Création d'un outil d'analyse des tweets politiques lors de campagnes politiques

Hassine Nader

Création d'un outil d'analyse des tweets politiques lors de campagnes politiques

HASSINE
Nader

Sous la direction de Julien Longhi, Claude Ponton et Claudia Marinica

Laboratoire : AGORA et ETIS

UFR LLASIC (UFR Langage, Lettres, Arts du Spectacle, Information et Communication)
Département d'Informatique pour les Lettres, Langues et Langage

---

Mémoire de master Science Du Langage 2 - 20 crédits

Parcours : Industries De la Langue, orientation recherche

Année universitaire 2016-2017

Création d'un outil d'analyse des tweets politiques lors de campagnes politiques

HASSINE
Nader

Sous la direction de Julien Longhi, Claudia Marinica et Claude Ponton

Laboratoire : AGORA et ETIS

UFR LLASIC (UFR Langage, Lettres, Arts du Spectacle, Information et Communication)
Département d'Informatique pour les Lettres, Langues et Langage

---

Mémoire de master Science Du Langage 2 - 20 crédits

Parcours : Industries De la Langue, orientation recherche

Année universitaire 2016-2017

Remerciements

D’abord, je veux adresser mes remerciements à mon directeur de mémoire, Julien Longhi, pour sa grande disponibilité, pour son aide précieuse, pour le temps qu’il m’a consacré et pour ses encouragements tout au long de la rédaction de ce mémoire.

Je remercie également mes encadrants, Claude Ponton et Claudia Marinica pour leur suivi et pour m’avoir guidé, conseillé et aidé durant la réalisation de ce mémoire.

Je remercie Boris Borzic et Abdulhafiz Alkhouli pour m’avoir guidé et aidé sur la partie technique du projet et de partager avec moi leur expérience.

Un merci tout particulier à Mussab Zneika et Ines Bannour pour les heures qu’on a partagé ensemble dans le même bureau et les différents sujets qu’on a abordés.

Je tiens à remercier Mounir Zrigui et toutes les personnes qui m’ont aidée, soutenue, encouragée au cours de ces deux années de master et qui ont rendu ce mémoire possible.

Enfin, j'adresse mes plus sincères remerciements à ma famille : Mes parents, mon frère et tous mes proches et amis qui m'ont accompagné, aidé, soutenu et encouragé tout au long de mon cursus.

UNIVERSITÉ Grenoble Alpes

Déclaration anti-plagiat
Document **à scanner** après signature
et **à intégrer** au mémoire électronique

DECLARATION

1. Ce travail est le fruit d'un travail personnel et constitue un document original.
2. Je sais que prétendre être l'auteur d'un travail écrit par une autre personne est une pratique sévèrement sanctionnée par la loi.
3. Personne d'autre que moi n'a le droit de faire valoir ce travail, en totalité ou en partie, comme le sien.
4. Les propos repris mot à mot à d'autres auteurs figurent entre guillemets (citations).
5. Les écrits sur lesquels je m'appuie dans ce mémoire sont systématiquement référencés selon un système de renvoi bibliographique clair et précis.

NOM : HASSINE PRENOM : NADER

DATE : 23 / 08 / 2017 SIGNATURE : [redacted]

Sommaire

Introduction

Les réseaux sociaux sont devenus une partie intégrante de notre quotidien. Leur objectif principal est de faciliter la communication avec les gens que l'on connaît d'un point de vue personnel et professionnel. Le service de microblogging, qui permet de publier des messages courts, a permis aux réseaux sociaux de prendre une nouvelle dimension : publication par les utilisateurs de leurs pensées, sentiments ou avis d'une manière courte.

Ouvert à tout le monde, Twitter[1] est devenu, ces dernières années, le numéro un dans le domaine du microblogging comme l'indiquent Kaplan et Haenlein (2010) dans leur recherche. Il peut être vu, comme un indicateur pour connaître les réactions de ses utilisateurs sur plusieurs sujets sociaux, politiques, économiques, etc. Par conséquent, on peut l'utiliser pour extraire les émotions, les sentiments ou les opinions de ses utilisateurs, Kristen et Dan (2016).

Dans le cadre du projet #Idéo2017, l'équipe s'est intéressé aux personnalités et partis politiques qui sont actifs sur ce réseau, en fonction des élections ou contextes, pour décrire et traiter leurs messages politiques en appliquant des analyses linguistiques afin d'extraire leurs émotions et sentiments en constituant un corpus en temps réel à partir des tweets[2] publiés dans leurs comptes officiels.

Le projet #Idéo2017, financé par la Fondation UCP[3], associe des chercheurs du laboratoire AGORA[4] et du laboratoire ETIS[5] (université de Cergy-Pontoise). L'objectif du projet est de créer un outil d'analyse des tweets politiques lors de campagnes politiques.

Comme indiqué dans la réponse à l'AAP de la Fondation de l'UCP, « Ce projet consiste en la création d'une application web en ligne qui permettra de traiter, avec des délais relativement courts, les messages produits en lien avec l'actualité politique (meetings, débats, émissions télévisées, etc.). Cet outil s'appuiera sur la méthodologie de constitution de corpus élaborée dans un précédent projet (corpus Polititweets) et l'implémentation d'outils de statistique textuelle et de visualisation de données. Les citoyens ou journalistes pourront ainsi effectuer leurs propres requêtes et obtenir des résultats compréhensibles grâce à cette interface qui rendra accessible des analyses issues de critères linguistiques et informatiques complexes. En fin de projet, ce travail aura aussi permis la constitution d'une archive du web social (tweets) autour des campagnes concernées » #Idéo2017[6].

Pour arriver à ces résultats, il faut connaître, décrire et analyser linguistiquement et statistiquement les messages politiques envoyés sur Twitter. Les analyses linguistiques qui vont être faites sont des analyses qui vont aider à extraire les principales thématiques des candidats, les « éléments de langage », leur manière de présenter le discours(neutre,

[1] https://twitter.com/
[2] c'est un message ou une publication sur Twitter
[3] http://fondation.u-cergy.fr/
[4] https://www.u-cergy.fr/fr/laboratoires/agora.html
[5] http://www-etis.ensea.fr/
[6] http://ideo2017.ensea.fr/

négative ou positive) et d'autres phénomènes. Aussi des analyses statistiques qui vont permettre à déterminer les sujets les plus évoqués, les relations et les temps forts de chaque candidat.

Description des laboratoires

Le projet associe deux laboratoires de l'Université Cergy-Pontoise[7]. Le premier, qui est le cadre de ce stage, est le laboratoire AGORA, dont sa directrice est Isabelle Prat. AGORA est un centre de recherche en Lettres, Sciences Humaines et Sociales qui résulte de la fusion du CICC (Civilisations et identités culturelles comparées) et du CRTF (Centre de Recherche Textes et Francophonies). Il réunit, autour de l'analyse des sociétés et de leurs écritures, 74 enseignants-chercheurs et 80 doctorants : historiens, civilisationnistes, littéraires, spécialistes du langage et de la communication, archéologues. La fusion des deux centres de recherches permet une approche interdisciplinaire qui, tout en respectant les différents outils méthodologiques disciplinaires, débouche sur des combinaisons fécondes et des coopérations novatrices. Le stage s'intègre dans les travaux de la sous-équipe « discours, communication, numérique ».

Le deuxième laboratoire est ETIS, Équipes Traitement de l'Information et Systèmes ; son directeur est Mathias Quoy. ETIS est une unité de recherche commune au CNRS, à l'ENSEA [8] Cergy et à l'Université de Cergy-Pontoise. Elle est rattachée principalement à l'Institut des sciences informatiques et leurs interactions (INS2I) et elle est structurée en quatre équipes de recherche qui sont l'équipe MIDI (Indexation Multimédia et Intégration de données), l'équipe ICI (Information, Communications, Imagerie), l'équipe ASTRE (Architectures, Systèmes, Technologies pour les unités Reconfigurables Embarquées) et, enfin, l'équipe Neurocybernétique.

Cadre du travail

1. Twitter

Créé le 21 mars 2006 par Jack Dorsey, Evan Williams, Biz Stone et Noah Glass, et lancé en juillet de la même année, Twitter est un outil de microblogage géré par l'entreprise Twitter Inc. Il permet à un utilisateur d'envoyer gratuitement de brefs messages (limités à 140 caractères), appelés tweets, sur internet, par messagerie instantanée ou par SMS [Wikipédia[1]].

De nos jours, *Twitter* est considéré comme l'un des plus fameux réseaux sociaux dans le monde à côté de *Facebook*. Au travers des comptes officiels de personnalités publiques, il est même considéré comme une source fiable d'information sur l'actualité de ces personnalités.

[7] https://www.u-cergy.fr
[8] http://www.ensea.fr/

1.1 Les utilisateurs de Twitter en France

En France, les chiffres parlent d'un nombre élevé d'utilisateurs de Twitter. D'après les statistiques qui sont faites en septembre 2016 sur le site du blogdumoderateur, le nombre de visiteurs uniques par mois sur desktop est environ 5,74 millions ce qui donne 653 000 visiteurs uniques par jour. Par contre, le nombre augmente sur mobile à 15,27 millions de visiteurs uniques par mois. Ceci montre qu'un quart de la population française utilise ce moyen de communication.

Ces utilisateurs sont d'âge, de sexe et de niveau différents [blogdumoderateur] :

- 55% des utilisateurs sont des hommes et 45% sont des femmes
- 33% ont entre 16 et 24 ans, les 25-34 ans sont à 26%, les 35-44 ans sont bien présents avec 25%. Les utilisateurs qui ont un âge entre 45 et 64 ans sont à 16%.
- 36% des utilisateurs sont allés à l'école jusqu'à 18 ans, 19% ont un BTS ou Bac Pro et 29% un diplôme universitaire

Pour conclure, Twitter semble bien constituer un espace libre d'expression pour tous.

1.2 Les tweets politiques

L'une des caractéristiques principales de Twitter est la limitation des messages à seulement 140 caractères. On ne peut donc pas voir de longues publications. Par conséquent, chaque utilisateur doit choisir ses mots soigneusement et doit utiliser des abréviations pour les mots longs ou des initiales pour les mots composés.

Parmi les utilisateurs, on peut trouver des personnalités et des partis politiques qui, à travers leurs comptes Twitter, partagent leurs activités quotidiennes avec leurs abonnés. Twitter semble donc constituer un prisme susceptible de fournir un reflet de la vie politique.

D'après Longhi J. 2013, les tweets politiques sont de vrais discours politique et ne sont pas un moyen secondaire de transmission de l'information, ils se caractérisent par le passage d'un ethos discursif dans les discours politiques vers un ethos technodiscursif qui est le fruit de la petite taille des tweets qui provoque l'existence d'une intensité sémantique, par condensation et décontextualisation partielle, voire une recontextualisation par les moyens technologiques.

Ainsi, Julien Longhi déclare dans une de ses interviews "J'étudie les comptes Twitter des principaux responsables politiques depuis quatre ans, mes recherches montrent que ce média social condense et reflète la substance de la parole politique". Twitter devient donc un moyen très important pour les politiciens pour publier et présenter leurs idées au grand public d'une manière rapide et efficace.

2 Les analyses possibles sur les tweets politiques

Notre projet se concentre sur les analyses linguistiques qui vont être appliqué sur les tweets des personnalités et des partis politiques, pour cela nous avons décidé de faire un état de l'art sur les analyses pour qu'on voie ce qui est possible de mettre en place.

2.1 Nature des mots

En traitement automatique du langage naturel, les deux niveaux de base pour commencer à traiter la langue sont :

- L'étiquetage
- La lemmatisation

L'étiquetage[9] (*Part of speech* en anglais) est également appelé analyse morpho-syntaxique. Il consiste à attribuer un rôle grammatical ou une catégorie à chaque mot. Il est considéré comme l'un des traitements de base pour des analyses syntaxiques ou sémantiques futures pour arriver à travers ces derniers à extraire le sens d'une phrase ou un texte. Voici un exemple d'étiquetage d'un tweet de Marine Le Pen:

"Je suis très sensible aux questions de l'énergie nucléaire."

Je	suis	très	sensible	aux	questions	de	l'	énergie	nucléaire	.
PP	V	ADV	ADJ	PREP	N	PREP	DET	N	ADJ	PUNC

Il existe plusieurs outils d'étiquetage du texte. On peut citer le *LIA TAG*[10]*, Cordial Analyseur*[11]*, le Stanford Tagger*[12].

Le deuxième niveau est la lemmatisation[13]. Comme son nom l'indique, il permet d'attribuer un lemme à chaque mot, c'est-à-dire de prendre sa forme canonique (i.e. infinitif pour les verbes, singulier pour les noms, masculin singulier pour les adjectifs). Donc, les lemmes peuvent prendre des nombreuses formes en fonction du nombre, genre ou le mode.

En prenant l'exemple précédent, la lemmatisation donne ce résultat:

Je	suis	très	sensible	aux	questions	de	l'	énergie	nucléaire	.
JE	ÊTRE	TRES	SENSIBLE	AU	QUESTION	DE	LE	ENERGIE	NUCLEAIRE	.

[9] https://fr.wikipedia.org/wiki/%C3%89tiquetage_morpho-syntaxique (consulté le 25/08/2017)
[10]http://lia.univavignon.fr/fileadmin/documents/Users/Intranet/chercheurs/bechet/download_fred.html
[11] http://www.cordial.fr/Cordial_Analyseur/Presentation_Cordial_Analyseur.htm
[12]http://nlp.stanford.edu/software/tagger.shtml
[13] https://fr.wikipedia.org/wiki/Lemmatisation (consulté le 25/08/2017)

Il existe aussi des outils qui permettent de faire l'étiquetage et la lemmatisation en même temps, comme le *TreeTagger*[14] qui est le plus connu dans le domaine du TAL.

En relation avec les tweets politiques, d'après Djemili *et al.* (2014), ces méthodes vont nous permettre de déterminer la nature de chaque mot dans les tweets : verbe, adjectif, nom, préposition, etc. Ensuite, ils ont comparés *Stanford POS Tagger, Apache Open NLP Tagger*[15] et *Wikimeta Tagger*[16]. Et c'était *Wikimeta Tagger* le plus performant, car il permet de détecter tous les éléments nécessaires. De plus, il est capable de donner des détails concernant les noms propres, et de distinguer les lieux et les personnes par la détection des entités nommées.

2.2 Détection des relations syntaxiques

Pour arriver à analyser les tweets politiques, il faut, d'une part, les extraire et, d'autre part, les mettre sous forme d'un corpus ou une base de données pour les traiter. Le premier niveau d'analyse que l'on peut avoir est l'analyse syntaxique. Elle permet d'étudier la structure des tweets pour identifier et localiser les erreurs existantes en fournissant une structure hiérarchisée des groupements structurels et des relations fonctionnelles. Elle permet donc, de connaître l'organisation des lexèmes, et, par conséquent, si le texte répond ou pas à la syntaxe du langage. En TAL, pour effectuer l'analyse syntaxique et d'après Tutin A. et Dini L. 2016, il nous faut un analyseur syntaxique ou un parseur. Aussi, d'après Chardon Baptiste *et al.* (2016), dans le cas des tweets, les phrases étant courtes, il est alors, un peu difficile de déterminer la signification de la phrase. Pour pallier cela, les twittos utilisent la notion d'*hashtag*. Par exemple, le tweet de Philippe Poutou.

Figure 1 : Tweet de *@PhilippePoutou* - Capture d'écran

La figure précédente montre un tweet[17] de Philippe Poutou, qui est un candidat à l'élection présidentielle 2017 en France, publié par son compte officiel *@PhilippePoutou*

[14] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/
[15] https://opennlp.apache.org/
[16] http://www.wikimeta.fr/
[17] https://twitter.com/PhilippePoutou/status/855121289073315841

le 8 mars 2017. Poutou a utilisé le *hashtag* « #migrants » qui est en relation avec le contexte du tweet et le *hashtag* « #JeVotePoutou » pour encourager les gens de voter pour lui.

Pour conclure, le rôle d'une analyse syntaxique est « d'identifier pour chaque mot d'une phrase de quel autre mot il dépend syntaxiquement et via quelle relation syntaxique » [Synomia]. Donc, cette tâche va nous permettre d'étudier la structure de corpus de chaque candidat.

2.3 La reconnaissance d'entités nommées (REN)

La détection des entités nommés (EN) est une étape souvent indispensable dans les différents domaines du traitement automatique de la langue : analyse syntaxique, traduction automatique, recherche d'information, etc.

D'après Chinchor (1998), les entités nommées sont tous les éléments du langage qui font référence à une entité unique et concrète, appartenant à un domaine spécifique:

- Humain
- Économique
- Géographique

Selon Martineau C. *et al.* 2007, Les EN peuvent être aussi des noms propres au sens classique ou noms propres dans un sens élargi mais aussi des expressions de temps et de quantité. L'objectif principal de cette tâche, est de repérer et catégoriser les entités nommées d'un texte.

Dans le cas des tweets politiques, les entités nommées peuvent être:

- Les noms des politiciens (Emmanuel Macron, Marine Le Pen, Manuel Valls, Jean-Luc Mélenchon, etc.)
- Les noms des partis politiques (Les Républicains, Groupe Socialiste et apparentés, Union des démocrates et indépendants, etc.)
- Les événements politiques (Élections présidentielle, Élections municipales)
- Etc.

On peut citer trois approches pour la détection des entités nommées, selon Talha Meryem *et al.* (2014):

- Approche symbolique
- Approche statistique ou à base d'apprentissage
- Approche hybride

Toujours selon Meryem *et al.* (2014), la première approche s'appuie sur l'utilisation de grammaires formelles construites à la main. Elle se fonde sur la description des EN grâce à des règles qui exploitent des marqueurs lexicaux, des dictionnaires de noms propres et parfois un étiquetage syntaxique.

La seconde approche fait usage de techniques statistiques ou encore dites à base d'apprentissage pour apprendre des spécifiés sur de larges corpus de textes (corpus d'apprentissage) où les entités-cibles ont été auparavant étiquetées et, par la suite, adapter

un algorithme d'apprentissage qui va permettre d'élaborer automatiquement une base de connaissances à l'aide de plusieurs modèles numériques (CRF, SVM, HMM …). Cette méthode a été envisagée pour avoir une certaine intelligence lors de la prise des décisions. Ceux sont principalement certains paramètres qui peuvent être manipulés dans le but d'améliorer les résultats du système, ce qui n'est pas le cas pour les approches symboliques qui n'appliquent que les règles préalablement injectées.

Une troisième approche existe. Elle représente une combinaison des deux précédentes. Elle utilise des règles écrites manuellement mais construit aussi une partie de ses règles en se basant sur des informations syntaxiques et des informations sur le discours extraits de données d'apprentissage grâce à des algorithmes d'apprentissage et des arbres de décision.

La reconnaissance des entités nommées va nous permettre d'extraire des informations de tweets, par exemple :

Figure 2 : Tweet de @benoithamon - Capture d'écran

La figure précédente montre un tweet[18] de Benoît Hamon, qui est un candidat à l'élection présidentielle 2017 en France, publié par son compte officiel @benoithamon.

Au niveau de la REN, « campagne » nous permet de savoir que Hamon parle de sa campagne électorale, et « Toulouse », c'est le lieu de fin de cette campagne.

Durant notre projet, nous n'avons pas utilisé les systèmes de reconnaissance d'entités nommées à cause de l'utilisation d'un outil d'analyse textuelle qui ne prend pas en charge la REN, par contre, nous comptons utiliser la REN dans les prochains travaux pour améliorer nos résultats.

2.4 Détection des thématiques

L'analyse sémantique va nous permettre de comprendre la signification de la phrase en se basant sur le sens de ses mots, c'est le principe de la compositionnalité de Frege[19]. Également, au niveau de l'analyse des tweets politiques, on est censé analyser le contenu de ces derniers, ce qui va nous permettre de connaître la structure du discours de chaque candidat, leurs préférences thématiques, leurs idéologies, etc. Pour cela, on peut utiliser une méthode d'analyse de contenu qui est l'analyse logico-sémantique[20].

[18] https://twitter.com/benoithamon/status/822574916683833349 (consulté le 27/08/2017)
[19] https://en.wikipedia.org/wiki/Principle_of_compositionality (consulté le 27/08/2017)
[20] Qui s'en tient au contenu manifeste, ne considérant que le signifié immédiat, accessible. Elle comprend trois moments. (Source : analyse-du- discours.com)

Une méthode qui peut nous permet de détecter les thématiques est celle de James Benhardus et Jugal Kalita (2013) qui proposent d'analyser les relations entre les mots avec les uni-grammes, bi-grammes et les tri-grammes pour extraire les thématiques.

Guille A. et Favre C. (2014) ont proposé une autre méthode fondée sur la modélisation de l'anomalie dans la fréquence de création de liens dynamiques entre utilisateurs pour détecter les pics de popularité et extraire une liste ordonnée de thématiques populaires.

Enfin et surtout, Ratinaud et Marchand (2012) ont utilisés la méthode *ALCESTE* du logiciel *IRaMuTeQ* qui aide à la détection manuelle des thématiques d'un corpus en entrée.

2.5 Détection des relations entre les mots

« La cooccurrence est la coprésence ou présence simultanée de deux unités linguistiques (deux mots par exemple ou deux codes grammaticaux au sein d'un même contexte linguistique) » Mayaffre, D. (2008). Par exemple :

Élections et candidat / Twitter et publication / joueur et entraîneur

Cette analyse permet de détecter les cooccurrences entre les mots d'un corpus textuelle à l'aide des logiciels de textométrie qui vont proposé une représentation graphique de la cooccurrences, et elle permet de déterminer la catégorisation automatique de textes et aide à extraire les classes lexicales pour déduire la variété de leur emploi, et aussi, pour déterminer l'utilisation de certains termes avec d'autres et de connaître leur fréquence. (Martinez et coll. 2010).

2.6 Détection d'événement

Le domaine de la politique est axé sur les événements qui sont organisés par les différents candidats, par conséquent, ces derniers ne vont pas rater l'occasion sans parler de leurs actualités dans leurs tweets. Les événements peuvent être sur des courtes périodes comme les débats télévisés, des manifestations ou des visites etc. Aussi, sur des longues périodes comme l'élection présidentielle ou législatives. « Un évènement sera représenté par un ensemble de termes dont la fréquence augmente brusquement à un ou plusieurs moments durant la période analysée. Comme les hashtags permettent de donner une idée générale sur les sujets discutés dans un tweet. » Dridi H-E. et LEPALME, (2014).

Figure 3 : Tweet de @EmmanuelMacron - Capture d'écran

La figure précédente montre un tweet[21] d'Emmanuel Macron, qui est un candidat à l'élection présidentielle 2017 en France, publié par son compte officiel *@EmmanuelMacron* le 8 mars 2017. Macron a utilisé le *hashtag* « #JournéeDesDroitsDesFemmes » pour montrer sa participation à la journée des droits des femmes.

2.7 Détection de l'émotion

L'apprentissage supervisé permet de traiter un phénomène pour construire un modèle qui sera capable de prédire ce phénomène avec une manière automatique. Avec cette technique on a une cible à prédire. Par exemple, dans leur recherche, Hasan Maryam *et al. (*2014) ont utilisés le modèle circumplex d'émotions de Russell (1980), (voir figure 4), qui contient 28 émotions.

Figure 4 : Modèle circumplex d'émotions de Russell (1980)

Selon ce modèle, le système peut prédire le pourcentage d'émotion de chaque candidat pour chaque catégorie d'émotion en traitant les tweets de ces derniers. Pour cela, on peut créer un modèle qui contient les champs lexicaux des différentes idéologies. On peut également citer les travaux d'Andranik Tumasjan *et al.* (2010) qui sont arrivés à montrer que Twitter est un outil de prédiction de résultats des élections à l'aide de la détection de l'émotion et du sentiment en utilisant LIWC2007, qui est un outil d'analyse textuelle payant. Par contre, Choy *et al.* (2011) ne sont pas arrivés à déterminer les résultats des élections de 2011 à Singapour à l'aide de l'extraction de l'émotion.

[21] https://twitter.com/EmmanuelMacron/status/839563734922244096 (consulté le 27/08/2017)

3 Travaux autour des tweets

L'importance de ce phénomène vient à travers plusieurs études scientifiques pertinentes parmi lesquelles on peut citer les travaux menés par Johnson K. et Goldwasser D., (2016) qui sont intitulés "*Identifying Stance by Analyzing Political Discourse on Twitter*". Comme le titre l'indique, ces deux chercheurs américains de l'Université Prudue ont travaillé sur l'identification du comportement des politiciens durant les élections présidentielles 2016 aux États-Unis à travers leurs discours sur Twitter sur des problèmes politiques et leur réaction aux événements internationaux. La méthode utilisée durant cette recherche est une méthode faiblement supervisée qui leur permet de déterminer les comportements les plus probables des candidats. D'après eux, leur travail est le premier de son genre pour prédire les comportements des politiciens en utilisant des données de Twitter qui sont basées sur le contenu, les frames et les activités temporelles, Johnson K. et Goldwasser D., (2016).

Une autre recherche est celle menée par Mohammad Saif M. *et al.* (2016). Elle est basée sur la détection de l'opinion que peut avoir une personne et qu'il peut exprimer d'une manière positive ou négative. Il s'agit de connaître leur position vis-à-vis d'une cible. Ils ont, pour cela, utilisé pour la première fois un ensemble de données de paires tweet-cible annotées sur l'opinion et le sentiment. Ils ont montré que la connaissance d'un sentiment exprimé par un tweet est un bénéfice pour la classification des opinions. Et durant cette recherche, ils disposaient de données non étiquetées, à travers des méthodes supervisées et la représentation distribuée de mots pour améliorer la classification de la position. Concernant les tweets politiques, les recherches de Mohammad Saif M. *et al.* (2015) se sont concentrées sur l'analyse automatique des tweets politiques et l'extraction du sentiment, l'émotion, le style et l'objectif. Ils ont utilisé un corpus de tweets qui a été constitué durant les élections présidentielles américaines en 2012. Leur but était de comprendre comment le sentiment public est façonné, le suivi du sentiment et sa polarisation par rapport aux candidats et aux enjeux, et comprendre l'impact de ces tweets, etc. Durant leur recherche, ils ont utilisé trois méthodes d'annotation automatique : *random base line*, *majority base line* et *SVM*[22]. Le résultat le plus performant est le SVM, qui est une méthode de classification binaire par apprentissage supervisé. Enfin, ils ont montré que les tweets transmettent des émotions négatives deux fois plus que positives et que même si le classificateur d'objectif profite des caractéristiques d'émotion, la détection d'émotion seule peut ne pas faire la distinction entre plusieurs types différents d'objectifs.

L'apprentissage supervisé est l'approche utilisée par Hasan Maryam *et al*. (2014) pour déduire l'état émotionnel des utilisateurs. Les chercheurs ont utilisé les émoticônes et les hashtags comme une base pour détecter les sentiments et les émotions. Pour nettoyer les textes, ils ont remplacé les mots qui commencent par @ en "USERID", et les liens url par "URL". Ils ont, ensuite, remplacé les mots qui contiennent plus de deux occurrences d'une même lettre par une seule ; par exemple : « *happyyyy* » en « *happy* ». Ils ont également supprimé les tweets qui contiennent des hashtags de deux types opposés. Par example: « *Got a job interview today with At&t... #nervous #excited.* », ici, on observe

[22] SVM: http://www.support-vector.net/

que *#nervous* appartient à la classe *Unhappy*, par contre, *#excited* appartient à la classe *Happy*. Ils ont appliqué le même traitement aux tweets qui contiennent des émoticônes de deux types opposés et les tweets qui contiennent un émoticône et un hashtag de types opposés. Le dernier traitement qui sera fait, est de supprimer les hashtags qui se trouvent à la fin des tweets car les classificateurs vont mettre une grande quantité de poids sur les étiquettes, ce qui peut nuire à la précision. Cependant, les balises au début ou au milieu du tweet sont laissées car elles font partie du contenu de la phrase. Enfin, ils ont comparé la précision de plusieurs algorithmes de *machine learning*, y compris SVM, KNN, *Decision Tree* et *Naive Bayes* pour le classement des messages Twitter. Leur technique a une précision de plus de 90%.

On peut encore citer la recherche de Vidak *et al. (*2016) qui s'intéresse à analyser les pratiques discursives à travers le fonctionnement des outils textuels multimodaux[23] du tweet. Ils ont utilisés un corpus de tweets pour effectuer une étude qualitative et quantitative, et ils ont montré que les outils multimodaux ne sont pas utilisés seulement pour un rôle technique dans les tweets, mais aussi, ils jouent également un rôle important dans l'expression des sentiments, des partis-pris ou en alimentant la polémique.

Les travaux précédents sont d'une manière générale un échantillon représentatif de l'existant, mais nous allons aussi détailler quelques travaux scientifiques menés à Cergy (lieu de déroulement du stage) qui ont influencé la manière dont le projet #idéo2017 est construit.

Dans Longhi J. *et al*. (2016), intitulé *l'extraction automatique des phénomènes linguistiques dans un corpus de tweets politiques*, le travail était concentré sur la détection de la négation en utilisant plusieurs méthodes et critères de recherche pour mettre en avant la diversité de ses formes, à la fois du point de vue des paliers de l'analyse (syntaxe, sémantique, énonciation) mais aussi de ses spécificités.

Ensuite, la détection de l'idéologie dans les tweets politiques (Djemili S. *et al*, 2014) a permis d'utiliser les critères discursifs de Sarfati G. E. (2014) pour la création de règles linguistiques qui vont être implémentées dans un outil de traitement automatique de la langue pour détecter l'idéologie dans les corpus de tweets politiques.

De plus, des méthodes textométriques qui ont été utilisées par Longhi J. et Saigh D. 2016 pour la détection des réactions des twittos sur l'annonce du nouveau système d'assurance et du chômage en France à l'aide de l'outil d'analyse textuelle *IRaMuTeQ*. Et en dernier lieu, Longhi J. 2017 explique l'efficacité d'un tweet en fonction du nombre de fois où il était retweeté en utilisant les analyses proposées par *IRaMuTeQ*.

Ces différents travaux ont servi de base au stage : constitution de corpus de tweets, implémentation de règles linguistiques issues de travaux d'analyse du discours, recours à la textométrie et à sa capacité à rendre compte visuellement des résultats.

[23]le mot-dièse, le lien internet et les liens permettant l'intégration des supports multimédia

4 Choix de l'outil d'analyse textuelle

4.1 Comparaison

Dans notre projet, on va appliquer des analyses linguistiques sur l'ensemble des tweets collectés pour chaque candidat ou parti politique, pour cela nous allons choisir un outil d'analyse textuelle qui sera le plus compatible avec nos besoins. Ce choix sera fait en se basant sur plusieurs critères. Parmi ces critères, on trouve :

Les analyses qui sont proposés par ces outils.

- le code doit être open source
- une API qui permet de l'exploiter en version Web
- exécution en mode batch

Pour commencer nous avons étudié quatre outils d'analyses linguistiques qui sont :

- *IRaMuTeQ*[24]
- *Hyperbase*[25]
- TXM[26]
- Lexico3[27]

Car ce sont des outils connus et utilisés en analyse du discours assistée par ordinateur.

Le premier outil est *IRaMuTeQ* version 0.7 alpha 2 : il permet de faire des analyses statistiques sur des corpus textuels et sur des tableaux individus/caractères. Il propose quatre fonctionnalités principales[28] :

1. Statistique sur le corpus
2. Spécificités et AFC à partir de segmentation définie
3. Classification selon la méthode de Reinert
4. Analyse de similitude sur les formes pleines d'un corpus
5. Nuage de mots

Le deuxième outil est *Hyperbase*, est un logiciel documentaire et statistique pour l'exploration des textes. Il propose six fonctionnalités principales[29] :

1. Calcul des spécificités et graphes de distribution des unités linguistiques du corpus
2. Indices de richesse lexicale et d'accroissement du vocabulaire
3. Traitement et représentation factoriels de matrices lexicales ou grammaticales complexes dans la lignée des travaux de Jean-Paul Benzécri
4. Calcul de distances entre textes, classification et représentation arborées

[24] http://www.iramuteq.org/
[25] http://ancilla.unice.fr/
[26] http://textometrie.ens-lyon.fr/
[27] http://lexi-co.com/
[28] http://www.iramuteq.org/documentation/html (consulté le 27/08/2017)
[29] https://fr.wikipedia.org/wiki/Hyperbase#Fonctionnalit.C3.A9s (consulté le 22/08/2017)

5. Extraction des phrases typiques et des segments répétés
6. Calcul et représentations des cooccurrences et réseaux thématiques

Le troisième outil est TXM Heiden, S. (2010b). C'est une plateforme qui aide couramment les utilisateurs à construire et à analyser tout type de corpus textuel numérique éventuellement étiqueté et structuré en XML. Il propose cinq fonctionnalités principales[30] :

1. Construction de sous-corpus
2. Cooccurrences
3. Statistique
4. Spécificités et AFC
5. Classification

Le quatrième outil est Lexico3, c'est un logiciel de statistique textuelle. Il propose trois fonctionnalités principales[31] :

1. Statistique
2. Spécificités et AFC
3. Navigation lexicométrique

Le tableau suivant présente une comparaison entre quatre outils d'analyses textuelles (IRaMuTeQ, *Hyperbase*, TXM et Lexico3) pour les trois premiers critères :

	Open source	**API**	**Mode batch**
IRaMuTeQ	+	-	+
Hyperbase	+	-	-
TXM	+	-	-
Lexico3	+	-	-

« + » : disponible
« - » : non disponible

Tableau 1 : Ce tableau présente la comparaison entre quatre outils d'analyses textuelles

Comme le montre le tableau 1, premièrement tous les outils sont en *open-source*, donc, la modification au niveau de leur code source est autorisée. Deuxièmement, comme

[30] http://txm.sourceforge.net/doc/manual/manual1.xhtml (consulté le 22/08/2017)
[31] http://lexi-co.com/ressources/manuel-3.41.pdf (consulté le 22/08/2017)

notre plateforme est une application web, il serait préférable de trouver une API (*Application Programming Interface*) pour l'un de ces outils afin de l'intégrer directement à notre code et exécuter facilement ces fonctionnalités; malheureusement qu'aucun outil ne propose ce service. Le troisième critère est le *mode batch*[32] qui n'est disponible qu'avec le logiciel *IRaMuTeQ* mais seulement avec la fonctionnalité de statistique.

Nous avons décidé de choisir *IRaMuTeQ* puisque son code est *open-source*. Mais le plus important qu'il est exécutable en mode batch et avec des modifications au niveau de son code qui est développé en Python, on a réussi à exécuter quelques fonctionnalités d'*IRaMuTeQ* (Spécificité et AFC, classification et l'analyse de similitude. Plus la fonctionnalité de statistique qui est déjà existante) en utilisant les lignes de commande (mode batch) et sans passer par son interface graphique pour que nous arrivions à exploiter les résultats de ces fonctionnalités en version web et en temps réel.

4.2 IRaMuTeQ

IRaMuTeQ est un outil d'analyse textuelle libre développé par Pierre Ratinaud au sein du laboratoire LERASS[33]. « Il permet de faire des analyses statistiques sur des corpus texte et sur des tableaux individus/caractères » *IRaMuTeQ*[34]. Il propose 5 types d'analyses sur les corpus : statistiques, spécificités et AFC, Classification selon la méthode de Reinert, analyse de similitudes et des nuages de mots.

Avant de commencer la description des fonctionnalités d'*IRaMuTeQ*, on va présenter le format des données en entrée, le dictionnaire qu'il utilise ainsi les différentes catégories grammaticales existantes.

4.2.1 Format d'entrée et syntaxe

Comme tous outils, *IRaMuTeQ* propose son propre format d'entrée qui est « .txt ». Ce fichier doit être encodé en UTF8 et doit respecter la mise en forme « Alceste ». Cette mise en forme indique qu'avant chaque texte introduit il faut mettre quatre étoile suivies de variables étoilées et séparées par un espace comme le montre la figure 5 :

```
**** *EmmanuelMacron *tweet1
Je veux ensemble que nous croyions à nouveau dans l'Europe,
l…
**** *EmmanuelMacron *tweet2
Je veux une vraie politique de baisse de prix des billets, a
**** *EmmanuelMacron *tweet3
Aujourd'hui, les billets de et vers La Réunion sont trop che
**** *EmmanuelMacron *tweet4
Que vous gagniez deux fois ou cinq fois le SMIC, vous payez
injust…
**** *EmmanuelMacron *tweet5
Simplifier la vie des entrepreneurs en simplifiant le droit
renvoyan…
```

Figure 5 : Extrait du corpus d'Emmanuel Macron

[32] https://fr.wikipedia.org/wiki/Traitement_par_lots (consulté le 22/08/2017)
[33] https://www.lerass.com
[34] http://www.iramuteq.org

Un corpus doit contenir au moins un texte pour qu'il soit accepté par le logiciel.

4.2.2 Nettoyage

Passant maintenant à la partie de nettoyage du corpus, et au niveau des configurations d'*IRaMuTeQ*, on trouve une option de nettoyage qui peut être appliquée sur le corpus. Cette option permet de faire plusieurs traitements sur le fichier :

- Passer le corpus en minuscule
- Permet de retirer une liste de caractères au choix (par défaut, *IRaMuTeQ* conserve les caractères alphanumériques et accentués)
- Remplacement des apostrophes ou/et les tirets par des espaces
- Conservation de la ponctuation

4.2.3 Lemmatisation

Comme *IRaMuTeQ* est un outil d'analyses linguistiques, donc il propose d'appliquer la lemmatisation sur le corpus d'entré. Cette option permet de regrouper plusieurs formes ou dérivés d'un mot et de les assigner le lemme correspondant. Il lemmatise les noms au singulier, les verbes à l'infinitif et les adjectifs au masculin singulier.

- traité, traitais, traitions → traiter
- grand, grande, grandes → grand
- université, universités → université

Après la phase de la lemmatisation, *IRaMuTeQ* accorde à chaque mot une catégorie grammaticale selon le tableau suivant :

Étiquette	**Catégorie grammaticale**
adj_sup	Adjectif supplémentaire
art_ind	Article indéfini
adj_pos	Adjectif possessif
adv_sup	Adverbe supplémentaire
pro_dem	Pronom démonstratif
art_def	Article défini
con	Conjonction

pre	Préposition
ono	Onomatopée
adj_dem	Adjectif démonstratif
nom_sup	Nom supplémentaire
adv	Adverbe
ver	Verbe
adj_num	Adjectif numérique
pro_rel	Pronom relatif
adj_ind	Adjectif indéfini
pro_ind	Pronom indéfini
pro_pos	Pronom possessif
aux	Auxiliaire
ver_sup	Verbe supplémentaire
adj	Adjectif
adj_int	Adjectif interrogatif
nom	Nom commun
num	Chiffre
pro_per	Pronom personnel
nr	Non reconnue

Tableau 2 : Tableau des étiquettes d'*IRaMuTeQ*

4.2.4 Les fonctionnalités

Dans cette partie, on va donner une description pour les différentes fonctionnalités qui nous intéressent chez *IRaMuTeQ*. Tout d'abord, on va commencer par la fonctionnalité des statistique, après on va expliquer les spécificités et AFC, ensuite, la classification, puis, l'analyse de similitude et enfin le nuage de mots. Nous avons quelques descriptifs des fonctionnalités de la documentation d'*IRaMuTeQ* [Documentation1] [Documentation2].

4.2.4.1 Statistiques

La première fonctionnalité est la statistique, cette analyse propose des statistiques simples sur les corpus textuels :

- Effectif de toutes les formes.
- Effectif des formes actives et supplémentaires.
- Liste des mots avec une seule occurrence.
- Liste des hapax
- Un graphique qui présente en abscisse les logarithmes des rangs et en ordonnées les logarithmes des fréquences des formes et elle se base sur la loi de Zipf[35].

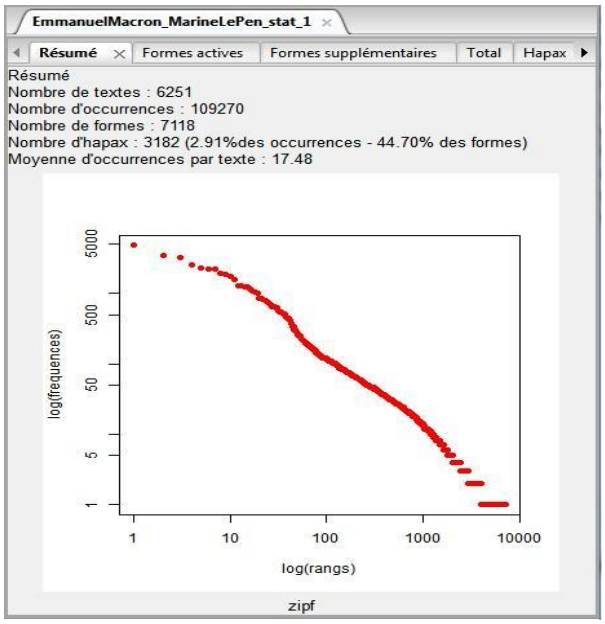


Figure 6 : Les résultats de la fonctionnalité de statistique

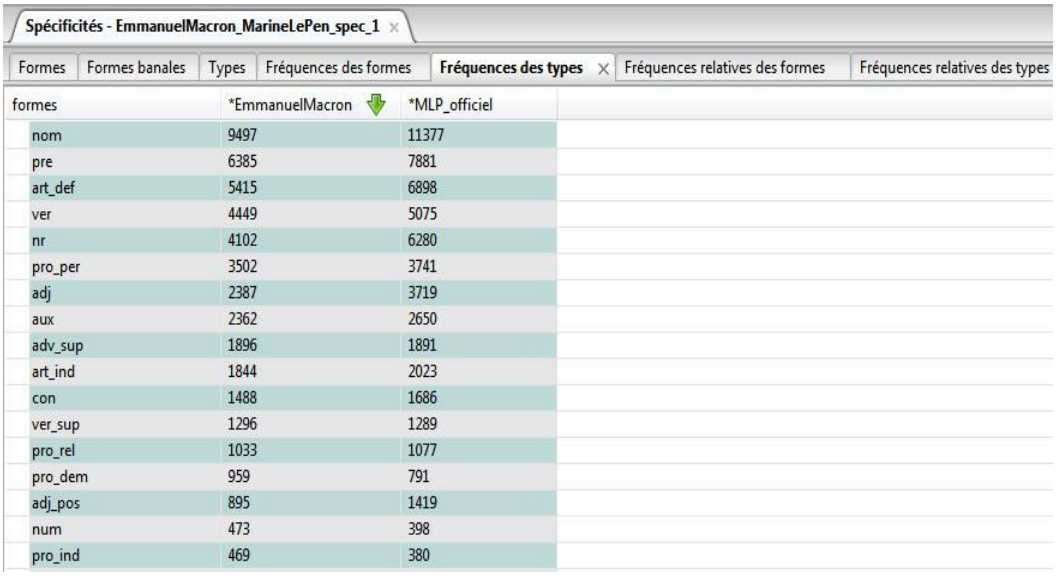
4.2.4.2 Spécificité et AFC

La deuxième fonctionnalité est la spécificité et AFC, Cette analyse permet d'identifier les mots spécifiques par sous-catégories et réalise une analyse factorielle sur un tableau lexical agrégé (TLA) construit avec les variables ou les modalités sélectionnées en utilisant la loi hypergéométrique.

Elle donne comme résultat (Figure 7) :

- La liste des formes, des formes banales et des catégories grammaticales, et leurs scores par modalité
- Effectif de chaque forme/lemme (ou catégorie grammaticale)
- Des graphiques (Histogramme et Diagramme)

[35] https://fr.wikipedia.org/wiki/Loi_de_Zipf (consulté le 21/08/2017)



formes	*EmmanuelMacron	*MLP_officiel
nom	9497	11377
pre	6385	7881
art_def	5415	6898
ver	4449	5075
nr	4102	6280
pro_per	3502	3741
adj	2387	3719
aux	2362	2650
adv_sup	1896	1891
art_ind	1844	2023
con	1488	1686
ver_sup	1296	1289
pro_rel	1033	1077
pro_dem	959	791
adj_pos	895	1419
num	473	398
pro_ind	469	380

Figure 7 : Capture de l'interface de Spécificité et AFC

4.2.4.3 Classification Méthode Reinert

La troisième fonctionnalité est la classification, cette analyse utilise la méthode « Alceste » de Max Reinert. Cette méthode classe les phrases du corpus, en fonction de la distribution du vocabulaire présent dans ces unités de contexte, et elle repère le vocabulaire dans les différentes unités de contexte et les met en relation. Autrement dit, il relie les contextes qui ont des mots communs. (Valérie Delavigne 2014)

Le logiciel propose trois types de classification (documentation[36]) :

- Classification simple sur texte : Ici, les Textes resteront dans leur intégralité, la classification permettra ainsi de regrouper les Textes les plus proches.
- Classification simple sur segments de texte : La classification portera sur les segments de textes (ST).
- Classification double sur RST : La classification est menée sur deux tableaux dans lesquels les lignes ne sont plus des segments de texte mais des regroupements de segments de texte (RST). Le même traitement est ainsi fait deux fois, mais en changeant le nombre de formes actives par RST.

La classification permettra de regrouper les textes les plus proches et elle propose une répartition par classe (par thème), un graphe qui permet de savoir la relation entre ces

[36]http://www.iramuteq.org/documentation/fichiers/documentation_19_02_2014.pdf, 2.5.3Classification Méthode Reinert (consulté le 22/08/2017)

classes (Figure 8) et un troisième graphe qui donne la liste des mots pour chaque classe ce qu'il va aider l'utilisateur à interpréter le thème.

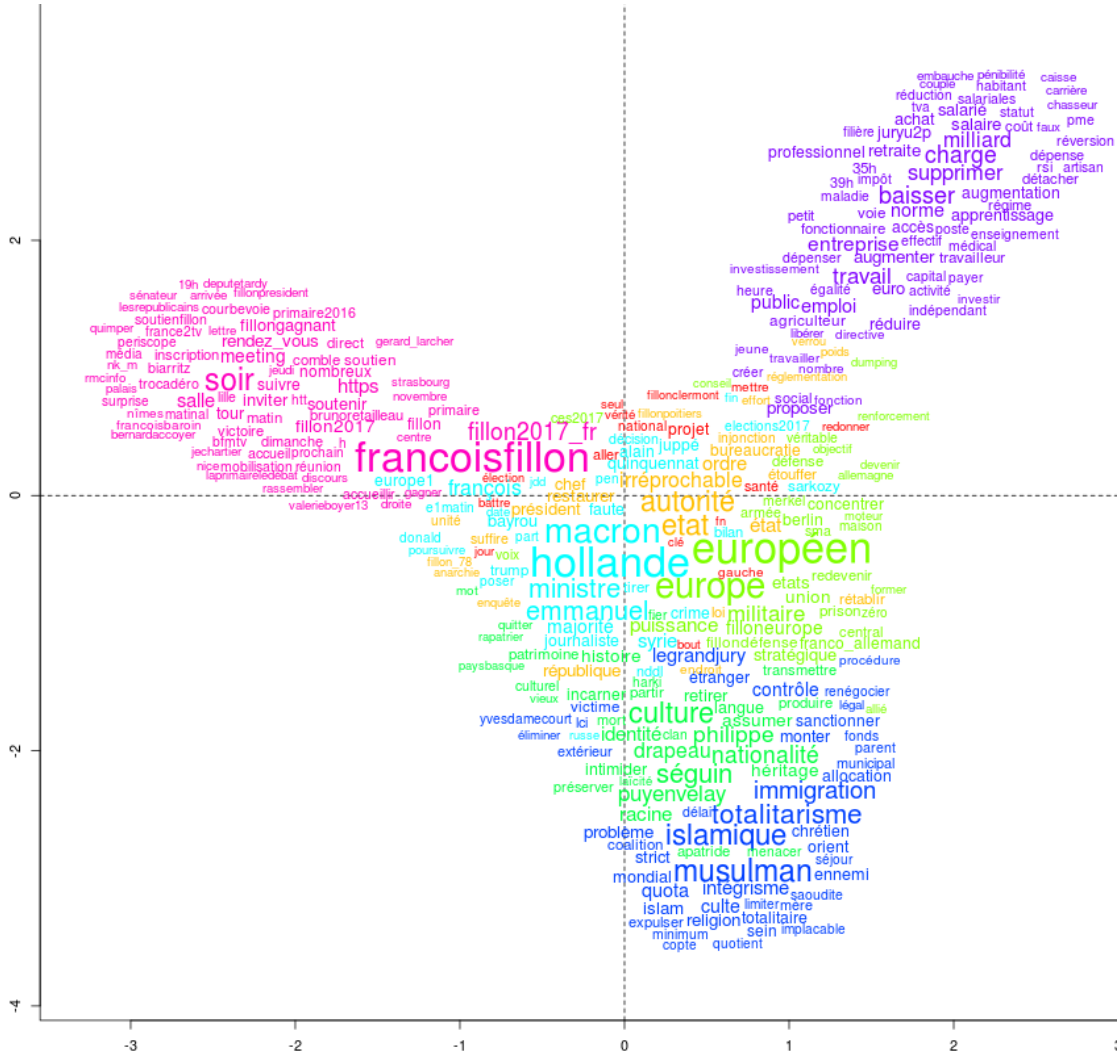


Figure 8 : Graphe des relations entre les thèmes

4.2.4.4 Similitude

L'objectif de l'analyse de similitude (ADS) est d'étudier la proximité et les relations entre les éléments d'un ensemble, sous forme d'arbres maximum : le nombre de liens entre deux items évoluant «comme le carré du nombre de sommets» (Flament et Rouquette, 2003: 88), l'ADS cherche à réduire le nombre de ces liens pour aboutir à «un graphe connexe et sans cycle» (Degenne et Vergès, 1973: 473). D'après Marchand P., Ratinaud P. (2012), l'analyse de similitude d'une matrice textuelle a été intégrée au logiciel IRaMuTeQ (développé par Pierre Ratinaud) et permet de décrire des classes lexicales, des profils de spécificités ou même des corpus entiers.

Il s'agit d'une analyse des cooccurrences présentée sous formes de graphiques de mots associés. Les indices de similitudes proposés dans *IRaMuTeQ* sont ceux disponibles dans la librairie proxy de R.

Comme le montre la figure 9 Plus la taille des mots est grande, plus ils sont fréquents dans le corpus, plus les liens/arêtes sont épais, plus les mots sont co-occurrents.

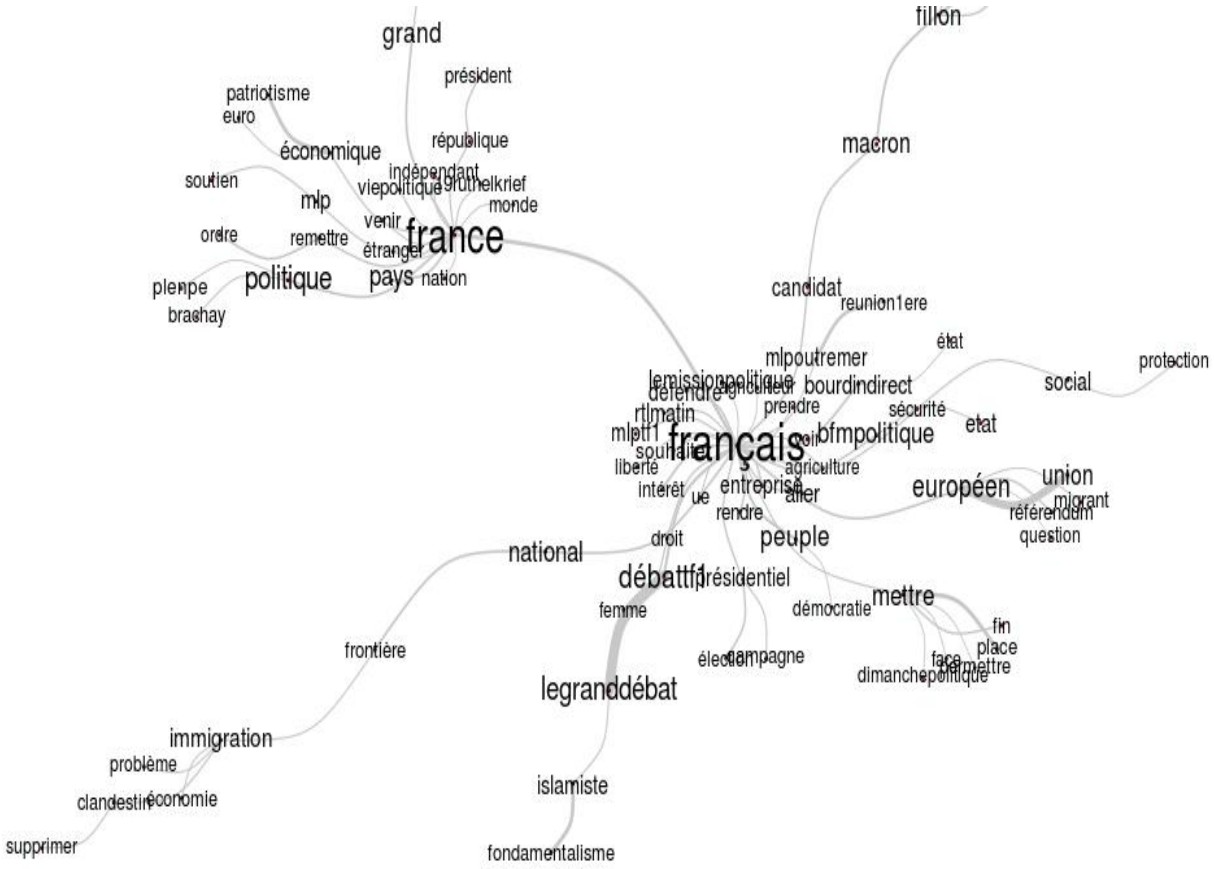


Figure 9 : Graphique de l'analyse de similitude

4.2.4.5 Nuage de mots

La dernière fonctionnalité est le nuage de mots, cette analyse permet d'afficher le lexique des mots associés au corpus sur la forme d'un graphique appelé « nuage de mots » où la taille des formes/mots est proportionnelle à leur fréquence. Les mots les plus cités sont placés au centre.

Figure 10 : Graphe de nuage de mots

5 Conclusion

Nous avons montré qu'avant de débuter avec les analyses sémantiques, il faut passer par l'étiquetage et la lemmatisation, car ce sont les pièces maîtresses de chaque outil de traitement automatique de la langue, et qu'ils nous permettent d'avoir des résultats satisfaisants en appliquant les différentes méthodes de détection de l'outil qu'on a choisi qui est *IRaMuTeQ*.

Dans la section suivante, nous allons décrire notre plateforme #Idéo2017 en passant par les différentes analyses linguistiques proposées et le moteur de recherche développé, et aussi les techniques de développement de cette plateforme qui nous a permis de rendre nos résultats disponibles au public sous forme graphique et statistique.

Tâches effectuées

6 Description et développement de la plateforme #Idéo2017

6.1 Introduction

Dans cette partie nous allons présenter la plateforme #Idéo2017 innovante par l'accès à l'information qu'elle permet aux citoyen. Ainsi, la plateforme #Idéo2017 a pour objectif de proposer un outil qui permettra aux citoyens d'analyser par eux-mêmes les dires des candidats à l'élection présidentielle[37] en France en 2017, les partis politiques qui participent aux élections législatives[38] 2017 et quelques personnalités politiques pour le quinquennat [39] . #Idéo2017 propose des analyses linguistiques et des visualisations graphiques de données des comptes des personnalités et des partis politiques candidats aux élections, afin de décrire et traiter leurs messages, en constituant un corpus en quasi temps réel, car la mise à jour se fait chaque 24 heures à partir des tweets publiés dans leurs comptes officiels et à partir du premier septembre 2016 pour la présidentielle et du 7 mai 2017 pour les législatives. Ce traitement doit pouvoir rendre compte, par l'utilisation d'outils et de fonctionnalités issues de la linguistique outillée, des principales caractéristiques de ce corpus, et permettre notamment des comparaisons entre les différents candidats. A la fin de la campagne présidentielle, ce recueil de tweets va permettre la constitution d'une archive de la campagne 2017 (finalisation du corpus en septembre grâce à l'obtention d'1 mois de CDD d'ingénierie par le consortium CORLI).

Dans un premier temps, nous présentons le développement de l'outil, en présentant la structure qui a été choisie, les justifications de ces choix, ainsi que les choix technologiques. Nous proposons ensuite une présentation plus précise de l'outil, et des analyses rendues possibles, et enfin on va présenter les visualisations graphiques de données.

6.2 Description de l'outil #Idéo2017

Dans cette section, nous allons présenter la structure globale de l'outil #Idéo2017. Comme écrit précédemment, l'outil #Idéo2017 est une plateforme web en ligne qui permet de traiter les messages produits en lien avec l'actualité politique (meetings, débats, émissions télévisées, etc.). Son objectif est de rendre disponibles des analyses issues notamment d'outils de statistique textuelle et de visualisation de données, sous forme web (et non logicielle), afin que des utilisateurs non spécialistes de ces outils puissent avoir accès à certains résultats (sans passer par les phases de constitution de corpus, de balisage, etc.). Les citoyens peuvent ainsi effectuer leurs propres requêtes et obtenir des résultats compréhensibles grâce à cette interface qui rend accessible des analyses issues de critères

[37] http://ideo2017.ensea.fr/plateforme/
[38] http://ideo2017.ensea.fr/legislatives2017/
[39] http://ideo2017.ensea.fr/quinquennat/

linguistiques (calculs de spécificités pour les mots les plus employés par les personnalités politiques, analyses de similitudes, algorithme Alceste, etc.).

6.3 Description de la chaîne de traitement

Pour la mise en place de l'outil #Idéo2017, nous avons dû suivre plusieurs étapes, comme présenté dans le schéma de la figure 11 : (1) l'extraction de l'ensemble de tweets des 11 personnalités politiques qui se sont déclarées en tant que candidats à l'élection présidentielle (neuf partis politiques pour les législatives), (2) la mise en place d'un sauvegarde des tweets, (3) l'indexation des tweets pour faciliter la recherche dans l'ensemble de tweets, (4) l'application d'un ensemble d'analyses linguistiques sur les tweets, (5) la mise en place d'un moteur de recherche sur l'ensemble de tweets, et (6) l'affichage des résultats sur une page web.

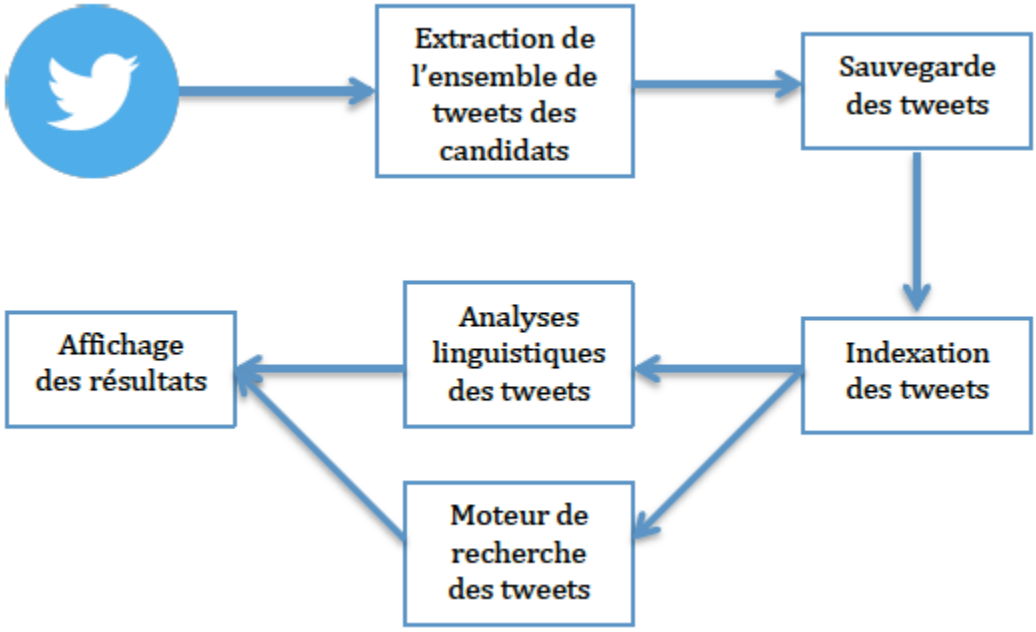


Figure 11 : Schéma de la plateforme #Idéo2017

Dans un premier temps, nous nous intéressons à l'extraction des tweets des candidats. Pour l'outil #Idéo2017, nous souhaitons extraire les tweets chaque jour et proposer aux utilisateurs des analyses sur les tweets de la journée précédente ; par exemple, le 4 avril 2017 les utilisateurs pourront analyser les tweets envoyés par les candidats jusqu'au 3 avril 2017. À cette fin, nous avons profité des travaux qui ont été réalisés dans le cadre de l'extraction du corpus Polititweets (Longhi *et al*, 2014).

Au moment de l'extraction des tweets et puisque ces derniers ne contiennent pas que du texte, on a décidé de nettoyer les tweets récupérer et de garder que le texte et les *hashtags*. Pour cela nous avons utilisés les expressions régulières pour supprimer les liens et les émoticônes.

Étroitement liées à l'étape d'extraction des tweets, dans les étapes de sauvegarde et indexation des tweets nous nous intéressons au stockage et à la mise en place d'un système d'indexation ; ces deux points nous permettront de faciliter l'accès et la recherche sur les tweets et ainsi la mise en place d'un moteur de recherche intelligent à l'aide de filtres.

Dans l'étape d'analyse linguistique nous souhaitons proposer à l'utilisateur un ensemble d'analyses à réaliser sur l'ensemble de tweets. Ces analyses, décrites de manière détaillée dans la section suivante, concernent plus particulièrement : l'emploi d'un mot spécifique et ses dérivés par les différents candidats, les mots associés à un mot spécifique, le nuage de mots, les thématiques, les relations entre les mots, les spécificités des différents candidats et une partie pour la visualisation graphique de données qui a été intégré avec les analyses linguistiques.

Finalement, l'étape de mise en place du moteur de recherche, nous avons développé un moteur de recherche intelligent à base de filtres qui permet à l'utilisateur de réaliser des recherches sur les tweets en utilisant des filtres spécifiques ce qui permet une liberté de navigation sur la base de tweets.

6.4 Description des analyses linguistiques effectuée

Pour une meilleure compréhension de l'outil #Idéo2017, nous proposons dans la Figure 12 l'interface graphique de la plateforme. Nous pouvons remarquer que deux analyses différentes sont proposées à l'utilisateur[40] (auxquelles nous pouvons ajouter le moteur de recherche) :

- *« J'analyse les tweets qui contiennent le mot… »* : Cette analyse permet à l'utilisateur de choisir un mot parmi les 13 mots qui sont souvent employés dans les débats politiques (Alduy, 2017). Cette analyse donne accès à quatre analyses possibles : l'usage de ce mot par les différents candidats, les mots associés à ce mot, l'emploi de ce mot et ses dérivés par les différents candidats et le nuage de mots.

- *« J'analyse les tweets de....[candidat] »* : Cette analyse permet à l'utilisateur de choisir un candidat parmi les onze candidats déclarés (neuf partis politiques) afin de réaliser les analyses linguistiques suivantes sur ses tweets : les mots les plus utilisés, les thématiques, les relations entre les mots, nuage de mots, les spécificités des différents candidats (si l'utilisateur a choisi d'analyser tous les candidats au même temps), la visualisation des données et l'extraction d'un corpus au format *IRaMuTeQ*.

- *« Je navigue dans tous les tweets par filtre »* : Cet outil permet à l'utilisateur de chercher librement sur toute la base des tweets à l'aide des facettes.

[40] Ici on vise le grand public, les citoyens et les chercheurs qui ont un intérêt à l'analyse du discours politique.

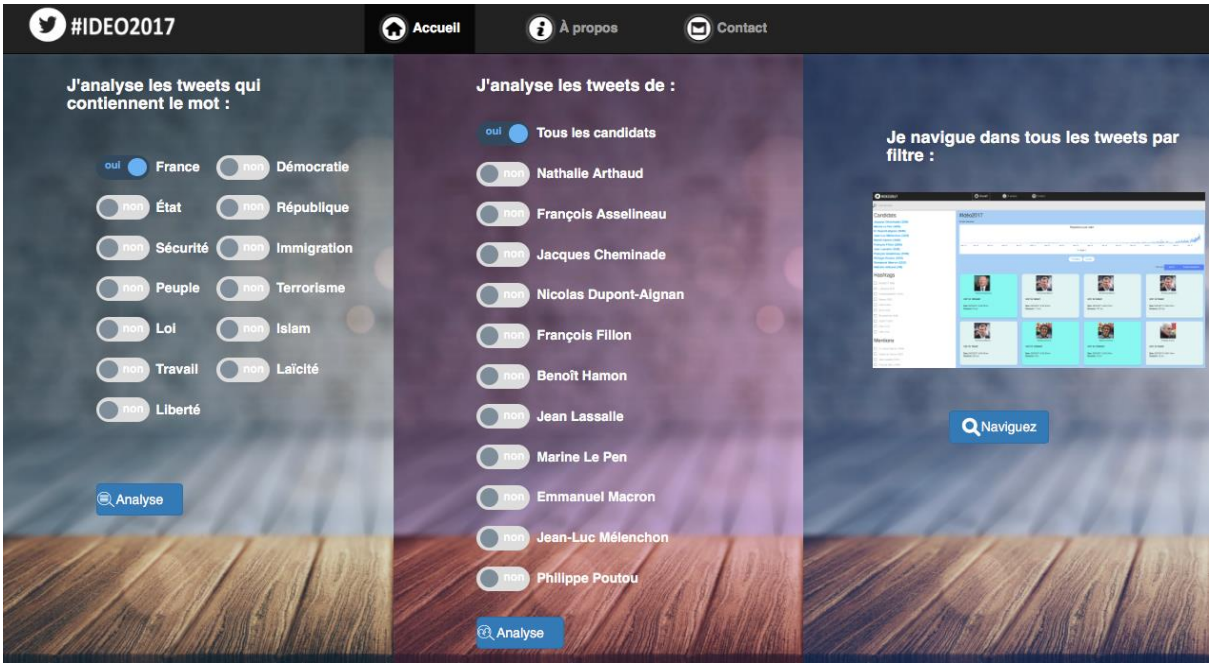


Figure 12 : Interface graphique de la plateforme #Idéo2017 (#présidentielle2017)

6.4.1 « J'analyse les tweets qui contiennent le mot… »

Durant cette analyse on propose :

- pour la plateforme #présidentielle2017 13 mots (Alduy, 2017) sur lesquels les analyses seront faites: France, état, république, peuple, loi, travail, liberté, démocratie, sécurité, immigration, terrorisme, islam et laïcité.
- pour la plateforme #législatives2017 on propose 14 mots (Alduy, 2017) : France, état, sécurité, peuple, entreprise, travail, liberté, démocratie, république, immigration, terrorisme, islam, laïcité et gouvernement.
- pour la plateforme #quinquennat2017 on a proposé aussi 14 mots (Alduy, 2017) décomposer sur trois chapeaux : (1) les sujets de fond : impôt, sécurité, culture, immigration, terrorisme, laïcité, (2) les réformes en cours : travail, salaire, social, (3) la présidence : France, démocratie, république, liberté, réforme.

Le choix de ces mots était varié pour toucher les principaux thèmes.

La recherche sur les tweets se fait sur le mot entière, mais aussi on peut trouver par exemple « travailleur, travailler, etc. » pour le mot « travail »

Par ailleurs, selon le mot choisi, on peut avoir quatre types d'analyse :

- **L'usage de ce mot par les différents candidats** : cette analyse va nous permettre d'identifier l'emploi de ce mot par les différents candidats. Le résultat de cette analyse est généré sous forme de deux graphiques, un pour le calcul des spécificités qui indique le sur-emploi et sous-emploi du mot par les candidats

(Figure 13) et l'autre pour la fréquence d'emploi de ce mot par les candidats (Figure 14).

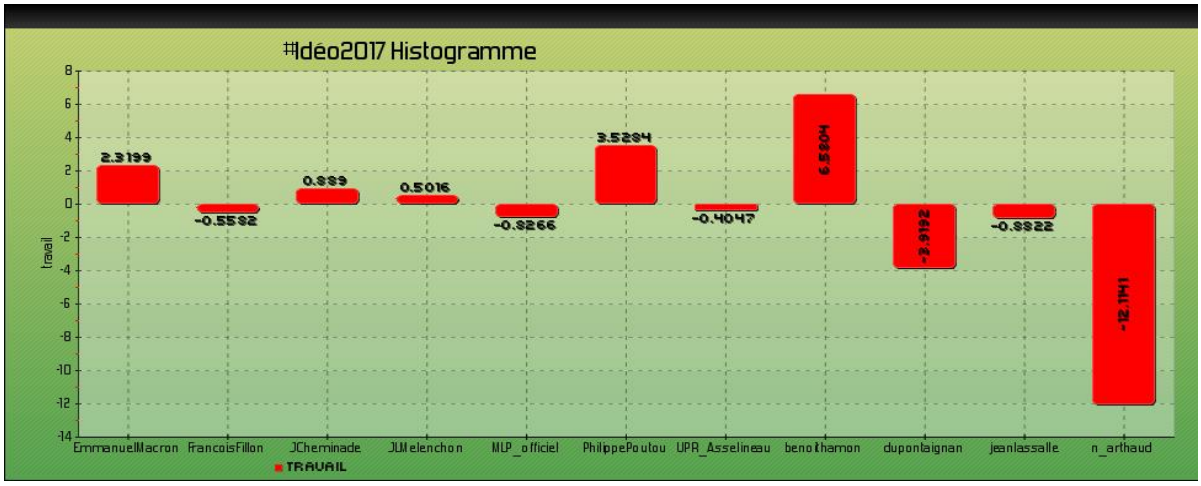


Figure 13 : Sur et sous emploi du mot « travail » par les différents candidats

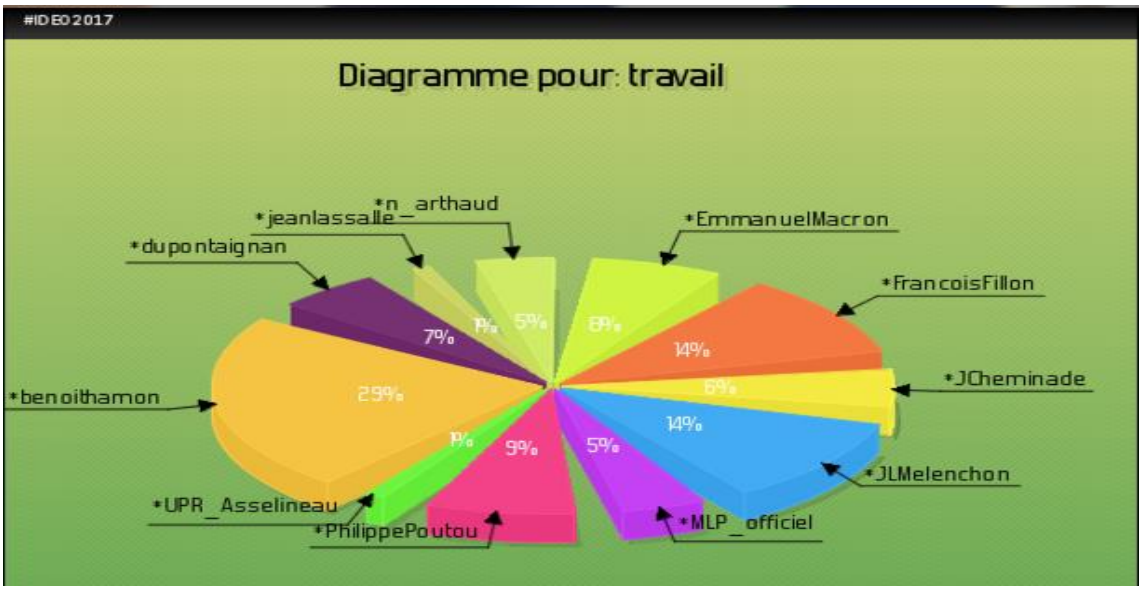


Figure 14 : L'usage du mot « travail » par les différents candidats

- **Les mots associés à ce mot pour tous les candidats** : c'est l'analyse de cooccurrences, elle est présentée sous forme d'un graphe de mots associés (Figure 15).

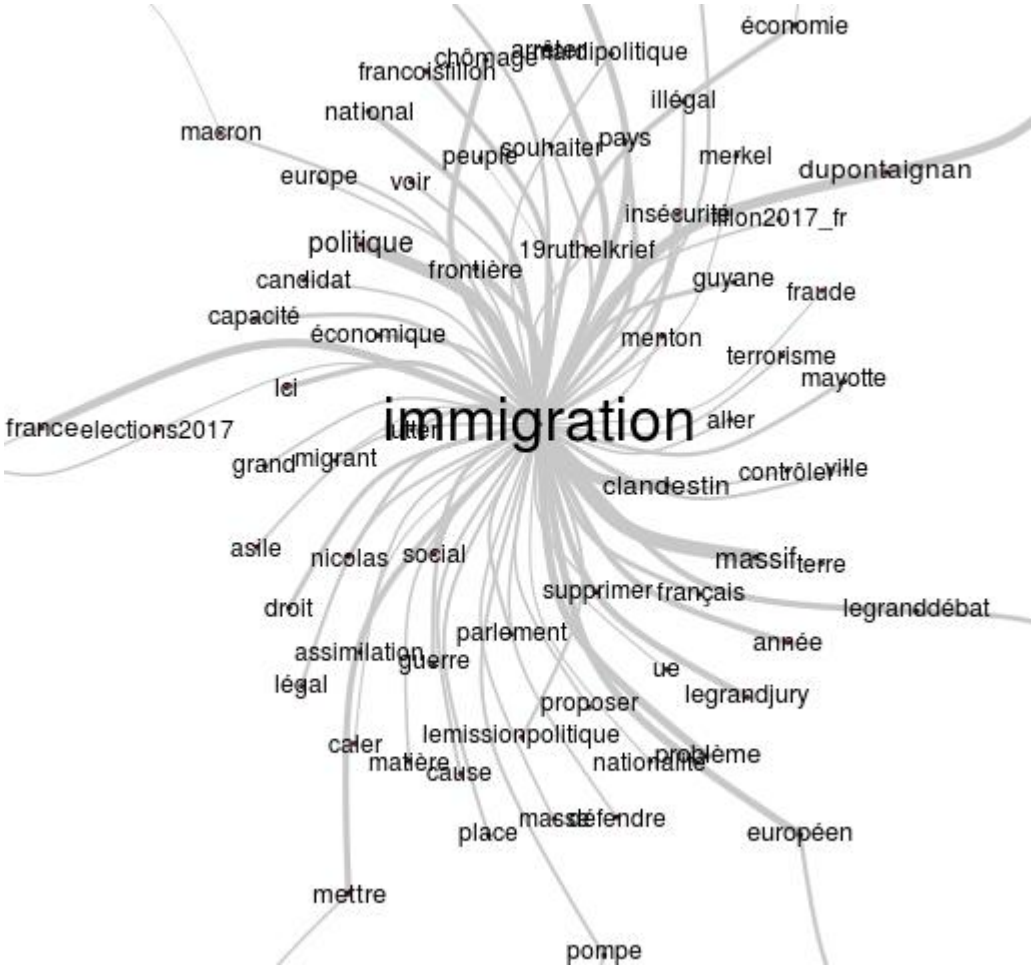


Figure 15 : Les mots associés au mot « immigration » pour tous les candidats

- **L'emploi de ce mot et ses dérivés par les différents candidats** : contrairement à la première, cette fonctionnalité nous permet d'identifier l'emploi de ce mot et ses dérivés et non pas seulement le mot exact (Figure 16). Ces trois premières analyses se font à l'aide des scripts du logiciel *IRaMuTeQ*.

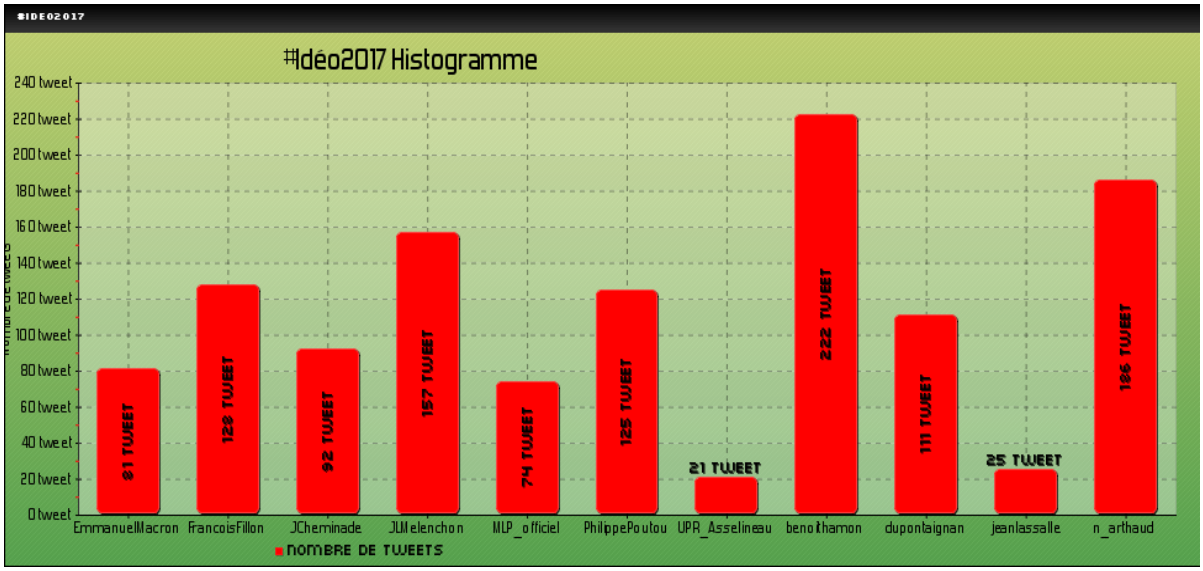


Figure 16 : L'emploi du mot travail et ses dérivés

- La dernière analyse est « **Le nuage de mots** », elle permet d'afficher le lexique des mots des tweets sur la forme d'un nuage de mots. (Figure 17)

Figure 17 : Le nuage de mots pour le mot « travail »

6.4.2 « J'analyse les tweets de... [Candidat] »

Cette fonctionnalité nous permet d'analyser les tweets :

- des 11 candidats officiels à la présidentielle 2017 : Nathalie Arthaud, François Asselineau, Jacques Cheminade, Nicolas Dupont-Aignan, François Fillon, Benoît Hamon, Jean Lassalle, Marine Le Pen, Emmanuel Macron, Jean-Luc Mélenchon et Philippe Poutou; elle permet également une analyse sur tous les candidats.
- de neuf partis politiques pour les législatives 2017 : La France insoumise, En Marche Debout la France, Union des Démocrates et Indépendants (UDI), Les Républicains, Front National, Parti socialiste, Parti Communiste Français (PCF) et Europe Écologie Les Verts (EÉLV).
- de neuf personnalités pour la plateforme #quinquennat2017 : Edouard Philippe, Emmanuel Macron, Gérard Collomb, Jean-Luc Mélenchon, Laurent Wauquiez, Marine Le Pen, Nicolas Hulot et Olivier Faure.

Une fois un des candidats choisi, les analyses linguistiques possibles sont :

- **Les mots les plus utilisés** : elle propose des statistiques simples sur les mots les plus utilisés de chaque corpus de tweets (Figure 18).

Les mots les plus utilisés de @jeanlassalle

Forme	Fréquence	Type
presidentielle2017	88	Non reconnue
jeanlassalle	74	Non reconnue
lassalle	37	Non reconnue
jean	34	Nom
france	34	Non reconnue
retrouver	30	Verbe
grand	29	Adjectif
itele	26	Non reconnue
candidat	25	Nom
français	23	Adjectif
politique	21	Adjectif

Figure 18 : Les mots les plus utilisés pour le candidat « Jean Lassalle »

- **Thématiques** : Cette analyse permet de regrouper les mots les plus proches pour aider à percevoir les différents thèmes de chaque candidat. Elle donne comme résultat un tableau qui contient la liste des mots pour chaque thème (figure 19) et un graphe qui permet de faire émerger les grands domaines lexicaux et d'en dégager, après lecture et analyse, les principaux thèmes (figure 20).

La liste des mots de chaque thème :

Le thème 1	Le thème 2	Le thème 3	Le thème 4	Le thème 5	Le thème 6	Le thème 7
jeanlassalle	jeanlassalle	presidentielle2017	presidentielle2017	homme	legranddebat	france
france	jjbourdin_rmc	jeanlassalle	retrouver	france	français	sudradio
politique	bourdindirect	lassalle	soir	itele	pays	presidentielle2017
lci	entretiendembauche	jean	inviter	presidentielle2017	mouv	venir
jeuneslassalle	bfmtv	candidat	partir	onpc	minutespourconvaincre	service
presidentielle2017	campagne	présidentiel	jeanlassalle	peuple	trouver	national
voir	direct	maire	politique	pays	retirer	monde
lelive	aller	amf2017	lci	jeanlassalle	président	agriculture
remettre	france	élection	itele	syrie	syrie	grand
aller	https	france	matin	prendre	france	passer
connaître	français	parrainages	passage	démocratie	entreprise	permettre
français	bayrou	inviter	émission	perdre	élire	commun
amour	grand	franceinfo	ami	grève	troupe	temps
rencontre	syrie	lcp	direct	berger	voir	pays
beau	presidentielle2017	débat	interview	faim	premier	jeune
lassalle2017	langue	répondre	onpc	seul	jeanlassalle	jeanlassalle
nouveau	nicolassarkozy	https	cher	dernier	ensemble	andrebercoff
homme	candidat	question	france2	gagner	état	parler
lcimatin	jour	campagne	radioclassique	directferrari	demander	état
marcher	mettre	député	h30	libre	donner	rencontrer
pays	écouter	letempsestvenu	suite	résister	ambassade	agriculteur
vie	homme	programme	heure	conscience	chercher	français
gg_rmc	reconstruire	lci	lien	femme	aller	voir
résister	parler	commun	tf1	entreprise	politique	territoire

Figure 19 : La liste des mots de chaque thème pour le candidat « Jean Lassalle »

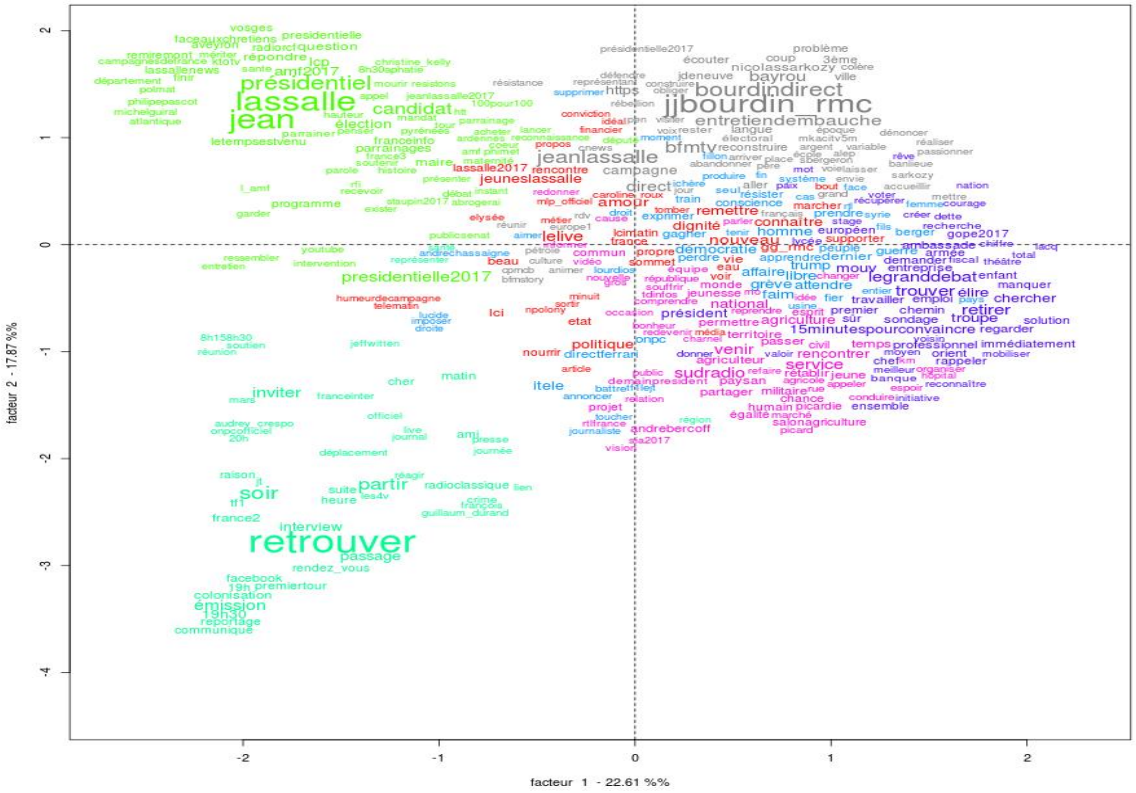


Figure 20 : Le graphe des thématiques pour le candidat « Jean Lassalle »

- **Les relations entre les mots** : permet d'analyser la similitude des mots de chaque candidat sous forme graphique (figure 21).

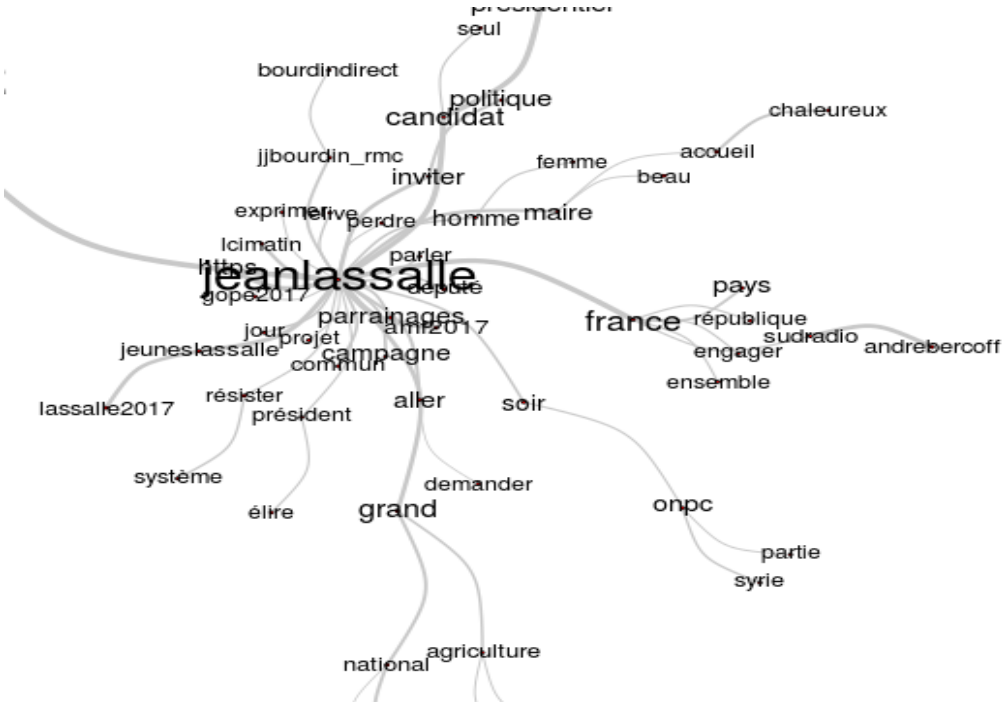


Figure 21 : Les mots associés pour le candidat « Jean Lassalle »

- **Nuage de mots** : permet d'afficher le lexique des mots des tweets sur la forme d'un nuage de mots (figure 22).

Figure 22 : Le nuage de mots pour le candidat « Jean Lassalle »

- La dernière analyse est consacrée pour le choix d'analyser tous les candidats, c'est « **Les spécificités des différents candidats** », elle permet d'identifier les mots et catégories spécifiques des différents candidats, et de les comparer.

En plus des analyses linguistiques, nous avons ajoutés des visualisations graphiques de données pour les candidats et les partis politiques choisis à l'aide de l'outil de visualisation de données *Kibana*. Pour cela, nous avons proposés quatre graphes comme suit :

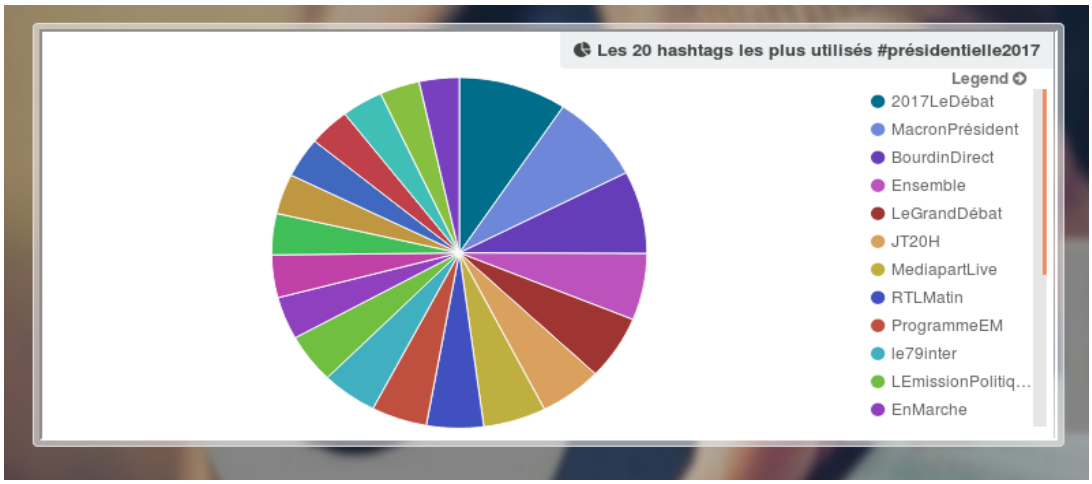


Figure 23 : Les 20 hashtags les plus utilisés par Emmanuel Macron

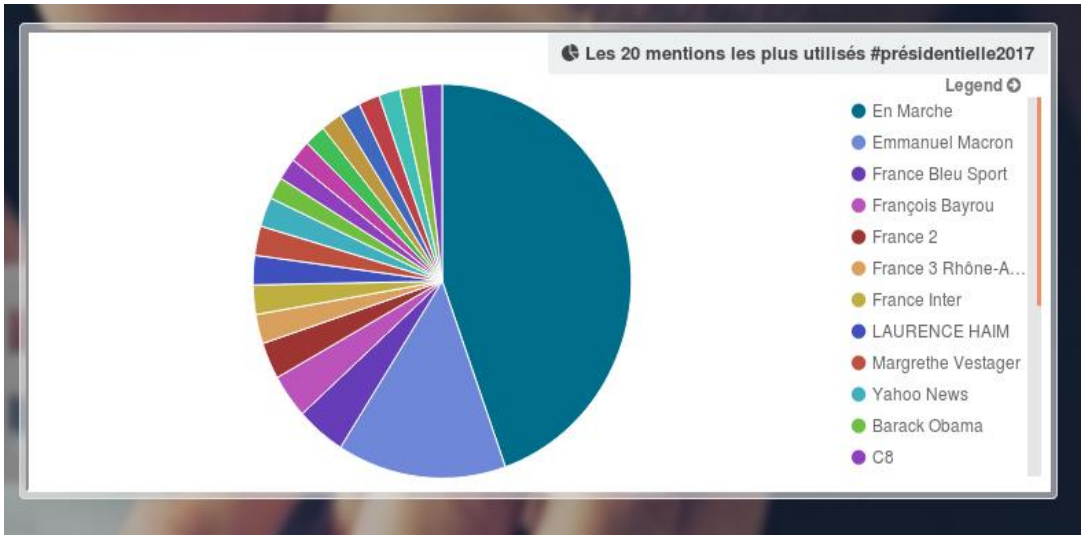


Figure 24 : Les 20 mentions les plus utilisés par Emmanuel Macron

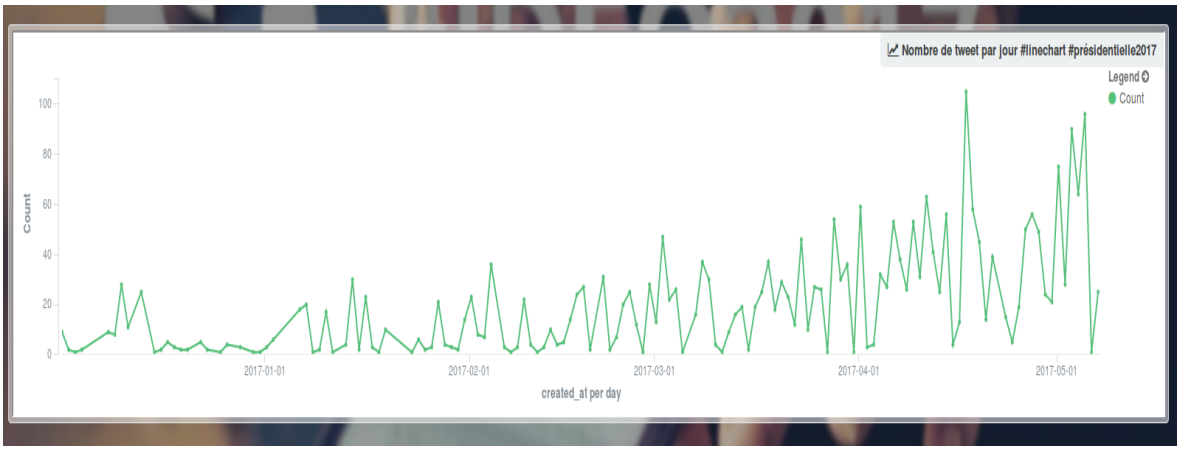


Figure 25 : Le nombre de tweets par jour pour Emmanuel Macron

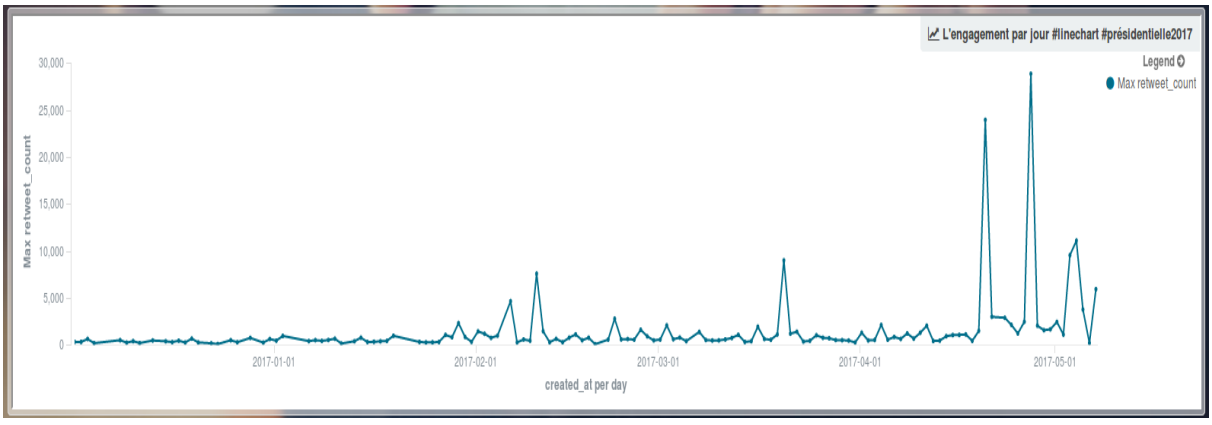


Figure 26 : Le nombre de retweets par jour pour Emmanuel Macron

6.5 Problèmes rencontrées et réflexion

Comme nous avons mentionné qu'IRaMuTeQ est exécutable en mode batch qu'avec la fonctionnalité de statistique, donc, le problème principal était de comment modifier le code source de ce logiciel pour qu'on arrive à exécuter le reste de ces fonctionnalités en ligne de commande.

Pour cela, on a appliqué des modifications au niveau de son code source, et on va faire une comparaison entre l'exécution à l'aide de l'interface graphique et l'exécution en mode batch comme suit :

6.5.1 Chargement du corpus

La première étape est de charger le corpus destiné :

Interface :

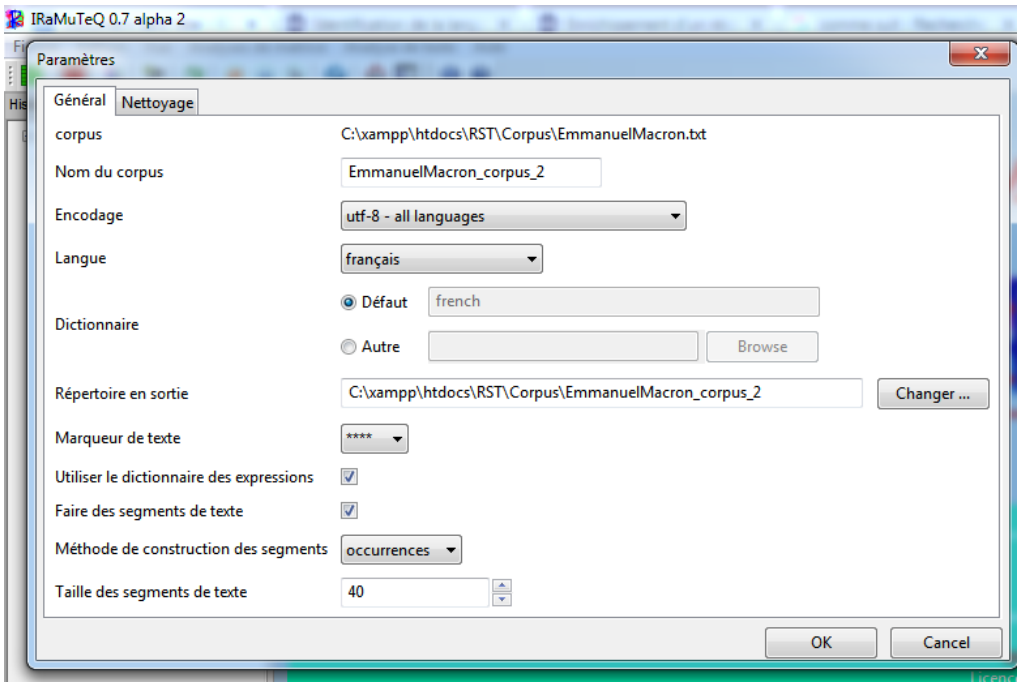


Figure 27 : L'interface du chargement du corpus d'*IRaMuTeQ*

Mode Batch

A l'aide de cette commande on peut remplacer l'interface de chargement de corpus

« *iramuteq\iracmd.py -f NomCorpus -e utf-8 -l french -b* «

- *iramuteq\iracmd.py :* Le chemin du fichier exécutable d'*IRaMuTeQ*
- *-f NomCorpus :* désigne le paramètre du nom de corpus
- *-e utf-8 :* désigne le paramètre de l'encodage

- *-l french :* désigne le paramètre qui détermine le dictionnaire (français, anglais, etc.)
- *-b : (building)* désigne le paramètre de construction d'un corpus au format .cira qui sera utilisé par *IRaMuTeQ* lors des analyses.

6.5.2 Statistiques

Interface :

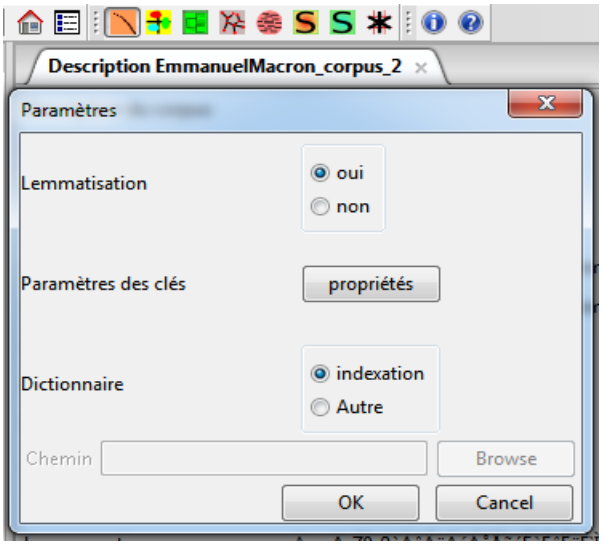


Figure 28 : L'interface de la fonctionnalité de statistique d'*IRaMuTeQ*

Mode Batch

A l'aide de cette commande on peut remplacer l'interface de la fonctionnalité de statistique :

« *iramuteq\iracmd.py -r NomCorpus_corpus_1\Corpus.cira -e utf-8 -l french -t stat* «

- *iramuteq\iracmd.py :* Le chemin du fichier exécutable d'*IRaMuTeQ*
- *-r NomCorpus_corpus_1\Corpus.cira :* désigne le paramètre du chemin du corpus.cira qui est construit lors du chargement du corpus original.
- *-e utf-8 :* désigne le paramètre de l'encodage.
- *-l french :* désigne le paramètre qui détermine le dictionnaire (français, anglais, etc.).
- *-t stat* : désigne le type d'analyse (ici statistique)

La modification au niveau du code source :

```
elif options.type_analyse == 'stat' :
    self.Text = Stat(self, corpus, parametres = {'type':'stat', 'lem':1})
```

On distingue que la lemmatisation prend la valeur 1 qui remplace l'option « oui ou non « de l'interface graphique.

À ce stade, le développeur du logiciel donne cette proposition comme solution seulement pour le chargement de corpus et la fonctionnalité de statistique. Pour le reste des fonctionnalités, nous avons modifié le code source pour arriver à faire fonctionner trois analyses qui sont : spécificité et AFC, classification et l'analyse de similitude.

6.5.3 Spécificité et AFC

Interface :

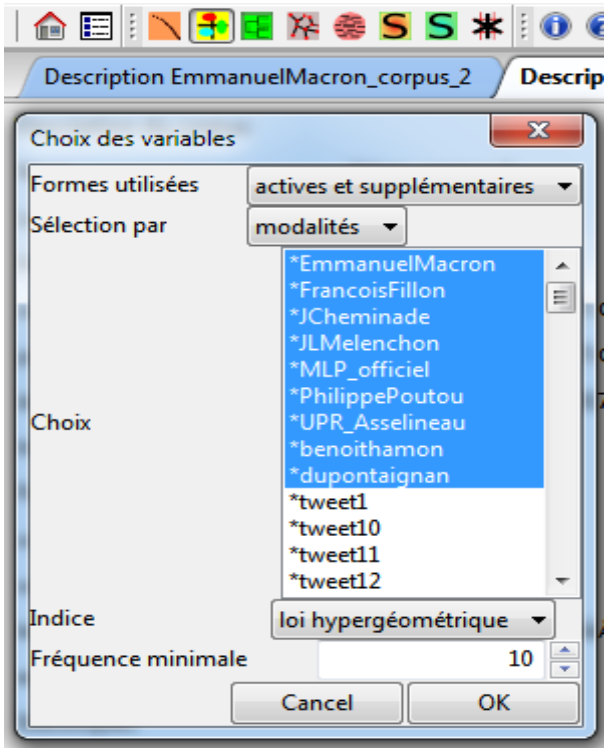


Figure 29 : L'interface de la fonctionnalité de spécificité et AFC d'*IRaMuTeQ*

Mode Batch

A l'aide de cette commande on peut remplacer l'interface de la fonctionnalité de spécificité et AFC :

« *iramuteq\iracmd.py -c Analyse.ira -r NomCorpus_corpus_1\Corpus.cira -e utf-8 -l french -t spec* «

- *-c Analyse.ira :* le nom du fichier qui contient la configuration de cette fonctionnalité
- *-t spec:* désigne le type d'analyse (spécificité et AFC)

- La modification au niveau du code source :

```
#Création d'un fichier qui va contenir la liste des variables
        etoilLi= corpus.make_etoiles()

        etoilLi.sort()
        # print etoilLi
        output=open("etoilLi.txt","w")
        for line2 in etoilLi:
            output.write(line2)
            output.write("\n")
        output.close()
elif options.type_analyse == 'spec' :
    self.Text = Lexico(self, corpus, parametres = {'type':'spec', 'lem':1, 'typeformes'
    :0, 'indice':'hypergeo', 'clnb':2, 'mineff':10})
```

Comme le montre le code ci-dessus, on récupère la liste de variables de notre corpus et on la stocke dans un fichier (etoilLi.txt). Ensuite, on lance l'analyse avec les paramètres nécessaire :

- *type* : spec (spécificité et AFC)
- *lem 1* : lemmatisation « oui »
- *typeformes 0* : les formes actives et supplémentaires prennent la valeur 0, les formes actives seulement prennent la valeur 1 et les formes supplémentaires seulement prennent la valeur 2.
- *indice* : le calcul sera avec la loi hypergéométrique ou la chi2.
- *mineff* : seuil de l'effectif de chaque forme.

Le dernier paramètre est le choix des modalités, comme le montre la figure 19, nous avons choisi neuf modalités qui sont les noms des candidats qui existent dans ce corpus. Pour cela, et au moment de la création du fichier « *etoilLi.txt* », nous allons récupérer la liste de modalités dans un autre fichier « *select.txt* ».

Ensuite, nous avons rajouté la ligne « *ListEt=RecupererSelec.recupererSelection()* » qui va permettre à la fonction *prefetences(self)* du fichier « *textaslexico.py* » d'*IRaMuTeQ* de récupérer la liste des modalités sans passer par l'interface graphique.

```
def preferences(self) :
    listet= self.corpus.make_etoiles()
    listet.sort()
    #ListEt = [listet[i] for i in dial.list_box_1.GetSelections()]
    ListEt=RecupererSelec.recupererSelection()
    self.listet = ListEt
    self.listet.sort()
    self.parametres['clnb'] = len(ListEt)
    return self.parametres
```

6.5.4 Classification

Interface :

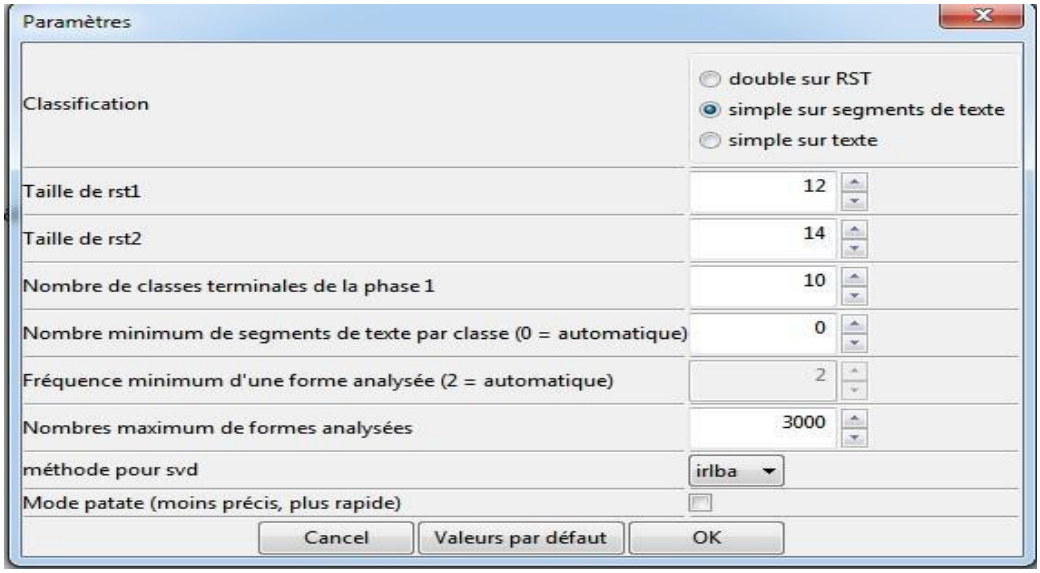


Figure 30 : L'interface de la fonctionnalité de classification d'*IRaMuTeQ*

Mode Batch:

A l'aide de cette commande on peut remplacer l'interface de la fonctionnalité de classification :

« *iramuteq\iracmd.py" -c Analyse.ira -r NomCorpus_corpus_1\Corpus.cira -e utf-8 -l french -t alceste*»

- *-c Analyse.ira :* le nom du fichier qui contient la configuration de cette fonctionnalité.
- *-t alceste:* désigne le type d'analyse (Classification).

Au niveau du code source, puisque cette fonctionnalité demande à chaque fois un nombre terminal de classes et un nombre minimum de segment du corpus, nous avons rajouté des différentes configurations dans le fichier « *iracmd.py* » d'*IRaMuTeQ* selon le nombre de tweet de chaque corpus :

```
if etoilLiG <= 20:
    self.Text = Reinert(self, corpus, parametres = {'type':'alceste', 'lem':1, 'mincl':2, 'nbcl':4,
    'max_actives':3000, 'nbcl_p1':4, 'tailleuc1':12, 'tailleuc2':14, 'classif_mode':0, 'mode.patate':0,
    'nbforme_uce':0, 'expressions':1, 'svdmethod':'irlba', 'nbactives':6, 'clnb':3, 'minforme':2,
    'eff_min_forme':3})
if (etoilLiG <= 30) and (etoilLiG > 20):
    self.Text = Reinert(self, corpus, parametres = {'type':'alceste', 'lem':1, 'mincl':3, 'nbcl':4,
    'max_actives':3000, 'nbcl_p1':5, 'tailleuc1':12, 'tailleuc2':14, 'classif_mode':0, 'mode.patate':0,
    'nbforme_uce':0, 'expressions':1, 'svdmethod':'irlba', 'nbactives':6, 'clnb':3, 'minforme':2,
    'eff_min_forme':3})
if (etoilLiG <= 60) and (etoilLiG > 30):
    self.Text = Reinert(self, corpus, parametres = {'type':'alceste', 'lem':1, 'mincl':3, 'nbcl':4,
    'max_actives':3000, 'nbcl_p1':3, 'tailleuc1':12, 'tailleuc2':14, 'classif_mode':0, 'mode.patate':0,
    'nbforme_uce':0, 'expressions':1, 'svdmethod':'irlba', 'nbactives':6, 'clnb':3, 'minforme':2,
    'eff_min_forme':3})
if (etoilLiG <= 80) and (etoilLiG > 60):
    self.Text = Reinert(self, corpus, parametres = {'type':'alceste', 'lem':1, 'mincl':3, 'nbcl':4,
    'max_actives':3000, 'nbcl_p1':9, 'tailleuc1':12, 'tailleuc2':14, 'classif_mode':0, 'mode.patate':0,
    'nbforme_uce':0, 'expressions':1, 'svdmethod':'irlba', 'nbactives':6, 'clnb':3, 'minforme':2,
    'eff_min_forme':3})
if (etoilLiG <= 110) and (etoilLiG > 80):
    self.Text = Reinert(self, corpus, parametres = {'type':'alceste', 'lem':1, 'mincl':15, 'nbcl':4,
    'max_actives':3000, 'nbcl_p1':10, 'tailleuc1':12, 'tailleuc2':14, 'classif_mode':0, 'mode.patate':0,
    'nbforme_uce':0, 'expressions':1, 'svdmethod':'irlba', 'nbactives':6, 'clnb':15, 'minforme':2,
    'eff_min_forme':3})
if (etoilLiG > 110):
    self.Text = Reinert(self, corpus, parametres = {'type':'alceste', 'lem':1, 'mincl':0, 'nbcl':4,
    'max_actives':3000, 'nbcl_p1':15, 'tailleuc1':12, 'tailleuc2':14, 'classif_mode':0, 'mode.patate':0,
    'nbforme_uce':0, 'expressions':1, 'svdmethod':'irlba', 'clnb':15, 'minforme':2, 'eff_min_forme':3})
```

On voit bien que le paramètre du nombre de classes terminales « *nbcl_pl* » change de valeur à chaque fois ce qui nous a permis de contrôler le nombre de classes selon le nombre de tweets de chaque corpus.

6.5.5 Similitude

Interface :

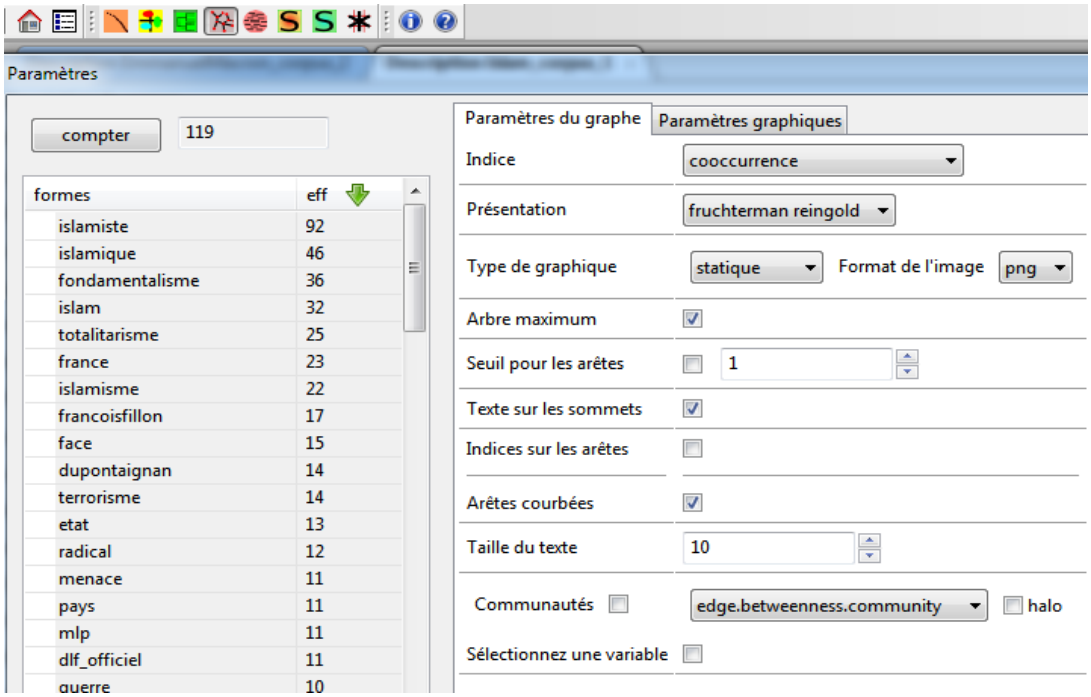


Figure 31 : L'interface de la fonctionnalité de similitude d'*IRaMuTeQ*

Mode Batch

A l'aide de cette commande on peut remplacer l'interface de la fonctionnalité de classification :

« *iramuteq\iracmd.py" -r NomCorpus_corpus_1\Corpus.cira -e utf-8 -l french -t simitxt*»

- *-t simitxt:* désigne le type d'analyse (Similitude).

```
elif options.type_analyse == 'simitxt' :
    self.Text = SimiTxt(self, corpus, parametres = {'type' : 'simitxt', 'lem':1,
    'keep_coord':0, 'coeff':1, 'layout':2, 'cols':(255, 0, 0, 255), 'cola':(200, 200,
    200, 255), 'type_graph':1, 'arbremax':1, 'coeff_tv':1,'coeff_tv_nb':0, 'tvmin':5,
    'tvmax':30, 'coeff_te':1, 'coeff_temin':1, 'coeff_temax':10, 'vcex':1, 'label_v':1,
     'label_e':0, 'seuil_ok':0, 'seuil':1, 'edgecurved':1, 'cex':10, 'film':0, 'width':
    1000,  'nbactives' : 100, 'height':1000, 'alpha':20, 'vcexmin':10, 'vcexmax':25,
    'com':0, 'bystar':0, 'communities':0, 'halo':0, 'cexfromchi':0, 'sfromchi':0, 'svg'
    :0})
```

6.6 Description du moteur de recherche

La troisième fonctionnalité de notre application est sous forme d'un moteur de recherche autonome et en temps réel. Il est censé faire des recherches sur la base de tweets avec des facettes sur le candidat, sur les hashtags ou les mentions et il permet aussi de trier les résultats selon la date ou l'engagement. L'utilité de ce moteur est de donner une liberté totale de recherche à l'utilisateur pour qu'il puisse comparer entre les candidats à l'aide des requêtes complexes.

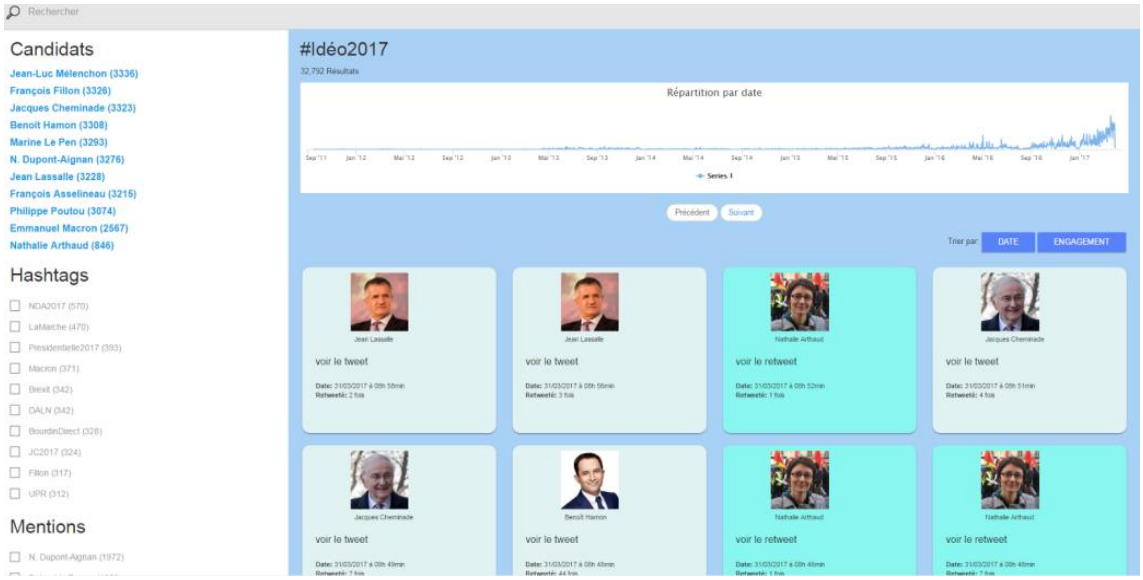


Figure 32 : Le moteur de recherche développé

6.6.1 Elasticsearch

Elasticsearch est un moteur de recherche et d'analyse *RESTful* distribué, capable de résoudre un nombre grandissant de cas d'utilisation. Élément clé de la suite *Elastic*, il stocke de manière centralisée les données et il permet d'effectuer et de combiner des recherches variées sur des données structurées, non-structurées, de géolocalisation ou indicateurs. Aussi, les agrégations d'*Elasticsearch* nous permettent d'explorer les tendances et d'identifier des modèles à partir de nos données. (Site officiel[41])

6.6.2 Elasticsearch et l'analyse linguistique ?

Dans le cadre de l'analyse linguistique, nous proposons deux types d'analyse : *une analyse par candidat* - l'utilisateur peut analyser tous les tweets d'un candidat choisi parmi les 11, et *une analyse par mot* (ou thème) - l'utilisateur peut analyser tous les tweets qui contiennent un mot spécifique choisi parmi une liste de 13 mots prédéfinie.

Parmi les problèmes techniques que nous avons rencontrés lors du développement de la fonctionnalité d'analyse linguistique, le plus important concerne la recherche d'un mot dans un texte (utilisée dans notre plateforme par exemple dans l'*analyse par mot* décrite ci-dessus). En effet, la recherche en plein texte risque de récupérer des résultats non pertinents et cela peut influencer les résultats des analyses linguistiques. Par exemple : une recherche avec le mot « loi » peut produire des résultats qu'on appelle « faux positifs » du type « emp**loi »**, « exp**loi**tation ». De plus, étant donné que nous mettons en place des analyses linguistiques, il serait nécessaire de pouvoir réaliser des recherches prenant en compte l'aspect linguistique des mots (par exemple, si nous cherchons le mot « travail », nous souhaiterions avoir comme retour les tweets contenant les mots « travail », « travailleur », etc.). Également, dans notre plateforme, nous souhaitons stocker les tweets dans une base de données, mais malheureusement les bases de données classiques ne proposent pas une fonctionnalité efficace de recherche d'un mot dans un texte ; ainsi, nous avons besoin d'un outil puissant qui nous permettrait de réaliser des recherches linguistiques en plein texte dans les tweets.

Lors de nos recherches, nous nous sommes tournés vers *Elasticsearch* car il répond aux problèmes présentés ci-dessus par l'utilisation d'un algorithme de pertinence basé sur le modèle TF/IDF[42] qui est très utilisé dans la recherche d'information. Cela nous permet de récupérer, lors d'une recherche, que les tweets les plus pertinents par rapport à la recherche faite.

6.6.3 Un moteur de recherche intelligent avec Elasticsearch

Le moteur de recherche développé dans la plateforme #Idéo2017 a pour but de proposer à l'utilisateur des recherches intelligentes à facettes sur la totalité des tweets. Une recherche à facettes permet à l'utilisateur de filtrer les tweets en choisissant un ou

[41] https://www.elastic.co/fr/products/elasticsearch (consulté le 22/08/2017)

[42] https://www.elastic.co/guide/en/elasticsearch/guide/current/scoring-theory.html (consulté le 22/08/2017)

plusieurs critères (les facettes). Dans notre plateforme, nous avons intégré trois types de facettes : (1) la première est une facette par candidat (donc une recherche que dans les tweets d'un candidat spécifique est possible), la deuxième est par hashtag, et la dernière par mentions. Par exemple: si nous cherchons le mot « université », nous pouvons savoir qui sont les candidats qui parlent plus de l'université, les hashtags liés à ce mot ou les utilisateurs qui ont été mentionnés dans les tweets qui sont liés au mot « université ».

Pour le développement du moteur de recherche, nous nous sommes questionnés sur plusieurs points. Premièrement, nous avons souhaité utiliser un système de stockage de nos tweets qui nous permettra de réaliser des recherches efficaces à facettes. Deuxièmement, comme notre projet est destiné au grand public, la plateforme peut recevoir un nombre important de requêtes en même temps, et donc, le système choisi devra être capable de gérer ces requêtes sans temps d'attente.

Pour répondre aux problèmes exprimés ci-dessus, nous avons opté pour l'utilisation d'*Elasticsearch* qui propose une représentation de l'information sous la forme d'un index clustérisé ce qui nous permet non seulement de faire des recherches textuelles, mais aussi d'agréger ces données sur plusieurs facettes. De plus, pour garder notre application fonctionnelle même en cas d'un nombre important de requêtes, *Elasticsearch* propose la création d'un cluster avec plusieurs nœuds en répartissant la charge des requêtes entre les nœuds, et en réalisant une sauvegarde automatique et répliquée des données.

6.6.4 Sécurité

L'interface du moteur de recherche développé est basée sur le framework AngularJS, et, par conséquent, les données d'*Elasticsearch* seront ouvertes au public. Afin d'empêcher l'accès à nos données, nous avons mis en place un contrôle d'accès à nos données en appliquant une restriction sur les permissions d'accès à *Elasticsearch* avec l'attribution de la permission lecture-seule au public.

Elasticsearch et *Kibana* (dont l'utilisation sera décrite ci-dessous) utilisent respectivement les ports 9200 et 5601 ; ainsi, pour éviter l'accès direct du client à l'index à travers les ports, nous avons mis en place, comme indiqué dans la Figure 2, un *reverse proxy* qui permet de contrôler toutes les demandes d'accès directes de l'extérieur et qui n'autorise que la méthode GET. Ce proxy d'Apache devient un intermédiaire pour crypter le trafic entrant. Donc, le trafic externe sera converti en trafic interne entre Apache et *Elasticsearch.*

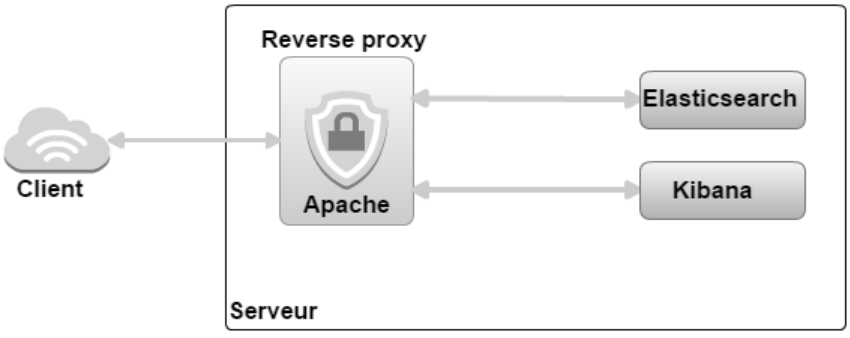


Figure 33 : Schéma du contrôle d'accès client-serveur

6.7 Visualisation de données

6.7.1 Kibana

Nous utilisons *Kibana* dans le but de réaliser des analyses sur nos données textuelles sous forme de graphes. La figure 34 montre l'évolution de la moyenne par mois du nombre de retweets pour chaque candidat durant les six derniers mois. Nous pouvons remarquer un pic de retweets pendant le dernier mois qui précède l'élection présidentielle pour plusieurs des candidats, les premiers étant Philippe Poutou, Jean-Luc Mélenchon et Benoît Hamon.

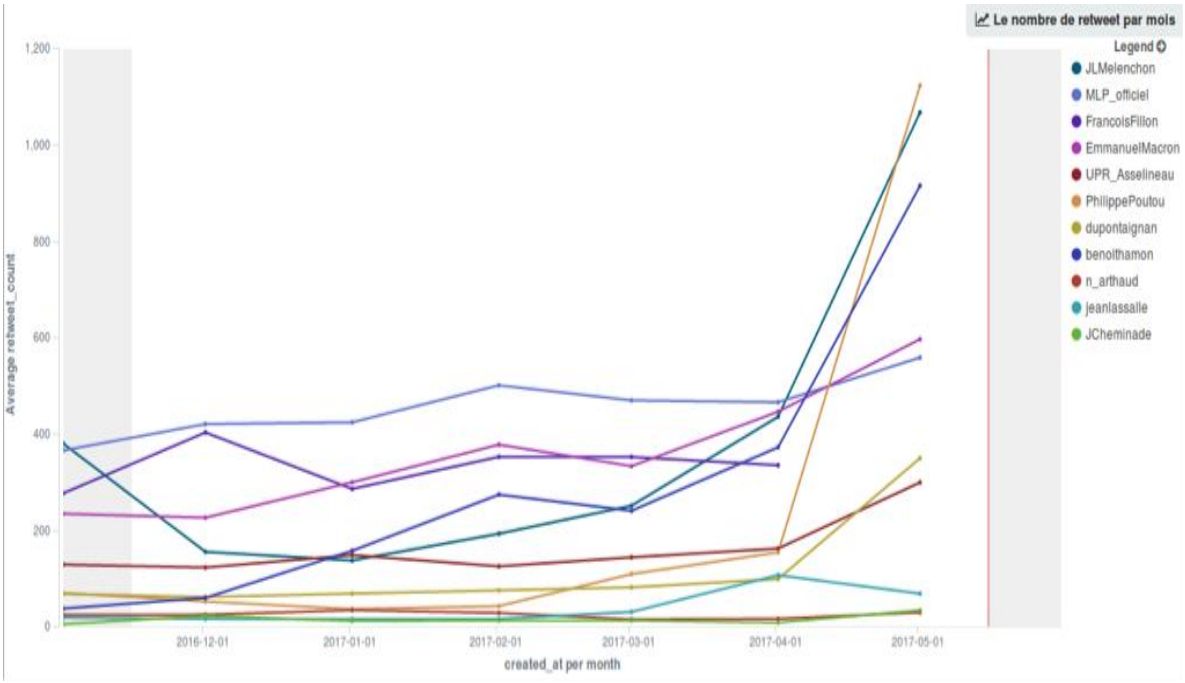


Figure 34 : Évolution de la moyenne par mois du nombre de retweets pour chaque candidat durant les six derniers mois

La figure 35 se compose de deux diagrammes de type camembert : celui de l'intérieur représente la décomposition des 11 candidats selon le nombre de tweets et celui de l'extérieur représente le pourcentage des cinq utilisateurs Twitter les plus mentionnés par les candidats.

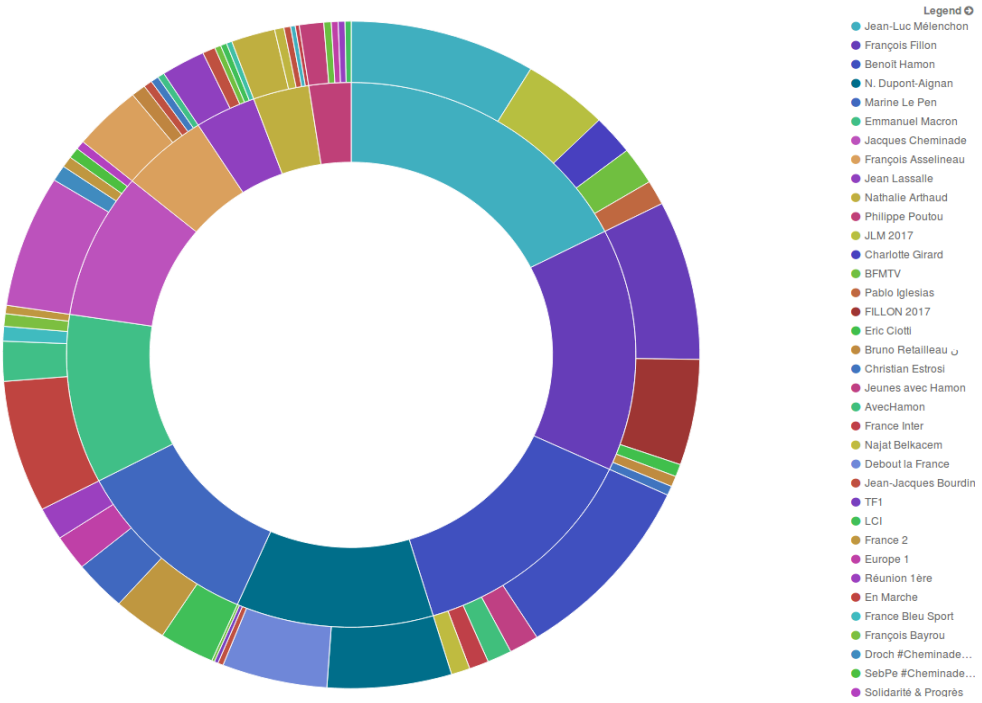


Figure 35 : Décomposition des candidats selon le nombre de tweets et le pourcentage des cinq utilisateurs Twitter les plus mentionnés par les candidats

La figure 36 est un diagramme à bandes verticales ; ici nous comparons les 11 candidats par rapport aux cinq hashtags les plus utilisés : #BourdinDirect, #Macron, #Presidentielle2017, #Fillon, #LeGrandDébat.

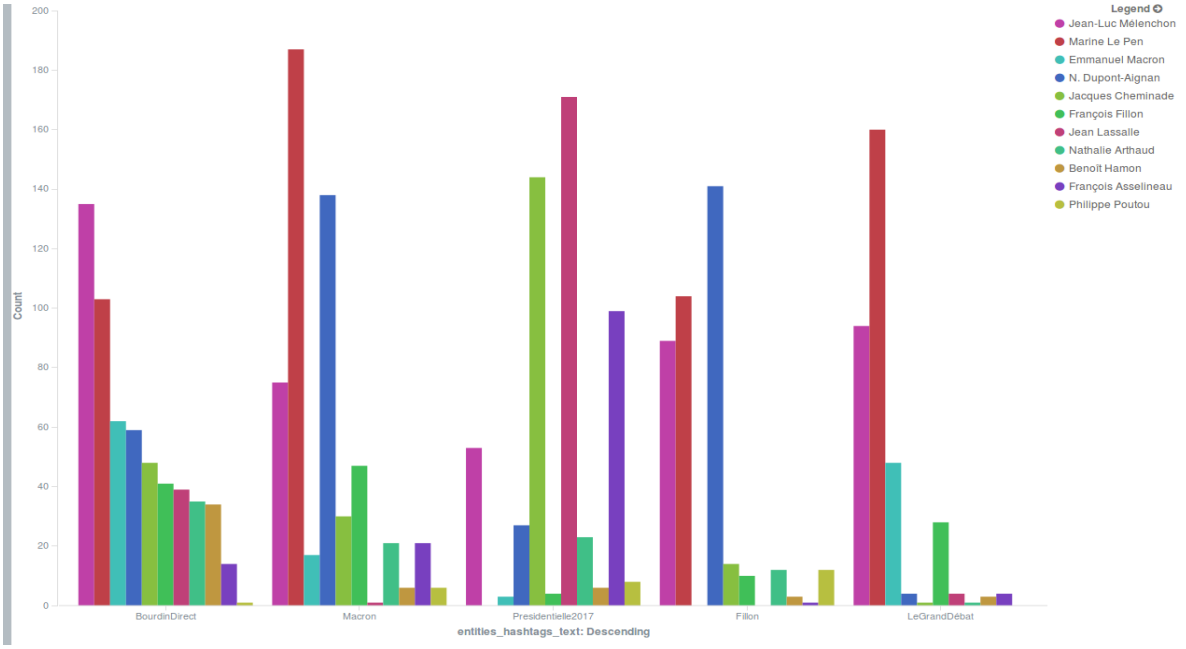


Figure 36 : Comparaison des candidats par rapport aux 5 hashtags les plus utilisés

6.8 Développement de l'outil #Idéo2017

Dans la figure 37, nous présentons l'architecture globale de la plateforme développée, ainsi que la relation entre les éléments technologiques centraux.

Après la récupération des tweets, nous stockons toutes les informations liées à ces derniers dans une base de données centrale NoSql qui est MongoDB, son avantage consiste en une structure flexible orientée documents qui ne nécessite pas des requêtes complexes pour l'accès aux données. Ensuite, nous sélectionnons les informations dont nous avons besoin pour indexer les tweets, comme par exemple :

- Le contenu du tweet
- Le compte de la personne qui a posté le tweet (twittos)
- La date de création du tweet
- Le nombre de retweets
- Les mentions
- Les hashtags
- etc.

Pour la mise en place de l'index, notre choix s'est porté vers *Elasticsearch* car pour l'indexation, il propose plusieurs types de configurations pour réaliser le mapping (qui indique comment les données seront stockées et indexées) :

- La configuration *analyzer* est proposée pour définir la langue du corpus et l'analyseur peut décomposer le texte en tokens selon la langue choisie. Pour notre plateforme nous avons utilisé la langue française.
- La configuration *normalizer* permet de transformer tous les mots en minuscules ou en code ASCII, etc.
- Le choix des champs qui seront traités lors de la recherche.
- Le type de chaque champ
- Le format de la date
- Etc.

L'index construit ci-dessus sera exploité dans un premier temps dans le développement des analyses linguistiques. A cette fin, nous avons utilisé l'API *Elasticsearch*-PHP pour faire les analyses en temps réel. Quand l'utilisateur choisi un candidat/un parti ou un mot, le corpus se crée automatiquement (en sélectionnant uniquement les tweets du candidat/parti choisi ou les tweets qui contiennent le mot choisi), et il sera mis à jour pour pouvoir appliquer nos analyses sur un sous-ensemble du corpus de tweets.

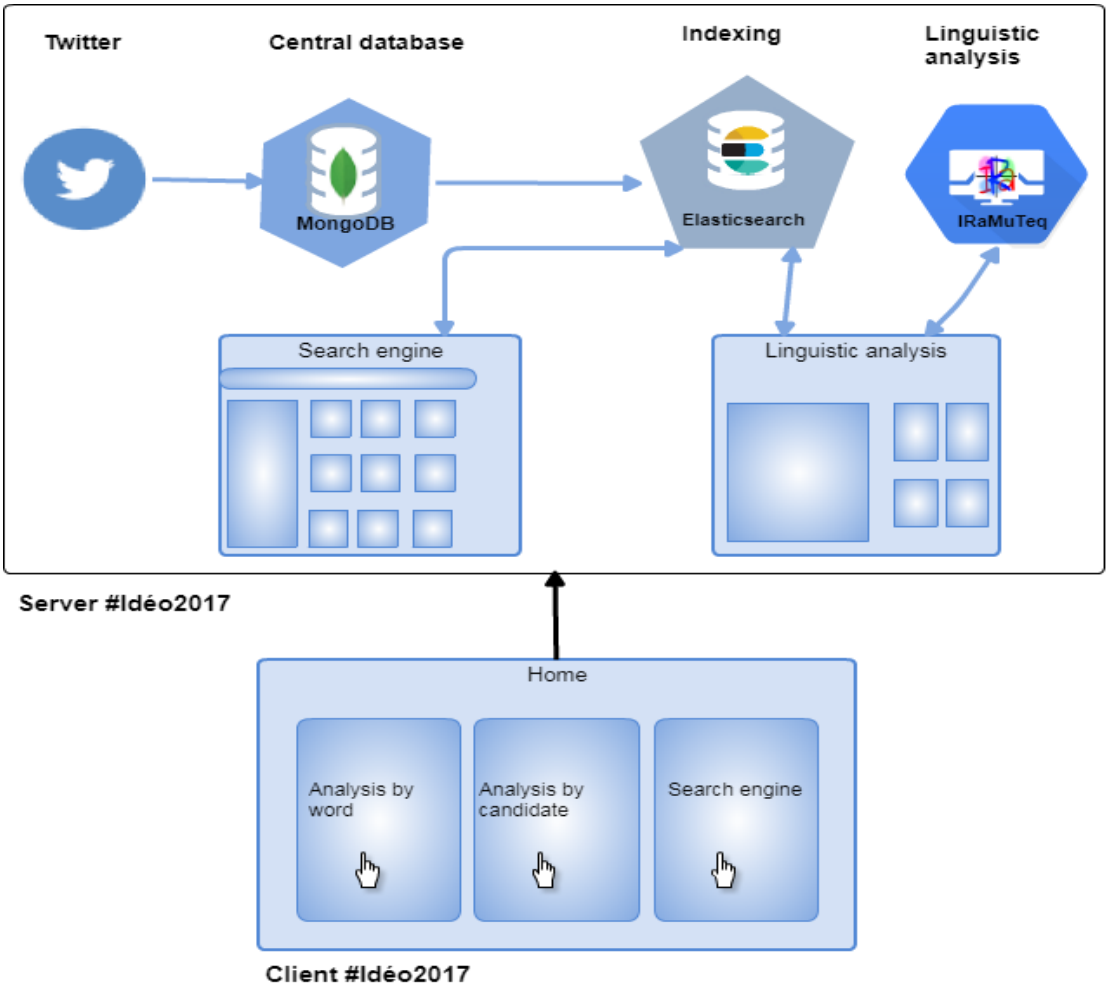


Figure 37 : Architecture globale de la plateforme #Idéo2017

Dans un deuxième temps, l'index construit est utilisé dans la mise en place de notre moteur de recherche à facettes décrit dans la section précédente. Pour cela, nous avons utilisé ElasticUI[43] qui est développée en AngularJS et qui nous a permis de mettre en place nos facettes et les résultats des recherches. L'utilisateur a le choix entre faire des recherches simples sur un mot, ou croiser les recherches simples avec le choix d'un candidat, d'un hashtag ou d'une mention, et les résultats seront affichés en temps réel.

Aussi, nous avons opté pour un cluster avec deux nœuds pour profiter de la gestion efficace du cluster réalisée par *Elasticsearch*, comme par exemple : la sauvegarde automatique et répliquée des données, l'interrogation via les API REST et la répartition de la charge entre les deux nœuds. Nous envisageons d'augmenter le nombre de nœuds dans notre cluster pour les prochaines utilisations de notre plateforme.

En plus de la précision, l'une des caractéristiques les plus importantes pour le choix d'*Elasticsearch*, c'est sa rapidité. Au niveau de la récupération d'une grande quantité de données. *Elasticsearch* nous a permis de surmonter le problème de lenteur qu'on peut avoir au niveau de page web lors de la récupération d'un tel volume de données.

[43] http://www.elasticui.com/

Afin de nous différencier du moteur de recherche présent sur l'interface de Twitter, nous avons conçu notre outil de recherche comme un système hybride, associant les réponses des tweets d'une recherche en temps réel à une synthèse de plusieurs tweets par agrégation de l'information via les facettes et les calculs linguistiques de *clustering* ou de nuages de mots.

Ainsi pour un mot ou un thème particulier, notre objectif est de donner accès aux tweets originels pour chaque candidat mais également de connaître la répartition exacte du nombre de tweets par candidat ou par grande thématique.

Cette connaissance de la distribution de tweets nous offre une contextualisation globale pour chaque requête car notre objectif est autant de réaliser un moteur de recherche que d'offrir à nos utilisateurs une plateforme de veille concurrentielle entre les différentes stratégies de communication des candidats. Rappelons que ces deux applications (moteur et système de veille) partent de postulats opposés. En effet, si le moteur de recherche ambitionne de lutter contre le bruit (toutes les réponses doivent être le plus pertinentes possibles), un système de veille comparative, type benchmark, se fixe comme ambition de réduire le silence (aucun tweet pertinent ne doit échapper à l'usager). Or lorsque l'on tente de réduire le silence, on augmente le bruit, et plus on lutte contre le bruit, plus le silence devient assourdissant. En sciences de l'information, on évalue cette complexité par le taux de précision et de rappel qui sont deux équations qui modélisent parfaitement ces valeurs diamétralement opposées que sont le bruit et le silence.

Pour relever ce défi, nous nous sommes inspirés des applications de BI (business intelligence), d'outils de reporting et de système de cartographie de l'information. Habituellement, réservés à des outils de *dashboarding* ou de *back office*, nous proposons au grand public une extension des résultats de requêtes par l'intégration visuelle et progressive d'une information synthétique par analyse linguistique directement accessible sur notre front office.

7 Conclusion

Pour conclure, #idéo2017 est la continuité d'un travail qui a commencé depuis 2014 autour des élections municipales, et nous avons réussi à proposer une plateforme qui analyse le discours des politiciens et les partis politiques sur Twitter à travers l'extraction des mots les plus utilisés, des thématiques, les relations entre les mots, un nuage de mots et la visualisation des données, et aussi, de proposer un moteur de recherche par facette qui facilite la recherche sur l'ensemble des tweets.

De plus, #idéo2017 a pris l'attention des médias, notamment avec un reportage tourné au sein du laboratoire ETIS par *VOnews 95*[44] et un article sur Le Parisien[45], LA

[44] http://95.telif.tv/2017/04/20/ideo2017-le-site-qui-decortique-les-tweets-des-candidats-a-la-presidentielle-video/
[45] http://www.leparisien.fr/val-d-oise-95/ideo2017-une-plate-forme-d-analyse-des-tweets-politiques-made-in-val-d-oise-05-03-2017-6734049.php

MONTAGNE[46], France Culture[47], etc. Cela montre que le projet a reçu l'intention du public au niveau local.

8 Constitution du corpus

Durant la dernière phase de ce projet, l'équipe #Idéo2017 est en train de constituer un corpus de données de format TEI[48]. Pour cela, et durant la campagne présidentielle 2017, nous avons rassemblé les tweets politiques à partir des comptes officiels des candidats en temps réel. Ces tweets sont les publications de chaque candidat, donc, cela va nous permettre de créer une archive de leurs discours.

Un exemple des tweets qui se trouve dans les comptes de quelques candidats :

Figure 38 : Tweets de @EmmanuelMacron[49], @FrancoisFillon[50], @MLP_officiel[51] et @JLMelenchon[52] - Captures d'écran

Pour constituer notre corpus, on va utiliser la méthodologie de constitution de corpus élaborée dans un précédent projet qui est *CoMeRe*. Il était piloté par Thierry Chanier. Le corpus est intitulé « *Polititweets* », Longhi Julien *et al.* (2014).

Le corpus *Polititweets* a été constitué en 2014. Ses données sont exploitables sous la forme d'un fichier XML qui permet de baliser les informations selon l'importance et de

[46] http://www.lamontagne.fr/paris/internet-multimedia/politique/2017/04/16/ce-que-leurs-tweets-revelent-des-candidats-a-la-presidentielle_12367032.html
[47] https://www.franceculture.fr/emissions/le-numerique-et-nous/le-tweet-un-genre-du-discours-politique
[48] Format XML de représentation de texte : http://www.tei-c.org/
[49] https://twitter.com/EmmanuelMacron/status/822125404769153025
[50] https://twitter.com/FrancoisFillon/status/822167782519349248
[51] https://twitter.com/MLP_officiel/status/822344928852410368
[52] https://twitter.com/JLMelenchon/status/823784187924742144

manière organisée, et il était constitué selon les différents standards du format TEI comme le montre la figure suivante :

```
<post xml:id="cmr-politweets-a388206741545959424" who="#cmr-politweets-p109320501"
 when="2013-10-10T09:37:54" xml:lang="fra">
 <p>Comment développer chez nos enfants l'esprit d'entreprendre ? vos témoignages et
  expériences m'intéressent ...<ref target="https://t.co/FbOlpkfvRl"
   >https://t.co/FbOlpkfvRl</ref></p>
 <trailer>
  <fs>
   <f name="medium">
    <string>web</string>
   </f>
   <f name="favoritecount">
    <numeric value="3"/>
   </f>
   <f name="retweetcount">
    <numeric value="18"/>
   </f>
  </fs>
 </trailer>
</post>
```

Figure 39 : Exemple d'analyse d'un tweet du corpus Polititweets

Cet extrait du corpus nous permet de visualiser un tweet bien balisé:

- <post></post>: c'est la balise qui englobe un tweet avec ses informations comme l'id, who (qui), when (quand), langue
- : où on peut trouver le tweet en texte
- <trailer></trailer>: cette balise contient les informations supplémentaire pour un tweet, comme: medium (l'outil utilisé pour accéder au site et partager le tweet comme : web, iphone, etc.), favoritecount, retweetcount, inReplyToUserId, inReplyToStatusId, etc.

Après le stockage de données durant les élections présidentielles dans une base, on va passer à la finalisation du corpus selon le manuel de Longhi J. *et al* (2014) par un mois d'ingénieur et la création d'une interface qui va permettre de faire des recherche sur le corpus prévu.

Pour conclure, ce corpus est une bonne piste pour les utilisateurs intéressés par l'analyse du discours, parce qu'il sera considéré comme une archive de l'élection présidentielle en France de 2017.

Conclusion

Nous avons réussi à mettre en ligne trois versions[53] de notre plateforme qui sert à analyser le discours des politiciens sur Twitter et à proposer des analyses linguistiques qui exigent une interprétation manuelle de l'utilisateur dans certains cas, des analyses statistiques et de la visualisation de données. Cette plateforme présente également un moteur de recherche par facettes pour la recherche sur la base de tweets a également été développé

Durant la première partie, nous avons montré l'existence de plusieurs types d'analyses possible sur les tweets et particulièrement les tweets politiques. Nous avons fait une comparaison entre quelques outils d'analyses textuelles qui nous a conduits à choisir *IRaMuTeQ*. Il est en effet le plus compatible avec nos critères techniques et fonctionnels, notamment face au besoin de le convertir en version web. Par conséquent, l'utilisation de cette version est plus facile à comprendre que le logiciel en question. Nous avons également décrit les différentes fonctionnalités de cet outil pour avoir une idée claire sur ce qu'il peut proposer comme analyses linguistiques.

Dans la deuxième partie, nous avons fait une description du projet et de ses étapes de développement. Tout d'abord, la description de l'outil #Idéo2017, ensuite, une description de la chaîne de traitement, puis, nous avons décrit les analyses linguistiques proposées par les deux parties principales : l'analyse des tweets d'un candidat et l'analyse par mots. Ensuite, nous avons montré les problèmes rencontrés avec l'outil d'analyse textuelle choisi et les modifications que nous avons apportées à son code source. De même, nous avons expliqué le fonctionnement du moteur de recherche développé et sa relation avec la linguistique et le choix d'*Elasticsearch* et ses avantages. Aussi, nous avons proposé des visualisations graphiques de nos données avec des diagrammes de type camembert et des *line chart,* ce qui nous a permis de faire une fusion entre le texte et le graphique. Finalement, nous avons expliqué le développement de l'outil avec une architecture qui montre les enchaînements entre les différentes étapes.

Ce que je peux retenir de ce stage, c'est que j'ai appris l'esprit d'équipe et le partage des tâches entre les différents contributeurs à ce projet. J'ai également pu avoir une bonne expérience au niveau de la recherche avec la participation à deux articles (l'un d'eux sera publié prochainement à la conférence *cmccorpora*[54]). Côté développement, cette partie a été très enrichissante au niveau personnel puisque nous avons utilisé plusieurs outils et langages de programmation, notamment *IRaMuTeQ, Elasticsearch, Kibana, MongoDB, MySQL, Apache, AngularJS, Bootstrap, PHP, Java, Python* et le mode batch, et cela m'a permis d'améliorer mes compétences techniques.

Par contre, concernant le développement de la plateforme, nous avons été pressés par le temps, puisque le stage a commencé le 1er février 2017 et l'élection présidentielle se déroule le 7 mai 2017, donc, nous étions obligés d'accélérer le rythme du travail, et nous avons réussi à lancer la première version de la plateforme #présidentielle2017, le 29

[53] #Présidentielle2017, #Législatives2017 et #Quinquennat2017
[54] https://cmc-corpora2017.eurac.edu/speakers/

mars 2017. Et par la suite nous avons lancé la deuxième version #législatives2017 et la troisième #quinquennat2017.

Cependant, et statistiquement[55] parlant, entre le premier jour du lancement et le dernier jour des élections législatives (du 29 mars 2017 jusqu'au 18 juin 2017), la plateforme a été consultée par 1105 utilisateurs partagés sur 38 pays où la France qui est en tête du classement avec 940 utilisateurs, les États-Unis avec 18 utilisateurs, le Canada 17 utilisateurs et la Tunisie avec 12 utilisateurs, etc. 11351 pages ont été vues, avec 5 min comme durée moyenne des sessions. Ces résultats sont très encourageants et montre l'intérêt qu'il y a eu pour le travail réalisé dans ce stage avec l'équipe d'#Idéo2017.

[55] D'après les statistiques de *Google Analytics*

Perspectives

Dans cette partie, nous allons évoquer les différents possibles pistes d'améliorations à la suite de ce stage.

D'abord, nous pourrions améliorer les analyses existantes comme le nuage de mots et la relation entre les mots, et les rendre dynamiques ou en 3D, de telle façon que l'utilisateur puisse naviguer dans les différents graphes.

Deuxièmement, la migration vers une version supérieure d'*Elasticsearch* et *Kibana* serait intéressante parce qu'ils proposent d'autres fonctionnalités plus robustes que la version utilisée.

Troisièmement, l'utilisation d'autres outils d'analyses textuelles permettrait d'autres développements. Toutefois, il faut travailler sur leur code source pour que nous puissions exploiter les résultats obtenus en version web. Cela va nous permettre de proposer aux utilisateurs qui ont un intérêt à l'analyse de discours politique une plateforme riche et diversifiée.

À long terme, nous pourrions analyser non seulement les discours politiques sur Twitter, mais aussi les discours télévisés produits par des spécialistes, les articles des journaux, etc. afin de faire une comparaison avec les analyses faites sur le discours sur Twitter sur les centres d'intérêt, les thèmes et les différents sujets abordés par une seule personnalité politique.

Bibliographie

[Alduy C. 2017] : livre « Ce qu'ils disent vraiment. Les politiques pris aux mots » http://tempsreel.nouvelobs.com/rue89/rue89-nos-vies-connectees/20170125.OBS4322/ordre-immigration-camarade-comment-ont-ete-analyses-les-mots-des-politiques.html (consulté le 27/07/2017)

[Andranik Tumasjan *et al.,* 2010] : Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner, Isabell M. Welpe (2010). *Predicting Election s with Twitter- What 140 Characters Reveal about Political Sentiment.*

[Balahur *et al.,* 2010] : Balahur, A., Steinberger, R., Kabadjov, M., Zavarella, V., Goot, E. v. d., Halkia, M., et al. (2010). *Sentiment analysis in the news. Proceedings of the Seventh conference on International Language Resources and Evaluation, Retrieved May 25, 2010 from*: http://www.lrec-conf.org/proceedings/lrec2010/pdf/2909_Paper.pdf.

[Benhardus J. et Kalita J., 2013] : '*Streaming trend detection in Twitter', Int. J. Web Based Communities*, Vol. 9, No. 1, pp.122–139.

[Chinchor N. ,1998] : Chinchor N. (1998), « *MUC-7 Named Entity Task Definition (version 3.5) », in Proceedings of the 7th Message Understanding Conference (MUC-7), 19 April-1 May 1998, Fairfax, VA*.

[Choy *et al.* 2011] : Choy, M., Cheong, L. F. M., Ma, N. L., & Koo, P. S. (2011). *A sentiment analysis of Singapore Presidential Election 2011 using Twitter data with census correction.*

[Djemili S. *et al,* 2014] : Sarah Djemili, Julien Longhi, Claudia Marinica, Dimitris Kotzinos, Georges-Elia Sarfati, (2014). *What does Twitter have to say about ideology?. Gertrud Faaß & Josef Ruppenhofer. NLP 4 CMC: Natural Language Processing for Computer-Mediated Communication / Social Media - Pre-conference workshop at Konvens 2014, Oct 2014, Hildesheim, Germany. Universit¨atsverlag Hildesheim*, 1, http://www.uni-hildesheim.de/konvens2014/data/konvens2014- workshop-proceedings.pdf: p.16-25, 2014.

[Dridi H-E. et LEPALME, 2014] : Houssem Eddine DRIDI *et* Guy LAPALME, 2014, Détection d'évènements à partir de Twitter.

[Françoise Fessant] : Cours d'apprentissage non supervisé, http://www.vincentlemaire-labs.fr/cours/2.2-ApprentissageNonSupervise.pdf

[Guille, A. et Favre, C. 2014] : Une méthode pour la détection de thématiques populaires sur Twitter

[Hasan Maryam *et al.,* 2014] : Maryam Hasan, Elke Rundensteiner, Emmanuel Agu (2014) *Detecting Emotions in Twitter Messages*.

[Heiden S. 2010b] : *The TXM Platform : Building Open-Source Textual Analysis Software Compatible with the TEI Encoding Scheme. In Ryo Otoguro, Kiyoshi Ishikawa,*

Hiroshi Umemoto, Kei Yoshimoto, Yasunari Harada (Ed.), 24th Pacific Asia Conference on Language, Information and Computation - PACLIC24 (p. 389-398). Institute for Digital Enhancement of Cognitive Development, Waseda University, Sendai, Japan.

[Johnson K et Goldwasser D, 2016] : Kristen Johnson & Dan Goldwasser, (2016): *Identifying Stance by Analyzing Political Discourse on Twitter, Related work,* p.67.

[Kaplan et Haenlein, 2010] : Andreas Kaplan, Michael Haenlein 2010 *Users of the world, unite! The challenges and opportunities of social media, Business Horizons*, vol. 5, n°1, 2010

[Longhi, 2012] : Longhi Julien. Discours, style, format : contraintes et niveaux de structuration de la textualité des Tweets de Mouloud. 3e Congrès Mondial de Linguistique Française, Jul 2012, France. pp.1127-1141, 2012. <halshs-00944636>

[Longhi Julien *et al.* 2014] : Longhi, J., Marinica, C., Borzic, B., Alkhouli, A., 2014, Polititweets, corpus de tweets provenant de comptes politiques influents. In Chanier T. (ed) Banque de corpus CoMeRe. Ortolang.fr : Nancy. [cmr-polititweets-tei-v1]

[Longhi Julien *et al.*, 2016] : Longhi Julien, Claudia Marinica *et* Haddioui Naoual, (2016): Extraction automatique de phénomènes linguistiques dans un corpus de tweets politiques : quelques éléments méthodologiques et applicatifs à propos de la négation, *Conclusion*, p.14.

[Longhi J. et Saigh D. 2016] : Longhi Julien *et* Saigh Dalia. 2016, *A textometrical analysis of French arts workers « fr.Intermittents » on Twitter. 4th conference CMC and Social Media Corpora for the Humanities*, Sep 2016, Ljubljana, Slovenia.

[Longhi J. *et al*, 2014] : Proposition pour l'acquisition d'un corpus de Tweets (cmr-polititweets-teiv1-manuel.pdf)
https://repository.ortolang.fr/api/content/comere/v3.3/cmr-polititweets/cmr-polititweets-tei-v1-manuel.pdf

[Martinez et coll. 2010] : Martinez, W. ; Daoust, F. ; Duchastel, J. Un service Web pour l'analyse de la cooccurrence. JADT 2010.

[Martineau C. *et al.* 2007] : Martineau C., Tolone E., Voyatzi S. (2007). Les Entités Nommées : usage et degrés de précision et de désambiguïsation. In Proceedings of the 26th International Conference on Lexis and Grammar, Bonifacio, pp. 105--112.

[Mayaffre, D. 2008] : « De l'occurrence à l'isotopie. Les cooccurrences en lexicométrie

[Marchand P. et Ratinaud P. 2012] : Marchand P., Ratinaud P., 2012. L'analyse de similitude appliquée aux corpus textuels : les primaires socialistes pour l'élection présidentielle française (septembre-octobre 2011). In : Actes des 11eme Journées internationales d'Analyse statistique des Données Textuelles. JADT 2012. Liège. p. 687-699.

[Mohammad Saif M *et al.* 2015] : Saif M. Mohammad, Xiaodan Zhu, Svetlana Kiritchenko, and Joel Martin (2015): *Sentiment, Emotion, Purpose, and Style in Electoral Tweets*

[Mohammad Saif M. *et al.*, 2016] : Saif M. Mohammad, Parinaz Sobhani *et* Svetlana Kiritchenko, (2016): *Stance and Sentiment in Tweets*

[Ratinaud et Marchand, 2012] : Pierre Ratinaud et Pascal Marchand, 2012, Application de la méthode ALCESTE aux « gros » corpus et stabilité des « mondes lexicaux » : analyse du « CableGate » avec IRAMUTEQ.

[Rosenthal *et al.,* 2014] : Rosenthal Sara, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter and Veselin Stoyanov ,(2014): *SemEval-2015 Task 10: Sentiment Analysis in Twitter*

[Russell, 1980] : J. A. Russell, *« A circumplex model of affect, » Journal of Personality and Social Psychology,* vol. 39, pp. 1161– 1178, 1980.

[Sarfati G. E. 2014] : Georges-Elia Sarfati, Les discours institutionnels en confrontation. Contributions a l'analyse des discours institutionnels et politiques, chapter L'emprise du sens: Note sur les conditions théoriques et les enjeux de l'analyse du discours institutionnel, pages 13–46. L Harmattan.

[Talha Meryem *et al.*,2014] : Talha Meryem, Siham Boulaknadel, Driss Aboutajdine1, (2014): RENAM: Système de Reconnaissance des Entités Nommées Amazighes, p.2

[Thelwall Mike *et al.*, 2010] Thelwall Mike, Kevan Buckley, Georgios Paltoglou, Di Cai (2010). *Sentiment strength detection in short informal text. Journal of the American Society for Information Science and Technology*.

[Tutin A. et Dini L. 2016] Cous Syntaxe/sémantique : présentation générale

[Valérie Delavigne, 2014] : Valérie Delavigne. Alceste, un logiciel d'analyse textuelle. Texto ! Textes et Cultures, Équipe Sémantique des textes, 2003, pp.n.a. <hal-00924168>

[Vidak *et al,* 2016] : Vidak Marko *et* Jackiewicz Agata, « Les outils multimodaux de Twitter comme moyens d'expression des émotions et des prises de position », Cahiers de praxématique [En ligne], 66 | 2016, mis en ligne le 01 janvier 2016, (consulté le 21 janvier 2017). URL : http://praxematique.revues.org/4247

Sitographie

blogdumoderateur: http://www.blogdumoderateur.com/chiffres-twitter/ (07/01/2017).

ideo2017: http://ideo2017.ensea.fr/(08/01/2017).

Synomia : https://www.synomia.fr/fr/notre-metier/la-technologie-synomia/analyse-syntaxique (22/01/2017).

Wikipédia1: https://fr.wikipedia.org/wiki/Twitter (06/01/2017).

Wikipédia2 : https://fr.wikipedia.org/wiki/Analyse_des_donn%C3%A9es(11/01/2017).

Wikipédia3: https://fr.wikipedia.org/wiki/Apprentissage_automatique(12/01/2017).

[Documentation1] :http://www.iramuteq.org/documentation/fichiers/documentation_19_02_2014.pdf P. 11 (Consulté le 28/07/2017)

[Documentation2] :http://www.iramuteq.org/documentation/fichiers/Pas%20a%20Pas%20IRAMUTEQ_0.7alpha2.pdf) (Consulté le 28/07/2017)

Table des illustrations

Table des tableaux

Table des matières

MOTS-CLÉS : TALN, réseaux sociaux, textométrie, analyse de discours, fouille de tweets.

RÉSUMÉ

Ce mémoire présente le travail réalisé pendant 6 mois de stage au laboratoire ETIS au sein de l'équipe MIDI (Indexation Multimédia et Intégration de Données) en association avec le laboratoire AGORA à Cergy. Le travail portait sur l'analyse du discours politique sur Twitter durant la campagne politique 2017 en France (présidentielle et législative) en quasi temps réel. L'objectif était d'étudier, dans un premier temps, les analyses qui peuvent être réalisées sur des tweets politiques, et, dans un deuxième temps, de créer une application web qui permet de traiter, avec des délais relativement courts, les messages produits en lien avec l'actualité politique. Cette application permettra aux citoyens d'analyser les tweets des candidats à l'élection présidentielle et législative 2017 en France, et d'utiliser un moteur de recherche par facette qui facilite la recherche sur la base de tweets. L'utilisation d'#Ideo2017 fournit les principales caractéristiques du corpus de chaque candidat, comme notamment, les champs lexicaux, les thématiques, les relations entre les mots, etc. Elle fournira également des visualisations graphiques des données. Cela permet aux utilisateurs de faire des comparaisons entre les différents comptes ou de vérifier l'utilisation de certains mots par les différents candidats.

KEYWORDS : NLP, social media, textometry, discourse analysis, tweets mining.

ABSTRACT

This report illustrates the work done during my internship of 6 months at the Lab ETIS with the team MIDI (Multimedia Indexation and data integration) which works in association with the Lab AGORA in Cergy. The work focuses on analyzing the political discourse in Twitter during the election's campaign of 2017 in France (presidential and legislative) in near real time. The first goal of this work is to study the analyses that could be produced in political tweets; the second goal is to create a web application which will provide, in a relatively short time, treatment of messages in relation to the current political scene. This will enable citizens to analyze the tweets of candidates for the presidential and legislative election of 2017 in France and to use a search-facetted engine which facilitate the search in the tweets base. The use of #Ideo2017 provides the principal characteristics of each candidate corpus, mainly, in the lexical and thematic areas, the relationship between words, and the graphic data visualization. This application enables the users to compare between different accounts and to verify the use of certain words by different candidates.