HEC MONTRÉAL

Consultation citoyenne: Une analyse bonifiée pour la ville intelligente

Par

Nicola Tiberio

Sciences de la décision

(Intelligence d'affaires)

Mémoire présenté en vue de l'obtention

du grade de maîtrise ès sciences

(M.Sc.)

Décembre 2018

Sommaire

La participation citoyenne est un enjeu de plus en plus important pour les villes partout à travers le monde. Les villes veulent connaître l'opinion de leurs citoyens, et ceux-ci ne demandent qu'à pouvoir l'exprimer facilement. À l'heure actuelle, la participation des citoyens aux consultations citoyennes est très basse étant donné les contraintes de temps qu'elles impliquent. L'entreprise B-Citi y a vu une opportunité d'affaires intéressante, développant ainsi une plateforme permettant entre autres de sonder la population en ligne.

Afin de laisser les citoyens s'exprimer librement et pleinement, la plateforme de B-Citi permet l'utilisation de questions ouvertes. Ces questions permettent aux citoyens d'exprimer non seulement leur opinion, mais aussi des idées auxquelles les gestionnaires des villes n'auraient pas toujours pensées. Cependant, de telles questions donnent un défi d'envergure en ce qui concerne l'analyse des résultats. En effet, les réponses sont sous forme de données non structurées, soit du texte, ce qui est très long à lire s'il y a beaucoup de répondants. De plus, ces données nécessitent un travail supplémentaire pour être analysées automatiquement par un ordinateur. C'est sur cet aspect que ce concentre plus particulièrement ce mémoire.

L'analyse des réponses écrites par les citoyens se sépare en deux volets : la classification, où l'on tente d'identifier les idées exprimées par les citoyens en réponse à une question, et l'analyse de sentiments, où l'on cherche plutôt à voir si le sujet de la question est abordé de façon positive ou négative par les citoyens. De plus, il faut simplifier et automatiser au maximum la tâche car le niveau de connaissances en analyse de données, structurées ou non, des gestionnaires des villes est inconnue. Il faut donc que la solution proposée soit robuste et facile de compréhension.

C'est avec ces critères en tête que l'analyse de regroupement a été la technique retenue pour le volet classification. Cette méthode a l'avantage d'être entièrement non-supervisée. Brièvement, cette méthode consiste à nettoyer les données, traiter le texte afin qu'un ordinateur ou un algorithme puisse le comprendre et ensuite regrouper ensemble les textes qui se ressemblent le plus. Dans le cas présent, les textes sont les commentaires laissés par les citoyens. Une fois les textes jugés similaires par l'algorithme de classification regroupés, il est possible de faire ressortir les mots qui définissent ce groupe. Chaque groupe obtient aussi un poids, correspondant au nombre de personnes faisant partie du groupe. Au final, en combinant les mots saillants de chaque groupe et leur poids dans l'ensemble des réponses, le gestionnaire pourra voir quelles idées exprimées sont les plus populaires, mais aussi quelles combinaisons

d'idées viennent souvent dans un même groupe ou dans des groupes différents. Cela permet entre autres au gestionnaire d'essayer de répondre à des idées exprimées par des personnes différentes et ainsi satisfaire une plus grande partie de la population.

Pour ce qui est du volet analyse de sentiments, la technique retenue est l'entraînement d'un réseau de neurones qui estime ensuite la probabilité qu'un commentaire donné soit positif. Le réseau de neurones est entraîné sur un large corpus de commentaires exprimant des opinions sur des sujets variés, et séparés entre positifs et négatifs. Cela permet au réseau de neurones d'apprendre ce qui constitue un commentaire positif par rapport à un commentaire négatif. On utilise les bi-grams dans le traitement du texte, soit chaque sous segment de deux mots trouvés dans un commentaire. Cela permet entre autres de bien considérer l'impact d'une marque de négation dans une phrase. La probabilité mesurée qu'un commentaire soit positif est donc sa positivité. Un gestionnaire d'une ville pourra alors trouver la positivité moyenne pour une question donnée, et même voir si elle diffère selon différents segments de la population.

Le mémoire propose aussi des solutions pour bien représenter les résultats à des réponses de sondage plus classiques, ainsi que plusieurs avenues intéressantes pour utiliser la géolocalisation dans les consultations citoyennes.

Cette recherche vient faciliter le passage de la consultation citoyenne au numérique, ce qui aura pour conséquence d'améliorer la participation citoyenne et d'offrir la possibilité aux villes d'utiliser davantage les opinions et les idées des citoyens dans leurs décisions.

Remerciements

Avant toute chose, je tiens à remercier François Bellavance, mon directeur de maîtrise. En tant qu'étudiant, le mémoire est bien souvent le premier projet de cette envergure auquel on fait face. François aura su bien m'orienter et me conseiller tout au long de ce long parcours. Sans cette aide, je sais que je ne serais pas aussi fier du travail accompli, c'est pour cette raison que je le remercie.

Je remercie également B-Citi pour m'avoir donné l'occasion de participer à un projet stimulant auquel je crois sincèrement. Je souhaite à toute l'équipe que l'entreprise connaisse le succès pour lequel tous travaillent très fort. J'aimerais particulièrement remercier Patrick Munroe, qui m'a aidé à trouver des bonnes façons d'attaquer le problème, Paul Molins, qui a su amener le produit à un autre niveau pour la qualité de l'analyse textuelle, et Carle Beauchamp, pour son feedback constructif en fin de projet.

J'aimerais également remercier Mitacs, qui m'ont permis de me consacrer à ce projet à temps plein, et mes parents qui ont toujours fait tout ce qui était en leur pouvoir pour faciliter mon accès aux études. J'ai été très privilégié de me rendre aussi loin dans mes études avec autant de support, j'en suis pleinement conscient et j'en serai éternellement reconnaissant.

Finalement, un énorme merci à Kim, ma copine, qui a enduré mes séances de rédaction pendant nos journées de congé, mes questionnements incessants sur la méthodologie, et pour sa capacité à écouter et sa volonté de comprendre tout ce que je fais, avec un intérêt sincère et franchement motivant. Son support dans tout ce que je fais est réellement inestimable.

À vous tous, je le redis encore, merci énormément!

Contenu

Chapitre 1: Introduction

1.1 Mise en contexte

De plus en plus, les villes à travers le monde cherchent à profiter du nombre toujours grandissant de données auxquelles elles ont accès. Plusieurs villes sont très avancées à ce niveau. On peut penser par exemple à la ville de Pittsburgh, qui a ouvert ses données à la population afin que ses citoyens aient accès au plus d'information possible. Ce système, baptisé *Burgh's eye view* permet aux citoyens d'accéder à de l'information qui n'avait jamais été rendue publique auparavant, dans le but de connecter le citoyen à sa ville (Grant, 2016). On peut entre autres y trouver de l'information sur le trafic, le transport en commun, mais aussi des lieux où des crimes ont été rapportés. C'est beaucoup dans un souci de transparence que la ville souhaite donner cette information aux citoyens. Cette transparence vise à amener un climat de collaboration et de communauté au sein d'une ville, et d'ainsi accroître l'engagement citoyen. Les villes cherchent aussi à prendre des décisions qui plairont aux citoyens afin de consolider cet engagement, certaines voulant même avoir l'opinion de ses citoyens.

Cette vision de connecter le citoyen à sa ville, c'est la même qui a motivé l'apparition de la plateforme B-Citi au Québec. B-CITI Solutions Inc. développent l'application B-CITI, qui est une plateforme intelligente multi-villes/multi-canaux qui a pour objectifs de connecter les systèmes des villes, centraliser et nettoyer les données, numériser les services aux citoyens et offrir l'analyse en temps réel des données. L'application B-CITI dispose d'un module de consultation citoyenne qui permet aux administrateurs d'une ville de créer des consultations citoyennes. Par consultation citoyenne, on entend le fait d'obtenir l'opinion de la population avant d'aller de l'avant avec des projets, ou encore de recueillir des idées provenant de la population. La consultation classique a souvent lieu les soirs de semaine, à l'hôtel de ville. Or, le taux de participation est bien souvent très bas, même trop bas pour en tirer des informations utiles à une prise de décision éclairée, et l'ambiance amenée par les citoyens a tendance à être cynique, comme rapporté par des représentants de villes contactés par B-Citi. Cela est confirmé dans un article paru dans L'Actualité, qui avance que le taux de participation aux budgets participatifs atteint rarement les 5% (Dubé, 2017).

C’est pourquoi B-Citi veut, par son module de consultation citoyenne, offrir la possibilité aux villes de consulter sa population en ligne, pour ainsi obtenir une plus grande participation et ce de façon beaucoup plus rapide et efficace.

1.2 Problématique

La problématique et le défi est le traitement et la visualisation des données recueillies par le module de consultation citoyenne de l’application de B-Citi. En effet, le grand volume de données anticipé sera quelque chose de nouveau et devra être bien traité. Ce sera d’ailleurs le sujet qui sera traité dans ce mémoire : Comment analyser et visualiser les données des consultations citoyennes de façon à ce que les gestionnaires des villes puissent prendre des décisions plus rapides et éclairées?

1.3 Objectifs

Les objectifs sont les suivants :

- Rendre au minimum le travail d’analyse et de traitement que les gestionnaires des villes auront à faire.
- S’assurer de l’exactitude des calculs et des représentations graphiques, et rendre le tout automatisé.
- On doit faire l’hypothèse que le niveau de connaissance en statistique et analyse de données des gestionnaires des villes est relativement faible, le tout devra donc être le plus simple possible. Vulgariser est la clé.
- Pouvoir traiter tous les types de questions mis en place par la plateforme B-Citi. La solution devra être flexible aux différentes consultations possibles.

1.4 Défis et contributions

La plupart des questions mises en place par B-Citi sont des questions de sondage classique. Leur traitement sera relativement similaire à ce qui est fait par les grandes firmes de sondage. D'ailleurs, B-Citi a déjà une bonne partie du travail de fait, il s'agit ici d'améliorer la visualisation de certaines données.

Cependant, B-Citi veut se démarquer dans la géolocalisation des données. En effet, les utilisateurs de la plateforme s'inscrivent et donnent quelques informations personnelles, dont leur adresse. Cela signifie que B-Citi peut représenter les réponses de consultations sur une carte, et vérifier s'il y a des tendances par quartier, ou district électoral, ce que les représentants des villes consultés par B-Citi ont souligné comme étant un point d'intérêt capital.

Là ou B-Citi peut aussi se démarquer par rapport à d'autres services de sondage comme Survey Monkey est dans le traitement des questions ouvertes. Ces questions posent un défi de taille, car le traitement du langage naturel de façon automatisée amène toujours son lot de difficultés et d'inexactitudes. Ce sera donc le plus gros des défis et là où il y aura le plus de contributions, tant pour aider B-Citi, que pour faire avancer le traitement automatisé des questions ouvertes en général. Le mémoire portera donc principalement sur ce sujet.

1.5 Structure du mémoire

Le mémoire est divisé en quatre chapitres. Le premier présente les faits saillants de la revue de littérature pertinente à la réalisation du projet. Cette revue permet d'en savoir plus sur ce qu'est une ville intelligente, en plus de relever les principaux apprentissages en ce qui concerne l'analyse du langage naturel. Diverses méthodes y sont présentées.

Le deuxième chapitre est consacré à la description et à la comparaison des différentes méthodes présentées pour les questions de sondage classique, de la géolocalisation des répondants et de l'analyse des commentaires dans les questions ouvertes.

Le troisième chapitre sélectionne les méthodes jugées les plus efficaces afin de répondre aux besoins des gestionnaires des villes. Il s'agit de trouver le meilleur équilibre entre qualité de l'information, facilité de

compréhension et rapidité d'exécution. Des exemples et simulations sont utilisées afin d'illustrer les méthodes d'analyse et la visualisation des résultats.

Finalement, le quatrième chapitre conclue en revenant sur tout le cheminement des chapitres précédents avec des commentaires finaux sur les analyses ainsi que de nouvelles pistes pour des recherches futures.

2. Revue de littérature

2.1 Traitement du texte

L'analyse de texte, ou plus précisément le *text mining,* comporte plusieurs concepts de base importants. En effet, afin que l'on puisse utiliser les mêmes méthodes d'exploitation de données (*data mining*) que lorsque les données sont structurées, le texte doit tout d'abord être traité. Bien souvent, l'objectif est de le convertir en vecteurs relatant la présence de différents mots (attributs, termes) dans différents documents (observations) (Weiss, Indurkhya et Zhang, 2010). Cette étape est communément appelée vectorisation. C'est d'ailleurs ce qu'accompli le TF-IDF, acronyme anglais se traduisant par fréquence du terme et fréquence inverse dans les documents; on observe combien de fois un mot est présent dans un document. S'il est très présent, cela aide à définir le document. S'il est présent dans trop de documents, son pouvoir de différenciation, nécessaire à la classification, en est alors réduit. C'est donc la principale caractéristique du TF-IDF qui permet d'attribuer un poids à chacun des mots (Weiss, Indurkhya et Zhang, 2010). Les poids des mots sont propres à chaque document, les poids étant déterminés par le produit de la fréquence dans le document (TF) et de la fréquence inversée dans l'ensemble des documents formant le corpus (IDF). La fréquence dans le document est simplement la fréquence du mot dans un document donné, divisé par la longueur du document. Si un document de 32 mots contient le mot chat 4 fois, la valeur de TF pour le mot chat sera de 4/32. La fréquence inversée d'un terme est définie comme le logarithme du nombre total de documents dans le corpus divisé par le nombre de documents comportant le terme. Par exemple, supposons un corpus de 100 documents dont 37 comportent le mot chat au moins une fois. La valeur de IDF sera donc égale au log(100/37). Le poids final TF-IDF du mot chat pour le document s'obtient donc en multipliant les deux valeurs. Dans l'exemple précédant du mot chat, on aurait (4/32)xlog(100/37)=0,054. La valeur TF-IDF d'un mot est donc spécifique à chaque terme, et à chaque document dans un corpus. Un poids élevé signifie donc une présence accrue du terme dans le document en question, et faible dans l'ensemble des documents (Weiss, Indurkhya et Zhang, 2010). À l'inverse, si un terme se retrouve dans presque tous les documents, son poids tendra vers 0 puisque la valeur de l'IDF sera proche de log(1)=0. C'est une façon de faire plus efficace que considérer seulement la présence ou non d'un mot ou encore un simple compte de mots. De plus, il est possible de filtrer les mots et ainsi

retirer les mots qui apparaissent dans trop de documents et ceux qui apparaissent dans trop peu d'entre eux. Cela permet de réduire la taille de la matrice TF-IDF et d'accélérer la vitesse de calcul pour obtenir l'information des textes ou encore prédire ce qu'ils contiennent. Le tableau 1 présente un exemple de matrice TF-IDF. Chaque ligne donne le poids TF-IDF d'un mot d'un document dans un corpus.

(# doc,#mot)	Poids TF_IDF
(0, 93)	0,240114196466
(0, 73)	0,32893112706
(0, 49)	0,421509007637
(0, 8)	0,356084677452
(0, 60)	0,290660347267
(0, 14)	0,383238227844
(0, 134)	0,36429745133
(0, 70)	0,406959132433
(1, 30)	0,502993482661
(1, 123)	0,266595644134
(1, 129)	0,266595644134
(…)	(…)

Tableau 1 Exemple de matrice TF-IDF pour un corpus de textes.
La première colonne donne la combinaison
document/mot et la deuxième le poids qui y est associé.

Une autre étape du traitement du texte, qui est préalable à la vectorisation est celle de la désuffixation. Cette étape consiste à enlever les suffixes des mots afin d'en garder seulement la racine sémantique (Bird, Klein et Loper, 2009). Par exemple, on souhaite que promener et promenades retournent tous deux la racine « promen » afin que l'ordinateur comprenne que ces deux mots différents réfèrent au même concept. Cela facilite beaucoup le travail de l'algorithme du TF-IDF et permet de mieux capter l'information cachée dans les textes, ce qui est d'ailleurs l'objectif principal en ce qui concerne l'analyse des commentaires pour une consultation citoyenne. La désuffixation vient réduire le nombre de mots différents présents dans les textes, réduisant ainsi la taille de la matrice TF-IDF. De plus, en réunissant les mots ayant le même sens, elle permet de mieux distribuer les poids de chaque terme dans cette même matrice.

Les textes peuvent aussi être traités avec la technique des n-grams. Un n-gram est une sous-séquence de longueur n dans une suite de mots. Il peut y avoir différents niveaux de granularité lorsque l'on regarde les n-grams d'un texte. On peut s'arrêter aux différents mots, aux différentes lettres ou même aux différents caractères, dépendamment de l'objectif de l'analyse. De décortiquer un texte en n-grams permet d'incorporer beaucoup de contexte dans l'analyse et de bien comprendre les nuances et complexités de la langue, beaucoup plus que si seuls les mots simples sont considérés (Ghiassi, Skinner et Zimbra, 2013). Par exemple, si des n-grams s'arrêtant aux mots sont utilisés, on peut facilement observer les n-grams ayant une marque de négation. Dans Ghiassi, Skinner et Zimbra (2013), les n-grams sont utilisés dans de l'analyse de sentiments. Ils avancent entre autres la capacité des n-grams à optimiser la qualité des variables retenues dans l'analyse (dans ce cas-ci, des chaînes de mots). En effet, en retenant les n-grams dépassant seulement un certain niveau de fréquence dans le corpus, on s'assure de n'utiliser que des variables de qualité. Il est aussi assez simple d'exclure des séquences de n-grams qui ne reflètent aucun sentiment. De plus, dans un cas où l'analyse de sentiments est sur les réseaux sociaux, utiliser des n-grams allant jusqu'au caractère permet de facilement traiter les émoticônes ou emojis comme un sourire (:)), surtout lorsque plusieurs sont utilisées l'un à la suite de l'autre, ce qu'un simple compteur de mots n'arriverait pas à faire. Faire augmenter n augmente la précision de la classification en analyse de sentiments. Cependant, plus le n est grand, plus la séquence de mots devient spécifique et il est donc moins probable de la retrouver dans un corpus de textes ou de commentaires (Ghiassi, Skinner et Zimbra, 2013). Il faut donc trouver un juste milieu entre précision et fréquence lors du choix de n menant à la sélection de variables.

2.2 Classification de documents

En ce qui concerne la classification de documents, plusieurs méthodes ont été utilisées dans les dernières années, avec un degré de succès variant d'efficace à très efficace. La classification de documents a pour objectif de distribuer les textes dans différentes catégories prédéfinies, appelées « *labels* » en anglais. Il faut donc avoir une certaine idée de l'information existant dans la banque de textes pour déterminer les catégories. Certains documents sont déjà classés dans différents dossiers, représentant une catégorie spécifique. On présente ensuite les nouveaux documents à un algorithme qui a été entraîné en fonction de ce qui a déjà été classé et qui détermine où devront aller les nouveaux documents (Weiss, Indurkhya et Zhang, 2010). En général, lorsqu'on veut utiliser un algorithme de classification, il est fréquent de voir des algorithmes tels Naive-Bayes, les séparateurs à vaste marge (*support vector machines*) ou encore les

réseaux de neurones. Les réseaux de neurones semblent être la méthode de classification de documents la plus efficace dans la plupart des cas (Manevitz et Yousef, 2007). Ces algorithmes font partie des méthodes dites supervisées, ou semi-supervisées, car elle nécessite une part d'intervention humaine, soit la classification des documents pour entrainer les algorithmes dans des catégories prédéfinies, ou des premiers documents dans le cas semi-supervisé.

Comment faire lorsque l'on n'a pas une idée ou une intuition de l'information que contiennent les documents pour définir les catégories? L'analyse de regroupement semble être une méthode appropriée (Weiss, Indurkhya et Zhang, 2010). Ici, on regroupe les documents qui se ressemblent le plus afin de les catégoriser ensuite. Afin de déterminer si les documents se ressemblent, on les compare selon les différents termes, et leur poids contenus dans la matrice TF-IDF. Il n'y a pas d'étape où l'on doit classer certains documents au préalable; il s'agit donc d'une méthode non-supervisée. Weiss, Indurkhya et Zhang (2010) suggèrent que cette méthode est en général moins précise qu'une classification semi-supervisée, mais que beaucoup d'information peut être captée rapidement par une bonne analyse de regroupement (*clustering*).

Afin de déterminer la qualité des regroupements, il est possible d'utiliser ce qu'on appelle le *silhouette score.* Le *silhouette score* est en quelque sorte une mesure de validité des regroupements qui est obtenue en mesurant la similarité des observations au sein d'un même groupe (cohésion) et les différences avec celles des autres groupes (séparation) (Rousseeuw, 1987). Ce score s'échelonne de -1 à 1, 1 étant la valeur optimale indiquant des regroupements de grande qualité. Il se calcule avec les mesures de distance classique comme la distance euclidienne ou encore la distance de Manhattan. On soustrait la distance moyenne des observations au sein de son regroupement, de la plus petite distance moyenne entre le regroupement en question et n'importe quel autre regroupement. On divise le tout par la distance maximale moyenne entre deux regroupements. Cela est répété sur chacune des observations. Il est donc possible ensuite de prendre la moyenne de tous ces scores ou encore de les visualiser en utilisant un diagramme en bâtons indiquant la distance euclidienne de chaque observation par rapport au regroupement le plus près de l'observation, sans être son propre regroupement. La moyenne des scores de l'ensemble nous donne un score spécifique pour cette analyse de regroupement. Si on ne change pas l'algorithme de regroupement ou les documents, le *silhouette score* restera toujours le même. Il variera si on demande un nombre de regroupements différents.

Il est difficile de savoir d'avance combien de documents sont nécessaires pour une bonne classification de textes. En effet, c'est presque du cas par cas (Weiss, Indurkhya et Zhang, 2010). Cela dépend entre

autres de la quantité de documents par catégorie. Si une catégorie est très petite, mais aussi très différente des autres, il sera facile de la classer. Cependant, si elle ressemble aux autres, il faudra d'autres documents pour aider à la classer correctement. Il n'y a donc pas vraiment de règle empirique sur le nombre de documents requis au total ou par catégorie pour bien entrainer un algorithme ou produire de bons regroupements. La variabilité d'un texte à l'autre est généralement beaucoup plus grande que dans des cas strictement numériques, rendant chaque cas différent des autres.

2.3 Analyse de sentiments

L'analyse de sentiments est souvent considérée comme un problème plus complexe que la classification de documents. La raison est que l'on doit vraiment décortiquer les subtilités du langage écrit, la signification des mots et surtout le contexte (Liu, 2015). Liu (2015) propose une façon de décortiquer une opinion en quatre parties distinctes : la cible, le sentiment ou l'opinion, le locuteur et le moment. La cible est l'objet ou l'individu sur lequel porte une opinion. Le sentiment ou l'opinion est ce qui en est dit, qu'on résume souvent à positif ou négatif, mais on peut aussi appliquer une échelle numérique pour mesurer le niveau de la polarité de l'opinion. Le locuteur est la personne qui assume ou exprime l'opinion émise. Dans un contexte de critique d'un produit, on a souvent l'opinion de plusieurs utilisateurs du produit dans un seul commentaire, ce qui rend l'analyse assez complexe. Finalement, il y a le moment où l'opinion a été émise. Encore dans un contexte de critique de produit, cela est très important afin de voir l'évolution que peut avoir la perception générale des clients.

Voici un exemple proposé par Liu (2015), traduit de l'anglais, qui illustre bien les quatre composantes de l'opinion :

John Smith, le 10 septembre 2011

J'ai acheté une caméra Canon G12 il y a six mois (1). Je l'adore (2)! La qualité de l'image est impressionnante (3). La durée de vie de la batterie est longue (4). Toutefois, ma femme la trouve trop lourde pour elle (5).

On remarque plusieurs choses dans cet exemple. Premièrement, il y a quatre opinions différentes dans ce court commentaire de cinq phrases, soient dans les phrases numérotées 2, 3, 4 et 5. Deuxièmement, elles réfèrent à différents aspects de la caméra. Troisièmement, l'opinion de deux personnes différentes est émise dans ce commentaire. Finalement, ces opinions divergent.

On peut décortiquer chacune des phrases selon les composantes de l'opinion. Notons que le moment est le même pour chacune des phrases, soit la date du commentaire. Dans la phrase 2, John donne une opinion positive sur la caméra en général. John en est le locuteur, la caméra est la cible, et l'opinion est positive. Dans la phrase 3, l'opinion est maintenant dirigée plus spécifiquement sur la qualité de l'image, tout en restant positive. Dans la phrase 4, la cible change encore, allant maintenant vers la durée de vie de la batterie. Finalement, dans la phrase 5, le locuteur devient la femme de John, la cible le poids de la caméra, et l'opinion exprimée est plutôt négative.

Cet exemple permet de montrer d'autres concepts de l'opinion avancés par Liu (2015). L'opinion peut être rationnelle ou émotive. La phrase 2 est une opinion émotive, tandis que les phrases 3, 4 et 5 sont des opinions rationnelles. Les opinions rationnelles sont généralement considérées comme étant plus fortes que les opinions émotives. Cependant, ces dernières sont plus faciles à identifier.

Plusieurs méthodes ont été proposées au cours des années afin d'extraire l'opinion exprimée dans les données textuelles. Hu et Liu (2004) proposent une classification supervisée visant à donner une opinion générale des commentaires. Plusieurs méthodes sont testées pour arriver à un résultat satisfaisant. C'est la façon la plus simple de traiter l'analyse de sentiments vu qu'on se rapproche grandement de la classification. Cependant, Hu et Liu (2004) démontrent la pertinence de diviser les commentaires par phrase, afin de capturer les opinions pouvant s'appliquer sur des cibles différentes. De plus, ces auteurs proposent de regrouper les textes par sujet avant de les classer par sentiment, afin d'avoir une opinion plus précise qu'une simple vue d'ensemble. Cela permet non pas de seulement savoir si on considère l'objet positivement ou négativement, mais aussi quels aspects sont bien ou mal vus. Cependant, il est à noter que cette méthode nécessite une intervention humaine pour classer un bon nombre de documents de façon positive ou négative, donc de façon binaire ou encore de les classer comme neutre si c'est pertinent pour l'analyste.

Taboada et al. (2011) ont proposé une méthode non-supervisée qui diffère grandement. On a comme base deux lexiques, un positif et un autre négatif. Ces lexiques contiennent des mots à connotation positive ou négative. On passe un par un tous les mots d'un commentaire et on ajoute des points pour chaque mot positif, et on en enlève pour chaque mot négatif. Il est possible d'inclure des mots venant bonifier la valeur d'un mot. Par exemple, *très bien* vaudrait plus que *bien* seul. De plus, on peut aussi considérer que les marques de négation viennent inverser la valeur d'un mot. Par exemple, *pas fameux* rend négatif le mot *fameux.* Finalement, il est aussi possible de donner plus de points pour des mots plus fort afin de venir raffiner l'analyse. Par exemple, *incroyable* représente une opinion clairement plus

marquée que *bien.* Cette méthode a pour avantage de venir donner une forme de degré à la polarité de l'opinion. On peut facilement distinguer des commentaires très positifs et d'autres très négatifs et les faire ressortir plus par rapport à ceux dont l'opinion est plus modérée. Dans un contexte d'entreprise par exemple, on peut vite identifier les cas problématiques qui nécessitent une action rapide.

Martineau et Finin (2009) proposent une variation du TF-IDF, soit le Delta TF-IDF qui a été grandement efficace dans l'identification du sentiment dans les documents. Le principe est de séparer une partie des documents selon leur polarité, positive ou négative. Ensuite, on génère une matrice TF-IDF pour les documents positifs, et une autre pour les documents négatifs. Ensuite, on soustrait l'une de l'autre. Les termes se retrouvant à la même fréquence dans les deux types de documents auront des poids similaires; lors de la soustraction, ils auront tendance à s'annuler. C'est un résultat souhaitable, car agissant de la même façon que le texte soit positif ou négatif, ils ne sont pas discriminants. Les termes qui seront discriminants en ce qui concerne la polarité d'un texte ou d'un commentaire auront un poids élevé dans l'un ou l'autre des types de documents seulement, ce qui se traduira par un poids élevé dans la matrice du Delta TF-IDF. Cette nouvelle matrice servira à entrainer un algorithme de classification qui sera plus apte à détecter les variations de sentiment que s'il avait été entraîné par la méthode TF-IDF régulière.

2.4 Correcteur de fautes

Un aspect de l'analyse textuelle qui est plus répandu dans les applications au grand public est la capacité de corriger des fautes automatiquement. On peut penser à Google qui suggère assez facilement des corrections à des fautes de frappe lors d'une recherche (Norvig, 2007), à des logiciels de correction comme ceux inclus dans Word et Antidote, ou encore la suggestion de mots d'un clavier de téléphone intelligent. La plupart des correcteurs fonctionnent en ayant de prime abord une banque de mots existants. Tous les mots n'y figurant pas sont alors identifiés comme étant potentiellement erronés. Une fois le mot identifié, on peut prédire si le mot était effectivement erroné et prédire ce qui aurait dû être écrit. Bien souvent, cette prédiction sera un mot de la banque de mot qui a un seul caractère différent de celui qui est écrit. Certains systèmes peuvent même prédire une faute de frappe selon la proximité des lettres sur un clavier. Cependant, la plupart de ces systèmes dépendent de la banque de mots, souvent annotée, bâtie bien souvent manuellement. Antidote et Word réussissent bien à corriger en fonction du contexte des mots.

Whitelaw et al. (2009) propose une approche différente. Elle ne dépend pas d'une banque de mots, ce qui la démarque déjà des autres correcteurs. Pour cette méthode, un énorme corpus de textes a été bâti

à partir du Web, bien souvent avec des articles de journaux. Plus d'un milliard de pages Web ont été utilisées pour y arriver. La méthode suppose que la plupart des mots y sont bien écrits, mais que certains ne le sont pas. Ce corpus permet de recenser les mots les plus utilisés sur le Web, en s'arrêtant à 10 millions de mots. Ensuite, on bâti des n-grams avec ces mots. Un n-gram, comme mentionné dans la section 2.1, est une sous-séquence de n éléments à partir d'un mot, d'une phrase, ou d'un paragraphe. Dans le cas présent, on utilise un n-gram dont la granularité s'arrête au caractère près. Par exemple, dans la phrase « Ceci est une phrase », « ceci », « eci » et « e ph » sont tous des n-grams où n = 4. L'utilité de cette méthode pour un correcteur est que l'on pourra tenir compte du contexte dans lequel les mots sont utilisés, et donc tenter de corriger les erreurs d'accord en plus des fautes de frappe ou d'orthographe. En effet, les auteurs ont bâti ce qu'ils appellent un *n-gram language model (LM)*. C'est ce LM qui est entraîné avec les textes du Web.

Lorsque l'on demande au correcteur de corriger un paragraphe, le contexte est utilisé. En plus du contexte, la fréquence des mots dans le corpus (donc sur le Web) est considérée. Si un mot est écrit d'une façon mais qu'un mot prédit comme étant similaire apparait 10 fois plus souvent dans un contexte similaire, on peut alors substituer avec le mot plus fréquent. Des tests ont montré que cette méthode peut faire passer un corpus de textes contenant un taux d'erreur de 10,8% à un taux d'erreur de 3,8%. Whitelaw et al. (2009) ont remarqué que plus il y avait d'éléments contextuels présents, plus ils pouvaient faire confiance au n-gram LM.

2.5 Traduction

En ce qui concerne la traduction d'un texte dans une autre langue, encore une fois le grand public y est plus souvent exposé que les autres aspects de l'analyse textuelle. Beaucoup de gens ont déjà utilisé l'application Google Translate et constaté son évolution, offrant de plus en plus de langues et des résultats de plus en plus exacts. Les systèmes de traduction à travers le temps ont effectivement grandement évolué. Au début, on utilisait un dictionnaire des deux langues concernées, et remplaçait le mot de la 1^{e} à son équivalent de la 2^{e}. Ensuite, la grammaire a été ajoutée au processus pour prendre en considération la structure spécifique de chaque langue. Un exemple est l'anglais qui place le sujet d'une phrase après son descriptif et l'espagnol qui fait le contraire. Les traducteurs ont aussi tenu compte de groupes de mots au lieu de mots individuels dans le processus.

Récemment, une nouvelle technique d'apprentissage machine (*machine learning*) a été développée, plus spécifiquement les réseaux de neurones récurrents (Massaron, 2018). Un réseau de neurones récurrents (RNR) prend en entrée l'état du réseau lors de son dernier calcul. C'est donc dire que les résultats d'anciens calculs peuvent influencer les nouveaux. Cela est très utile lorsque l'on veut faire de l'apprentissage en séquence ou encore faire apprendre à un réseau de neurones un énorme patron compliqué, comme le langage humain. C'est pourquoi cette technique prend de l'ampleur dans le monde de l'analyse textuelle (Medium, 2016). Le RNR apprend grâce à des mêmes textes ou phrases qui sont publiés en plusieurs langues, comme des publications de l'Union Européenne. Ensuite, le RNR prend une nouvelle phrase en entrée, et en sortie donne une version codée de la phrase en une série de chiffres représentant les caractéristiques de la phrase. Cette série de chiffres est ensuite envoyée dans un deuxième RNR qui sert à décoder la phrase préalablement codée dans une nouvelle langue. Cela évite de devoir connaître les règles du langage humain, l'algorithme découvre le tout lui-même (Medium, 2016). C'est une méthode dont l'efficacité dépend grandement de la grandeur du corpus de textes traduits (Massaron, 2018).

3. Méthodologie

3.1 Principes généraux pour l'analyse de données et la présentation des résultats de consultations citoyennes

Afin de bien visualiser les données, il est important de tenir compte de plusieurs principes généraux pour présenter les résultats de façon claire et efficace. Il faut aussi tenir compte du fait que les gestionnaires des villes n'ont pas nécessairement une formation ou une connaissance suffisante en statistique. Il faudra donc que tous les graphiques ou représentations visuelles soient simples à comprendre de tous. C'est d'ailleurs le premier conseil donné par Baer et al. (2009), il faut savoir à qui on s'adresse, et comment aborder cette cible.

Baer et al. (2009) mentionnent l'importance de choisir des outils visuels simples, mais qui reflètent bien la réalité que l'on veut présenter. Par exemple, il est important que les ordres de grandeur des figures représentent bien la différence réelle entre deux entités. Le choix des couleurs est aussi important, il ne faut pas trop en mettre, et tenir compte de la possibilité que la personne ciblée soit daltonienne. Pour faire un bon graphique, Baer et al. (2009) ont établi une liste des points à respecter. Un bon graphique doit :

- Attirer l'attention du lecteur.
- Présenter l'information simplement, clairement et de façon précise.
- Ne porte pas à confusion, ou ne trompe pas.
- Montre les données de façon concentrée (ex : un graphique linéaire plutôt que plusieurs diagrammes circulaires).
- Facilite la compréhension des données et met de l'avant les tendances et différences.
- Illustre un message ou une idée cohérente avec le sujet de la question ou du texte.

Dans bien des cas, Baer et al. (2009) suggèrent d'utiliser le diagramme en bâtons. Il a tendance à être le meilleur graphique pour respecter tous les points mentionnés ci-dessus. Ce sera donc le choix privilégié pour présenter les résultats des consultations citoyennes.

Afin de bien marquer les différences entre plusieurs groupes de données, Baer et al. (2009) conseillent de les présenter en ordre croissant ou décroissant des valeurs des statistiques (p.ex. fréquence, moyenne), afin de bien marquer les différences et d'éviter les pièges liés aux illusions d'optique. Ce principe ne sera toutefois pas appliqué dans le cas où les variables indépendantes sont ordinales afin de maintenir l'ordre de grandeur.

Pour ce qui est de visualiser les résultats sur une carte, les mêmes principes s'appliquent. Il faut qu'elle soit simple, qu'elle permette de voir des différences ou tendances et qu'elle soit à une échelle juste géographiquement.

C'est avec tous ces conseils en tête que les représentations visuelles des données seront construites et présentées aux utilisateurs de la plateforme B-Citi.

Construction de sondages

Avant de se lancer dans la méthodologie et l'analyse des divers types de questions, il est important de discuter de la construction de sondages et des avenues futures qui permettront à B-Citi de perfectionner leur produit. Il serait intéressant pour B-Citi, et surtout pour les utilisateurs de la plateforme, d'ajouter un outil ou un guide de construction de sondage. L'Institut du Nouveau Monde (INM) pourrait d'ailleurs grandement les guider dans cette avenue. Qualtrics offre aussi un guide des bonnes pratiques en construction de sondages qui donne un bon point de départ. En effet, afin d'améliorer la qualité des réponses et de réduire le nombre de questionnaires incomplets, il existe plusieurs pratiques pour rendre le sondage ou le questionnaire plus agréable à compléter. Il est important que le sondage ne soit pas trop long, afin que les répondants prennent le temps de compléter le questionnaire. Il est recommandé que le sondage prenne au plus cinq minutes à répondre afin de maximiser la qualité des réponses (Qualtrics, 2015). Qualtrics mentionne aussi qu'un répondant moyen répond à trois questions à choix multiple par minute. Bien qu'il ne s'agisse que d'une règle générale et non d'une vérité absolue, cela constitue un bon indicateur pouvant éviter les sondages trop demandant. Certaines questions nécessitent plus de réflexion que d'autres, il est important de ne pas en abuser. De plus, il est important que le répondant ait le sentiment de progresser au fur et à mesure qu'il répond aux questions. Des questions courtes et faciles peuvent grandement aider, surtout en début de sondage pour que le répondant s'intéresse rapidement (Qualtrics, 2015). Une barre de progression est aussi une bonne source de motivation pour le répondant.

Afin de guider les villes dans la construction d'une consultation citoyenne, on suggère à B-Citi de bâtir un système de points pour chaque type de question. Les questions plus complexes comme les questions

ouvertes ou les choix multiples avec plusieurs combinaisons ont plus de points que les questions où l'on donne une appréciation d'un énoncé simple sur une échelle de type Likert. Le but est qu'au fur et à mesure de l'ajout de questions, lorsque le questionnaire atteint ou dépasse un certain seuil de points, l'application suggère au gestionnaire de la ville de simplifier ou réduire son questionnaire, sans toutefois l'obliger. Cela permettrait d'aider les villes qui n'ont pas d'experts ou d'habitués de la construction de questionnaires dans leur équipe.

3.2 Types de questions

Avant de se lancer dans les analyses, il est important d'identifier tous les types de questions qui ont été pensés et mis en place par B-Citi dans leur application afin de savoir quels types de résultats seront générés. Pour le moment, B-Citi a élaboré et programmé huit types de questions pouvant aller chercher de l'information différente chez le citoyen, dépendamment de la question. La plupart des différents types de questions viennent surtout changer la façon avec laquelle le répondant interagit avec l'outil de sondage afin d'éviter la redondance, mais amène quand même quelques occasions où l'analyse sera un peu différente. Il est donc bien important de comprendre les types de questions ainsi que les bonnes façons de les analyser et de présenter les résultats. Afin de bien tester le tout, et parce qu'aucune vraie consultation citoyenne n'a été entreprise avec l'outil B-Citi avant la rédaction du mémoire, une consultation comportant 500 répondants a été simulée, avec des réponses et profils de répondants générés au hasard. Le tout a été fait sur Microsoft Excel. La consultation simulée pose des questions aux citoyens afin d'avoir une idée de ce qu'ils aimeraient voir dans le nouveau parc que la ville prévoit aménager, et où il devrait se trouver. Pour ce qui est du travail sur les données textuelles, il devient plus compliqué de générer des commentaires différents et pertinents pour une analyse de données. C'est pourquoi des données libres sur Internet, principalement sur Kaggle, seront utilisées. Ces données peuvent être des critiques de films, de produits achetés en ligne, etc. En ce qui concerne l'obtention de textes en français, l'INM a fourni quelques commentaires sur des consultations citoyennes passées, qui pourront servir à vérifier la viabilité des méthodes testées sur une langue différente que l'anglais.

Pour ce qui est du contenu graphique, des calculs et des algorithmes utilisés pour la réalisation du mémoire, tout a été fait avec le langage de programmation Python. Les librairies utilisées seront mentionnées au fur et à mesure qu'elles apparaissent dans les prochaines sections et les prochains

chapitres du mémoire. Il est à noter que pour des raisons pratiques de programmation, Python identifie les premiers items d'un tableau de données comme étant la ligne 0 au lieu de 1.

3.2.1 Questions de sondages classiques

Les données de sondage simulées ont été sauvegardées dans un fichier CSV (Comma-Separated Values) qu'il a été possible de lire en Python grâce à la librairie Pandas. Cette même librairie a permis de traiter les données, calculer des moyennes et autres indicateurs utiles à l'analyse. Les graphiques ont été produits avec la librairie Matplotlib, qui permet d'arriver rapidement à des graphiques clairs et faciles à modifier ou ajuster au besoin. Pandas et Matplotlib permettent de produire une multitude de graphiques différents qui répondent aux besoins de la plateforme de B-Citi pour la consultation citoyenne.

Le premier type de question est une question classique à choix multiples. Le créateur de la question peut décider si le répondant a l'option de choisir plusieurs réponses ou non. On peut donc diviser cette question en deux cas distincts, soit celui où il est possible de choisir plus d'une option, et celui où il est impossible de le faire. En général, chaque option prendra une valeur binaire qui dépend si cette option a été choisie par le répondant ou non. Dans le premier cas, la représentation visuelle sera un diagramme en bâtons, illustrant soit le nombre absolu de fois qu'une réponse a été choisie, ou la proportion des gens l'ayant choisie.

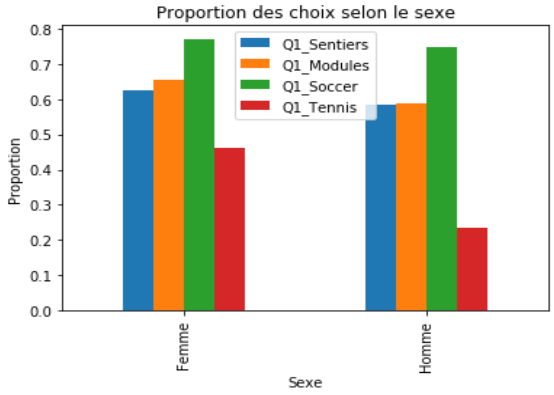

Figure 1- Exemple d'un diagramme en bâtons pour la visualisation des résultats d'une question à choix multiples où il est possible à un répondant de sélectionner plus d'un choix

À la figure 1, on peut voir un exemple des résultats à une question à choix multiples qui aurait pu être posée par une ville. On demande au citoyen de choisir les installations qu'il utiliserait dans un nouveau parc, en lui permettant d'en choisir plus d'une. Les résultats sont présentés ici sous forme de proportion des répondants ayant choisi cette option. La base de données de B-Citi permet facilement de séparer les résultats selon certaines caractéristiques des répondants, comme par exemple le sexe. L'utilisation du diagramme circulaire est à proscrire quand il est possible de choisir plus d'une réponse, car le total peut dépasser 100%.

Dans le cas où les répondants ne peuvent choisir qu'une seule réponse parmi les choix proposés, le diagramme circulaire peut être utilisé pour présenter les résultats. La figure 2 est un exemple de la même question où il est cette fois impossible de choisir plus d'une réponse.

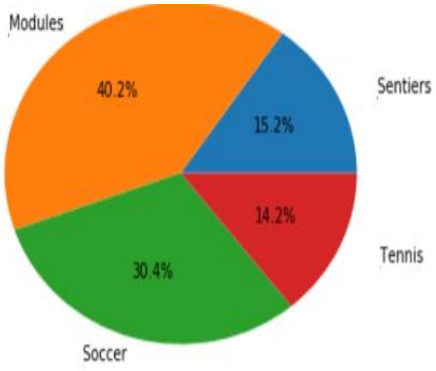

Figure 2- Exemple d'un diagramme circulaire pour la visualisation des résultats d'une question à choix multiples où chaque répondant doit sélectionner une seule option

On peut remarquer à la figure 2 un des inconvénients des diagrammes circulaires mentionnés par Baer et al. (2009). En effet, en un coup d'œil, il est très difficile de voir si la section « Sentiers » est plus grande que la section « Tennis ». Pour remédier à cela, il est bon d'afficher les valeurs en pourcentage associé à chaque section du graphique. De plus, si on désire présenter les résultats selon les modalités d'une caractéristique des répondants comme à la figure 1, il faut alors créer un diagramme circulaire pour chacune des modalités, ce qui est un désavantage par rapport au diagramme en bâtons lorsqu'on veut comparer les résultats.

Le deuxième type de question consiste à présenter des éléments (souvent visuels) au répondant afin que celui-ci leur donne une note d'appréciation sur une échelle de Likert de 1 à 5 par exemple. En général, le score moyen par option est calculé pour ce type de question. Un haut score signifie que c'est une option prisée par les citoyens. Or, la moyenne peut parfois présenter des pièges. En effet, une option ayant une moyenne de 3 peut avoir 500 personnes ayant répondu 3, mais aussi 250 personnes ayant répondu 5 et 250 autres ayant répondu 1. Ce sont pourtant deux situations complètement différentes. Un gestionnaire pourrait choisir l'option polarisée dans le deuxième cas, sachant qu'elle plaira grandement à la moitié de la population. Dans le premier cas cependant, il est moins évident que ça en vaille la peine. Pour remédier à ce problème, il peut être pertinent d'utiliser en plus de la moyenne les mesures de dispersion, ou encore la distribution des notes spécifiques attribuées à cette option lorsqu'évaluer sur une échelle de Likert.

Prenons un exemple où on présente trois photos, avec trois styles de parcs différents. On demande alors aux citoyens de mettre une note reflétant leur appréciation des divers styles de parcs.

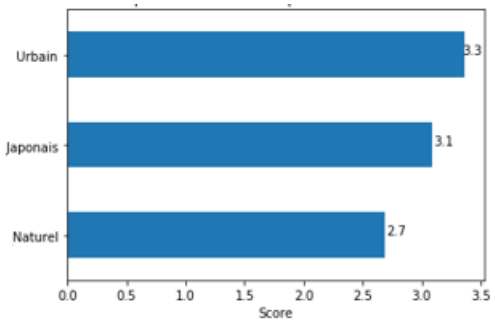

Figure 3-Scores moyens de l'appréciation des citoyens des divers styles de parcs proposés, la note maximale étant 5.

La figure 3 présente les scores moyens, sur 5, pour les trois options différentes. Comme expliqué préalablement, ces chiffres ne disent pas toute l'histoire. La façon dont les scores des répondants se dispersent est très riche en information utile pour le décideur. Prenons par exemple l'option 2, soit le style de parc urbain ci-dessus. Son score moyen de 3,3 indique une appréciation du style présenté, mais rien qui saute aux yeux, même si c'est l'option la plus prisée par les répondants. Si on regarde le nombre de fois que chaque score a été choisi, on obtient la distribution présentée dans la figure 4.

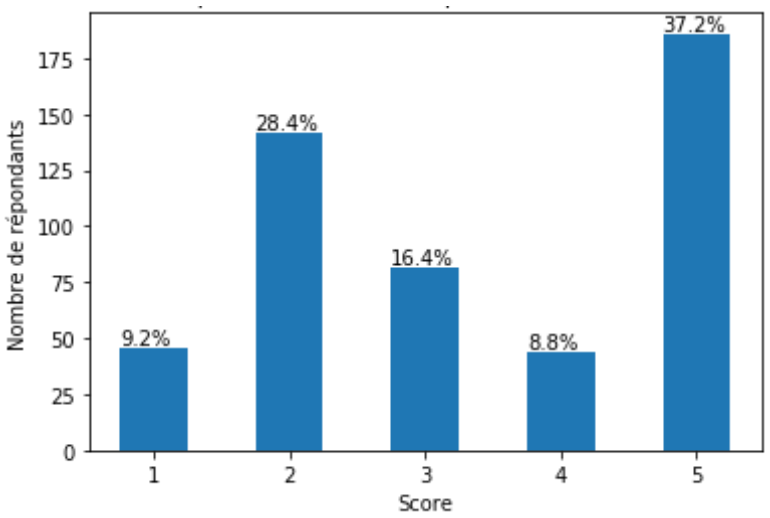

Figure 4- Répartition du choix urbain. Le pourcentage indiqué est la proportion des citoyens ayant choisi de donner un score en particulier au style de parc urbain tel que proposé par la ville

On remarque rapidement que la fréquence de la note maximale est assez loin devant toutes les autres, ce qui porte à croire qu'une bonne partie de la population sera satisfaite si le nouveau parc ressemble au style urbain. De voir que 37,2% des répondants seront satisfaits de façon maximale par ce type de parc parle plus que de dire qu'en moyenne, leur satisfaction s'élève à 3,3/5. Il faut donc avoir accès à toute cette information pour vraiment prendre une décision éclairée. On estime dans cet exemple que plus du tiers de la population serait très satisfaite.

Un troisième type de question consiste à demander au répondant de classer les choix présentés en ordre de préférence. Ce type de question se traite de manière similaire à la précédente. En effet, on attribue un score à chaque rang, soit n pour le premier rang, n-1 pour le 2^{e}, et ainsi de suite. Dans ce système, n représente le nombre d'options présentées dans la question. On arrive facilement à un score moyen pour chaque option, et le reste de l'analyse des résultats est identique à celle présentée aux figures 3 et 4. Le nombre de premiers rangs donne de l'information supplémentaire au décideur qu'un score moyen pourrait mal représenter. Cette question diffère surtout pour le répondant et la manière qu'il interagit avec le module de consultation.

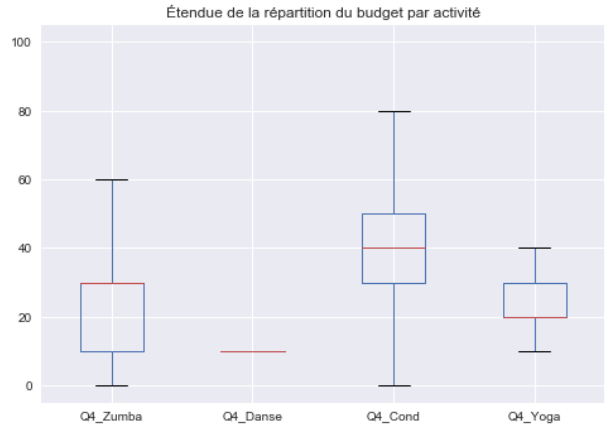

Figure 5- Boîte à moustache montrant l'étendue des résultats. On peut voir que les résultats pour l'option Conditionnement (Cond) varient beaucoup plus que ceux des options Yoga ou Danse.

Pour le quatrième type de question, on offre un budget X au citoyen et il doit le répartir sur diverses activités ou utilisations proposées par la ville. On a donc un budget moyen par option; cela ressemble quelque peu aux questions précédentes sauf que cette fois, l'étendue des réponses est encore plus grande, ce qui rend très important les représentations comme la boîte à moustache (*boxplot*) (voir figure 5) ou l'histogramme. Il importera aussi de voir s'il est fréquent pour une utilisation qu'elle reçoive la totalité du budget, ou à l'inverse, rien du tout. L'avantage de cette question sera surtout pour une ville qui voudra sonder sa population quant à son propre budget. Il sera possible de le faire avec l'interface pensé par B-Citi.

Le cinquième type de question en est un où on demande au répondant de choisir parmi des points prédéterminés sur une carte. L'analyse faite pour le premier type de question s'applique de la même façon, c'est une question à choix multiples. La différence est vraiment au niveau du visuel et de l'interaction pour le répondant, qui pourra visualiser sur une carte où sont les emplacements possibles proposés par la ville, comme par exemple le lieu du nouveau parc. Cela peut aussi servir à trouver quels lieux sont plus souvent fréquentés par la population.

Le sixième type de question consiste simplement à donner son appréciation sur un énoncé, avec une note sur une échelle au choix du gestionnaire de la ville. On arrive encore une fois avec un score moyen et une distribution des notes données par les répondants. Il n'y a donc rien de vraiment nouveau au point de vue de l'analyse et de la présentation des résultats.

3.2.2 Questions avec géolocalisation

Un des deux types de question qui n'a pas été traité dans la section précédente consiste à laisser le répondant choisir un point librement sur une carte. On peut également laisser un commentaire avec le point sur la carte, mais cet aspect sera traité plus en profondeur dans la section suivante. Le libre choix sur une carte servira aux villes, dans le cas d'un parc par exemple, afin de savoir si les citoyens ont d'autres idées quant à l'emplacement du nouveau parc. On considérera deux façons de présenter les résultats sur une carte, soit la carte de chaleur et la carte avec marqueur. Cela s'appliquera à deux cas, soit lorsque le répondant choisis un point sur une carte, ou dans le cas où le gestionnaire souhaite voir s'il existe des tendances géographiques dans les réponses pour une question donnée. Les représentants des villes contactés par B-Citi ont démontré un intérêt marqué pour ce genre de contenu.

Carte de chaleur

La carte de chaleur s'applique bien dans les deux cas.

Supposons comme exemple une question où on demande aux citoyens s'ils utilisent des terrains de soccer de façon régulière. Il peut être utile de placer sur une carte tous les gens ayant répondu oui, dans

l'éventualité où on veut ajouter des terrains de soccer dans la ville. Il sera possible de le faire pour B-Citi pour tous les répondants dont l'adresse civique est entrée dans leur profil. La figure 6 montre sur une carte la concentration des réponses à la question sur le soccer. Plus on s'approche du rouge foncé, plus la densité de répondants qui utilisent des terrains de soccer et qui habitent à cet endroit est forte. La figure 6 nous indique qu'il existe deux, peut-être trois endroits où il est évident que de placer des terrains de soccer sera plus apprécié par les citoyens que s'ils étaient placés ailleurs. Il s'agit donc d'une bonne façon d'optimiser l'utilisation des ressources en aménageant des installations qui seront assurément utilisées. C'est donc de l'information pertinente pour la ville et représentée d'une façon très simple à comprendre. La carte de la figure 6 a été produite avec la librairie Python Gmaps, qui est un accès à Google Maps via Python. Les données y sont très précises, mais son utilisation par la plateforme de B-Citi nécessiterait de débourser une importante somme d'argent. Par conséquent, B-Citi a l'intention de développer une solution de géolocalisation interne. Cela leur éviterait de faire appel aux serveurs de Google à chaque utilisation, les forçant à débourser une importante somme allant de 4 à 5$ par tranche de 1000 appels au serveur, dépendamment du volume mensuel. Si l'utilisation mensuelle dépasse les 500 000 appels, Google demandera alors de fonctionner sur une base contractuelle.

Figure 6- Utilisateurs de terrains de soccer selon leur adresse. Des coordonnées de longitude et de latitude ont été générées au hasard dans un intervalle fixe afin de simuler des résultats réalistes

Afin d'appuyer encore plus la décision des gestionnaires, il est possible d'améliorer l'information de la carte de la figure 6. En effet, il serait très pertinent que la carte indique les terrains de soccer déjà existants, afin de faire un choix logique plus facilement. Il serait facile de le faire si la ville intègre les coordonnées des parcs existants. Pour aller plus loin, la carte de chaleur pourrait refléter une fonction combinant la densité de la population utilisant les terrains, et la distance par rapport aux autres terrains. L'objectif serait donc de maximiser les deux composantes de la fonction.

Carte avec marqueur

La carte avec marqueur est très similaire à la carte de chaleur, mais elle offre un niveau de précision supplémentaire. Elle sera utilisée dans les cas où un répondant doit choisir un point librement sur une carte. Pour rester dans la thématique d'un nouveau parc, supposons que la ville demande à ses citoyens de choisir où ils aimeraient voir le nouveau parc. On leur présente une carte de la ville, et ils peuvent cliquer sur l'endroit qui leur semble le plus approprié. Ils peuvent même laisser un commentaire pour des précisions ou des souhaits liés à leur choix.

Comme on peut le voir sur la figure 7, lorsque la précision est minimale, ou lorsque nous adoptons une vue d'ensemble de la carte, les points qui sont les plus prêts se regroupent en un point dense, et le nombre de points dans ce regroupement est indiqué. Cela permet de facilement visualiser les choix faits par les citoyens.

Figure 7- Exemple de précision minimale d'une carte avec marqueur

Lorsqu'on clique sur un des regroupements, par exemple celui avec 492 répondants, la précision de la carte augmente afin de s'approcher des points qu'il renferme. Le regroupement se subdivise en de plus petits regroupements, qui à leur tour peuvent se subdiviser au fur et à mesure qu'on augmente la précision de la carte. Au final, on peut en arriver à une carte très précise avec des points individuels, et il est même possible de voir les commentaires laissés par les citoyens. C'est ce qui est illustré par les figures 8 et 9. Cette précision de géolocalisation soulève quelques débats éthiques quant à la vie privée des gens. D'ailleurs, Google a déjà eu à se défendre devant les tribunaux pour cette raison.[1] C'est pourquoi il est recommandé que B-Citi n'affiche la position du répondant seulement s'il a accepté de mettre son adresse dans le système et que celle-ci est utilisée avec son consentement, ou encore qu'on limite la précision de

[1] Les Affaires (2018), La (pas si) petite histoire de la géolocalisation, Les Affaires, récupéré le 11 octobre 2018 de https://www.lesaffaires.com/techno/internet/la-pas-si-petite-histoire-de-la-geolocalisation-/604407

l'affichage de façon à se limiter à des groupes d'au moins cinq à dix personnes à fin de conserver une forme d'anonymat.

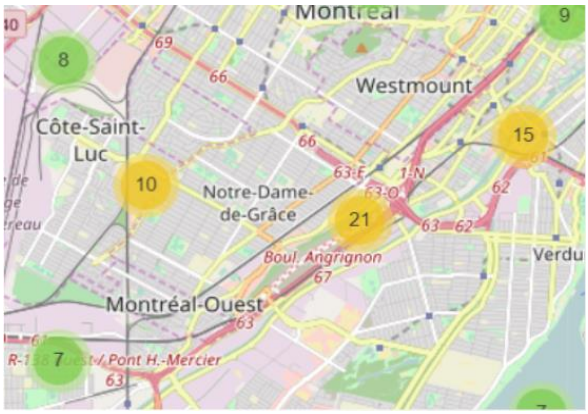

Figure 8- Précision moyenne. Les répondants se séparent tranquillement lorsqu'on s'approche de la précision maximale.

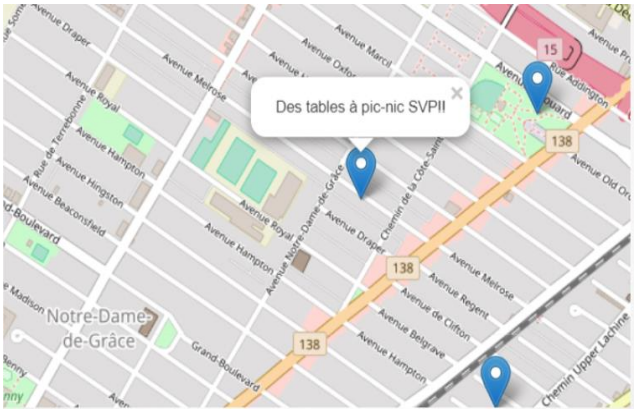

Figure 9- Précision maximale. Il ne reste que des points individuels. On peut explorer divers endroits de la ville de cette façon.

C'est la librairie Folium qui a été utilisée pour produire les cartes des figures 7 à 9, qui utilise Leaflet, une librairie ouverte de création de cartes. Folium permet d'accéder à toutes les fonctionnalités de Leaflet en Python, alors qu'elles sont normalement adaptées pour le langage de programmation JavaScript. Cette librairie a l'avantage d'avoir la fonctionnalité *Marker Cluster* qui a été utilisée et de ne pas engendrer les mêmes coûts d'utilisation que Gmaps.

Carte avec polygones

La présentation de résultats à l'aide d'une carte avec polygones n'est pas encore incluse dans la plateforme et n'est pas prévue pour la prochaine mise à jour. Cependant, cet ajout améliorerait grandement la qualité du produit offert par B-Citi et a déjà été discuté. Lors de séances de discussion entre B-Citi et des représentants des villes, ces derniers ont mentionné qu'ils aimeraient beaucoup voir les réponses aux questions en fonction de l'adresse des répondants, particulièrement par district électoral. Le tout pourrait être facilement représenté sur une carte. Pour ce faire, il faudra que les villes fournissent les coordonnées délimitant les polygones de leurs différents districts électoraux afin de les représenter sur la carte. Ensuite, il faudra associer chaque adresse à son district électoral. Ensuite, il suffit de colorer

chaque polygone en fonction de l'intensité ou la fréquence d'une réponse, en se servant de l'adresse du répondant. Un exemple des possibilités de cette représentation est présenté à la figure 10.

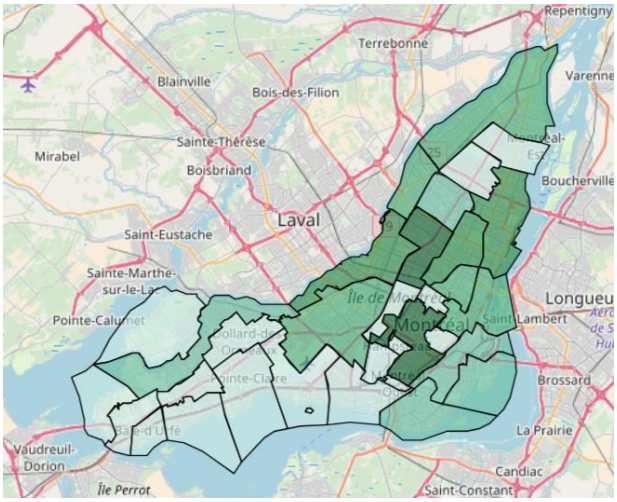

Figure 10- Carte de Montréal avec les polygones des différents quartiers. Les couleurs varient en fonction de la population du quartier, un vert plus foncé indiquant une population plus forte.

3.3 Questions ouvertes

Maintenant que les questions classiques et de géolocalisation ont été abordées, il reste l'imposant défi que représentent les questions ouvertes. Il est très pertinent pour une ville de laisser ses citoyens exprimer leur opinion librement. En effet, les questions ouvertes enlèvent la limite des réponses qu'imposent les questions de sondage classique comme les questions à choix multiples. On peut donc explorer différentes avenues assez rapidement. Les citoyens peuvent amener un point de vue que les gestionnaires de la ville n'avaient pas anticipé. De plus, dans un contexte électoral, toute information sur ce que les citoyens aiment ou aimeraient voir est importante afin de les satisfaire et augmenter ses chances d'élection ou de réélection. Pour les citoyens, cela augmente leurs chances d'avoir une ville à leur image où ils sont bien, tout en leur donnant le sentiment d'être consultés et d'avoir une certaine influence sur la prise de décision. C'est une situation gagnant-gagnant.

Les questions ouvertes peuvent aussi remplacer certaines séances de discussion organisées par la ville. Au lieu que les citoyens se déplacent à l'hôtel de ville à une heure et une date précise, qui ne leur convient pas nécessairement, ils peuvent répondre à une ou plusieurs questions ouvertes à un moment qui leur

convient mieux dans une fenêtre temporelle. Cela devrait augmenter drastiquement le taux de participation aux consultations citoyenne et ainsi donner une bien meilleure idée à la ville de l'opinion des citoyens.

Or, l'analyse d'une grande quantité de commentaires ou de réponses à une question ouverte engendre un défi de taille. Dans une grande ville comme Montréal, ou ses banlieues environnantes, il n'est pas farfelu de penser que des milliers de réponses seront compilées lorsque la plateforme B-Citi sera adoptée et utilisée par beaucoup de citoyens. De lire tous les commentaires un à un peut être une entreprise fastidieuse. L'information obtenue ne sert à rien si la ville est incapable de l'analyser et d'en tirer des enseignements, faute de temps ou de ressources. C'est pourquoi B-Citi souhaite automatiser l'analyse du texte dans son module de consultation citoyenne. Cela permettra d'obtenir le pouls de la population beaucoup plus rapidement, tout en allant chercher des idées supplémentaires.

L'analyse textuelle liée aux commentaires ou questions ouvertes se divise en deux types d'analyse : la classification de documents et l'analyse de sentiments. Des méthodes proposées dans la littérature seront testées afin de trouver une façon juste et efficace d'accomplir ces deux tâches. Toutefois, avant de se lancer dans ces tâches, le texte doit être traité. Étant donné la réalité linguistique du Québec, il faudra être en mesure d'accomplir toutes ces tâches tant en français qu'en anglais.

3.3.1 Traitement du texte

Le traitement du texte est une étape primordiale pour bien réussir une analyse automatisée du texte. Le langage naturel, le sens des mots et des phrases n'étant pas capté par une machine, il faut manipuler le texte et le transformer en ce qu'on appelle un vecteur de mots, qui permettra à l'algorithme de capter les subtilités et différences, ou ressemblances d'un commentaire à l'autre. La librairie Python NLTK (Natural Language Tool Kit) est utilisée dans ce mémoire pour une bonne partie du traitement du texte.

La première étape consiste en la création de *token* et de *stems*. Un *token* est simplement une séparation des phrases en liste de mots, séparés par une virgule et tous en minuscule. Les langages de programmation étant habituellement sensibles à la case, il est primordial que tous les mots soient en minuscule afin que l'algorithme de vectorisation ne fasse pas de différence par exemple entre un mot placé en début ou en milieu de phrase. De plus, cela permet de faciliter le travail du traitement du texte dans les étapes qui suivent. NLTK possède un *tokenizer* qui se charge de cette tâche.

La deuxième étape est le traitement des apostrophes. Cette étape diffère d'une langue à l'autre. En français, l'utilisation de l'apostrophe sert souvent à lier un déterminant à un mot commençant par une voyelle (l'aube par exemple). En anglais, elle apparaît souvent en fin de mot, pour marquer la possession ou la négation (*The man's cat* ou *he shouldn't*). Les algorithmes de vectorisation comptent par exemple *altitude* et *l'altitude* comme deux mots différents étant donné l'absence d'espace entre le déterminant et le nom. Pourtant, il est bien clair pour un humain qu'il s'agit de la même chose. La stratégie sera donc de simplement remplacer l'apostrophe par un espace et d'éliminer la lettre restante du déterminant lorsqu'on enlève dans une étape subséquente les mots vides de sens. Par exemple, *l'altitude* deviendra *l altitude*. Déjà à cette étape, l'algorithme reconnaîtra le mot *altitude* de la même façon partout dans le texte.

En anglais, on procède plutôt en remplaçant certaines combinaisons de caractères incluant une apostrophe par les mots qu'ils sous-entendent, ou un espace dépendamment du cas. Voici les trois traitements supplémentaires qui ont été pensés et appliqués dans le mémoire:

- Remplacer *'ve* par un espace et *have* (would've → would have)
- Remplacer *n't* par une espace et *not* (wouldn't → would not)
- Remplacer les apostrophes de possession par un espace (park's → park s)

Les librairies n'ont pas nécessairement une fonction ou une option permettant de faire ce travail automatiquement, il faut donc ajouter ces fonctions de remplacement manuellement dans le code à cette étape pour améliorer la performance du traitement du texte.

La troisième étape est celle où l'on enlève les mots sémantiquement vides. On pense ici aux déterminants, pronoms et marqueurs de relation qui n'amènent rien au niveau du contenu. NLTK a une liste de ces mots dans plusieurs langues, dont le français et l'anglais. On peut alors retirer tous les mots des commentaires qui se retrouvent sur cette liste. Cependant, la liste disponible dans NLTK est quelque peu insatisfaisante, laissant certains mots inutiles dans les commentaires, et ce dans les deux langues. Pour pallier ce problème, deux nouvelles listes de mots sémantiquement vides ont été créées, une pour chaque langue, à partir de celles fournies par NLTK. Cette nouvelle liste comporte plusieurs ajouts de mots jugés non pertinents dans le cadre d'une consultation citoyenne. Par exemple, en anglais, toutes les lettres seules résultant du retrait des apostrophes ont été ajoutées. On peut penser au « *s* » issu des marqueurs de possession à l'étape précédente du traitement de l'apostrophe, comme dans *park's*, qui devient *park s*. Le *s* étant dans la liste à exclure, il nous reste que le mot *park*, ce qui est très souhaitable dans notre cas.

Cette liste sera modifiable par les villes afin qu'elles puissent l'adapter aux diverses questions et applications possibles. Il est important de faire cette étape avant celle de la racinisation à l'étape suivante, car l'inverse compliquerait grandement le retrait des mots sémantiquement vides.

La quatrième étape est la création de *stems*. Un *stem* est un mot dont on enlève tous les suffixes afin d'en garder uniquement la racine, processus appelé racinisation. Par exemple, *promènerais* et *promenades* donneront tous deux la racine *promen*, ce qui est désirable, car les deux mots expriment essentiellement la même idée. Il faut donc que l'algorithme comprenne que ces deux mots portent le même sens dans le texte. Cela est très utile, surtout en français avec les multiples conjugaisons et accords possibles des mots. NLTK offre un *stemmer* tant en anglais qu'en français.

La cinquième étape du traitement du texte est celle de la vectorisation et de la création de la matrice TF-IDF. Comme mentionné précédemment, c'est une bonne façon de convertir les mots en vecteurs et de leur attribuer un poids relatant l'importance du mot quant à son pouvoir de différenciation (Weiss, Indurkhya et Zhang, 2010). Rappelons que le poids prend une valeur allant de 0 à 1, 0 étant pour un mot très peu important et 1 un mot très important pour définir un document ou un commentaire dans le contexte d'une consultation citoyenne. La valeur du poids TF-IDF diffère pour un même mot d'un commentaire à l'autre vu que la portion TF n'est pas constante. La librairie Python SK Learn possède une fonction TF-IDF Vectorizer qui permet assez facilement de créer un vecteur de mots et une matrice TF-IDF à partir d'un corpus de textes. Afin de bien illustrer son fonctionnement, considérons les trois phrases suivantes :

0. Ceci est une phrase sans animal.

1. Celle-ci comporte un chat.

2. Ici, le chat est avec un autre chat, un chat de race. Le chat n'est plus seul.

Dans cet exemple, chaque phrase correspond à un document ou commentaire différent dans le corpus. La matrice TF-IDF, après les cinq étapes de traitement du texte, retient un vecteur de 7 mots : animal, phrase, chat, comporte, ici, race, seul. Comme on peut le voir dans les termes restants, certains ne s'y trouvent plus. Cela est dû aux mots à exclure. Par exemple, il ne reste que deux mots à la première phrase (document 0) : animal et phrase. Le tableau 2 présente la matrice TF-IDF de ce petit corpus.

(Phrase, Mot)	Longueur du document après traitement	Fréquence brute dans le document	TF	IDF	poids TF-IDF
(0, animal)	2	1	0,5	0,477	0,239
(0, phrase)	2	1	0,5	0,477	0,239
(1, comporte)	2	1	0,5	0,477	0,239
(1, chat)	2	1	0,5	0,176	0,088
(2, chat)	7	4	0,57	0,176	0,100
(2, ici)	7	1	0,14	0,477	0,067
(2, race)	7	1	0,14	0,477	0,067
(2, seul)	7	1	0,14	0,477	0,067

Tableau 2- Matrice TF-IDF de l'exemple afin d'en illustrer le fonctionnement

Si on s'attarde sur le mot animal, on voit qu'il est présent dans la phrase 0 une seule fois. Le TF devient donc ½ = 0,5 selon la formule de la section 2.1. Il n'est pas dans les autres phrases, la valeur de IDF est donc égale à log(3/1)=0,477. Le poids TF-IDF qui serait calculé par SK Learn différerait de celui dans le tableau 2, car il utilise une formule quelque peu différente de celle présentée dans la section 2.1 pour des raisons de codage et d'efficacité. L'algorithme ajoute 1 au IDF (IDF=log(*n/ni*)+1, où n= le nombre de documents et *ni* le nombre de documents contenant le mot *i*). Cela permet d'éviter des multiplications par 0 dans les calculs. Dans un cas où un mot est dans tous les documents, on aurait log(1) qui donnerait une valeur de 0 au IDF et au poids TF-IDF du mot dans le document. Par défaut, les mots qui ne sont pas présents dans un document ont un poids TF-IDF de 0 et n'apparaissent pas dans la matrice. Malgré ces différences de calcul, le principe est le même : les mots discriminants ont un poids plus élevé par rapport aux autres. Dans l'exemple, le mot animal a un poids plus élevé, car il est présent dans une seule phrase. Cela le rend discriminant et permet de différencier cette phrase des autres. De plus, le mot chat a un poids plus élevé dans la phrase 2, vu qu'il est présent quatre fois, contre une seule fois dans la phrase 1. Les mots ici, race et seul sont présents que dans une seule phrase, mais celle-ci est plus longue, ce qui réduit leurs poids.

Le mot auquel l'identifiant fait référence dans la matrice TF-IDF est sous forme de racine pour les raisons énoncées dans la section 2.1. Rappelons que la racine regroupe les différents types de mots référant au

même concept, comme promenade, un nom, et promener, un verbe. Il est cependant possible d'obtenir la liste des mots retenus en racines lisibles pour un humain ainsi que la forme de la matrice. Par exemple, une forme (191, 138) indique la présence de 191 documents et de 138 mots (racines) importants. Les 138 mots sont identifiés par un chiffre, mais il est possible de retrouver la racine du mot associé à cet identifiant numérique assez facilement grâce au *TF-IDF Vectorizer* de Python. On pourrait par exemple trouver que le mot à l'identifiant 23 est la racine *promen*, donc le mot promenade. Un mot jugé important en est un qui sera discriminant, qui permettra de bien différencier les commentaires. C'est donc ce qui reste après le filtre des mots à exclure.

Il est possible de filtrer davantage si jugé nécessaire. La fonction *TF-IDF Vectorizer* de la librairie SK Learn peut prendre plusieurs paramètres intéressants en entrée afin d'améliorer la qualité des termes retenus. Parmi ceux-ci on retrouve le min et max df (fréquence dans le document), soit la proportion maximale ou minimale des documents comprenant un terme en particulier. Si on sort de cette étendue, le terme sera exclu. Par exemple, si un mot se retrouve dans plus de 95% des documents, mais n'est pas dans la liste des mots exclus, on peut l'exclure en indiquant un max df de 0,95. À l'inverse, un mot qui est présent dans moins de 1% des documents peut être exclu en indiquant un min df de 0,01. Dans le cas d'une consultation citoyenne, si on veut s'assurer que plusieurs personnes commentent un sujet particulier, on peut utiliser le min df. On s'assure donc de ne garder que des idées partagées par plusieurs personnes, donc présentes dans plusieurs commentaires, et non celles mises de l'avant par une infime minorité. Par contre, cela peut facilement taire une idée pensée par une seule personne, mais qui est hautement innovatrice. Il faut donc faire attention lorsque l'on utilise cette fonction, car il y a bien des cas où elle nuit plus qu'elle n'aide.

Une fois toutes ces étapes accomplies, il ne reste plus qu'à choisir comment déterminer les faits saillants de ces commentaires. On présentera deux méthodes pour y arriver, la classification et l'analyse de regroupements. Par la suite, des méthodes d'analyse de sentiments seront aussi présentées.

Correcteur de fautes

Un autre ajout pertinent pour la plateforme développée par B-Citi serait un correcteur de fautes. On cherchera ici à corriger les fautes communes en remplaçant le mot mal orthographié par sa bonne version. Les fautes amènent un problème similaire à l'accord des pluriels dans la classification des textes. Par exemple, la racinisation est une façon que la machine comprenne que *promenade* et *promenades* sont la même chose. Il faut donc trouver aussi une façon qu'un algorithme comprenne qu'une erreur comme *promnade* réfère aussi au même concept. Idéalement, il s'agirait d'un correcteur similaire à ce que l'on

trouve sur Word ou pour la suggestion de mots d'un téléphone intelligent. Il devra donc prédire le mot qui a voulu être écrit à partir de ce qui a réellement été écrit. Une solution a été envisagée où on itérait sur chaque mot dans les commentaires pour voir s'il se trouvait dans le dictionnaire. S'il n'y était pas, on le remplace par le mot qui lui ressemble le plus. Cependant, après avoir constaté le temps de calcul et le manque de précision de cette méthode, elle a été mise de côté afin de prioriser d'autres projets plus pressants.

Un autre aspect de la correction à considérer est celui des mots où deux orthographes sont acceptées. Par exemple, la langue française accepte deux orthographes pour le mot *évènement*, soit cette dernière ou encore *événement.* Cela amène le même problème où la machine pense qu'il s'agit de deux mots différents. Il faudra donc choisir une des deux orthographes, et, lorsque l'autre est présente, la remplacer par la première façon d'écrire le mot. Un système similaire à ce qui a été discuté dans la section 2.4 pourrait bien répondre aux besoins de B-Citi, car elle nécessite très peu de supervision et donne des résultats assez satisfaisants.

Traduction

Un autre ajout futur qui serait intéressant dans le traitement du texte serait un traducteur. Cela permettrait de réunir sous un seul bloc les réponses en français et celles en anglais afin qu'elles puissent toutes être comparées ensemble. Cela permettrait de simplifier l'analyse, au lieu d'avoir des analyses séparées dans chacune des deux langues, et permettrait d'agrandir les échantillons pour améliorer la performance des algorithmes utilisés. La méthode discutée en 2.5 pourrait bien s'appliquer ici, même que le corpus pourrait rapidement être bonifié par quelques traductions humaines des commentaires.

De plus, il est possible d'aller plus loin en laissant les citoyens écrire dans plusieurs autres langues et de traduire automatiquement vers l'anglais ou le français pour analyser ces réponses. En effet, avec la réalité linguistique et culturelle du Québec, principalement dans la région de Montréal, il existe beaucoup de citoyens qui écrivent beaucoup mieux dans une langue autre que le français et l'anglais. Par contre, leur opinion n'en est pas moins intéressante. Une plateforme qui leur permettrait de s'exprimer dans leur langue maternelle et d'être bien compris quand même pourrait grandement motiver leur motivation à être impliqués et à répondre aux consultations citoyennes, et la ville serait en meilleure posture pour répondre à leurs besoins. La méthode discutée en 2.5 serait moins applicable ici pour le moment simplement, car il serait difficile d'avoir un corpus de traductions assez grand et précis dans toutes ces langues afin qu'un réseau de neurones récurrents puisse fonctionner.

3.3.2 Classification de documents

La classification de documents sert à détecter le ou les sujets traités dans les commentaires ou réponses textuelles des citoyens. On peut aussi trouver s'il existe des thèmes récurrents dans les réponses écrites par les citoyens. La classification de documents s'applique bien lorsque la question demande aux citoyens de donner des idées ou des aspects plus ou moins appréciés de la ville. On peut penser à des questions comme « *Qu'aimez-vous faire lorsque vous allez ou parc? »* ou encore « *Que manque-t-il dans le service de transport en commun de la ville? »*

3.3.2.1 Algorithmes de classification

Comme dans les cas classiques de données structurées, les méthodes de classification de textes nécessitent la séparation des documents en deux échantillons, un d'entraînement, et un autre de test. Dans le mémoire, des données déjà classées serviront afin de tester les algorithmes. Cependant, dans le cas pratique des villes, les commentaires reçus n'auront pas de classes préalables. Il faudra donc qu'au moins une personne en classe une partie. La classification humaine servira ensuite à entraîner l'algorithme pour apprendre les patrons de classement et par la suite à classer automatiquement le reste des documents.

Comme mentionné dans Weiss et al. (2010), il est très difficile de savoir combien de documents seront nécessaires afin d'en arriver à une bonne classification. Il est très difficile de savoir combien de classes existeront. Ce sera à la discrétion de la personne responsable du classement manuel d'un échantillon, et cela dépendra aussi si les différences entre les classes sont marquées ou subtiles. Classer manuellement 50 à 100 commentaires devrait suffire dans bien des cas, mais il faut s'attendre à devoir en faire plus si on dépasse 1000 commentaires à classer. Bien évidemment, il ne faut pas s'attendre à devoir classer manuellement des centaines de commentaires à chaque question.

Cinq algorithmes de classification seront testés :

- Les arbres de décision
- Les forets aléatoires
- L'algorithme Naïve-Bayes
- Séparateurs à vaste marge (en anglais *support vector machines, SVM*)

- Les réseaux de neurones.

Dans tous les cas, les algorithmes prendront en entrée la matrice TF-IDF et la classe associée à l'échantillon d'entraînement, et tenteront de prédire la classe des commentaires en test en se fiant à la matrice TF-IDF qui leur est associée.

Deux jeux de données serviront à tester ces algorithmes, un en français provenant de l'INM de 191 commentaires, et un autre trouvé sur Kaggle où 644 documents en anglais ont été conservés, ayant tous une seule classe. Celui en français porte sur une consultation citoyenne liée à un parc montréalais (fourni par courriel par l'INM à B-Citi), et celui en anglais sur des textes de blogues[2]. Dans les deux cas, les textes sont déjà classés par sujet. La répartition des textes est présentée dans les tableaux 3 et 4. Dans les deux cas, trois classes sont dominantes, dont une qui se démarque grandement. Dans les données en français, le peu de commentaires dans les trois dernières catégories pourrait facilement nuire à la qualité de la classification. Le reste des classes ont un nombre de commentaires similaire. On mesurera le taux de bonne classification afin de trouver l'algorithme qui performe le mieux. La librairie Python SK Learn offre une version de chacun des cinq algorithmes mentionnés ci-dessus. Puisque la taille des échantillons de ces deux exemples est relativement petite, une validation croisée de type *k-fold* a été utilisée, avec une valeur de k égale à 10. Le principe est de créer 10 sous-échantillons dans le jeu de données, chacun d'entre eux servant une fois de données tests prédites à partir des algorithmes de classification entraînés avec les 9 autres échantillons. On utilisera la moyenne des 10 taux de bonne classification comme mesure d'évaluation de la performance de l'algorithme.

[2] Disponible sur https://www.kaggle.com/wpncrh/marginal-revolution-blog-post-data

Catégorie	Nombre de commentaires
Books	167
Data	42
Games	38
History	50
Education	148
Law	107
Music	45
Philo	47

Tableau 3- Résumé des données en anglais. On peut y voir une forte asymétrie dans la distribution des textes.

Catégorie	Nombre de commentaires
Promenade	74
Activités culturelles	41
Sport/jeux	40
Détente	13
Nature	12
Sécurité	11

Tableau 4- Résumé des données en français. On peut y voir une forte asymétrie dans la distribution des commentaires. Les trois dernières catégories peuvent potentiellement être très difficiles à identifier par un algorithme vu leur petit nombre d'occurrences.

Avant de présenter les grandes lignes de chacun des cinq algorithmes utilisés, on peut observer lesquels ont le mieux performé dans le tableau 5. Naïve-Bayes est l'algorithme ayant le mieux performé sur les données en français. Par contre, on remarquera plus tard que cela est en grande partie dû au fait que l'algorithme a classé tous les commentaires dans les trois catégories les plus populaires, augmentant ainsi grandement ses chances de bien classer dans l'ensemble. Pour l'exemple avec les commentaires en anglais, le SVM a été le plus efficace suivi de près par le réseau de neurones. Dans la plupart des cas, la qualité des résultats diverge grandement d'une langue à l'autre.

Algorithme	Taux de bonne classification moyen (Moyenne de la validation croisée K-fold, k=10)	
	français (n=191)	**anglais** (n=644)
Naive-Bayes	59,48%	51,22%
Séparateur à vaste marge	40,86%	61,97%
Réseau de neurones	46,66%	59,67%
Arbre de décision	38,79%	51,95%
Forêt aléatoire	44,83%	53,69%

Tableau 5- Résultats selon l'algorithme de classification ainsi que les données utilisées.

Le tableau 6 est un exemple de matrice TF-IDF avec le premier commentaire des données de l'INM. La matrice TF-IDF issue de la validation croisée a retenu 71 mots différents. Dans tous les cas, une matrice TF-IDF est construite à partir des commentaires inclus dans chaque itération du *k-fold*. Dans le cas des données de l'INM, chaque commentaire dans les données de test aura donc une valeur TF-IDF pour les 71 mots relevés.

(Id commentaire, Mot)	Poids TF-IDF
(0, promen)	0,217709
(0, patin)	0,273148
(0, aller)	0,616985
(0, asseoir)	0,295055
(0, lir)	0,255655
(0, banc)	0,324385
(0, écout)	0,295055
(0, oiseau)	0,324385

Tableau 6- Matrice TF-IDF du premier commentaire des données de l'INM. On y trouve l'identifiant du commentaire et le mot (raciné) dans la première colonne, et la deuxième colonne indique la valeur du poids TF-IDF pour ce mot dans le commentaire en question. On peut voir que le mot ayant le plus de poids dans ce commentaire est le mot aller. Ce commentaire est de la catégorie Promenade.

Arbre de décision

L'arbre de décision est un algorithme de classification qui a l'avantage d'être facile à comprendre lorsqu'on observe ses résultats. L'arbre de décision considère la relation entre chaque caractéristique d'une observation et son influence sur la variable à prédire (Dahan et al., 2014). Dans le cas présent, on pense donc à l'influence qu'a chaque combinaison TF-IDF sur la classe du document. L'algorithme classe donc les différents documents selon la probabilité qu'ils appartiennent à une classe selon la combinaison d'attributs constituant le texte. On peut utiliser le critère de Gini comme mesure de la qualité des partitions à chaque nœud de l'arbre. La figure 11 donne un exemple abrégé avec les données en français de l'INM pour un échantillon de 172 commentaires sur les 191; les 19 autres commentaires sont pour l'un des 10 sous-ensembles tests de la validation croisée.

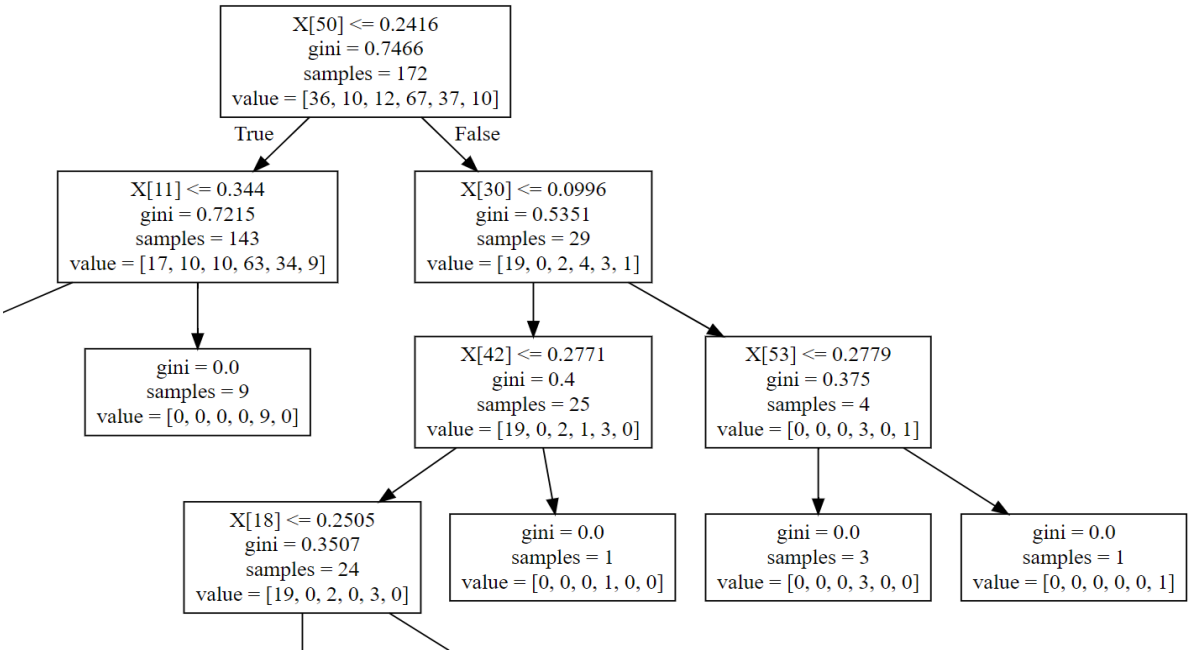

Figure 11- Aperçu d'une partie d'un arbre de décision obtenu pour classer les commentaires en français de l'exemple de l'INM. Cet arbre correspond à un exemple k parmi les k-fold.

L'arbre se lit de haut en bas. Au haut de l'arbre, on peut observer que l'arbre de décision utilise la variable X50 en premier, qui correspond au mot *théâtr* (racine de théâtre) dans la matrice TF-IDF. L'arbre commence avec l'attribut ou mot avec le plus grand gain informationnel selon le critère de Gini et continue de cette façon à chaque nœud enfant. Dans le cas où la valeur du poids TF-IDF n'est pas plus petite ou égale à 0,2416, on se dirige vers la droite de l'arbre. Il y a 29 commentaires sur les 172 qui ont

répondu à ce critère (samples=29 au nœud suivant). Au nœud suivant, on observe la variable X30, soit le mot *march* (racine de marcher). Si la valeur du poids TF-IDF des commentaires dans ce nœud dépasse 0,0996, le terme qui est sélectionné dans le nœud enfant pour la prochaine étape est X53, *verdur* (racine de verdure). Si la valeur du poids TF-IDF de X53 dépasse 0,2779, les commentaires se voient assigner la 6e catégorie dans cette feuille terminale, soit sports/jeux. Il y a un seul commentaire dans cette feuille terminale parmi les 172 commentaires.

Forêts aléatoires

Les forêts aléatoires consistent essentiellement à classer les observations plusieurs fois avec plusieurs arbres de décision en utilisant des sous-échantillons différents à chaque fois pour construire les arbres, et d'ensuite choisir le mode ou la moyenne de toutes les classifications ou prédictions comme classification ou prédiction finale (Breiman, 2001). Les sous-échantillons créés sont généralement tous de la même taille que le fichier d'entraînement, avec la particularité d'utiliser la technique d'échantillonnage bootstrap, paramètre par défaut de SK Learn. Cette technique rééchantillonne avec remise les données à partir du jeu de données d'entraînement. À chaque nœud, un sous-ensemble de mots de la matrice TF-IDF est sélectionné aléatoirement. Le mot de ce sous-ensemble avec le plus grand gain d'information est ensuite sélectionné pour donner les deux nœuds enfants. La forêt aléatoire utilisée ici comportait 10 arbres, soit la valeur par défaut de SK Learn, et la moyenne des prédictions des arbres a été utilisée pour la classification finale.

Naïve-Bayes

Pour l'algorithme Naïve-Bayes, la fonction Multinomial NB de SK Learn est utilisée. Cette fonction se sert habituellement de nombres entiers, mais elle fonctionne aussi bien avec des comptes de mots fractionnaires comme dans le cas d'une matrice TF-IDF. L'algorithme Naïve-Bayes est très populaire dans les problèmes de classification de textes. Il s'agit d'une méthode probabiliste (Manning, Raghavan et Schütze, 2008). Cette méthode est dite naïve, car elle suppose l'indépendance des variables explicatives entre elles étant donné la classe. C'est donc dire que la présence d'une caractéristique pour une observation donnée n'influence pas la probabilité que d'autres caractéristiques soient présentes (Rish, 2001). Dans le cas de la classification de textes par exemple, on fait l'hypothèse que le fait qu'un mot soit

présent dans un texte ou commentaire n'influencerait donc pas la probabilité qu'un autre y soit. Par exemple, de voir le mot pomme apparaître ne fait pas augmenter les chances de voir le mot rouge selon cette hypothèse. C'est une hypothèse qui dans bien des cas peut sembler irréaliste, mais qui historiquement donne des résultats intéressants, pouvant souvent rivaliser avec des algorithmes plus sophistiqués (Rish, 2001).

Pour expliquer brièvement comment fonctionne cette méthode, désignons par C= {1, …, m} l'ensemble des m catégories dans lesquelles sont classés les documents d'un corpus, et X_i= {X_{i1}, …, X_{ip}} l'ensemble des valeurs des p variables explicatives pour le document i dans le corpus, i=1,…,n. Dans notre contexte, X_i correspond au vecteur des poids TF-IDF des p mots de la matrice TF-IDF associé au document i. Le classificateur Naïve-Bayes applique le théorème de Bayes de la façon suivante pour obtenir la probabilité que le document i appartienne à la catégorie c (Manning et al., 2008; Ray, 2017a; Rish, 2001):

$$P(C_i = c|X_i = x) = \frac{P(X_i=x|C_i=c)P(C_i=c)}{P(X_i=x)} \ .$$

En supposant naïvement l'indépendance conditionnelle des p variables explicatives, nous avons

$$P(X_i = x|C_i = c) = P(X_{i1} = x_1|C_i = c) \times P(X_{i2} = x_2|C_i = c) \times \cdots \times \left(X_{ip} = x_p\middle|C_i = c\right).$$

Puisque P(X_i= *x*) est identique pour toutes les catégories c= 1,…, m, la catégorie prédite pour le document X_i= *x* est donc donnée par :

$$\hat{c} = \underset{c}{argmax}\, P(X_{i1} = x_1|C_i = c) \times \cdots \times \left(X_{ip} = x_p\middle|C_i = c\right) \times P(C_i = c) \ .$$

La probabilité *P(C_i = c)* dans la dernière équation est simplement estimée par la proportion de documents appartenant à la catégorie *c* sur l'ensemble des documents dans l'échantillon d'entraînement. De plus, la fonction Multinomial NB de SK Learn estime la probabilité *(P X_{ij}= x_j | C_i= c)*, j=1,…,p, par le ratio N_{cj}/N_c, qui correspond à l'estimateur du maximum de vraisemblance, où N_{cj} est égale au nombre de documents dans la classe c qui contient le mot x_j , et N_c = (N_{c1} + N_{c2} + … + N_{cp}). Cependant, cette formule donne la possibilité de multiplier par 0 si un mot n'est pas présent dans une catégorie. C'est donc dire qu'un seul mot pourrait annuler la possibilité qu'un commentaire soit classé dans une catégorie c indépendamment des autres mots du commentaire. Pour pallier ce problème, on utilise la technique du *Laplace Smoothing*, qui consiste à additionner un à N_{cj}, j=1,…,p (Manning, Raghavan et Schütze, 2008).

Dans un contexte où le système de poids TF-IDF est utilisé au lieu de simplement relater de la présence ou non d'un mot pour une observation, les probabilités sont calculées en fonction des valeurs TF-IDF prises par chaque mot dans les différentes catégories c.

Cet algorithme a l'avantage de ne pas avoir besoin d'un échantillon d'entraînement trop grand pour donner des résultats satisfaisants. De plus, il exécute ses calculs très rapidement.

Séparateurs à vaste marge

Les séparateurs à vaste marge (support vector machines en anglais), couramment appelés SVM, sont une méthode de régression ou de classification supervisée. Son principe de base est plutôt simple : dans le cas d'une variable dépendante binaire et de deux variables explicatives, il s'agit de trouver comment séparer les deux classes, sur un plan à deux dimensions, de façon à maximiser la distance entre la droite qui sépare les deux classes et les points les plus près de cette droite pour chaque classe. Ces points sont en fait les vecteurs de support (Berwick, 2003; Kay, 2017b). La figure 12 est un exemple de séparation effectuée par un SVM.

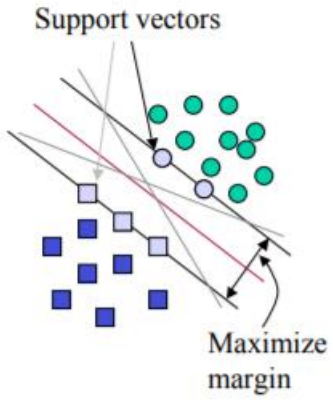

Figure 12- Image tirée de Berwick (2003). On voit la ligne centrale qui sépare les deux classes, ronds et carrés, en maximisant l'écart par rapport aux deux classes. Les vecteurs de supports sont déterminés par les points les plus près de cette séparation.

Cette droite de séparation sert en fait de point de décision pour la classification. En effet, les valeurs pour les deux variables indépendantes placées sur un plan à deux dimensions déterminent de quel côté de la droite se retrouve chacune des observations. Essentiellement, la droite de décision est calculée sur un

échantillon d'entraînement, et les nouveaux points représentant les nouvelles observations sont classés en fonction de leur position par rapport à cette droite.

Cependant, qu'en est-il lorsque la classification ne peut se faire de façon linéaire ? La méthode du SVM suggère de modifier les données afin de transformer un problème non linéaire en un problème linéaire, qui sera ensuite facile de séparer linéairement (Berwick, 2003; Kay, 2017b). Si on reprend un plan à deux dimensions, la figure 13 illustre un exemple où il n'est pas possible de séparer parfaitement les deux classes par une droite.

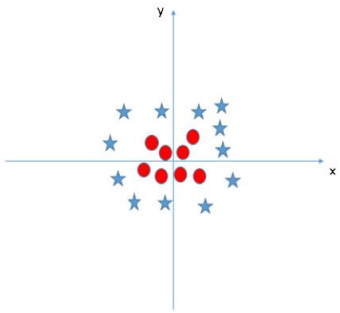

Figure 13- Image tirée de Ray (2017b). Les données ne peuvent être séparées efficacement de façon linéaire.

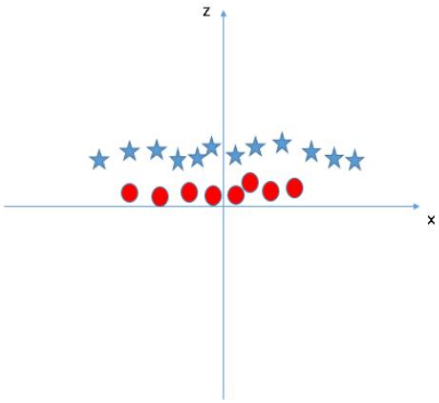

Figure 14- Image tirée de Ray (2017b). Les données sont maintenant facilement séparables linéairement grâce à l'introduction de la dimension z, qui est représentée sur l'axe vertical, et la transformation $z= x^2 + y^2$.

Cependant, si on ajoute une nouvelle dimension z au plan xy, où $z= x^2+y^2$, cette transformation permet en quelque sorte de réorienter le problème de classification de façon linéaire, tel qu'illustré à la figure 14. Cela témoigne donc de la flexibilité de la méthode SVM. De façon générale, cette méthode consiste à utiliser une transformation des données originales pour les projeter dans un espace de dimension supérieure où il sera possible de séparer les classes avec un hyperplan. Bien qu'il soit difficile pour le cerveau humain de se le représenter, il est effectivement possible de trouver une transformation et un

séparateur linéaire lorsqu'il y a plus de deux dimensions. Par exemple, s'il y a trois dimensions ou plus, au lieu d'une droite, ce sera un hyperplan qui agira comme séparateur.

Il existe quelques fonctions de transformation, communément appelées fonctions à noyau (*kernel functions*), qui sont plus fréquemment utilisées et disponibles dans Scikit Learn pour la méthode SVM (Berwick, 2003; Ray, 2017b).

Étant donné que les SVM sont généralement des algorithmes de classification binaires, l'algorithme utilise la technique du *one-vs-one* afin de classer les observations lorsqu'il y a plusieurs catégories à classer. Cette technique consiste à créer un classificateur SVM pour chacune des combinaisons de deux classes possibles (Scikit Learn 1, 2017). S'il y a cinq classes, il y aura donc 10 combinaisons possibles. Chaque classificateur se spécialise pour sa combinaison de classes. La classe ayant été choisie le plus souvent sera celle qui sera choisie au final.

Cette méthode fonctionne bien lorsqu'il y a plusieurs variables indépendantes (dimensions), comme dans le cas présent, où il y a une dimension par mot. Les valeurs TF-IDF permettent alors de placer chaque document dans l'espace. L'algorithme calcule l'hyperplan servant à la décision et peut ensuite classer les documents dont la classe est inconnue.

Réseaux de neurones

Les réseaux de neurones s'inspirent de ce que nous connaissons du fonctionnement des neurones du cerveau humain pour l'apprentissage supervisé. On utilise ici le perceptron multicouche qui est organisé de sorte que l'information circule d'une couche d'entrée vers une couche de sortie (Tang et al., 2007). Chaque couche cachée entre la couche d'entrée et la couche de sortie comporte un nombre de neurones pouvant varier. L'algorithme utilise une fonction de combinaison des valeurs des variables à l'entrée d'un neurone et y applique une fonction d'activation qui est généralement une transformation non linéaire. La valeur obtenue est passée aux neurones de la couche suivante pour éventuellement aboutir à une prédiction à la couche de sortie. Comme avec les autres algorithmes, le réseau de neurones prend en entrée la matrice TF-IDF issue des commentaires traités. Ce sont ces valeurs qui seront passées dans la première couche du réseau de neurones et ensuite elles seront transformées pour obtenir en sortie des prédictions. Dans le cas présent, le réseau de neurones a comme fonction d'activation l'unité de

rectification linéaire (*Rectified Linear Unit - ReLU*) et comme fonction de combinaison une combinaison linéaire des entrées. Le nombre d'itérations maximales est de 400, ce qui est suffisant pour permettre la convergence du modèle pour ce problème. On utilise la structure de base de SK Learn, soit une seule couche cachée de 100 neurones.

Analyse des résultats

Pour comparer les différents algorithmes dans leurs résultats, il peut être utile d'observer un cas particulier. Le tableau 7 présente la matrice TF-IDF pour un commentaire du jeu de données. Le commentaire était : *J'aime bien y aller pour me promener à vélo. C'est bien aussi pour un pique-nique près des gens qui déambulent, et aussi pour jouer à la pétanque.* Le tableau 8 présente la classification des différents algorithmes pour ce commentaire.

Mot	Valeur TF-IDF
Vélo	0,386078
Pétanqu	0,485499
Promen	0,358229
Pique-n	0,449452
Déambul	0,53376

Tableau 7- Matrice TF-IDF pour un commentaire de l'échantillon de test. 5 des 71 mots retenus à l'entraînement se retrouvent dans ce commentaire.

Algorithme	Classement
Arbre	Sports/Jeux
Forêt	Sports/Jeux
Naïve-Bayes	Promenade
Séparateurs à vaste marge	Promenade
Réseau de neurones	Promenade

Tableau 8-Classification des cinq algorithmes pour le commentaire du tableau 7.

Comme on peut le voir au tableau 8, la classification promenade a été choisie par trois des cinq algorithmes. Cela est grandement dû aux fortes valeurs TF-IDF des racines *promen* et *déambul.* Or, l'arbre

et la forêt ont choisi la catégorie sports et jeux, ce qui est aussi plausible vu la présence de racines comme *vélo* et *pétanqu.*

Au final, dans les conditions de ces tests, l'algorithme ayant le mieux performé sur l'échantillon français est de loin Naïve-Bayes. Le fait que l'algorithme nécessite moins de données dans l'échantillon d'entraînement que les autres algorithmes permet d'expliquer en partie ce phénomène.

Sur l'échantillon anglais, la meilleure classification a été obtenue avec SVM. Cet algorithme a été le seul à obtenir un taux de bonne classification au-delà de 60%. Le réseau de neurones était tout près à 59,67%.

Ces résultats sont très peu satisfaisants. En effet, arriver difficilement à un taux de classification de 60% n'est pas suffisant pour que quiconque puisse tirer des conclusions justes avec ces classifications. Cela peut être dû au fait qu'il a été supposé ici que chaque commentaire ou texte comportait un seul sujet ou classe. Or, il est fort probable, surtout si les textes sont plus longs, que les documents en question renferment deux classes ou plus encore. Ce n'est pas une information qui est captée ici. Un document se fait classer dans une catégorie, et toute autre information pouvant se rapporter à une autre est donc absente lorsqu'on regarde les classes finales. Ce n'est pas une situation qui reflète nécessairement bien la réalité des commentaires. D'ailleurs, les deux classifications dans l'exemple présenté aux tableaux 7 et 8 sont acceptables.

Dans un cas de consultation citoyenne, il est fort probable que les citoyens expriment plusieurs idées différentes dans leurs commentaires. Une autre méthode sera donc présentée afin de tenir compte de cette réalité, soit l'analyse de regroupement. Il aurait été également possible d'opter pour une méthode basée sur les thèmes des documents, soit l'allocation de Dirichlet latente (LDA en anglais). Très sommairement, il s'agit d'une méthode probabiliste générative qui itère sur une collection de documents. Chaque document est défini par un ensemble de mots, qui eux-mêmes font partie du vocabulaire de certains thèmes. La méthode LDA est donc en mesure de facilement identifier plus d'un thème qui probablement définit un document (Yu et al., 2017). C'est une méthode qui pourrait potentiellement être appliquée au cas ici.

3.3.2.2 Analyse de regroupement

Le but de l'analyse de regroupement (en anglais *clustering*) est de créer des classes de documents où les documents d'une même classe se ressemblent le plus possible, et diffèrent le plus possible des documents

des autres classes. Dans le cas présent, on utilisera le même traitement préalable du texte que lors de la classification de documents. Les regroupements se feront donc à partir d'une matrice TF-IDF. Les regroupements seront constitués de commentaires comportant les mêmes mots et ayant un poids TF-IDF similaire. C'est donc dire qu'un même regroupement comportera des commentaires où l'importance accordée à certains mots est similaire. Des commentaires de classes différentes peuvent facilement avoir quelques mots avec des valeurs TF-IDF similaires. Ce sont les autres mots du regroupement qui seront différents, chaque classe présentant donc des combinaisons de thèmes différentes. Par exemple, les commentaires regroupés dans une classe x peuvent avoir des valeurs TF-IDF similaires pour les mots promenade, pique-nique et baignade, alors que les commentaires regroupés dans la classe y auront des TF-IDF similaires pour les mots promenade, yoga et bruit. Le mot promenade revient dans les deux, mais ce sont les autres idées soulevées dans les commentaires qui différencient les deux classes. Cette méthode nous permet donc de capter plus d'une seule idée exprimée par les citoyens. On peut ainsi plus facilement identifier un segment qui partage une multitude d'idées. Qui plus est, cette méthode ne nécessite pas de classification humaine au préalable, elle est entièrement non supervisée. La méthode sera testée encore une fois sur les données de l'INM en français, et sur celles de blogues en anglais.

Il existe plusieurs algorithmes qui permettent de regrouper les commentaires. Les deux qui ont été essayés sont le K-Means et la classification hiérarchique selon le critère de Ward, ce dernier ayant été retenu, car il offrait des résultats similaires au K-Means en un peu moins de temps. Cela est un peu hors-norme, mais peut s'expliquer par la petite taille de l'échantillon et la nature itérative du K-Means. Dans le contexte de développement d'une plateforme pour les villes, ce critère s'est avéré important. Ensuite, la méthode du *Silhouette Score* permet de déterminer combien de classes permettront de maximiser la distance entre les classes et de minimiser celle à l'intérieur des classes. Le *Silhouette Score* utilise la distance euclidienne. Le nombre de classes possibles a été limité à 20 pour des raisons pratiques concernant l'analyse des résultats. La librairie SK Learn contient encore tout le nécessaire pour accomplir cette tâche.

On commence le test avec les données de l'INM, en français. Les commentaires que ces données contiennent sont en réponse à la question suivante : Qu'aimez-vous faire au parc *X* ? À titre indicatif, le tableau 9 résume les données, ainsi que les prédictions faites par l'algorithme Naïve-Bayes à la section 3.3.2.1.

Catégorie	Nombre de commentaires	Proportion réelle	Proportion prédite
Promenade	74	38,74%	54,31%
Activités culturelles	41	21,47%	25%
Sport/jeux	40	20,94%	20,69%
Détente	13	6,8%	0
Nature	12	6,28%	0
Sécurité	11	5,76%	0

Tableau 9- Répartition des données de l'INM selon leur classification initiale et le résultat des prédictions de l'algorithme Naïve-Bayes

L'INM avait donc relevé 6 catégories pour identifier les 191 commentaires, donnant une seule catégorie à chaque commentaire. Notre analyse de regroupement avec Ward obtient son meilleur regroupement à 19 classes. En théorie, c'est donc dire qu'il y a 19 combinaisons différentes de thèmes relevés par les citoyens qui sont assez différentes les unes des autres pour justifier qu'on les sépare. Il est probable qu'un autre nombre de regroupements donne de bons résultats, mais il est plus pratique de choisir automatiquement le nombre de regroupements optimal selon le *Silhouette Score* afin d'automatiser le plus possible le processus. Ces 19 regroupements indiquent que la classification permet de relever une grande variété de commentaires différents, exprimant probablement une multitude d'idées. L'algorithme de Naïve-Bayes, bien qu'il ait obtenu un bon taux de classification, n'a identifié que 3 des catégories dans ses prédictions, ce qui fait que l'on ignore un bon nombre d'idées.

Avant de décortiquer ce qui se retrouve dans les classes, il faut penser à faciliter la compréhension des résultats pour les gestionnaires des villes. Si on affiche les mots contenus dans les classes tels quels, ils devront comprendre les racines des mots vu que les classes sont générées à partir de la matrice TF-IDF qui a nécessité une racinisation des mots. Or, il est de loin préférable de présenter les résultats avec des mots complets qui se comprennent en un coup d'œil. Il y a moyen de remédier à ça.

Premièrement, il faut récupérer les commentaires complets, non traités. On leur assigne le regroupement auquel chaque commentaire appartient.

Deuxièmement, il faut traiter le texte, sans toutefois utiliser une technique de racinisation. Une technique de lemmatisation sera donc privilégiée. C'est une technique très similaire à la racinisation, mais qui au lieu de couper un mot jusqu'à sa racine, consiste à ramener un mot à sa forme la plus simple, généralement sa forme canonique qui existe dans le dictionnaire. Or, les outils de traitement du texte trouvés en Python ne fournissent pas de lemmatisation en français. Une solution à l'interne a été nécessaire. Avec l'aide d'un employé de B-Citi, un dictionnaire pour la lemmatisation a été développé à partir de ses propres travaux sur l'analyse du texte réalisés antérieurement. Le dictionnaire a pour objectif d'enlever tous les accords et conjugaisons dans les mots et ramener le tout à une forme de base, soit l'infinitif ou masculin singulier. Si le mot ne figure pas dans le dictionnaire, on le laisse tel quel. Le dictionnaire est facilement modifiable afin de rajouter des mots pouvant manquer au dictionnaire. Le tableau 10 est un exemple de la structure du dictionnaire :

Base	**Formes**
Promener	Promène
Promener	Promèneront
Promenade	Promenades
Retarder	Retardées

Tableau 10- Exemple de la structure du dictionnaire de lemmatisation. Chaque fois qu'un commentaire contient un mot de la colonne de droite, on le remplace par son équivalent de la colonne de gauche.

Troisièmement, on enlève les mots sémantiquement vides, comme on le faisait avant. Il est important de faire cette étape après la lemmatisation, afin que toutes les variations des mots sémantiquement vides soient vues comme un seul et même mot. On peut penser ici au verbe être qui est un mot important à retirer.

On peut maintenant aller voir ce qui se trouve dans les classes. Il existe deux représentations assez communes pour afficher un résumé des mots clés de chaque regroupement aux gestionnaires de la ville, une représentation avec un graphique classique (figure 10), et l'autre avec un nuage de mots (figure 11).

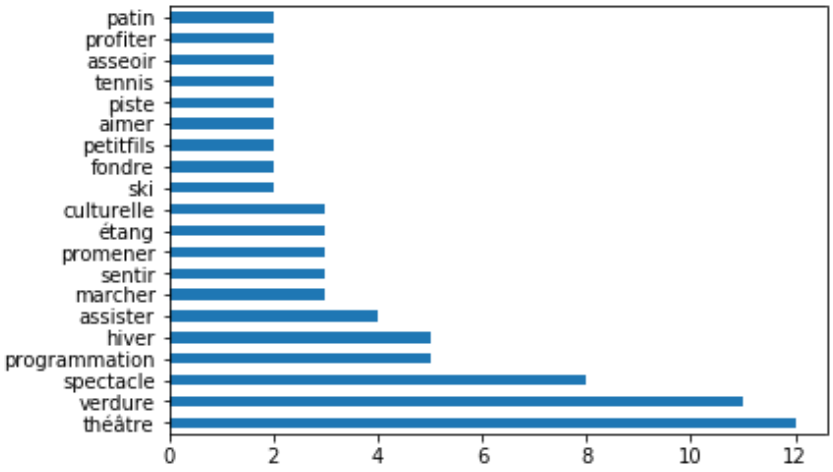

Figure 15- Sur l'axe vertical, on a les différents mots que l'on retrouve dans les commentaires du regroupement. Sur l'axe horizontal, on trouve le nombre de fois que le mot est apparu dans le regroupement.

En regardant la figure 15, on remarque rapidement que ces gens ont exprimé un fort désir d'aller au parc *X* pour des spectacles de théâtre. En plus du mot théâtre, on remarque les mots spectacle, programmation assister et culturelle, qui s'insèrent tous dans le champ lexical du théâtre. En second lieu, on remarque le mot verdure, marcher et étang, qui témoignent bien des raisons qu'ont les gens de fréquenter ce parc. On remarque aussi que la plupart des mots sont significatifs dans le sens qu'ils peuvent répondre à la question. Cela est signe que la liste des mots à exclure fait bien son travail.

Comme mentionné, il est aussi possible de présenter aux gestionnaires ces résultats sous forme de nuage de mots, fonctionnalité disponible dans la librairie Python WordCloud. Cette représentation est plus visuelle et tape-à-l'œil, mais ne permet pas de voir l'écart exact entre divers mots. La figure 16 en est un exemple.

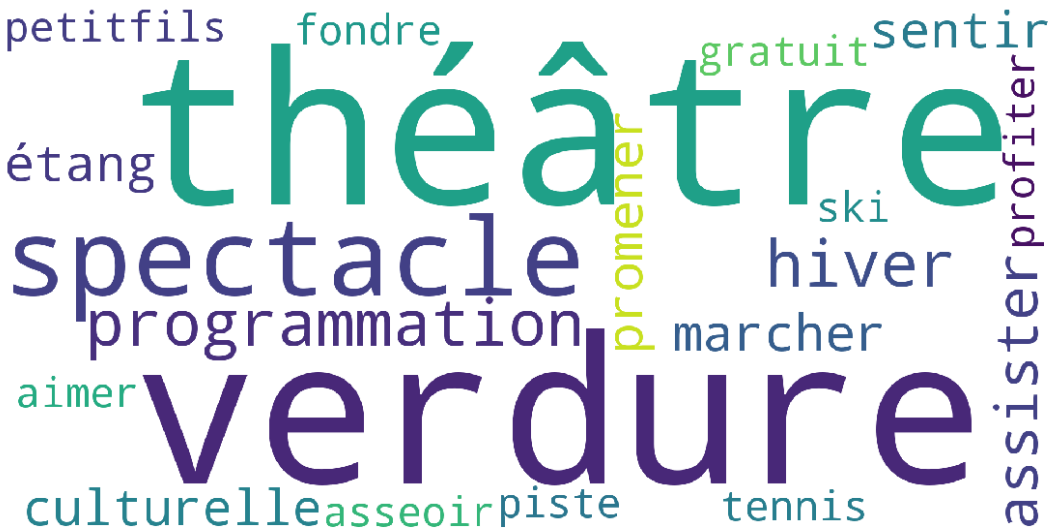

Figure 16- Nuage de mots pour le même regroupement que la figure 10. Les mots écrits en caractère plus gros sont présents plus souvent dans les commentaires. Cependant, s'ils dépassent grandement les autres mots, ils ne seront pas écrits plus gros encore, c'est plutôt une question de rang.

À titre comparatif, la figure 17 présente les résultats d'un autre regroupement.

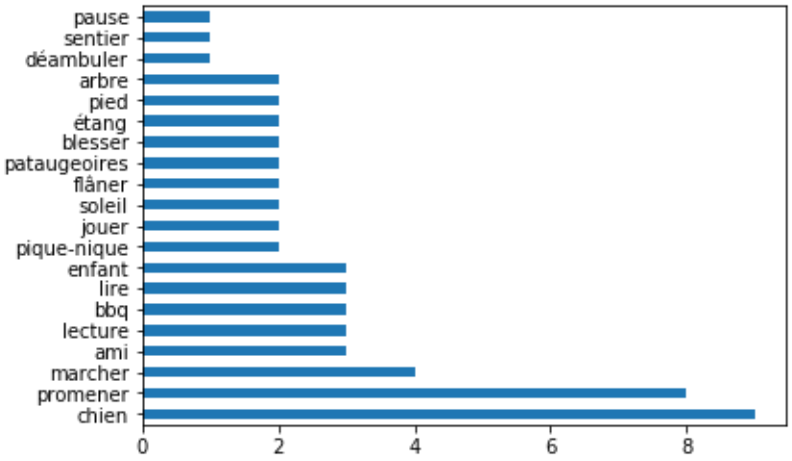

Figure 17- Résultats d'un autre regroupement

Bien qu'il soit pratique de voir les résultats groupés, la première chose que le gestionnaire voudra voir est probablement une vue d'ensemble. Afin de donner de l'importance aux diverses idées mentionnées dans les commentaires, un système de poids sera privilégié au lieu de prendre un simple compte de mots brut.

Pour ce faire, les cinq mots les plus populaires de chaque groupe sont choisis et seront traités comme les thèmes du groupe. Après plusieurs tests, il a été jugé que trois mots laissaient de côté trop d'information,

et que plus de cinq n'était pas pertinent vu que l'on attribue des poids proportionnels au rang des mots. Ces mots sont en ordre d'importance, le thème 1 est le mot le plus populaire, le 2 le 2^{e} et ainsi de suite. On a donc créé un tableau de six colonnes, une par thème, et une dernière avec le nombre de commentaires dans ce groupe. Le tableau 11 compte une ligne par regroupement.

	T1	T2	T3	T4	T5	nombre de commentaires
0	théâtre	verdure	spectacle	programmation	hiver	15
1	théâtre	spectacle	verdure	assister	marcher	19
2	marcher	vélo	regarder	canard	fontaine	21
3	promener	théâtre	libre	vélo	danser	10
4	activité	espace	café	culturelles	déambuler	14
5	vélo	patin	traverser	pique-nique	lire	18
6	promener	vélo	seul	nature	gens	9
7	course	pique-nique	yoga	pied	lecture	7
8	profiter	verdure	bruit	théâtre	ami	7
9	promener	hiver	jouer	observer	étang	6
10	installation	étang	planter	passer	animaux	17
11	asseoir	écouter	promener	musicien	banc	10
12	gens	rencontrer	marcher	pique-nique	verdure	4
13	chien	promener	marcher	ami	lecture	9
14	jeu	air	aire	enfant	hiver	6
15	théâtre	verdure	banc	détente	piste	6
16	pratiquer	slackline	montréal	personne	randonnée	3
17	public	grouper	cour	danser	projection	3
18	promener	chien	laisser	pétanque	détente	7

Tableau 11- Exemple du tableau des résultats. Une colonne par thème du regroupement, en ordre d'importance ainsi qu'une colonne indiquant le nombre de personnes dont le commentaire se situe dans ce groupe.

Certains groupes peuvent sembler similaires comme les groupes 13 et 18. Le groupe 18 a probablement été séparé du reste étant donné qu'il est le seul à mentionner la pétanque ce qui le différentie bien. Cependant, il est clair que le nombre de groupes est probablement trop élevé, il serait donc envisageable de forcer un nombre plus petits de regroupements afin de simplifier l'analyse des résultats. Par ailleurs, il pourrait être intéressant d'utiliser un dictionnaire des synonymes afin d'éviter que des mots comme promener et marcher soient vus comme étant différents alors qu'ils réfèrent à la même activité. Cela contribuerait également à réduire le nombre de regroupements.

Ensuite, on transforme ce tableau en tableau codé en binaire, relatant de la présence d'un mot dans la colonne T1, T2, T3 et ainsi de suite comme présenté au tableau 12. Cela permet d'obtenir plus facilement les résultats tels que présentés aux figures 18, 19 et 20.

Regroupement	T1_Théâtre	T2_Verdure	T3_Verdure	T2_Spectacle	…	T5_Hiver	T5_Marcher
0	1	1	0	0	…	1	0
1	1	0	1	1		0	1

Tableau 12- Exemple abrégé du tableau codé en binaire pour les regroupements 0 et 1. Le fait que ces regroupements ont le théâtre comme premier thème est représenté par le chiffre 1 dans la colonne T1_Théâtre. Il existe une colonne pour chaque combinaison de rang de thème et thème présents dans le tableau 11.

L'analyse de regroupement permet de facilement lier divers thèmes ensemble. En effet, ces regroupements nous permettent de trouver des combinaisons de thèmes qui sont populaires auprès des citoyens. Étant donné que les citoyens sont divisés en groupes, il est donc possible de voir quels groupes comportent différentes combinaisons. La figure 18 montre un graphique intéressant pour analyser ces combinaisons.

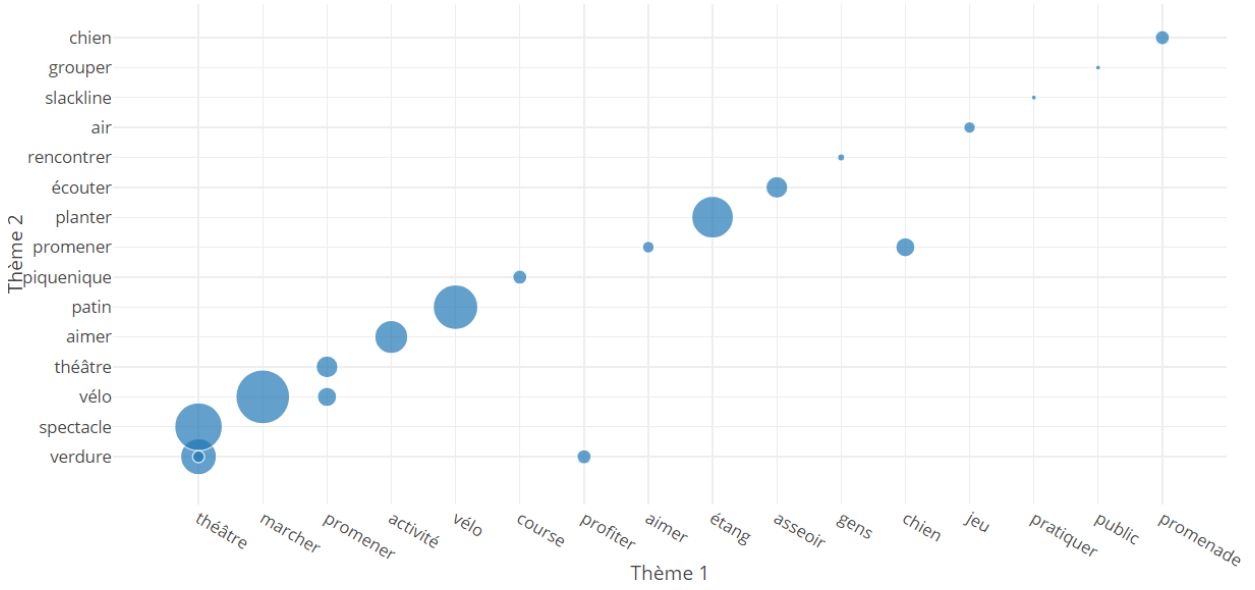

Figure 18- Figure présentant les combinaisons différentes pour les thèmes 1 et 2 de tous les regroupements.

Grâce à cette représentation, on peut voir quels thèmes appartiennent à un même regroupement. La taille des bulles est proportionnelle au nombre de personnes dans le regroupement. Si l'on regarde vers

la gauche par exemple, on peut voir qu'un gros regroupement de gens a choisi marcher comme thème principal, combiné avec vélo comme deuxième. On peut voir aussi que promener a été choisi deux fois comme premier thème, une fois avec vélo, et une autre avec théâtre. Cela peut, dans un premier temps, aider à orienter la lecture des commentaires si jamais un gestionnaire veut aller voir avec plus de précision ce qui est dit dans un cluster. Il est à noter qu'il est possible d'utiliser les thèmes trois, quatre et cinq afin de raffiner l'analyse.

Ces observations permettent aussi aux gestionnaires de prendre des décisions de deux façons différentes. Supposons qu'on veuille aménager un nouveau parc à partir de ces résultats. Si l'on prend le plus gros groupe au centre, qui a pour thème principal l'étang, on peut voir qu'il vaut la peine d'aménager un étang dans le nouveau parc. Or, le deuxième thème de ce groupe est planter. Si l'on plante des arbres comme deuxième aménagement du parc, on est certain de satisfaire grandement ce groupe de personnes. Or, il est possible que le gestionnaire de la ville préfère satisfaire plus de personnes, il pourra donc opter pour un aménagement qui conviendra à un autre groupe de personnes, comme du théâtre ou une piste cyclable afin de plaire à un public plus large. Cela permet de bien gérer ses ressources dans un contexte municipal.

Il est également possible d'avoir une vue d'ensemble de ce qui s'est dit dans les commentaires. Il faut tout d'abord multiplier toutes les colonnes de thèmes par la dernière colonne, soit celle du nombre de personnes dans le groupe dans le tableau 11. De cette façon, on tiendra compte de la quantité de personnes ayant mis de l'avant ces thèmes. Finalement, on applique un poids à la valeur de chaque colonne. Étant donné qu'il y a 5 thèmes, donc 5 poids, les poids seront divisés par 15 (5+4+3+2+1), afin qu'au total ils donnent 1. On aura donc pour T1 un poids de 5/15, 4/15 pour T2, et ainsi de suite. Il ne reste qu'à additionner le tout pour obtenir le score final pour chaque mot considéré. Par exemple, si l'on prend le mot théâtre au tableau 11, il est présent en première position du regroupement 0, ce qui se traduit par la colonneT1_Théâtre qui prend la valeur de 1 au tableau 12. Son poids pour ce regroupement sera donc de 5/15. Ce groupe comporte 15 personnes, le score final pour le mot théâtre dans le regroupement 0 sera donc de 1*5/15*15=5. Au groupe 3 du tableau 11, théâtre vient en deuxième position, dans un groupe comportant 10 personnes. On aura donc un score de 1*(4/5)*10=8. On additionne le score de chaque mot dans chacun des groupes de cette façon pour en arriver à un score

total qui peut se représenter en diagramme en bâtons ou en nuage de mots, comme présenté aux figures 14 et 15.

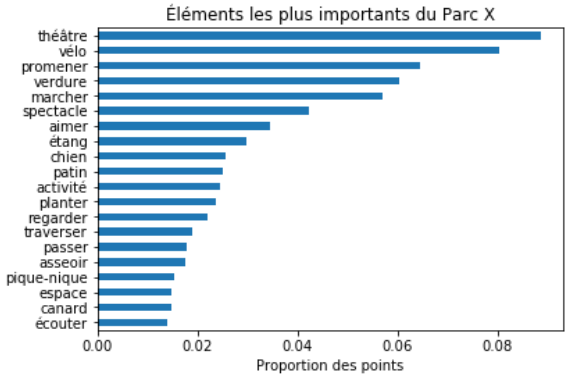

Figure 19- Graphique des résultats selon le score

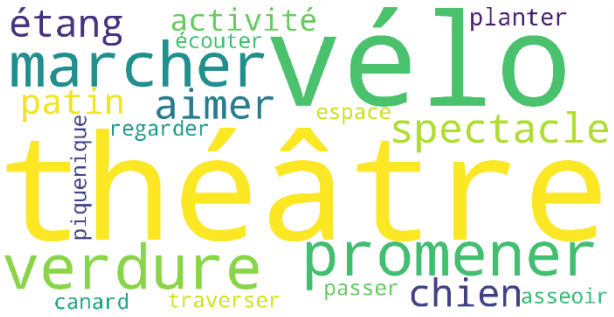

Figure 20- Nuage de mots représentant les mots ayant le score le plus élevé

On peut voir aux figures 19 et 20 que les résultats collent bien avec ce qui est présenté au tableau 4. Les catégories réelles de l'INM les plus populaires, soient la promenade, les activités culturelles et le sport sont bien présentes dans les mots les plus populaires relevés par la classification. Qui plus est, les catégories moins populaires comme la nature et la détente sont présentes, de façon un peu plus faible, ce qui s'aligne encore une fois avec les données réelles. Par ailleurs, la classification arrive bien à les identifier, contrairement à Naïve-Bayes qui n'a pas classé de commentaires dans les trois dernières catégories. Par contre, la catégorie la moins présente dans les données selon la classification, soit celle de la sécurité, n'a pas été relevée ici. Le modèle a plutôt semblé préciser les autres catégories au lieu de toutes les relever. Le système de pointage a aussi l'avantage de venir équilibrer les résultats par rapport à ce qui aurait été obtenu si l'on affichait un seul regroupement pour l'ensemble des commentaires. Cela laisse donc plus de place à la diversité des thèmes ou des idées.

De plus, dans le contexte de l'utilisation de la plateforme B-Citi, d'autres informations seront disponibles pour appuyer la prise de décision. Il sera possible de voir si certains groupes se démarquent de par les caractéristiques sociodémographiques des répondants y appartenant. Par exemple, il sera possible de voir si des groupes sont clairement composés de gens ayant un sexe particulier, un groupe d'âge différent des autres, ou encore utilisant des services de la ville particuliers, information disponible dans la base de données B-Citi. Cela donne une portée très grande à l'information qui sera disponible aux villes.

3.3.3 Analyse de sentiments

L'analyse de sentiments vise à voir si le sujet de la question posée par la ville est perçu de façon positive ou négative par les citoyens. Elle s'applique bien lorsqu'on demande aux citoyens de donner leur avis sur un sujet particulier. On peut penser à des questions comme « Qu'avez-vous pensé du service de déneigement du dernier hiver ? » ou encore « Donnez votre appréciation, en quelques phrases, du service de transport en commun. » Afin d'accomplir cette tâche, des algorithmes de classification seront utilisés, mais au lieu de simplement classer les commentaires, ils serviront à prédire la positivité des commentaires.

Positivité des commentaires

Afin de tester une technique d'analyse de sentiments dans ce mémoire, un corpus de plusieurs commentaires variés (n=676) a été bâti. Ces commentaires expriment des opinions sur plusieurs choses, comme des restaurants, des films, des services de la ville, des infrastructures ou encore des parcs. Ces commentaires ont été recueillis sur Google, ou encore écrits par l'équipe de recherche, afin de tester certains cas particuliers. Chacun des commentaires a été annoté manuellement d'un 0 s'il est négatif, et d'un 1 s'il est positif. Lorsque possible, la note associé au commentaire a été utilisée pour déterminer si un commentaire était positif, par exemple la note sur cinq étoiles des commentaires Google. Il n'y a pas de commentaires neutres dans le corpus. La stratégie sera donc de prédire si un nouveau commentaire s'ajoutant au corpus est positif ou négatif. Dans cette optique, nous afficherons aux gestionnaires de la ville la probabilité qu'un commentaire donné soit positif, ce qui sera interprété comme une mesure de la positivité du commentaire. Pour mesurer la qualité du modèle, le corpus sera divisé en un échantillon d'entraînement comportant 80% des commentaires, et un échantillon de test comportant l'autre 20%. On mesurera le taux de bonne classification binaire pour se donner une idée de sa performance. Ensuite, 50 nouveaux commentaires, hors du corpus, mais eux aussi annotés de 0 ou 1, seront utilisés pour qu'on mesure leur positivité. Environ 60% des nouveaux commentaires sont positifs. Afin de mesurer la qualité des prédictions, on regardera l'erreur moyenne de la prédiction en valeur absolue, et l'écart entre la positivité attendue et la positivité prédite. La positivité attendue est égale à la proportion de textes positifs, donc ici de 60%.

Avant de tenter de mesurer la positivité des commentaires, il faut encore une fois traiter le texte. Cela sera fait de façon très similaire à ce qui a été fait pour la classification des commentaires, mais avec quelques variantes.

Premièrement, la liste de mots à enlever sera beaucoup plus courte. On utilisera la liste de base fournie par la librairie NLTK, en y enlevant les marqueurs de négation comme *ne* et *pas* en français. En effet, l'exercice ici est différent qu'en classification de commentaires. En classification, ce sont les idées qui importent, donc surtout les noms communs et les verbes d'action comme promenade ou nager. Dans le cas de l'analyse de sentiment, ce sont plutôt les verbes de sentiments et les adjectifs qui seront importants, par exemple aimer et chaleureux. Qui plus est, les marques de négation permettent d'inverser le sentiment exprimé par ces mots.

Deuxièmement, la composition de la matrice sera différente. En effet, elle incorporera la technique des n-grams. Un n-gram, comme expliqué dans la section 2.1, est une sous séquence de mots de longueur n dans un texte. Par exemple, un bi-gram est une séquence de deux mots. Dans la phrase « On aime les chats », il existe plusieurs bi-grams : « On aime », « aime les » et « les chats ». Dans un contexte d'analyse de sentiments, il est attendu que d'utiliser les n-grams pour aller plus loin que le simple mot donne de meilleurs résultats (Tripathy, Agrawal et Rath 2016). En effet, cela permet de tenir compte des structures négatives dans les phrases. Par exemple, un bi-gram du bout de phrase « n'est pas bon » aura « n'est », « est pas » et « pas bon ». Cela permettra de mieux saisir les sentiments qui prennent plus d'un mot à exprimer. Le paramètre *n-gram range* du *TF-IDF Vectorizer* utilisé précédemment sera utilisé ici. Il prend en entrée une paire de chiffres, soit les différents n-grams qu'on veut utiliser. Dans notre cas, on lui demandera (1,2), donc tous mots seuls et tous les bi-grams dans les commentaires. Aller plus loin que les bi-grams ajouterait beaucoup de termes qui ne seraient pas forcément pertinents. Pour revenir à l'exemple précédent, un tri-gram donnerait « On aime les » et « aime les chats ». Ces 3-grams sont plus spécifiques et donc seraient plus nombreux dans un corpus de commentaires, ayant donc rarement un TF-IDF élevé. Utiliser des n-grams de trois mots ou plus ralentirait donc le processus sans ajouter beaucoup plus de précision au modèle. La matrice TF-IDF de l'exemple sera donc composée des mots seuls et des bi-grams dans les termes qui seront utilisés. Le reste ne change pas, le poids du TF-IDF se calcule de la même façon, sauf que les fréquences des bi-grams se calculent lorsque le bi-gram complet est trouvé dans le commentaire.

Finalement, l'ensemble des termes ne sera pas retenu pour le modèle de prédiction. En effet, en ayant les mots simples en plus des bi-grams donnent une liste très longue de termes pouvant être utilisés pour

prédire. On utilisera donc les termes ayant une fréquence plus élevée à travers le corpus de commentaires. Le paramètre *max features* du *TF-IDF Vectorizer* permettra d'accomplir cette tâche. Cela améliore grandement la qualité du modèle; il s'agit du même principe que celui d'enlever des variables expliquant très peu un phénomène en régression linéaire. Le modèle présenté ici aura 6000 termes, car c'est un bon compromis entre le nombre de termes utiles et le temps de calcul à grande échelle.

Une fois que la matrice TF-IDF est bâtie, on peut passer à la prédiction. Pour prédire, on utilise un réseau de neurones de six couches cachées comptant 33 neurones chacune. Cette structure a été choisie après plusieurs essais allant de 3 à 6 couches et de 20 à 100 neurones par couches. Les réseaux de neurones sont souvent suggérés dans ce type de prédiction, et cette structure s'est avérée la plus efficace, c'est-à-dire qu'elle a maximisé le taux de bonne classification tout en ayant la capacité de donner une plus forte probabilité qu'un commentaire soit positif s'il contient des mots qui sont sans aucun doute positifs. L'algorithme Naïve-bayes a été testé pour comparer, mais performait un peu moins bien. La fonction d'activation est la fonction logistique, étant donné la nature binaire de la variable à prédire. La fonction de combinaison est la combinaison linéaire.

Le modèle devra être en mesure de fournir des probabilités nuancées (autour de 50%), car certains commentaires sont plus neutres dans les commentaires de test, et il y en aura certainement dans les vrais commentaires des citoyens plus tard. C'est une information qui devra être capturée. L'échantillon d'entrainement ne comptait que très peu de ce genre de commentaires. C'est aussi une des raisons pourquoi on choisit de présenter les probabilités comme résultat, et non une décision de classification. Cette probabilité sera appelée positivité. Il faut éviter un modèle qui place toutes ses probabilités près des deux pôles, soient 0 et 1. Cela permet de capter correctement les commentaires qui sont plus nuancés. Il y a quelques commentaires dans ceux qui ont servi de test pour lesquels le réseau de neurones a attribué une probabilité qu'il soit positif autour de 50%. La plupart d'entre eux étaient effectivement des commentaires plus neutres, avec quelques exceptions où c'étaient des commentaires qui ont été plutôt mal identifiés.

Comme mentionné précédemment, le corpus de test était composé de 60% de commentaires positifs. C'est donc le score de positivité attendu. La moyenne de la probabilité calculée par le réseau de neurones sur l'échantillon de test est de 64%, ce qui est très près de ce qui est attendu. En donnant une valeur de 0.5 aux commentaires neutres, si l'on mesure l'écart entre la probabilité calculée et la valeur réelle du commentaire (0, 0,5 ou 1), on obtient une erreur absolue moyenne de 8%. Ce résultat est aussi satisfaisant, car il n'y a aucune mesure de positivité qui est exactement la classification préalable du

commentaire. Par exemple, l'erreur minimale dans ce test est de 2,5%, pour un commentaire négatif dont la probabilité qu'il soit positif a été mesurée à 2,5%.

L'idée dans l'application de B-Citi, sera d'utiliser ce même réseau de neurones dans toutes les consultations citoyennes. C'est pourquoi il est important que ce corpus soit le plus varié possible, afin de toucher plusieurs cas ou formulations qui sont très difficiles à anticiper. Si les gestionnaires de la ville le souhaitent, ils peuvent classer certains commentaires eux-mêmes, en termes binaires et non en termes de positivité, afin d'entraîner un nouveau réseau de neurones avec des commentaires de leur propre consultation en plus de ceux du corpus, mais ce n'est pas essentiel. Ces commentaires classés humainement pourraient toutefois s'ajouter au corpus qui a été construit et le rendre plus complet.

4. Choix et combinaison des méthodes pour la plateforme B-Citi

Maintenant que les méthodes de présentation des résultats de questions ouvertes ont été présentées, il faut maintenant choisir celles qui conviendront le mieux au contexte de consultations citoyennes. Pour bien choisir, il est bon de faire un rappel des objectifs du projet :

- Rendre au minimum le travail d'analyse et de traitement que les gestionnaires des villes auront à faire.
- S'assurer de l'exactitude des calculs et des représentations graphiques, et rendre le tout automatisé.
- On doit faire l'hypothèse que le niveau de connaissances en statistique et en analyse de données des gestionnaires des villes est relativement faible, le tout devra donc être le plus simple possible. Vulgariser est la clé.
- Pouvoir traiter tous les types de questions mis en place par la plateforme B-Citi. La solution devra être flexible aux différentes consultations possibles.

4.1 Classification des commentaires

Étant donné le critère de minimiser le travail des gestionnaires des villes, la méthode de regroupement des commentaires a été choisie par B-Citi, car elle donne des résultats satisfaisants, interprétable et sans

intervention nécessaire humaine à aucune étape du processus. La classification nécessite qu'on classe une portion des commentaires manuellement avant de lancer l'algorithme, ce qui répond moins aux demandes des clients de B-Citi. En effet, si une ville collecte des centaines, voire des milliers de commentaires, la tâche de classification manuelle devient très fastidieuse. Il faut en classer beaucoup pour espérer arriver à de bons résultats. De plus, l'exemple de la section 3.3.2.2 montre que l'analyse de regroupement fait un meilleur travail pour montrer la variété d'idées exprimées par les citoyens.

Par ailleurs, les résultats des algorithmes de classification n'ont pas été des plus concluants dans les deux exemples présentés dans la section 3.3.2 et qui ont servi de test dans le mémoire. Il aurait cependant été possible et même préférable de permettre la classification dans plusieurs classes, mais étant donné que le problème d'intervention humaine persiste, cette avenue n'a pas été explorée. Par contre, on présente ici une suggestion qui pourrait être utilisée dans une version future de la plateforme de B-Citi.

4.1.1 Combiner les méthodes de regroupement et de classification

L'idée proposée est de combiner l'analyse de regroupement et la classification. Ici, l'analyse de regroupement vient remplacer l'intervention humaine dans la classification, c'est donc elle qui classera un premier groupe de commentaires dans différents regroupements. L'analyse de regroupement va donc créer n classes avec ces premiers commentaires et les distribuer dans chacune des classes. Ensuite, les commentaires restants seront classés dans ces mêmes classes avec un algorithme de classification tel NaÏve-Bayes ou un réseau de neurones qui se sera entraîné avec les résultats de l'analyse de regroupement.

En utilisant le jeu de données de Kaggle (n=644), 400 commentaires ont été choisis au hasard afin de procéder au regroupement. Ces commentaires ont permis à un réseau de neurones d'apprendre ce qui constitue un texte d'une classe x par rapport à un texte d'une classe y. Le réseau de neurones en question a ensuite classé les commentaires restants en fonction de ce qu'il a appris. Il est structuré de la même façon que celui qui a servi à la classification de la section 3.2.2. De façon similaire à lorsqu'une simple analyse de regroupement est effectuée sur l'ensemble des données, le *Silhouette Score* trouve un nombre optimal de regroupements pertinents. C'est donc 13 catégories dans ce cas-ci pour lesquelles il sera possible pour le réseau de neurones de classer les commentaires. Le tableau 13 présente de quelle façon se répartissent les 644 commentaires dans les regroupements créés. Comme on peut le voir, certains

groupes ont très peu d'observations étant donné le grand nombre de catégories. Il est donc difficile pour le réseau de neurones de bien apprendre à classer étant donné le petit nombre d'observations par catégorie dans ce qui lui sert d'entraînement. Cette méthode gagnerait donc en efficacité dans un cas où le nombre de commentaires est important. En effet, elle performe plus rapidement qu'une analyse de regroupement sur l'ensemble des données.

No. du regroupement	Nombre de commentaires
10	214
4	206
2	53
3	26
11	26
1	23
12	23
5	17
9	14
0	12
8	12
7	10
6	8

Tableau 13- Nombre de documents par classes

Afin de voir si ces résultats collent avec la classification réelle des documents, il est possible de voir comment les documents par catégorie initiale se répartissent dans les 13 regroupements créés. C'est ce qui est présenté au tableau 14.

Catégorie	Groupe	Nombre	Proportion	Catégorie	Groupe	Nombre	Proportion
Books	0	3	1%	Education	0	3	2%
	1	3	1%		1	3	2%
	2	49	16%		2	1	1%
	4	67	23%		3	24	16%
	5	130	44%		4	37	25%
	6	6	2%		5	7	5%
	7	2	1%		10	43	29%
	8	1	0%		11	25	17%
	10	32	11%		12	5	3%
	11	1	0%	Law	0	1	1%
	12	3	1%		1	16	15%
DataSource	0	4	10%		3	1	1%
	3	1	2%		4	52	49%
	4	13	31%		5	4	4%
	7	3	7%		7	3	3%
	10	14	33%		8	1	1%
	12	7	17%		10	26	24%
Games	4	7	18%		12	3	3%
	5	2	5%	Music	2	2	4%
	9	13	34%		4	4	9%
	10	14	37%		5	2	4%
	12	2	5%		6	2	4%
History	0	1	2%		7	1	2%
	2	1	2%		8	9	20%
	4	20	40%		10	25	56%
	8	1	2%	Philosophy	1	1	2%
	9	1	2%		4	6	13%
	10	24	48%		5	2	4%
	12	2	4%		7	1	2%
					10	36	77%
					12	1	2%

Tableau 14-Tableau résumant la distribution des documents dans les différents groupes (cluster) selon leur classification initiale (label). La colonne proportion est la proportion de documents de la classe initiale se retrouvant dans un regroupement donné.

Comme on peut le voir, la méthode de classification a créé plus de classes qu'il n'en existait selon la classification initiale. La plupart des classes initiales semblent se retrouver majoritairement dans deux regroupements, avec le reste de leurs documents qui se répartissent un peu dans d'autres groupes. Par exemple, 67% des documents classés initialement dans *Books* se retrouvent dans les groupes 4 et 5,

laissant donc seulement 33% à répartir dans les 11 autres. Cette classe étant de loin la plus populaire, on en retrouve au moins une observation dans chaque regroupement. D'ailleurs, une classe livre est en fait très vaste comme classification et peut facilement regrouper plusieurs sous-thèmes. Il n'est pas difficile de s'imaginer que certains documents classés dans *Books* auraient pu se retrouver dans *Education* ou d'autres classes. Certaines classes ont été plus difficiles à regrouper, comme la classe *Education*. En effet, le regroupement ayant la plus forte proportion de documents de cette classe en compte moins de 30%.

4.2 Analyse de sentiments

Une seule méthode a été présentée pour l'analyse de sentiments, c'est donc elle que B-Citi utilisera sur sa plateforme. Il est toutefois possible de venir enrichir l'information qu'elle donne, c'est ce qui sera présenté ici.

4.2.1 Combiner analyse de sentiments et analyse de regroupement

Comme le nom l'indique, cette méthode consiste à combiner l'analyse de sentiments à l'analyse de regroupement. L'idée est de répondre aux questions suivantes : « *Qu'est-ce qui s'est dit dans les commentaires positifs ou négatifs ? »* ou encore *« Est-ce que les commentaires positifs et négatifs traitent des mêmes sujets? ».*

Le principe est simple. On applique l'analyse de sentiments telle que présentée à la fin du chapitre 3. Ensuite, on sépare les résultats en deux groupes en fonction des résultats de l'analyse de sentiments, les positifs, par exemple les commentaires ayant une positivité plus grande que 66%, et les négatifs, ayant une positivité plus petite que 33%. Une fois ces deux groupes créés, on applique une analyse de regroupement, encore une fois comme celle du chapitre 3. On aura donc, parmi les deux sous-groupes positif et négatif, des regroupements de textes similaires. On peut ensuite voir quels sujets ont été les plus populaires dans les deux sous-groupes et ainsi comparer. C'est une information qui peut être très intéressante pour les gestionnaires de la ville; ils sauront ainsi quels sont les facteurs ou sujets qui ont une tendance positive, et quels autres ont une tendance négative.

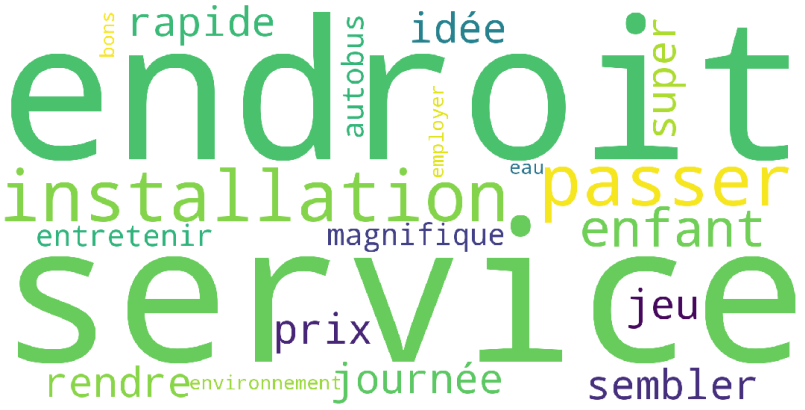

Figure 21- Nuage de mots avec les mots ressortant des commentaires positifs

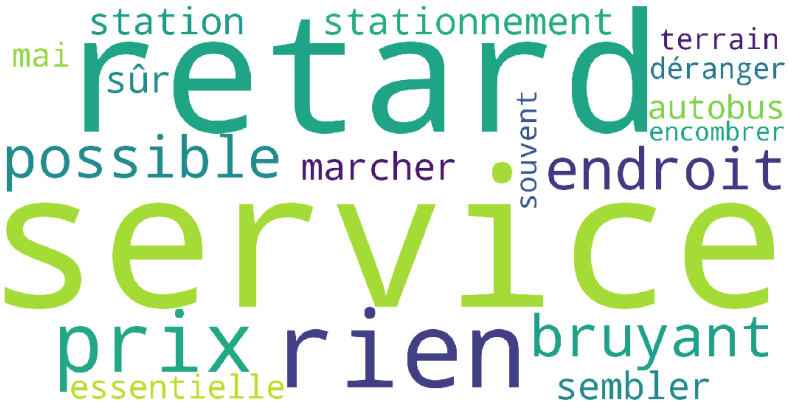

Figure 22-Nuage de mots avec les mots ressortant des commentaires négatifs

Les figures 21 et 22 sont issues du corpus de 676 commentaires ayant servi à illustrer l'analyse de sentiments dans la section 3.3.3. Comme on peut le voir, le service offre des opinions partagées, étant donné qu'il se retrouve comme un mot important dans les deux groupes. Les retards semblent être un grand problème pour les commentaires négatifs, alors que l'endroit en général et les installations semblent provoquer des réactions positives. Avec ces informations, il est facile de voir ce qui doit être amélioré, et ce qui fonctionne. C'est donc un pas plus loin que de simplement savoir si les commentaires sur un sujet donné sont positifs ou négatifs.

5. Conclusion

Pour conclure, il est bon de revenir sur les apprentissages et avancements que ce projet a permis, tant ceux qui sont personnels que ceux qui sont plus pour l'ensemble des domaines de l'analyse de données textuelles et de la consultation citoyenne.

Ce projet chez B-Citi a permis de prendre les notions de statistique, d'analyse de données et d'intelligence d'affaires telles qu'elles sont connues dans le domaine, et de les appliquer dans un contexte où les parties prenantes n'ont pas nécessairement ces connaissances. De plus, il s'est concentré sur un domaine de l'intelligence d'affaires qui est un peu moins exploité que d'autres, soit celui de l'analyse textuelle. Ce fut donc un important exercice de vulgarisation et de simplification, et cette considération a été le guide de tous les choix qui ont été faits. Il fallait limiter le travail de toutes les parties prenantes, tout en montrant des résultats clairs, pertinents et simples de compréhension.

De plus, ce projet a permis de tester les avancées récentes dans domaine de l'analyse de texte, et surtout, de les appliquer dans un contexte où il y a plus d'une langue à considérer. On a pu voir que l'anglais et le français nécessitent quelques ajustements différents au niveau du traitement du texte. Les marques de ponctuation par exemple ne jouent pas le même rôle d'une langue à l'autre. Avec la réalité linguistique des villes du Québec, il faut que les solutions proposées soient aussi efficaces en anglais qu'en français. Cela a demandé un effort supplémentaire que si l'anglais seul était nécessaire, car les solutions et la documentation disponibles librement sur Internet sont majoritairement faites pour l'anglais.

Finalement, le projet a permis d'explorer deux types d'analyse de textes nécessitant tous deux un travail quelque peu différent. En effet, il y a deux aspects des réponses à une question ouverte qui ont été considérés : ce qui est dit, donc le sujet, et comment c'est dit, donc la positivité. Étant donné que ce ne sont pas les mêmes mots qui nous donnent ces deux types d'information, forcément le travail d'analyse doit être différent. C'est ce qui a été accompli ici en utilisant des listes de mots à exclure spécifiques à chacune des analyses, en utilisant des algorithmes et une méthodologie propre à chaque problème, et en utilisant les mots seuls ou les bi-grams lorsque le cas le nécessitait. Par exemple, l'analyse de sentiments, donc de la positivité des commentaires, avait beaucoup à gagner avec l'utilisation des bi-grams, entre autres à cause de l'importance des marques de négation combinées avec les mots auxquels elles sont associées.

Bibliographie

Baer P., Blessing C., Capponi E., Cukier J., Duff K., Flanders J., Flannery C., Gardner J., Grenier M., Grossenbacher A., Marder D., Meyer K., Mitton T., St. John E., Schulz T. et Wanders A. (2009). Making Data Meaningful Part 2: A guide to presenting statistics. United Nations Economic Commission for Europe. Récupéré de http://www.unece.org/fileadmin/DAM/stats/documents/writing/MDM_Part2_English.pdf, le 7 février 2018.

Berwick R. (2003). An Idiot's Guide to Support Vector Machines (SVMs). Récupéré de http://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf, le 11 novembre 2018.

Bing L. (2015). *Sentiment Analysis Mining Opinions, Sentiments, and Emotions,* Cambridge University Press.

Bird S., E. Klein et E. Loper (2009). *Natural Language Processing with Python,* O'Reilly Media.

Breiman, L. (2001). Random forests*. Machine Learning, 45*(1), 5-32.

Dahan H., S. Cohen, L. Rokach et O. Maimon (2014). *Proactive Data Mining Using Decision Trees. In: Proactive Data Mining with Decision Trees*, Springer, New York.

Dubé C. (2017). Prenez le contrôle de votre ville! *L'actualité*, récupéré de http://lactualite.com/societe/2017/09/13/prenez-le-controle-de-votre-ville/ .

Ghiassi M., J. Skinner et D. Zimbra (2013). Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with Applications*, *40*(16), 6266-6282.

Grant J. (2016). *City of Pittsburgh Launches 'Burgh's eye view' application*. Récupéré de http://pittsburgh.cbslocal.com/2016/10/31/city-of-pittsburgh-launches-burghs-eye-view-application/ le 20 octobre 2017.

Hu M. et B. Liu (2004). Mining and Summarizing Customer Reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168-177.

Manevitz L. et M. Yousef (2007). One-class document classification via Neural Networks, *Neurocomputing,* 1466-1481.

Manning C., P. Raghavan et H. Schütze (2008). Text classification and Naive Bayes, *Introduction to Information Retrieval,* (pp. 234-265), Cambridge: Cambridge University Press.

Martineau J. et T. Finin (2009). Delta TFIDF: An Improved Feature Space for Sentiment Analysis. Récupéré de https://ebiquity.umbc.edu/_file_directory_/papers/446.pdf le 20 juin 2018.

Massaron L. (2018). Create and Train Machine Translation Systems, *TensorFlow Deep Learning Projects,* Packt Publishing, 320 p.

Medium (2016). Machine Learning is Fun Part 5: Language Translation with Deep Learning and the Magic of Sequences. Récupéré de https://medium.com/@ageitgey/machine-learning-is-fun-part-5-language-translation-with-deep-learning-and-the-magic-of-sequences-2ace0acca0aa le 1er aout 2018.

Norvig P. (2007). *How to Write a Spelling Corrector.* Récupéré de http://norvig.com/spell-correct.html le 31 juillet 2018.

Qualtrics (2015). *5 Easy Tips to Write an Effective Survey.* Récupéré de https://www.qualtrics.com/experience-management/research/survey-writing-tips/ le 19 juin 2018.

Ray S. (2017a). 6 Easy Steps to Learn Naive Bayes Algorithm (with codes in Python and R), *Analyticsvidhya*. Récupéré de https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained le 9 novembre 2018.

Ray S. (2017b). Understanding Support Vector Machine algorithm from examples (along with code), *Analyticsvidhya.* Récupéré de https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/ le 11 novembre 2018.

Rish I. (2001). An Empirical Study of the Naïve Bayes, *IBM*. Récupéré de https://www.researchgate.net/publication/228845263_An_Empirical_Study_of_the_Naive_Bayes_Classifier le 8 novembre 2018.

Rousseeuw P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics,* 20(1), 53-65.

Scikit Learn 1 (2017). *OneVsOneClassifier,* Scikit Learn. Récupéré de https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsOneClassifier.html# le 13 novembre 2018.

Scikit Learn 2 (2017). *Neural network models (supervised),* Scikit Learn. Récupéré de http://scikit-learn.org/stable/modules/neural_networks_supervised.html le 15 août 2018.

Taboada M. et al. (2011), Lexicon-based methods for sentiment analysis*, Computational Linguistics*, 37(2), 267-307. Récupéré de http://www.aclweb.org/anthology/J11-2001 le 26 mai 2018.

Tang H., K. Tan et Z. Yi (2007). *Neural Networks: Computational Models and Applications,* Springer-Verlag, Berlin Heidelberg.

Tripathy A., A. Agrawal et S. Kumar Rath (2016). Classification of sentiment reviews using n-gram machine learning approach*, Expert Systems with Applications,* 57, 117-126.

Weiss S.M., N. Indurkhya et T. Zhang (2010). *Fundamentals of predictive text mining,* Springer-Verlag, London Limited.

Whitelaw C., B. Hutchinson, G. Chung et G. Ellis (2009). Using the Web for Language Independent Spellchecking and Autocorrection*, Proceeding EMNLP '09 Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 2, 890-899.

Yu L., C. Zhang, Y. Shao et B. Cui (2007). LDA : a robust and large-scale topic modeling system, *Proceedings of the VLDB Endowment,* 1406-1417.

Annexe 1- Mots à exclure en français

Voici les mots de la liste d'exclusion en français. Les mots en gras sont des ajouts qui ne figuraient pas sur la liste de base.

a	à	afin	ai	**aie**
aient	**ailleurs**	**aime**	ainsi	**aller**
an	**année**	après	aussi	autre
aux	avant	avec	avoir	**bas**
beaucoup	bien	**bientôt**	**c**	ça
car	ce	ceci	cela	celles
celui	ces	**cest**	c'est	cet
cette	ceux	**chose**	ci	comme
côté	**d**	dans	**davantage**	de
déjà	depuis	des	**devoir**	**difficile**
difficilement	du	**également**	elle	en
entrer	et	**etc**	être	**exemple**
facilement	faire	falloir	**fin**	**fort**
gros	**haut**	il	**j**	je
l	la	le	les	leur
lieu	**long**	**lors**	**lorsqu**	**lorsque**
lui	**m**	ma,	mais	**manquer**
me	même	mes	**mettre**	**mieux**
moi	**moins**	mon	**n**	ne
nécessairement	**non**	nos	notre	nous
on	**ou**	**où**	**oui**	par
particularité	pas	**permettre**	**peu**	**plaire**
plus	plusieurs	pour	**pouvoir**	**prendre**
qu	quand	que	**quelqu**	quelqu'un

qui	quoi	**s**	sa	sans
se	selon	ses	seulement	si
son	suis	sur	surtout	**svp**
t	ta	tant	te	tes
toi	ton	**tôt**	tous	tout
toute	toutes	trop	tu	un
une	usage	utiliser	ville	**voir**
vos	votre	vous	**y**	

Annexe 2- Mots à exclure en anglais

Voici les mots de la liste d'exclusion en anglais. Les mots en gras sont des ajouts qui ne figuraient pas sur la liste de base.

a	about	**above**	**absolutely**	**actual**
after	**again**	**against**	**ain**	all
also	although	am	an	and
another	any	are	**aren**	around
as	at	**back**	**bad**	be
because	been	**before**	**being**	**below**
best	**better**	between	bit	both
but	by	can	choose	com
come	**could**	**couldn**	**couldnt**	d
date	day	**did**	**didn**	**didnt**
dislike	**do**	**does**	**doesn**	doing
don	**dont**	down	**during**	each
either	**else**	enough	etc	even
ever	**every**	**everything**	**everytime**	**everywhere**
feel	**few**	**fine**	for	forever
from	**further**	get	go	good
got	**had**	**hadn**	**has**	**hasn**
hasnt	have	**haven**	he	her
here	**hers**	herself	him	himself
his	hotel	how	I	**i**
ie	if	in	**include**	**instead**
into	is	**isn**	it	its
itself	just	**l**	led	let
like	**m**	ma	me	**missing**

more	most	mr	mrs	much
must	**mustn**	my	myself	**need**
needn	**new**	**nice**	nil	no
nor	not	**nothing**	now	of
off	**ok**	**okay**	**old**	on
once	**one**	**only**	or	other
otherwise	our	ours	ourselves	out
over	own	per	plus	poor
priority	put	re	**reality**	really
room	**s**	same	**say**	she
should	**shouldn**	**shouldnt**	so	some
something	**special**	stay	such	sure
t	than	that	the	**their**
theirs	them	**themselves**	then	there
therefore	these	they	**thing**	**think**
this	those	though	**through**	**time**
to	**today**	**told**	too	top
try	under	until	**unusual**	up
us	use	very	**was**	**wasn**
wasnt	we	**were**	**weren**	what
whatsoever	when	where	which	while
who	whom	why	**will**	with
without	**wont**	**worse**	**worst**	**would**
wouldn	**y**	**yesterday**	you	your
yours	**yourself**	**yourselves**		

Annexe 3- Exemples tirés du corpus d'apprentissage pour l'analyse de sentiment

Positifs	Négatifs
Les mesures écologiques prises par la ville sont rassurantes! J'aime penser que nous contribuons à améliorer le monde	Les autobus sont en mauvais état, brisés et pas du tout confortables
Les pistes cyclables font grandement du bien! J'aime beaucoup qu'on puisse circuler en ville de façon différente et y trouver de l'espace	Les taxes municipales sont trop chères. Je ne sais pas vraiment où va tout cet argent.
J'adore cet endroit, beaucoup de restaurants... un cinéma juste à côté aussi. Belle place pour prendre une petite marche.	Vous devriez réviser vos critères concernant vos chauffeurs. Couper les gens avec une autobus en transit et il n'as pas propriété de passage (car il veut tourner à gauche à une lumière verte non-clignotante)me coupe quand je voulais continuer tout droit Alors que j'étais déjà engagé j'ai eu un accident avec une autre automobiliste . J'allais oublier la vitesse maximum `sur industriel pourrait être à regarder, car je pense pas que c'est une autoroute ! ?
C'est bien, j'aime beaucoup l'idée d'avoir un système d'échange d'information entre la ville et ses citoyens. Ça rend notre ville moderne!	Bof! C'est de la le service et le cout $$ trop cher pour rien ! Inefficace ! Bref je préfère mon char ! Que le service de transport de ! En plus il manque d'abris bus!
Stationnement gratuit et spacieux. Grand choix de magasins et restaurants en plus de salles de cinéma.	Pas la bibliothèque la plus fournie de la ville. Pas beaucoup de livres pour ados, surtout en anglais. Si le bruit vous dérange, éviter d'y aller le samedi matin, il n'y a presque que des enfants.
oui, tout a été fait dans les temps et le travail a été de qualité	Le service à la clientèle est dégoûtant et irrespectueux. Le personnel n'est pas dutout courtois ou aidant. Les panneaux d'affiches ne sont représentatif des départs de train. Les dames présentent le lundi 25 septembre on préfèrer discutées entre elles plutôt que de nous offrir assistance. De plus les autobus arrivent

	dernières minutes et quitte la gare en moins de 5 secondes
Je suis surpris de la facilité avec laquelle on peut communiquer avec la ville. J'aime aussi que l'on ait des réponses rapides, ça donne envie de s'impliquer davantage	Mauvaise idée, le délai serait trop long et les coûts trop chers
J'aime les nouveaux festivals de musique en ville, ça donne de la vie à toute la population.	Y'a des nids-de-poule partout, on peut tu les réparer svp ??
J'aime beaucoup l'idée de pouvoir circuler à pied, en sécurité partout dans la ville. Merci! Je suis en fauteuil roulant. Aucun problème d'Accès. Très facile de se rendre sur le quai et d'embarquer ! Si toutes les stations pouvaient être comme cela ça serait génial !	L'information circule mal dans la ville, certains quartiers sont beaucoup plus privilégiés que d'autres, c'est assez injuste
Lieu familiale par excellence. Bonjour aux canards et bernaches! Grands espaces verts et jeux pour enfants.	Très loin... On prend rendez-vous par téléphone, mais il faut quand même patienter très longtemps avant de voir un agent! En plus de cette longue attente, j'arrive face à une bonne femme des plus désagréable! Impatiente, irrespectueuse... Une matinée entière de perdue