



HAL
open science

Identification de la langue & étiquetage morpho-syntaxique de tweets.

Sevil Zeynaligargari

► **To cite this version:**

Sevil Zeynaligargari. Identification de la langue & étiquetage morpho-syntaxique de tweets.. Sciences de l'Homme et Société. 2015. dumas-01260379

HAL Id: dumas-01260379

<https://dumas.ccsd.cnrs.fr/dumas-01260379>

Submitted on 22 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Identification de la langue & étiquetage morpho- syntaxique de tweets

Nom : ZEYNALIGARGARI

Prénom : Sevil

UFR LLASIC

Mémoire de master 2 professionnel – Sciences du langage

Spécialité ou Parcours : Industries de la langue

Sous la direction d'Olivier Kraif

Les encadrants du laboratoire : Jean-Philippe Magué, Matthieu Quignard

Résumé

Mots clés : tweets, analyse morpho-syntaxique, identification de la langue, clustering.

Ce rapport de stage représente 6 mois de travail au laboratoire ICAR, un laboratoire d'analyse de corpus complexe, situé à Lyon.

L'objet d'étude de ce travail est une analyse linguistique d'un grand nombre de tweets français. Avant cette analyse, un clustering est fait sur les utilisateurs de notre corpus en prenant comme principe la mesure de la modularité (en se basant sur les liens mutuels entre les utilisateurs). Ces clusters sont maximisés avec l'algorithme de Louvain, afin de mettre dans un cluster les utilisateurs qui ont les liens les plus forts entre eux.

Nous avons évalué quelques outils d'identification de la langue pour en choisir les meilleurs. Selon nos tests, les identificateurs automatiques Ldig et Langid sont plus performants pour l'identification de la langue des tweets de notre corpus.

Nous avons identifié la langue des tweets de notre corpus à l'aide des identificateurs automatiques testés les plus performants, afin de choisir les tweets français pour notre travail d'analyse morpho-syntaxiques.

Nous avons fait une analyse morpho-syntaxique des tweets français de notre corpus avec l'étiqueteur MElt.

Nous avons fait des analyses statistiques sur les tweets de notre corpus.

Ces analyses statistiques prouvent l'hétérogénéité de nos clusters. Nous pouvons voir qu'il y a des similarités linguistiques entre certains clusters. Nous avons ainsi des clusters qui ont des stratégies linguistiques tout à fait différentes.

Abstract

Keywords: tweets, analyse morpho-syntactic, language identification, clustering.

This report presents the internship of 6 months in ICAR laboratory in Lyon. ICAR is a laboratory of a complex corpus.

In this work we have analyzed linguistically a big data of French tweets. Before this analyze, we clustered the users of our text corpus. This clustering is based on mutual relations between the users. We used the modularity principles for these clustering. We extracted the maximized clusters with Louvain modularity.

We have tested some language identifiers to choose the best ones for our text corpus of tweets. We considered that Ldig and Langid are more performances. We identified the language of all tweets of our corpus of tweets with these two language identifiers.

We did morpho-syntactic analysis of all tweets with Melt tagger.

We choose the clusters that have more French tweets for some statistical analyses. This analyses shows heterogeneity of our clusters. We can see that there are some resemblances and differences between clusters. Each cluster has its own linguistic strategies.

Remerciements

J'avais envie d'adresser mes sincères remerciements à ceux qui ont contribué à l'élaboration de mon mémoire.

Je tiens tout particulièrement à remercier Monsieur Jean-Philippe MAGUE, qui m'a soutenue, encouragée, et avec qui j'ai établi une relation de confiance.

Merci à Monsieur Matthieu QUIGNARD pour son aide précieuse, pour le temps qu'il m'a consacré avec beaucoup de patience et de disponibilité pour répondre à mes questions.

Je tiens à exprimer toute ma reconnaissance à mon Directeur de mémoire Olivier KRAIF. Je le remercie de m'avoir encadrée, orientée, aidée et conseillée.

DÉCLARATION

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 : ZEYNALIGARGARI.....

PRENOM : Sevil.....

DATE : 03/12/2015.....

SIGNATURE :



Sommaire

Résumé	1
Abstract.....	3
1 Introduction	8
2 Contexte de la réalisation du stage.....	10
2.1 ICAR : un laboratoire d'analyse des corpus complexes.....	11
2.1.1 Les équipes de travail d'ICAR	11
2.2 Descriptif de la mission	12
2.2.1 Twitter : un réseau social	12
2.2.2 Besoins : demande d'ICAR pour le projet des tweets (SoSweet).....	12
3 Données du projet SoSweet	14
3.1 Corpus de SoSweet	15
3.2 Gestion et stockage des données.....	15
3.3 Clustering des utilisateurs.....	17
4 Identification de la langue des tweets	21
4.1 Enjeux et objectifs	22
4.2 Méthodologie	22
4.3 Annotations manuelles des tweets.....	22
4.3.1 Echantillonnage	22
4.3.2 Interface web créée en PHP pour l'annotation manuelle des tweets	22
4.3.3 Explication des 4 boutons proposés pour l'annotation des tweets et les critères de choix d'annotation :	23
4.3.4 Base de données MYSQL.....	25
4.3.5 Système de vote	26
4.3.6 Conclusion et résultats des annotations manuelles des tweets	27
4.4 Annotations automatiques de la langue des tweets.....	28
4.4.1 Identificateurs automatiques testés.....	28
4.4.2 Procédure d'évaluation des identificateurs automatiques de la langue	30
4.4.3 Les résultats des identificateurs automatiques.....	31
4.4.4 Rappel et précision pour la combinaison des identificateurs	32

4.4.5	Choix final de l'identificateur automatique et insertion des résultats des identifications dans MongoDB.....	33
4.5	Conclusion	34
5	Analyse morpho-syntaxique des tweets	35
5.1	Pourquoi l'analyse morpho-syntaxique des tweets ?.....	36
5.2	Analyseur MElt.....	36
5.2.1	Présentation de l'analyseur MElt	36
5.2.2	Présentation des étiquettes de MElt.....	37
5.2.3	Étapes d'étiquetage des tweets avec MElt.....	38
5.3	Echantillonnage des 12 plus grands clusters.....	40
5.4	Statistiques sur les 12 clusters	41
5.4.1	Longueur des tweets	41
5.4.2	Nombre d'étiquettes par cluster.....	46
5.4.3	Exemples des tokens spécifiques pour les 12 plus grands clusters	52
5.5	Conclusion	53
5.6	Perspectives	53
5.7	Bilan du stage.....	54
6	Bibliographie.....	57
7	Annexes.....	58

1 Introduction

Le développement des réseaux sociaux influence non seulement la vie quotidienne des gens mais leur vie sociale, politique et culturelle. Les chercheurs, dans les différents domaines, se sont intéressés à l'analyse de ce qui est diffusé dans ces réseaux sociaux pour pouvoir répondre à leurs problématiques.

Dans ce travail, nous nous sommes intéressés à la variation de la langue française sur le réseau social Twitter. Au 30 septembre 2015, il existe 320 millions d'utilisateurs d'actifs sur ce site. Cela montre la popularité de ce réseau social.

Pour faire une analyse linguistique sur une grande quantité de tweets, il faut avoir les outils automatiques et s'équiper des outils du TAL. La complexité de la langue utilisée sur ces réseaux, nécessite des outils assez performants qui pourront analyser les textes bruités.

Nos questions de base sont les suivantes : est-ce qu'on pourra faire un clustering des utilisateurs de Twitter ? Sur quel principe peut-on faire ce clustering ? Est-ce qu'on pourra trouver des similitudes sur le plan linguistique entre les utilisateurs d'un cluster donné ? Comment peut-on vérifier les ressemblances ou les différences linguistiques entre les clusters ?

Pour ce travail nous avons défini 4 chapitres :

Notre première partie consiste à présenter le contexte de la réalisation de ce stage : le laboratoire ICAR et ses différentes équipes ainsi que les attentes et la demande d'ICAR et le descriptif de la mission.

La deuxième partie est consacrée à la présentation du projet SoSweet, les enjeux et les objectifs de ce projet, le clustering des utilisateurs de notre corpus. Nous allons aussi présenter les données et notre corpus de travail.

Dans la troisième partie nous allons identifier manuellement la langue d'un échantillon de tweets. Nous allons nous servir de ces identifications manuelles comme référence pour évaluer le fonctionnement de quelques outils automatiques de l'identification de la langue. Nous allons présenter les résultats de ces évaluations et en trouver les identificateurs plus performants.

La quatrième partie présente les enjeux et l'importance d'une analyse morpho-syntaxique des tweets de notre corpus. Nous allons présenter l'étiqueteur MElt et les étapes de

l'étiquetage des tweets avec cet analyseur. Nous allons faire des statistiques sur les tweets, en nous basant sur les résultats obtenus par MElt. Nous allons présenter ces statistiques sous formes de diagrammes et d'histogrammes et nous allons les interpréter.

2 Contexte de la réalisation du stage

2.1 ICAR : un laboratoire d'analyse des corpus complexes

Le laboratoire ICAR, créé en 2003 et situé à Lyon, est un laboratoire d'analyse des corpus complexes. L'axe principal de recherche de ce laboratoire est l'analyse multidimensionnelle des corpus oraux interactifs et des corpus textuels. Les domaines concernés de la recherche sont : la linguistique interactionnelle, la linguistique de corpus... et aussi le traitement automatique des corpus écrits et oraux. Les langues de travail dans ce laboratoire sont : le français, l'anglais et l'arabe.

ICAR possède des équipes de recherches qui enregistrent les corpus oraux dans des situations réelles. Les chercheurs analysent les modalités des interactions langagières dans ces corpus et les usages de la langue dans ces interactions. Pour ce faire, ceux-ci font des études avancées sur les formes de la langue, son contenu langagier et la grammaire.

2.1.1 Les équipes de travail d'ICAR

Le laboratoire est composé de trois grandes équipes de travail: l'équipe InSitu, l'équipe ADIS et l'équipe CEDILLES.

L'équipe InSitu est constituée de chercheurs qui vérifient la langue selon deux aspects de leurs corpus oraux : les aspects linguistiques comme la syntaxe et la grammaire et les aspects visuels comme les gestes et le corps. Cette équipe est composée de trois sous-équipes :

1. Langues, Interactions, Situations [LIS]
2. Cognition, Collaboration, Interactions en Ligne [Cogcinel]
3. Systèmes d'Information, Ingénierie, Linguistique de l'Arabe et Terminologie [SILAT].

La deuxième sous-équipe, Cogcinel, dans laquelle ce stage a été effectué, développe une approche analytique des interactions humaine. Elle vérifie le rôle des émotions, les interactions argumentatives, les gestes, les regards, etc, dans les situations d'échanges. Cette équipe est composée de sept chercheurs permanents. Les maîtres qui ont encadré ce stage sont : Messieurs Matthieu Quignard et Jean-Philippe Magué qui font partie des membres permanents de cette équipe de recherche.

Toutes ces équipes ont, comme matière principale de recherche, des corpus oraux ou écrits et elles analysent la langue et ses particularités dans des situations variées comme l'apprentissage ou d'autres interactions variées.

2.2 Descriptif de la mission

Avant d'entrer dans le cœur de ma mission chez ICAR, nous allons présenter brièvement le site de Twitter et son fonctionnement.

2.2.1 Twitter : un réseau social

Twitter est un réseau social en ligne créé en 2006 par Jack Dorsey qui est actuellement le directeur de cette entreprise. La mission de ce site, selon le slogan publié sur la page de la présentation de cette entreprise, est de « donner à chacun le pouvoir de créer et de partager des idées et des informations instantanément et sans entraves. ». Nous sommes libres de twitter « tout ce que nous souhaitons mais l'entreprise ne prend pas en charge les risques éventuels de ce que nous publions sur leur site. » (Voir les termes et les conditions)¹.

Pour communiquer sur les réseaux sociaux, comme Twitter, les membres utilisent un langage spécifique. Pour économiser le temps et les caractères (140 caractères par *tweet*, limite donnée par Twitter) les utilisateurs de ces réseaux utilisent souvent des abréviations pour les mots courants. Au fur et mesure, ce genre de message est devenu familier pour les utilisateurs qui arrivent à se comprendre.

2.2.2 Besoins : demande d'ICAR pour le projet des tweets (SoSweet)

Le projet des tweets a été nommé SoSweet parce que nous faisons des recherches sur les aspects sociolinguistiques des tweets. SoSweet n'est pas un sigle mais représente la fusion de sociolinguistique et Twitter. Pour le projet de SoSweet, l'équipe de recherche concernée, chez ICAR, a pour but d'analyser une grande quantité de données de tweets et de développer une analyse plus détaillée sur les relations entre les utilisateurs de Twitter et la variabilité de la langue.

Ce stage a eu pour but d'effectuer les tâches suivantes :

1. identifier la langue des tweets par les identificateurs automatiques,
2. faire une analyse morpho-syntaxique des tweets par l'analyseur MELt,
3. stocker les informations sur les tweets dans la base de données,
4. étudier statistiquement la variation des langues entre les clusters différents.

¹<https://twitter.com/tos?lang=fr>

Toutes ces étapes sont les prétraitements qui permettent de préparer les données pour des analyses plus vastes dans le cadre d'un projet financé par l'ANR et d'une thèse financée par le Labex Aslan. Le nouveau doctorant, Clément Thibert, a commencé sa thèse depuis le mois d'octobre 2015.

3 Données du projet SoSweet

3.1 Corpus de SoSweet

Les tweets utilisés pour ce travail ont été rassemblés par le laboratoire ICAR auprès de l'entreprise DataSift.

Pour le choix des tweets utilisés dans SoSweet, deux critères ont été choisis :

- Un critère géographique : les utilisateurs qui habitent en GMT (Greenwich Mean Time) et GMT+1. Les pays situés en GMT : Burkina Faso, Cote d'Ivoire, Ghana, Grande Bretagne, Guinée, Guinée Equatoriale, Iles Canaries, Irlande, Islande, Liberia, Mali, Maroc, Mauritanie, Portugal, Sénégal, St-Hélène, Togo et les pays en GMT+1 Albanie, Algérie, Allemagne, Andorre, Angola, Autriche, Belgique, Bénin, Bosnie-Herzegovine, Cameroun, Centre Afrique, Congo, Croatie, Danemark, Espagne, France, Gabon, Gambie, Gibraltar, Hongrie, Italie, Liechtenstein, Luxembourg, Macédoine, Malte, Monaco, Niger, Nigeria, Norvège, Pays-Bas, Pologne, Rep. Tchèque, St- Marin, Slovaquie, Slovénie, Suède, Suisse, Tchad, Tunisie, Vatican, Yougoslavie, Zaïre¹. Nous avons choisi ces deux zones parce que selon l'agence de la francophonie, 75 % des francophones sont situés dans ces régions.
- Un critère linguistique : comme nous nous intéressons aux tweets en langue française, nous avons collecté les tweets des utilisateurs ayant déclaré tweeter en français durant l'inscription sur le site de Twitter, et ceux identifiés par DataSift comme étant du français.

Pour chaque utilisateur, auteur d'un tweet du corpus, ICAR a également collecté auprès de Twitter la liste des utilisateurs qu'il suit.

Dans ce stage nous avons travaillé sur deux versions de ce corpus : en début de stage, sur les tweets collectés entre juin 2014 et janvier 2015 (corpus A : soit 42 millions de tweets et 1.3 million d'utilisateurs), puis ce jeu de données a été étendu avec des tweets collectés jusqu'en juin 2015 (corpus B : avec un total de 70 millions de tweets et 1.7 million d'utilisateurs).

3.2 Gestion et stockage des données

Dans MongoDB, qui est une base de données NoSQL nous pouvons stocker des documents en format JSON, sans avoir prédéfini de schéma particulier pour ces documents. Comme nous avons dû traiter 70 millions de tweets qui font 132Go en format JSON et dans la base MongoDB, nous nous sommes heurtés à un problème d'espace sur le disque dur des ordinateurs du laboratoire. La capacité du serveur local de MongoDB étant limitée, nous avons utilisé pour cela un serveur de l'ENS.

¹ Pour plus d'information visiter le site :

http://www.alyabbara.com/moteurs_recherche/utilitaires/clock_mondial.html

Voici un exemple de nos documents dans MongoDB avec certaines informations (nous n'avons pas inséré toutes les informations dans notre base de données) fournies par DataSift:

```
{
  u '_id': ObjectId('55f282bdd0ba2008714cf643'),
  u 'date': datetime.datetime(2014, 12, 1, 0, 0,
7),
  u 'id': u '539207275077193728',
  u 'language': {
    u 'twitter': u 'und',
    u 'user': u 'fr'
  },
  u 'tweet': u '@asessinase xD',
  u 'user': 1538441737
}
```

Les informations données pour un tweet sont les suivantes en ordre d'apparition :

1. La date de la publication de ce tweet ;
2. L'identifiant du tweet ;
3. La langue identifiée par Twitter pour ce tweet diffusé ;
4. La langue définie par la personne qui a tweeté (durant l'ouverture d'un compte sur Twitter, on choisit la langue dans laquelle on va tweeter) ;
5. Le texte de ce tweet ;
6. L'identifiant de l'auteur de ce tweet.

Au fur et à mesure de ce stage, nous avons complété les informations dans la base de données, selon les besoins de notre travail.

Voici, ci-dessous, un exemple plus complet avec les analyses morpho syntaxique de MELT d'un document de notre base de données MongoDB. Par la suite, nous expliquerons les informations ajoutées.

```
{
  u '_id': ObjectId('55a906c142112e1c72390061'),
  u 'community': 255,
  u 'date': datetime.datetime(2014, 6, 19, 15, 59, 47),
  u 'id': u '479654787148165121',
  u 'language': {
    u 'Datasift': {
      u 'confidence': 92, u 'language': u 'fr'
    },
    u 'langid': {
      u 'conf': 0.9992384434371205, u 'lang': u
'fr'
    },
    u 'ldig': {
      u 'lang': u 'fr'
    },
  },
}
```

```

        u 'twitter': u 'fr',
        u 'user': u 'fr'
    },
    u 'melt': [{
        u 'normalization': u '',
        u 'probability': u '0.613696898985',
        u 'tag': u 'NPP',
        u 'token': u 'Balec'
    }, {
        u 'normalization': u '',
        u 'probability': u '0.141008409787',
        u 'tag': u 'NC',
        u 'token': u 'fr\xe8re'
    }],
    u 'tweet': u 'Balec fr\xe8re',
    u 'user': 2194245411 L
}

```

3.3 Clustering des utilisateurs

Une fois que nous avons collecté les tweets, nous avons eu besoin de regrouper les utilisateurs de notre corpus en cluster. Ce regroupement est fait selon les liens réciproques entre les utilisateurs, autrement dit les utilisateurs qui se suivent mutuellement sur Twitter.

Pour cela, ont été utilisés la mesure de la modularité et l’algorithme de Louvain¹.

Voici la définition de la modularité selon Jacques Cellier (2012):

« La modularité est un indice Q calculé à partir du réseau et d’une partition qui reflète le caractère plus ou moins « communautaire » des groupes ainsi obtenus. Il est toujours compris entre -1 et 1 et on peut l’interpréter ainsi :

$Q \leq 0$: les groupes ne forment pas du tout des clusters car ils possèdent, bien au contraire, plus de relations vers l’extérieur que de relations internes.

$0 < Q$: les groupes présentent un caractère communautaire d’autant plus marqué que la valeur de Q s’approche de 1.

Dans la pratique, une valeur de $Q > 0,5$ indique que les groupes présentent un caractère de cluster suffisamment marqué. »

La modularité d’un regroupement en clusters est calculée en comparant le nombre de liens inter clusters avec celui de liens intra cluster. Pour atteindre une modularité plus grande, il faut qu’il existe plus de liens internes entre les individus d’un même cluster que de liens externes vers les individus d’autres clusters. La méthode de Louvain propose un algorithme pour trouver un regroupement en clusters qui maximise la modularité.

¹ <https://sites.google.com/site/findcommunities/>

Après ce regroupement des utilisateurs en cluster, ces derniers sont, le cas échéant, positionnés spatialement sur la base de leurs tweets géolocalisés. Les cartes suivantes montrent la géolocalisation des utilisateurs du corpus ayant des tweets géolocalisés. Les clusters différents recouvrent des zones géographiques différentes. Ces cartes montrent les 24 clusters ayant plus de 1000 utilisateurs.

Sur ces cartes, chaque point représente un utilisateur. La géolocalisation de cet utilisateur est calculée à partir du barycentre (la moyenne) des coordonnées de géolocalisation de ses tweets. La distribution des points sur la carte, permet de visualiser que les utilisateurs d'un cluster sont répartis dans des régions géographiques différentes. Après le clustering des utilisateurs de notre corpus, nous avons inséré, pour chaque tweet, son numéro de cluster dans notre base de données MongoDB.

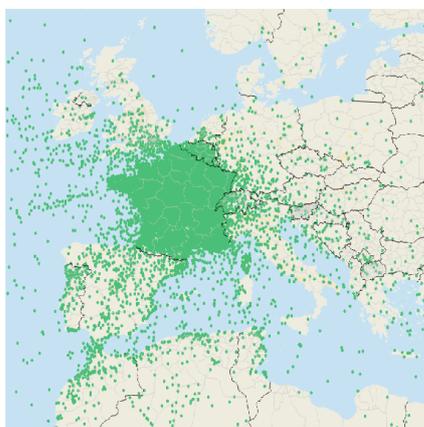


Figure 1, le cluster n° 1

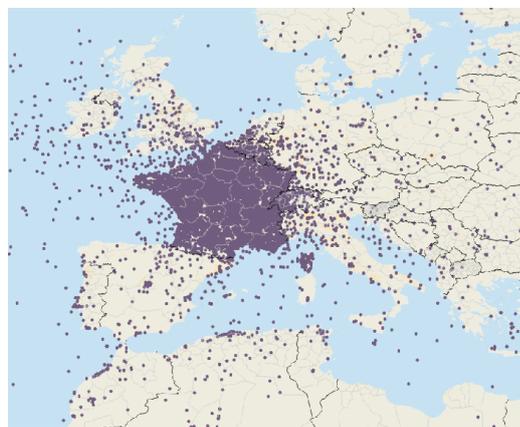


Figure 2, le cluster n° 2

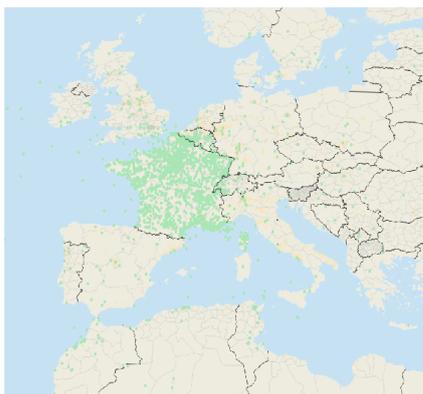


Figure 3, cluster n°6

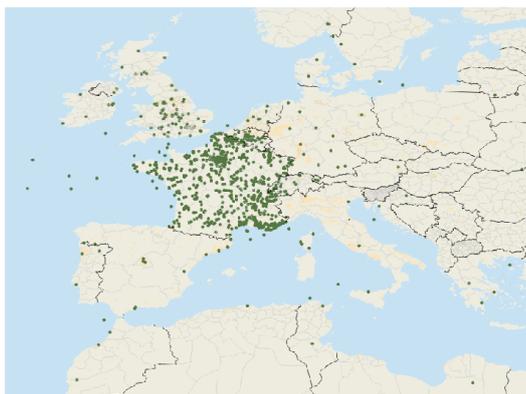


Figure 4, cluster n° 11

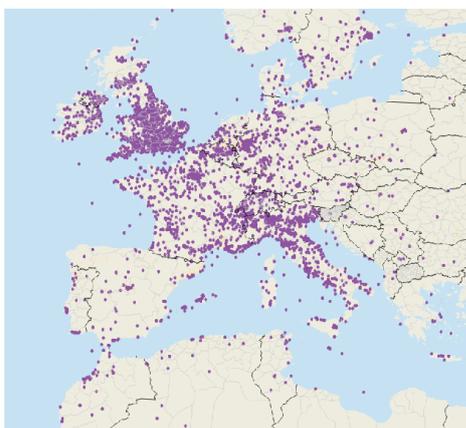


Figure 5, cluster n°3

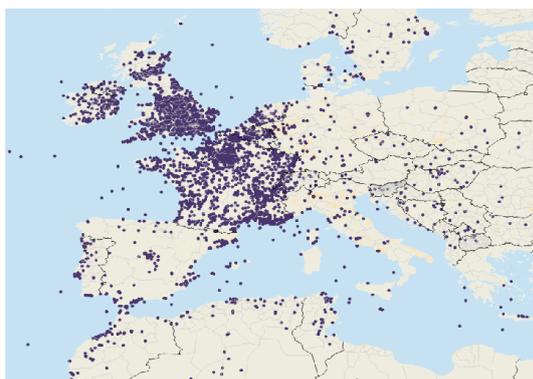


Figure 6, cluster n°4

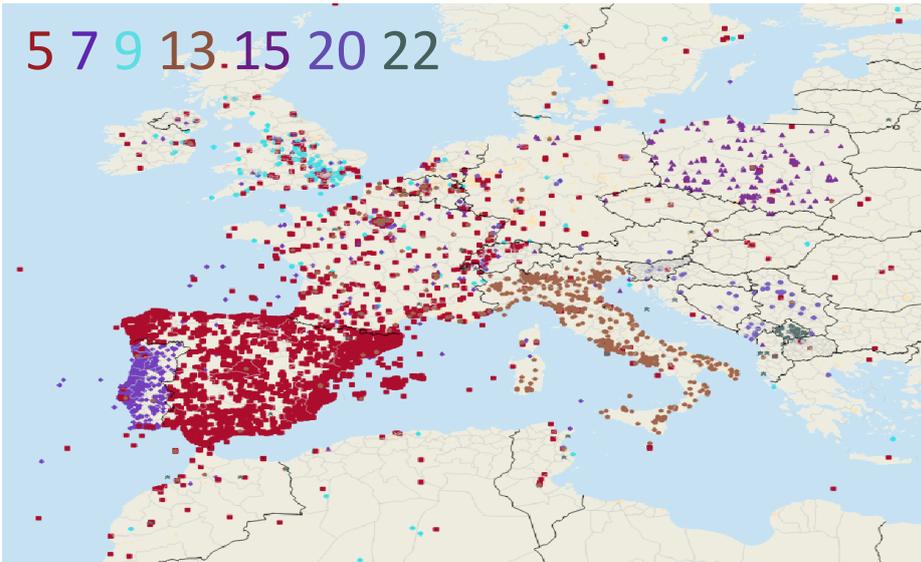


Figure 7, cluster n° 5,7,9,13,15,20,22

4 Identification de la langue des tweets

4.1 Enjeux et objectifs

Pour obtenir un maximum de tweets français, les tweets collectés respectent les deux critères que nous avons mentionnés dans la partie 2.1.

Malgré ces deux conditions, il existe des tweets, parmi les tweets collectés, qui ne sont pas en français. Pour notre travail d'analyse, nous avons besoin de sélectionner exclusivement les tweets en langue française. Afin d'arriver à sélectionner les tweets français nous avons besoin d'identifier la langue de ces tweets automatiquement.

4.2 Méthodologie

Pour identifier les tweets en français, nous avons besoin de vérifier, de connaître les identificateurs automatiques et d'évaluer leurs performances. Pour cela, nous avons besoin d'un échantillon de tweets pour lesquels la langue est connue. Dans une première étape nous avons donc annoté manuellement un tel échantillon de tweets.

Ces annotations ont été faites via une interface web en PHP que j'ai développée. Ces annotations nous ont servi comme référence pour les comparaisons entre les identificateurs automatiques de la langue. Nous avons pu ainsi évaluer les identificateurs de la langue existants en prenant comme référence les annotations faites à la main. Nous avons comparé les résultats des identifications avec les métriques Rappel, Précision et F-score qui seront détaillées plus loin. Avec les résultats de ces comparaisons nous avons pu distinguer l'identificateur le plus performant.

4.3 Annotations manuelles des tweets

4.3.1 Echantillonnage

Nous avons choisi, de façon aléatoire, 6 000 tweets de notre corpus A, soit 250 tweets dans chacun des 24 clusters ayant plus de 1 000 utilisateurs, pour les faire annoter manuellement. Nous avons créé une base de données pour stocker les tweets et les informations liées à ces tweets pour pouvoir ensuite ajouter les résultats des annotations manuelles. L'interface d'annotation en PHP nous a servi à alimenter notre base de données.

4.3.2 Interface web créée en PHP pour l'annotation manuelle des tweets

Nous avons pris un échantillon de tweets pour lesquels nous avons souhaité savoir s'ils ont été rédigés en français ou non. Pour effectuer ces annotations nous avons pris en compte le tweet dans sa globalité et non une partie de ce tweet. Cela veut dire que les

annotateurs ont dû trancher leur choix entre les tweets rédigés moitié en français et moitié dans une autre langue. Nous avons créé une page web, en PHP, pour demander la langue des tweets aux volontaires qui étaient les chercheurs du laboratoire ICAR. Nous avons choisi ces annotateurs, parce qu'ils étaient disponibles et qu'ils étaient habitués à ce genre de recherche. Nous avons fait défiler les tweets, les uns après les autres, pour les afficher sur l'écran. Ainsi, avec les 4 boutons « français », « pas français », « je ne sais pas » et « retour » nous avons demandé aux annotateurs d'annoter les tweets. Voici un imprimé d'écran de cette page web :



Figure 8, page PHP pour l'annotation manuelle des tweets

4.3.3 Explication des 4 boutons proposés pour l'annotation des tweets et les critères de choix d'annotation :

Le bouton « français » est pour indiquer que le tweet est en français. Il ne s'agit pas de juger si le tweet est en bon français ou pas, mais plutôt d'indiquer si, selon les annotateurs, l'intention de la personne qui a rédigé ce message était de s'exprimer en français. Dans certains cas nous pouvons facilement identifier qu'un tweet est écrit en français comme « bon, j'connais rien, tant pis, dodo ». Il existe aussi des tweets en deux

langues (français et une autre langue souvent l'anglais) qui peuvent être considérés comme du français. En voici quelques exemples :

```
@dansker78 Moin Moin! Alleeeeeeeeezzzzzz!!!!!!!!!!!! Go
Go Gooooooooo!!!! On y
@UntilthVeryEnd j'arrive !!!
Grégoire-ToiMoi IS NOW ON ALIX RADIO! it 's a french
hit! c'est un hit francais et c'est maintenant sur
alix radio
```

Ces tweets qui contiennent des mots étrangers sont annotés comme du français en tenant compte de leur contexte et de la globalité des phrases. Dans ces genres de tweets il y a souvent les mots basiques anglais comme *go*, *end*, *hit* etc. qu'on pourra reconnaître facilement et qui sont intégrés dans une phrase française. Dans ce cas le bouton *français* sera choisi.

Le bouton « Je ne sais pas » est pour indiquer que les annotateurs ne peuvent pas trancher. Par exemple, lorsqu'un tweet ne contient qu'un nombre, comme « 14 », il est bien difficile de dire si « 14 » est en français ou non. Les onomatopées, les hashtags, les noms propres non français, les chiffres, les noms des régions hors d'un contexte précis, peuvent provoquer une ambiguïté.

Un tweet comme « 15 minutes #ita ♥□♥□♥□????? » est considéré comme étant ambigu. Ou les tweets composés de deux langues comme le suivant : « Meexissy King - Mon Coeur <http://t.co/SxB56vJyFb> » ou « lolol mariage huh » sont aussi considérés comme étant ambigus. Donc le bouton « je ne sais pas » sera choisi dans ce cas.

Le bouton « pas français » est pour indiquer que les annotateurs sont certains que le tweet qu'ils ont sous les yeux est rédigé dans une autre langue que le français. En voici quelques exemples :

```
Lambert on now
Computer Science Schools Sao Tome Application Process
http://t.co/MlIKtFtcMr
```

Une fois qu'un tweet a eu suffisamment de vote ou que finalement ce tweet n'a pas pu être catégorisé par les annotateurs, nous arrêtons l'affichage sur la page web. Nous allons expliquer cette procédure en détail en partie 3.3.5.

Le bouton « retour » est pour revenir au tweet précédent, si l'annotateur réalise qu'il a besoin de revoir le tweet précédent ou de corriger son choix.

Les utilisateurs ont la possibilité de quitter la page d'annotation à un moment donné et de revenir sur le site puis de continuer à partir des tweets où ils se sont arrêtés.

A chaque fois, nous affichons le nombre de tweets restants et le nombre de tweets déjà annotés par un annotateur.

4.3.4 Base de données MYSQL

Dans notre base de données, nous avons trois tableaux: le tableau *annotateurs*, le tableau *annotations* et le tableau *tweets*. Le tableau *annotateurs* contient les mails des volontaires qui ont accepté d'annoter notre corpus. Il contient les colonnes : *id_user*, *mail*. Le tableau *annotations* contient *id_langue*, *langue*, *id_tweet*, *id_user*. Dans le tableau *tweets*, nous avons inséré les champs : *id_tweet*, *texte*, *cluster*, *statut*, *annotation_finale*. *Id_tweet* est l'information liée à un tweet donné, les ids des tweets sont tous uniques. *Texte* : est le contenu des tweets. Le *statut* est notre système de vote qui est par défaut zéro (nous allons expliquer les conditions de changement de *statut* en partie 3.3.5). Une fois qu'il y a suffisamment de votes (en tenant compte du champ de *statut*) pour la langue d'un tweet donné, nous pouvons lui attribuer une *annotation_finale*.

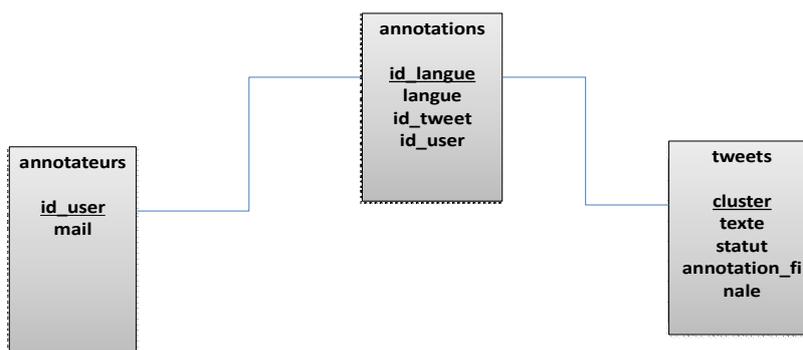


Figure 9, Les tables sur la base de données MYSQL

Le champ de *langue* dans la table *annotations* contient l'annotation d'un tweet par un annotateur. Si l'annotateur choisit le bouton français sur l'interface la valeur du champ *langue* dans la table *annotation* sera *fr*. Si l'annotateur clique sur le bouton *je ne sais pas* le champ de *langue* prendra la chaîne de caractère *ambigüe* comme valeur. Dans le cas du choix du bouton *pas français*, le champ *langue* aura comme valeur *non fr*.

4.3.5 Système de vote

Si les deux premiers annotateurs donnent un vote identique pour un tweet présenté, nous avons annoté ce tweet selon leur vote. Nous avons changé le statut en 1, et nous avons arrêté l’affichage de ce tweet pour les autres annotateurs.

Dans le cas d’avis différents des deux premiers annotateurs, nous avons réaffiché le tweet encore deux fois, afin de récolter deux autres votes. S’il y a eu trois votes d’accord, nous avons annoté ce tweet selon le vote des majorités, sinon nous avons considéré ce tweet comme un tweet qui doit être révisé et catégorisé à la fin par notre jury, donc, son statut change en 2 et on ne l’affiche plus aux autres annotateurs.

Après avoir fini l’affichage des tweets sur l’interface PHP, nous avons créé une page web pour pouvoir afficher tous les tweets qui ont finalement 4 avis mais qui n’ont pas 3 votes pareils. Nous avons construit une équipe de 3 juges francophones et linguistes pour l’annotation de ces tweets. Nous avons vérifié les tweets un par un et nous avons fait les drags and les drops¹ pour mettre les tweets dans les colonnes appropriées (français, pas français, ambigu). Ainsi, après avoir choisi la bonne colonne pour chaque tweet, le statut de ce tweet non catégorisé, change en 1 dans notre base de données et on lui attribue une annotation finale en fonction de la décision finale de l’équipe.

En voici quelques exemples avec les annotations finales attribuées par le juge :

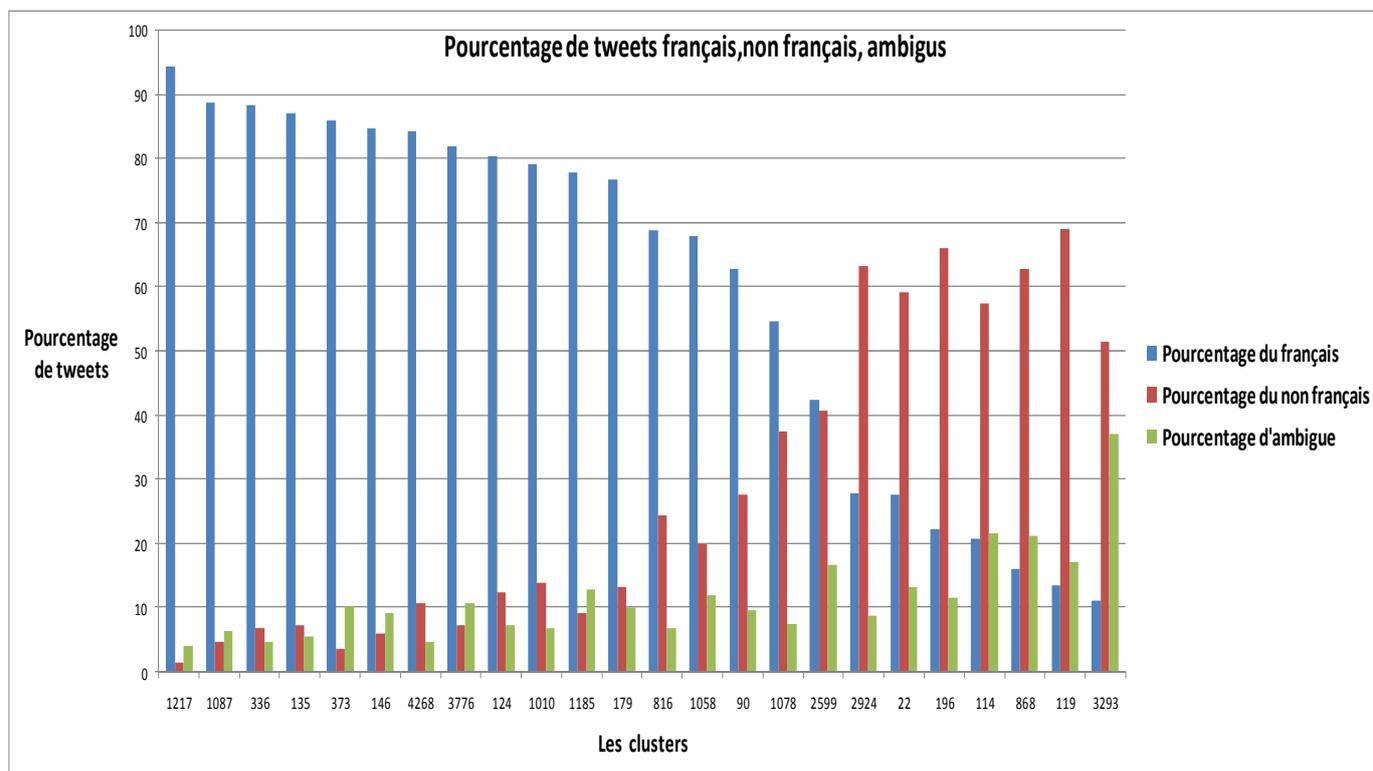
Tweets non catégorisés	français	Pas français	ambigu
Footballlllll http://t.co/mCVTpp5xqY			✓
@zeljko_vasic @b92vesti @blic_online @kurirvesti @telegrafrs no comment!!!		✓	
@zeljko_vasic @b92vesti @blic_online @kurirvesti @telegrafrs no comment!!!		✓	
come on cote d’ivoiremake us proud		✓	
#Nw The Dark Knight		✓	
@Adisa2002 nesor tklx ma mir live		✓	
@Fudzer @Olly_Baybee FAUD		✓	
@Louis_Tomlinson ilysm*			✓
@dansker78 Moin Moin!	✓		
@Michael5SOS hi bby ILYSM			✓
aqui pesa buééé!!!!		✓	

Tableau 1, annotation des tweets non catégorisés

¹Voir le code en Javascript concernant ces démarches en annexe I

4.3.6 Conclusion et résultats des annotations manuelles des tweets

Voici ci-dessous, présenté sous forme de diagrammes, le résultat des annotations manuelles :



graphique1, pourcentage de tweets français, non-français, ambigus

Comme nous le voyons dans ce diagramme, tous les clusters n'ont pas le même nombre de tweets français : 15 clusters ont plus de 60% de tweets français et 7 clusters en ont moins de 30%.

Si nous prenons comme exemple les clusters 1217 (la première barre) et 3293 (la dernière barre), nous constatons que le cluster 1217 a moins de 5% de tweets ambigus et qu'il a presque 90% de tweets français, tandis que le cluster 3293 qui a environ 40% d'ambiguïté, possède seulement 10% de tweets français.

Nous observons que le pourcentage de tweets français a une relation inverse aux tweets ambigus. Lorsqu'il y a plus de tweets français dans un cluster, il y a moins de tweets qui sont considérés comme ambigus (des tweets dont on ne pourra pas dire clairement s'ils

sont français ou pas).

Si nous regardons le pourcentage des tweets non-français, le cluster 3 293 a presque 50% de tweets non-français, ce qui est proche du pourcentage des tweets ambigus dans le même cluster. Le cluster numéro 1217 a moins de 10% de tweets non-français, chiffre qui est proche des tweets ambigus pour le même cluster.

Les clusters 2 924, 196 et 119 ont plus de 60% de tweets non-français, le pourcentage du français ne dépasse pas 30% et les ambigus représentent environ 15%. Nous pouvons dire que ces clusters ne tweetent pas beaucoup en français.

4.4 Annotations automatiques de la langue des tweets

4.4.1 Identificateurs automatiques testés

Nous avons testé 6 identificateurs de la langue. La langue identifiée par Twitter, User (la langue déclarée par l'utilisateur, ce n'est pas *stricto sensu* un identificateur) et DataSift qui sont les informations fournies par l'entreprise DataSift. Les trois identificateurs : Langid, Ldig et Cld2 sont des identificateurs Open Source et ils sont connus comme les plus performants par la communauté du TAL. Voici une explication brève de chacun de ces identificateurs :

- **Twitter**

Twitter possède son propre identificateur de langue. Au moment de tweeter, il pourra nous dire dans quelle langue est le tweet. Nous ne connaissons pas leur méthode d'identification de la langue et nous n'avons pas accès aux informations concernant leur méthode.

- **User**

La langue indiquée par l'auteur du tweet au moment de l'ouverture de son compte. User n'est pas un identificateur mais nous le prenons en compte pour pouvoir le comparer avec les détections manuelles et automatiques.

- **DataSift**

DataSift est l'entreprise qui a vendu les tweets au laboratoire ICAR. Cette entreprise possède son propre outil pour la détection des langues. Nous ne pouvons pas avoir accès aux informations sur la méthodologie de cette identification.

- **Langid**

Langid est un outil en libre-service pour l'identification des langues. Langid est formé selon le modèle classification naïve Bayésienne par les ngrammes ($1 \leq n \leq 4$). Cet identificateur utilise des corpus d'entraînement pour identifier la langue. Lui et T.Baldwin (2012). Les sources de ce corpus d'entraînement sont « JRC-Acquis, ClueWeb 09, Wikipedia, Reuters RCV2, Debian i18n »¹. Ces corpus sont dans 5 domaines différents: « Government documents, software documents, newswire, online encyclopedia and an internal crawl » Lui et T.Baldwin (2012).

La tokénisation se fait avec Aho-Corasick, qui est un algorithme de recherche de caractères dans un texte.

Nous pouvons utiliser Langid de trois manières : 1. Ligne de commande, 2. Librairie de Python 3. Directement sur le web.

Langid est un identificateur qui est construit et programmé pour les microblogs. Il est entraîné pour 97 langues (fa, fi, fo, fr, ga, gl, gu, he, hi, hr, ht, hu, hy, id, is, it,...)². Selon une étude (Lui et T.Baldwin (2014)) réalisée sur un corpus de 14 178 tweets dans 65 langues, le F-score de Langid est de 77%.

- **Ldig**

Ldig inclut une phase de normalisation pour l'identification de la langue des tweets. Ainsi les htag, url et les smileys qui contiennent des lettres de l'alphabet comme «:p », seront éliminés. Il existe également un dictionnaire de normalisation pour les mots orthographiés différemment sur Twitter comme « coolllllll ».

Selon la documentation de Ldig³, cet identificateur utilise la méthode « infinity-Gram ». Il possède une liste des mots ou des lettres qui sont très spécifiques dans une langue, comme le mot « ich » en allemand. Il utilise la méthode de la classification naïve bayésienne avec les caractères de n-gramme. Les 3 grammes sont utilisés assez souvent dans la méthode d'identification. Chaque langue a ses propres caractères et ses règles d'orthographe, cela justifie l'utilisation de n-gramme pour l'identification de la langue.

¹ <https://github.com/saffsd/langid.py>

² <https://github.com/saffsd/langid.py>

³ <http://fr.slideshare.net/shuyo/short-text-language-detection-with-infinitygram-12949447>

Selon le site de shuyo.wordpress.com¹, Ldig est un identificateur de langue spécifique pour Twitter. Ldig est capable de détecter 17 langues. Selon ce site, le rappel de cet identificateur pour la langue française est de 99.77%. Ldig est un identificateur construit et programmé pour les microblogs.

- **Cld2**

Compact Language Detection 2 ou Cld2 est l'outil de la détection de langue de Chrome (le navigateur développé par la société Google). Cld2 est basé sur la classification naïve bayésienne. Il identifie 80 langues en Unicode UTF8².

4.4.2 Procédure d'évaluation des identificateurs automatiques de la langue

Nous allons faire une analyse quantitative des identificateurs automatiques avec les métriques Rappel, Précision et F-score sur notre corpus de 6 000 tweets. Nous allons prendre comme référence les annotations faites à la main pour ces 6 000 tweets. Nous allons aussi combiner les résultats des identificateurs plus forts pour analyser leurs résultats et les comparer avec les identificateurs individuels.

4.4.2.1 Choix des aspects techniques pour l'identification automatique de la langue

Nous avons rencontré, à plusieurs reprises, un problème d'encodage pour les caractères spéciaux et les images ou les smileys. Nous avons résolu ce problème avant l'évaluation de nos identificateurs de la langue. L'encodage de Python étant strict, le code ASCII a été utilisé par défaut en Python, en lisant des articles³ sur ce sujet, nous avons pu ajouter des codes dans notre script Python pour pallier ce problème. Voici, en annexe II, quelques exemples de ces codes insérés dans nos scripts.

Afin d'utiliser les identificateurs de langue, nous avons utilisé le système d'exploitation Linux, parce que ces identificateurs sont incompatibles avec Windows.

Comme nous avons travaillé avec de gros volumes de données, il nous a souvent fallu quelques jours pour qu'un script puisse finir son travail. Nous avons donc consacré beaucoup de temps pour exécuter un script.

¹<https://shuyo.wordpress.com/2012/03/02/estimation-of-ldig-twitter-language-detection-for-liga-dataset>

²<https://code.google.com/p/cld2/>

³<http://sametmax.com/lencoding-en-python-une-bonne-fois-pour-toute/>

Nous avons également rencontré le problème des tweets qui sont entrés en doublon dans notre base de données. Il semblerait que DataSift nous ait fourni des tweets qui se trouvaient en double. A l'heure actuelle, nous n'avons pas pu corriger ce problème.

4.4.2 Définition des métriques Rappel, Précision et F-score pour les annotations manuelles des tweets

Voici les formules du rappel, de la précision et du F-score pour un identificateur donné dans notre travail :

$$\text{Rappel} = \frac{\text{Le nb de tweets correctement identifié comme du français}}{\text{Le nb de tweets vraiment en français}}$$

$$\text{Précision} = \frac{\text{Le nb de tweets correctement identifié comme étant du français}}{\text{Le nb de tweets français selon l'identificateur}}$$

$$\text{F-score} = \frac{2 \cdot (\text{précision} \cdot \text{rappel})}{(\text{précision} + \text{rappel})}$$

4.4.3 Les résultats des identificateurs automatiques

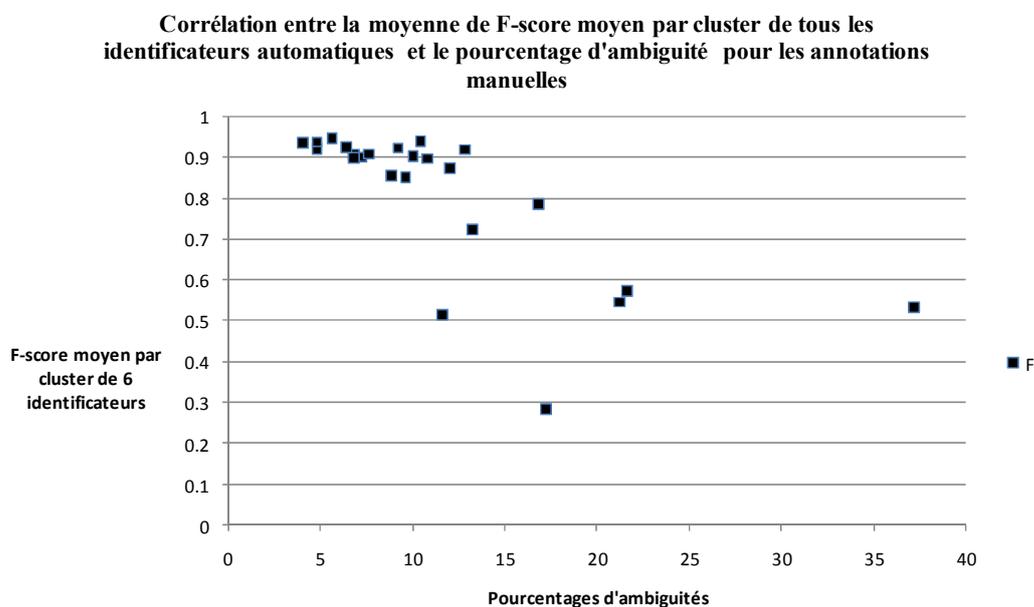
Le tableau suivant nous montre le rappel, la précision et le F-score de ces 6 identificateurs pour notre échantillon de 6 000 tweets.

Identificateurs	Rappel	Précision	F-score
Ldig	87%	87%	87%
Langid	77%	87%	82%
DataSift	72%	95%	81%
Twitter	94%	77%	80%
Cld2	67%	97%	79%
User	76%	59%	66%

Tableau 2, résultats des identifications automatiques

Ldig a le F-score le plus élevé. Langid est le deuxième dans notre tableau. User est le dernier de notre tableau. Les scores de Cld2 ne sont pas assez performants par rapport à Ldig et Langid. Le F-score est supérieur à 60% pour tous les 6 identificateurs.

Nous avons vérifié la corrélation entre les tweets annotés comme étant ambigus par les annotations manuelles et la moyenne de F-score de tous les identificateurs par cluster. Par la suite nous avons tracé le diagramme suivant :



graphique 2, corrélation entre le F-score et le pourcentage d'ambiguïté

L'axe horizontal étant le pourcentage d'ambiguïté de ses tweets (selon l'étiquetage manuel), chaque point représente un cluster. Nous avons en ordonnée la moyenne de F-score des 6 identificateurs par cluster. Plus le cluster a de tweets ambigus, plus le F-score moyen des identificateurs est bas dans ce cluster.

4.4.4 Rappel et précision pour la combinaison des identificateurs

Nous avons aussi combiné les identificateurs qui ont les précisions et les rappels les plus forts afin de vérifier le changement de F-score et voir si en les combinant nous pouvons améliorer les résultats.

Pour cette partie nous avons continué sur notre échantillon de 6 000 tweets annotés à la main.

Sachant que nous avons déjà inséré les résultats des annotations des identificateurs dans notre base de données, nous avons pu faire des requêtes pour trouver les 6 000 tweets de notre corpus avec leurs annotations automatiques par les identificateurs.

Nous avons ainsi fait des requêtes dans MongoDB avec les id de ces 6 000 tweets pour trouver les tweets qui sont annotés comme étant du français par à la fois les deux identificateurs (selon nos combinaisons). Nous avons ensuite comparé les résultats avec les détections manuelles et calculer le rappel et la précision et le F_score pour chaque couple. Le tableau suivant montre le F-score pour ces combinaisons d'identificateurs.

Identificateurs	F-score
Twitter et Ldig	85%
Twitter et User	78%
User et Ldig	77%
Twitter et User et Ldig	76%

Tableau 3, F-score des combinaisons des identificateurs de la langue

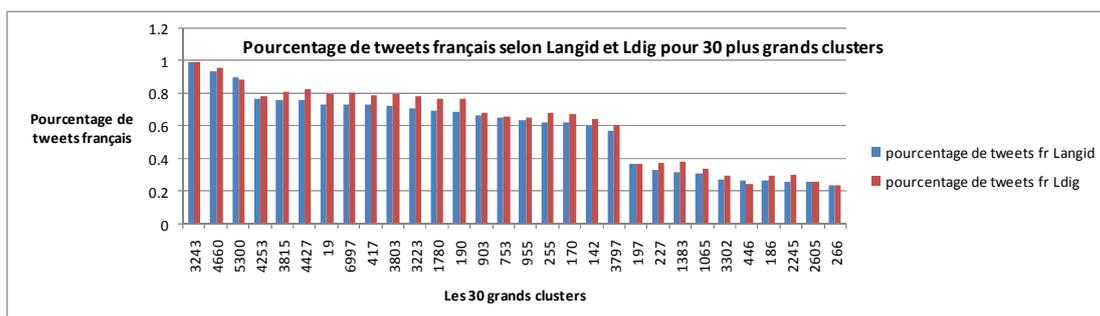
Nous n'avons pas eu de meilleurs résultats concernant le F-score avec ces combinaisons. Donc nous avons décidé de ne pas utiliser de ces combinaisons.

4.4.5 Choix final de l'identificateur automatique et insertion des résultats des identifications dans MongoDB

Après avoir travaillé sur un petit échantillon de nos données (les 6 000 tweets), nous avons considéré que les deux identificateurs Ldig et Langid sont plus performants au regard de leur F-score. Donc, nous avons utilisé ces deux, pour identifier la langue de tous les tweets du corpus B et nous avons éliminé Cld2 qui est moins performant.

Nous avons inséré ces résultats de l'identification, dans la base de données. Le diagramme ci-dessous montre, pour les 30 plus grands clusters, le nombre de tweets français par cluster selon les deux identificateurs Langid et Ldig.

Nous voyons que Langid et Ldig sont globalement d'accord pour les résultats obtenus.



graphique3, pourcentage de tweets français selon Ldig et Langid pour les 30 plus grands clusters

4.5 Conclusion

Nous voyons qu'il y a des clusters qui tweetent presque uniquement en français et les tweets diffusés par les utilisateurs de ces clusters sont moins ambigus. De plus, il existe des clusters qui sont plus complexes que d'autres pour l'identification de leur langue. Cette complexité est liée au mélange de deux langues ou à l'utilisation des hashtags ou mentions qui n'ont pas un contexte particulier. Quand le pourcentage des tweets ambigus augmente, les identificateurs ont plus de difficultés à détecter la langue des tweets. Quand nous trouvons plus de tweets ambigus pour un cluster, le F-score est plus bas. Cela veut dire que les identificateurs sont incapables de réagir envers les tweets ambigus. Cela a également été le cas pour la détection manuelle. Le problème des tweets ambigus reste le même dans les deux cas. Donc, il faut une sorte de normalisation ou de désambiguïsation pour ce genre de tweets. Par exemple, si un tweet contient uniquement un nombre, on va considérer qu'il est en français car il est géolocalisé dans une région française.

5 Analyse morpho-syntaxique des tweets

5.1 Pourquoi l'analyse morpho-syntaxique des tweets ?

Pour une analyse linguistique plus fine des tweets de nos clusters, nous avons besoin de faire une analyse morpho-syntaxique de ces tweets. Avec cette analyse, nous aurons la possibilité de vérifier les différences linguistiques qui existent entre les clusters au niveau de l'utilisation des différentes parties du discours ou le nombre de tokens par tweet ou la longueur des tokens. Pour cela nous avons choisi d'utiliser l'analyseur MElt.

5.2 Analyseur MElt

5.2.1 Présentation de l'analyseur MElt

MElt est un étiqueteur morpho-syntaxique libre-service, développé dans l'équipe d'Alpage par Pascal Denis et Benoît Sagot. Il est basé sur le modèle de chaîne de Markov à maximum d'entropie (MEMM). C'est un étiqueteur adapté pour l'analyse des textes bruités, il comporte donc une étape de normalisation. Chaque ligne étant une phrase, le format de sortie de MElt est une série de mots annotés mot/étiquette/probabilité séparés par des espaces. Voici un exemple de tweets annotés par MElt de cette manière :

```
La/DET/0.999997596102 batterie/NC/0.998097404522 de/P/0.999682540908  
mon/DET/0.819678627324 téléphone/NC/0.998192074148  
meurt/V/0.998281963964 plus/ADV/0.997822241042 vite/ADV/0.8338042831  
que/CS/0.954211330489 des/DET/0.976683310825 parents/NC/0.999859599512  
dans/P/0.999868242281 un/DET/0.993384823012 Disney/NPP/0.924351672686
```

Selon le site consacré à MElt¹, il est utilisé pour l'analyse des segments bruités comme les microblogs. MElt, faisant une normalisation de ces textes, aide à une analyse plus correcte des segments. Malgré ces étapes de normalisation nous avons constaté dans notre travail que cet analyseur n'arrive pas à annoter certains mots qui sont mal orthographiés ou qui sont oralisés, comme par exemple le mot « Bonjour » qui se présente sous la forme « Bonjourrrrrr » ou « Boooooonjour » est étiqueté comme une interjection par MElt.

¹ [https://www.rocq.inria.fr/alpage-wiki/tiki-index.php?page=MElt%20\(fr\)](https://www.rocq.inria.fr/alpage-wiki/tiki-index.php?page=MElt%20(fr))

5.2.2 Présentation des étiquettes de MElt

Voici, dans le tableau ci-dessous, les catégories grammaticales existantes dans MElt:

Étiquettes	
ADJ	adjectif
ADJWH	adjectif interrogatif
ADV	adverbe
ADVWH	adverbe interrogatif
CC	conjonction de coordination
CLO	pronom clitique
CLR	pronom réflexif clitique
CLS	pronom clitique
CS	conjonction de subordination
DET	déterminant
DETH	déterminant interrogatif
ET	mot étranger
I	interjection
NC	nom commun
NPP	nom propre
P	préposition
P+D	préposition+déterminant amalgame
P+PRO	préposition+pronomamalgame
PONCT	ponctuation
PREF	préfix
PRO	pronom personnel
PROREL	pronom relatif
PROWH	pronom interrogatif
V	verbe indicatif ou conditionnel
VIMP	verbe impératif
VINF	infinitif
VPP	participe passé
VPR	participe présent
VS	subjonctif

Tableau 4, les étiquettes de MElt

5.2.3 Étapes d'étiquetage des tweets avec MELt

Le schéma ci-dessous explique nos étapes d'analyse morpho-syntaxique des tweets avec MELt. Le script pour annotation par MELt contient la classe MELtWrapper qu'on va expliquer sous de ce schéma.

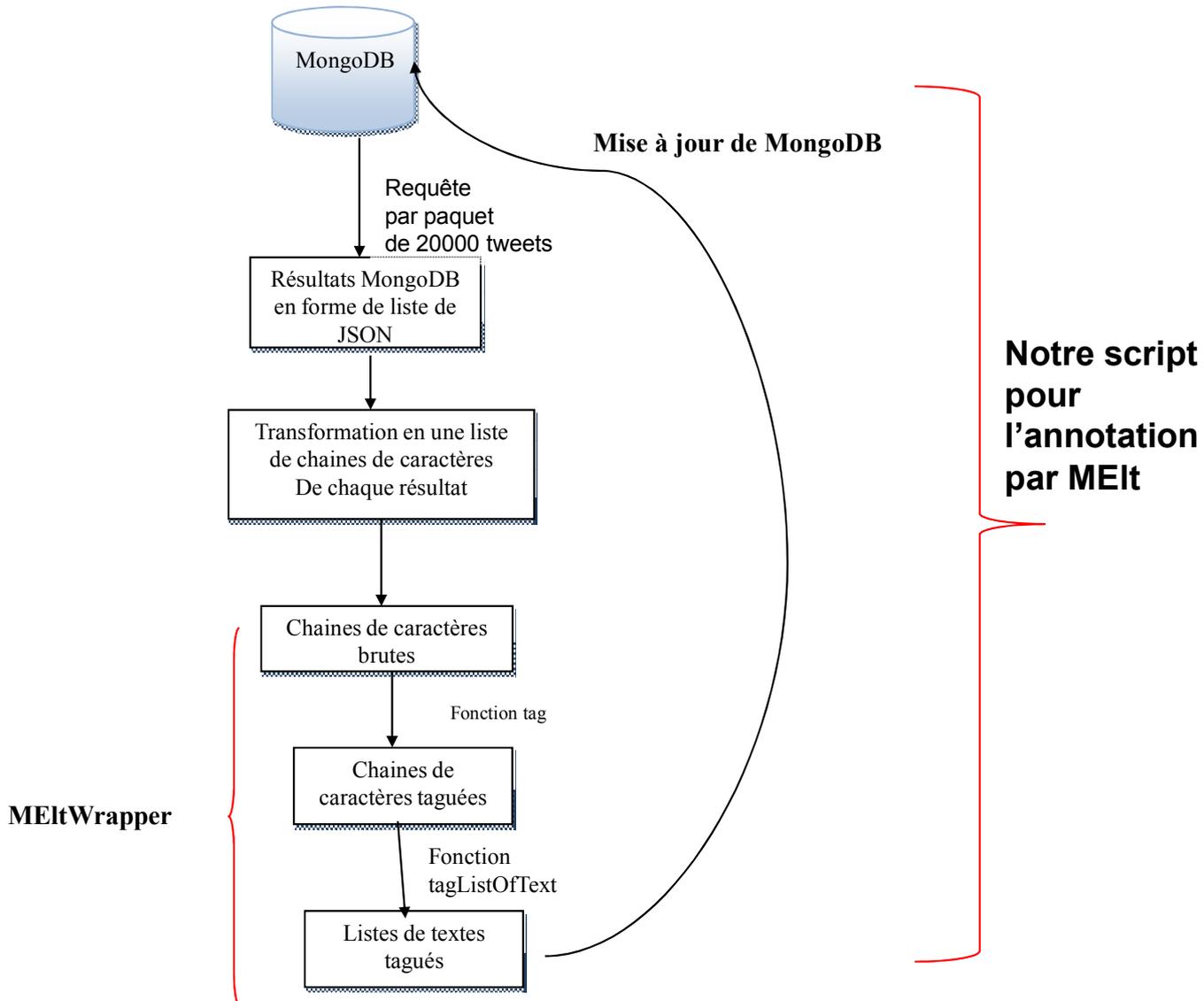


Figure 10, schéma des étapes d'analyse morpho-syntaxique avec MELt

Pour utiliser MELt dans des scripts python, nous avons défini une classe MELtWrapper. Elle fait le lien entre MELt et Python (Voir ce script en annexe III).

Dans cette classe nous avons défini les deux fonctions suivantes :

- **tagListOfTexts** : cette fonction prend comme paramètre une liste de textes à taguer. Elle les regroupe en chaîne de caractères à partir de retour chariot et elle renvoie les résultats.

```
def tagListOfTexts(self, liste):  
    chaine="\n".join(liste)  
    res=self.tag(chaine)  
    return res
```

- **tag** : Cette fonction prend comme paramètre un texte qu'elle transmet à MELt et renvoie les résultats de MELt sous forme d'une grande liste de liste de dictionnaires (au sens de Python, i.e. des tableaux associatifs).

Un dictionnaire représente un token. Un document est donc une liste de dictionnaires. Et une liste de listes de dictionnaires est donc une liste de documents tagés. Ces dictionnaires auront comme clé, les tags et comme valeurs les éléments tagués.¹

L'exemple suivant montre deux textes tagés par MELt :

```
[{'token': 'Au', 'tag': 'P+D', 'probability': '0.90305676909', 'normalization': ''}, {'token': 'tel', 'tag': 'ADJ', 'probability': '0.976219163052', 'normalization': ''}, {'token': 'avec', 'tag': 'P', 'probability': '0.99319435586', 'normalization': ''}, {'token': 'Stacy', 'tag': 'NPP', 'probability': '0.997467420283', 'normalization': ''}], [{'token': '@Alananas_', 'tag': 'NPP', 'probability': '', 'normalization': '_URL'}, {'token': 'a', 'tag': 'V', 'probability': '0.99946496138', 'normalization': ''}, {'token': 'mort', 'tag': 'VPP', 'probability': '0.605760338727', 'normalization': ''}, {'token': '...', 'tag': 'PONCT', 'probability': '', 'normalization': ''}]
```

Ensuite dans notre script d'annotation par MELt, nous importons MELtWrapper pour annoter les tweets. Comme nous avons 70 millions de tweets, si nous envoyons tous les tweets à la fois à MELt, cela prendra beaucoup de temps. Pour diminuer ce temps, nous avons choisi d'envoyer les tweets par paquet et de les insérer dans MongoDB par paquet. Ainsi, nous avons fait un calcul pour savoir combien de tweets nous pouvons mettre dans chaque paquet. En mettant un compteur de temps d'exécution sur notre script, nous avons sorti les résultats suivant concernant le temps d'exécution du programme:

¹ Voir le code en annexe IV.

Nombre de tweets à taguer	Nombre de tweets par paquet	Temps d'exécution (en seconde)	Temps moyen par tweet (en milliseconde)
1	1	6	6000
1000	100	82.94	83
1000	500	35.02	35
1000	1000	42.42	42
20000	10000	766.46	38
10000	5000	402.61	40

Tableau 5, calcul du temps d'exécution de MElt pour l'envoi par paquet des tweets

Si nous envoyons 1000 tweets à MElt en les mettant à chaque fois dans les paquets de 100 tweets, le temps d'exécution sera 82,94 secondes, tandis que si nous envoyons le même nombre de tweets dans un seul paquet de 1000 tweets le temps d'exécution sera divisé par deux. Selon le même principe nous envoyons les tweets par paquet de 20 000 à chaque fois pour diminuer le temps d'exécution.

Nous faisons une liste (*listeDeId*) avec tous les ids de ces tweets et nous faisons une autre liste (*listeDeTextes*) avec les tweets. Nous appelons la fonction *tagListOfTexts* et nous lui donnons comme paramètre *listeDeTextes*. Nous mettons les résultats de cette fonction dans MongoDB. Nous vidons *listeDeId* et *listeDeTextes* et nous recommençons la même procédure pour le paquet des 20000 tweets suivants et nous continuons comme ça jusqu'à l'annotation de tous les 70 millions de tweets¹.

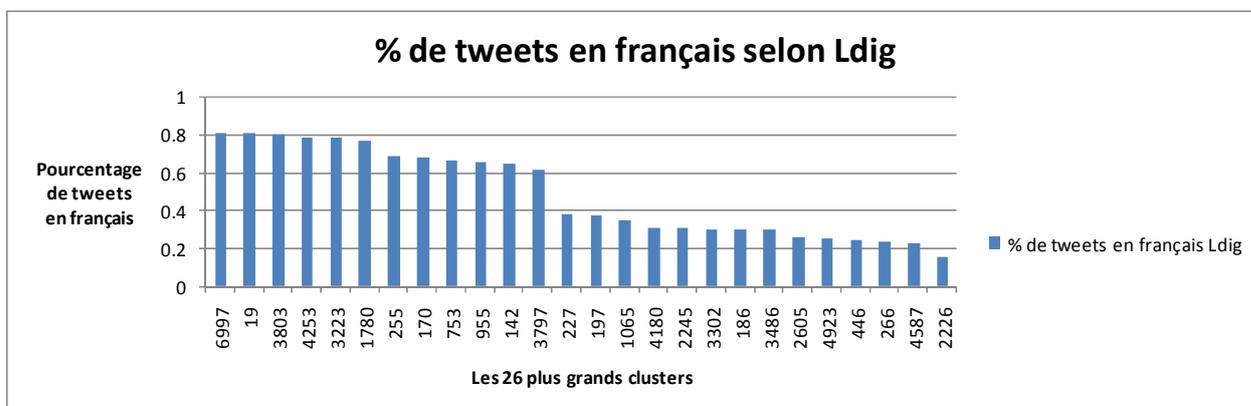
5.3 Echantillonnage des 12 plus grands clusters

Afin de diminuer le temps d'exécution de nos programmes pour les analyses statistiques sur les tweets, nous avons choisi un sous corpus de notre corpus B. Nous avons 7 000 clusters en total et 1 715986 utilisateurs au total. Tout d'abord nous avons fait un filtrage du nombre d'utilisateurs, nous avons trié les clusters qui ont plus d'un millier d'utilisateurs. Nous avons trouvé 26 clusters qui ont cette caractéristique. Le nombre d'utilisateurs de ces 26 clusters constitue un total de 1327980, qui représente 77% de tous les utilisateurs.

Le deuxième filtrage est fait sur les tweets de ces 26 clusters choisis au préalable, en prenant en compte le pourcentage de tweets français selon l'identificateur Ldig. Parmi ces clusters nous avons gardé ceux ayant plus de 50% de tweets en français. Ainsi, nous avons gardé 12 clusters.

¹Voir le code en annxe IV

Le diagramme suivant nous montre le pourcentage de tweets français, selon Langid, pour les 26 plus grands clusters.



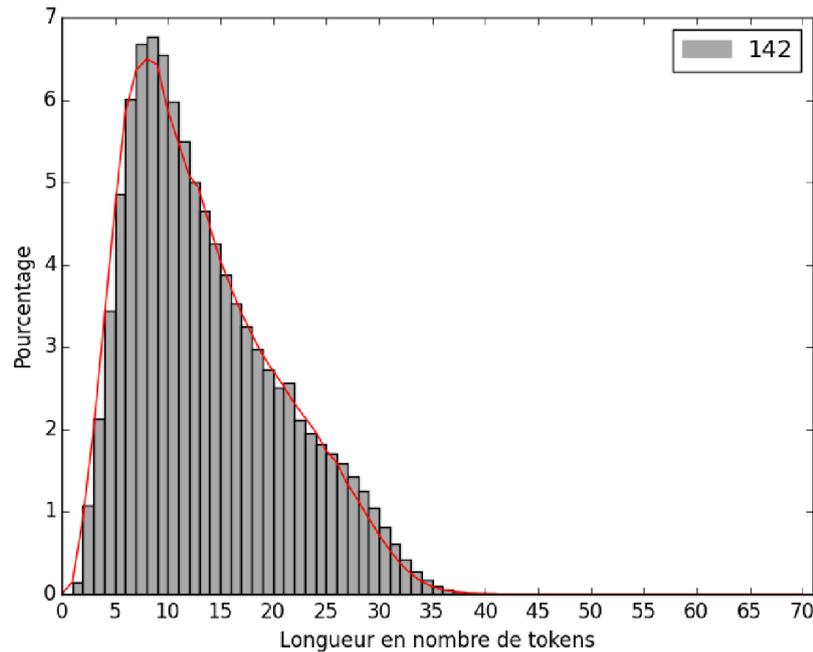
graphique4, pourcentage de tweets français selon Ldig

Nous voyons que ce pourcentage montre une chute après 12 ème cluster. Les 12 premiers clusters ont tous plus que 50% de tweets français. Ce chiffre arrive à 37% sur le 13 ème. Ainsi, dans la partie suivante, dédiée à l'étude statistique des tweets, nous allons nous concentrer sur ces 12 clusters et nous allons les étudier de plus près pour voir leurs différences et leurs particularités.

5.4 Statistiques sur les 12 clusters

5.4.1 Longueur des tweets

La longueur de chaque tweet est le nombre de tokens de ce tweet. Cette tokénisation est faite par MElt au moment de l'annotation de ces tweets. Comme nous avons expliqué dans le chapitre précédent, les utilisateurs des réseaux sociaux utilisent souvent le moins de caractères possible pour s'exprimer. Nous avons donc vérifié la longueur des tweets dans chaque cluster pour pouvoir les comparer entre eux. Nous avons dessiné ces histogrammes en Python avec le module Matplotlib. Voici les histogrammes de la distribution des longueurs de tweets pour le cluster numéro 142 qui suit la courbe de la distribution moyenne.



graphique 5, histogramme des longueurs de tweets pour le cluster 142

L'axe des abscisses de ce diagramme représente la longueur des tweets, ou le nombre de tokens qui varie de 0 à 70 (la limite donnée par Twitter étant de 140 caractères par tweets divisés par 2 en prenant compte les espaces). Sur l'axe des ordonnées nous avons le pourcentage de tweets. Les barres grises représentent la distribution des longueurs d'un cluster. La courbe rouge, identique sur tous les graphiques, est la distribution moyenne de longueur des tweets pour tous les clusters.

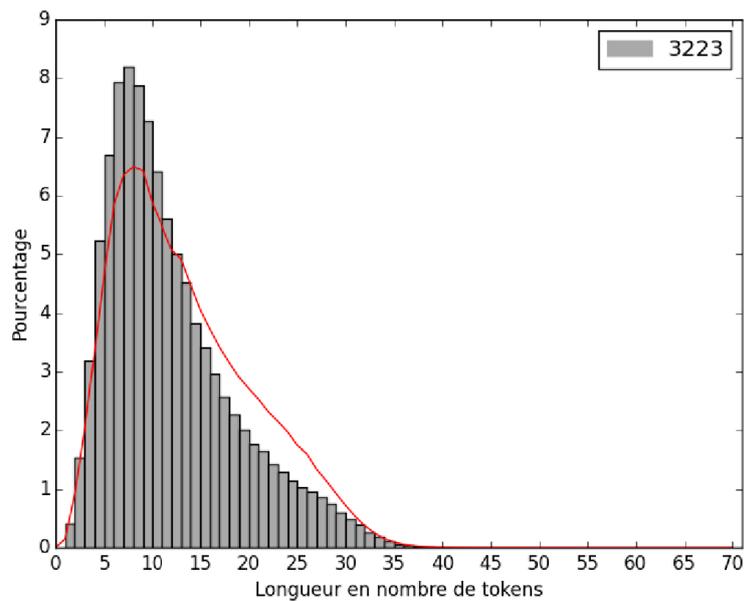
La courbe de la distribution moyenne montre que le pourcentage des tweets de longueur 1 c'est 3.25% de tous les tweets. Nous avons un sommet pour les longueurs de 7,8 et 9 tokens. Ensuite, nous avons une descente progressive à partir de la longueur de 10 qui a un pourcentage de 58% et cette descente continue jusqu'à une arrivée à zéro sur la longueur de 40 tokens. Nous pouvons en conclure que le maximum de 35 tokens fait 140 caractères dans notre corpus de tweets.

Le cluster 142 suit la courbe moyenne, nous avons regardé les tweets de la longueur 7 et 8 et 9 qui sont les sommets dans ce cluster. En voici quelques exemples :

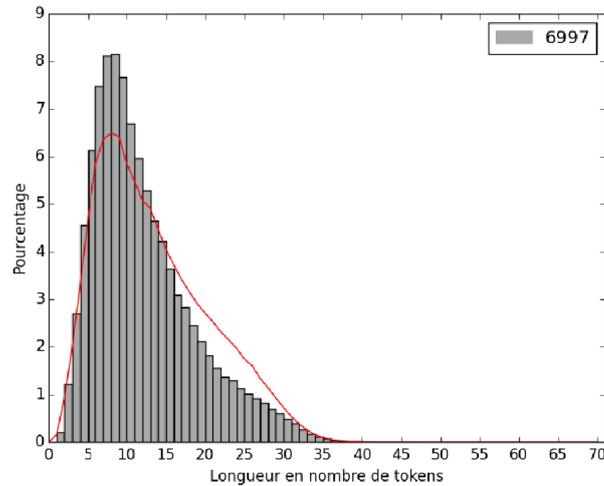
Numéro de Cluster	Ex. de tweets de longueur de token de 7	Ex. de tweets de longueur de token de 8	Ex de tweets de longueur de token de 9
142	@Aiko_SKR je vais toute la journée ☺	@weirdguurl C'est sûr à 100% !	Nn en fait j'sais pas c'est bizarre
	Sympa les high kick au foot ^^	@Punishouu Je fais ce que je veux.	@Cocacollique Même avec cette astuce j'ai jamais compris

Tableau 6, exemples de tweets de longueur 7,8 et 9 pour le cluster 142

Les clusters numéros 3223 et 6997 ont plus de tweets des longueurs de 7, 8 et 9 que la courbe moyenne. Voici les histogrammes tracés pour ces deux clusters :



graphique 6, histogramme des longueurs de tweets pour le cluster 3223



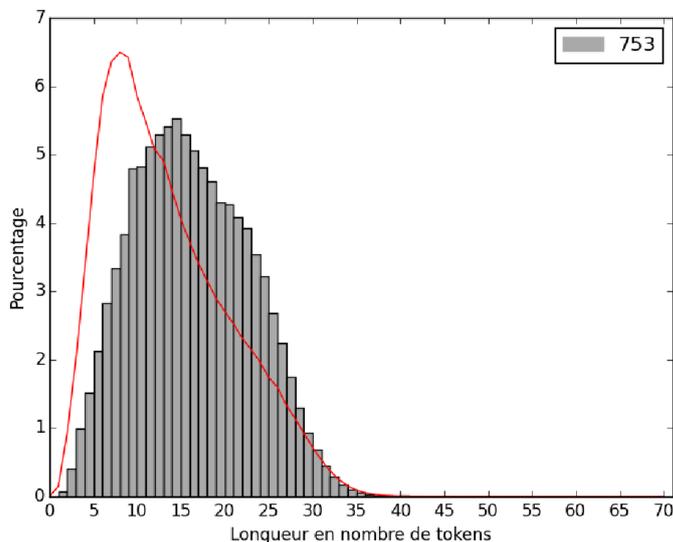
graphique 7, histogramme des longueurs de tweets pour le cluster 6997

Ces deux clusters diffusent plus de tweets de longueur de 7,8 et 9 que la moyenne. Nous avons trouvé quelques exemples de ces longueurs de token dans ces deux clusters :

Numéro de Cluster	Ex. de tweets de longueur de token 7	Ex. de tweets de longueur de token 8	Ex. de tweets de longueur de token 9
3223	@lachineblack ta supprimée les autres photos aha	Comme jsui presser de partir en vacances	je sais pas si je vais en cour demain
	Samedi je roule à 9:00	Je suis en stress total pour mon chéri	@QKeldenich mdrrr demain ça sera à qui le tour
6997	Une dixième année où ta présence manque	Aller go a l'entrainement rejoindre les autre	Assassin's Creed Unity sort pas sur play 3.
	@Romanelcq c'est deniss et qui ?	Elle sera toujours la meilleure de toutes!	Sans la géologie, l'SVT c'est intéressant

Tableau 7, exemples de tweets de longueurs 7,8 et 9 pour les clusters 3223,6997

Voici l'exemple d'un cluster qui a un pic sur la longueur 15. Nous avons regardé les tweets de cette longueur.



graphique 8, histogramme des longueurs de tweets pour le cluster 753

Numéro de Cluster	Ex. de tweets de longueur de token de 15
753	<p>"Une poule sur un mur - karaoke - Clipounets" : http://t.co/w7oPAVa9wq via @YouTube</p> <p>@quentin_be @lalibrebe et on ne parle pas de http://t.co/PQt9A4mXxI qu'on a du fermer...</p>

Tableau 8, exemples des tweets de longueur 15 pour le cluster 753

Avec ce cluster, nous voyons que le nombre de tweets de longueur entre 15 et 25 est nettement plus élevé que la courbe moyenne.

Voici par la suite tous les histogrammes de la longueur de tweets pour les 12 clusters en annexe VI. Ces observations nous permettent de distinguer trois types de cluster selon la longueur de tweets :

- Les clusters comme les numéros : 4253,170, 753,3797 qui font des tweets relativement longs par rapport à la distribution moyenne de la longueur.
- Les clusters qui font des tweets plus courts que la moyenne :6997, 19,3223,3803,1780,955.
- Les clusters qui suivent la courbe moyenne comme 255 et 142.

5.4.2 Nombre d'étiquettes par cluster

Comme nous avons annoté notre corpus avec l'analyseur MElt, il y a la possibilité d'analyser la répartition des catégories grammaticales des tokens par cluster.

Pour cela nous avons calculé la spécificité du nombre d'étiquettes par cluster avec le langage R.

Voici la définition de la spécificité selon Bénédicte Bommier-Pincemin(1999) dans sa thèse :

« spécificité : pour une unité caractérisante u d'un texte t , mesure de la réalisation de u dans t par rapport à toutes les autres réalisations de u dans les autres textes du corpus. ex. : forme d'écart réduit :

$$\frac{F_{ut} - \frac{F_u}{T}}{\sqrt{F_u}} \quad »$$

Elle a utilisé cette formule pour développer la spécificité en langage statistique R. Grâce à ce calcul de spécificité, nous pouvons caractériser la sur- ou la sous-représentation d'une catégorie par rapport aux autres. Quand la fréquence observée d'une catégorie n'a pas trop d'écart par rapport à la fréquence attendue globale, nous pouvons dire que la spécificité est proche de 0. Par contre, dans le cas des écarts extrêmes, comme quand un mot est sous-représenté dans le corpus ou qu'il est surreprésenté, nous avons des spécificités plus grandes en valeur absolue. Par ailleurs, une spécificité négative indique une sous-représentation, une positive une surreprésentation. Dans notre corpus, les scores de spécificités atteignent parfois des valeurs trop grandes pour être représentées en machine. R exprime ces écarts par Inf pour montrer la surreprésentation et -Inf pour le cas de la sous-représentation d'un mot dans le corpus étudié par rapport à toutes les autres réalisations de ce mot dans les autres textes du corpus.

Comme nous avons 29 catégories grammaticales, notre tableau de la spécificité est trop grand pour être présenté ici, nous nous sommes limités à 8 étiquettes pour donner un

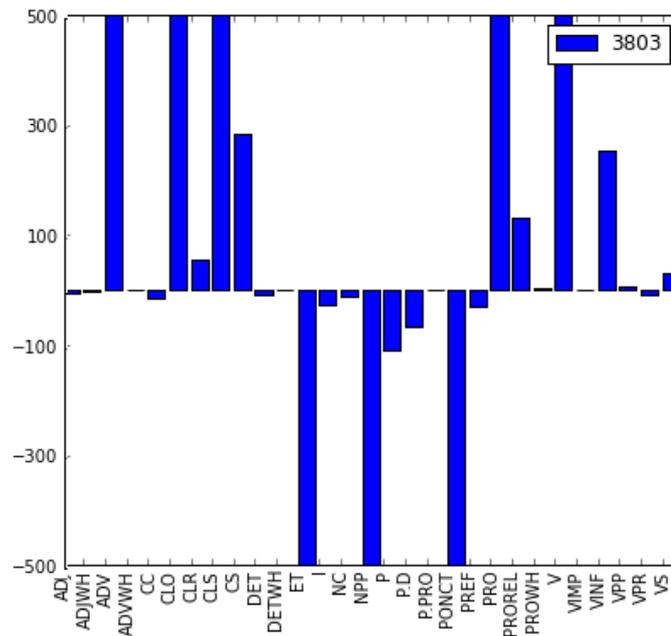
schéma de notre calcul. Voici un sous-tableau de notre tableau de spécificité pour les calculs des 20 plus grands clusters avec R.

	ADJ	ADJWH	ADV	ADVWH	CC	CLO	CLR	CLS
4253	Inf	65.9942	-Inf	-Inf	-Inf	-Inf	-Inf	-Inf
19	-3.9975	-1.2556	Inf	-2.0785	-8.5254	163.8318	47.2196	227.3974
6997	-0.3681	-1.7845	Inf	0.4807	-55.4627	267.32	40.5131	Inf
3803	-7.2276	-1.992	Inf	0.9216	-14.0181	Inf	55.1687	Inf
3223	0.3151	-3.159	Inf	-0.4968	-8.7225	Inf	63.79	Inf
1780	1.404	-7.2825	Inf	-1.382	-25.3492	132.6198	10.6556	215.6834
753	Inf	6.9688	-Inf	-243.027	-Inf	-Inf	-179.996	-Inf
955	-18.945	-0.8044	-212.297	-14.3818	-69.7623	-Inf	-15.1182	-Inf
255	-Inf	-45.2699	Inf	Inf	Inf	Inf	Inf	Inf
170	-0.4042	8.9913	-204.609	-3.4359	-Inf	-Inf	-3.1697	-Inf
142	-Inf	-4.3233	Inf	5.406	Inf	174.3425	-257.3	Inf
3797	-Inf	-3.1964	Inf	-1.6247	Inf	143.3558	0.4258	Inf

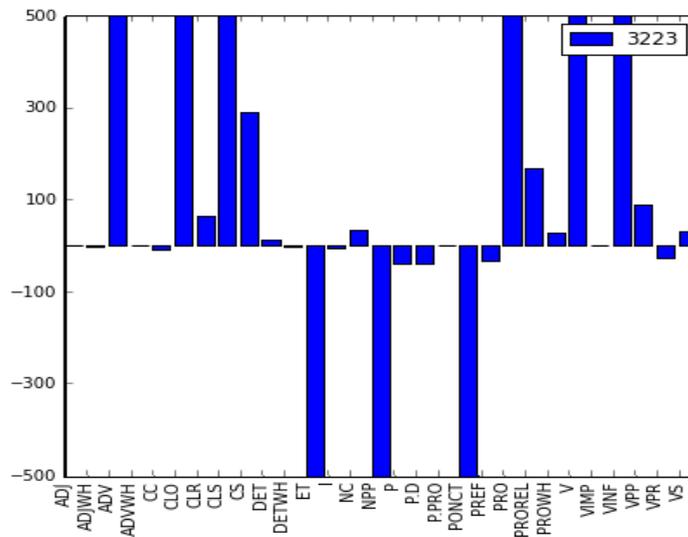
Tableau 9, spécificités de la distribution des étiquettes de MElt par cluster

A l'aide de ce tableau nous avons tracé les diagrammes suivants pour voir plus clairement ces distributions de catégories. Nous avons remplacé les valeurs « Inf » par « 500 » qui est une valeur suffisamment grande pour notre tableau. La valeur maximum étant 314.3993. Nous avons remplacé les valeurs « -Inf » par « -500 », La valeur minimum étant -257.3002.

L'axe des abscisses de ces diagrammes représente les étiquettes de MElt. Sur l'axe des ordonnées nous avons des scores de spécificité qu'on trouve dans ce tableau. Nous avons les clusters qui ont des profils identiques comme les numéros 3803 et 3223.

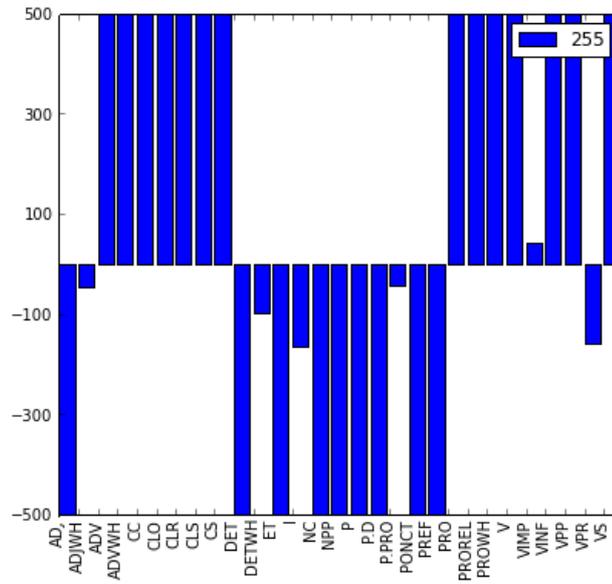


graphique 9, fréquences d'étiquettes pour le cluster 3803

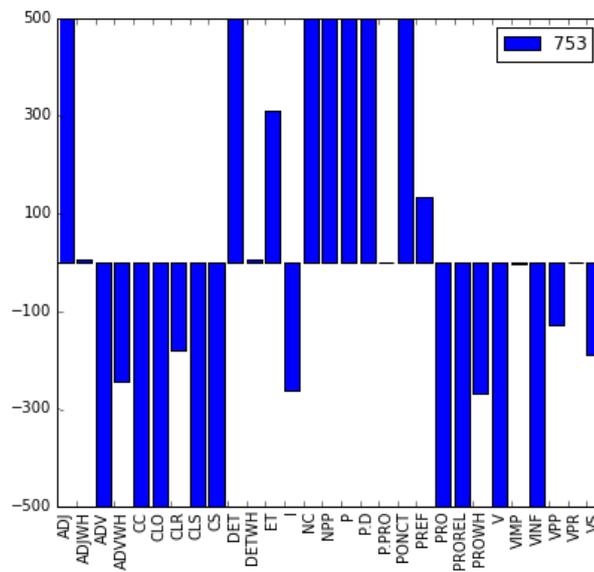


graphique 10, fréquences d'étiquettes pour le cluster 323

Il existe des clusters qui ont des profils exactement contraires. En voici un exemple :



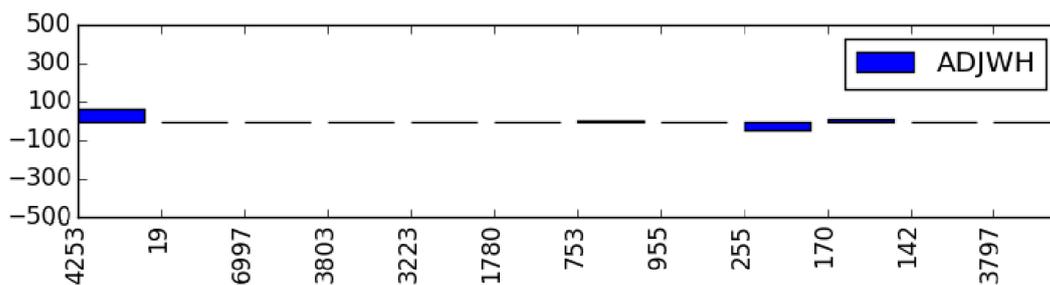
graphique 11, fréquences d'étiquettes pour le cluster 255



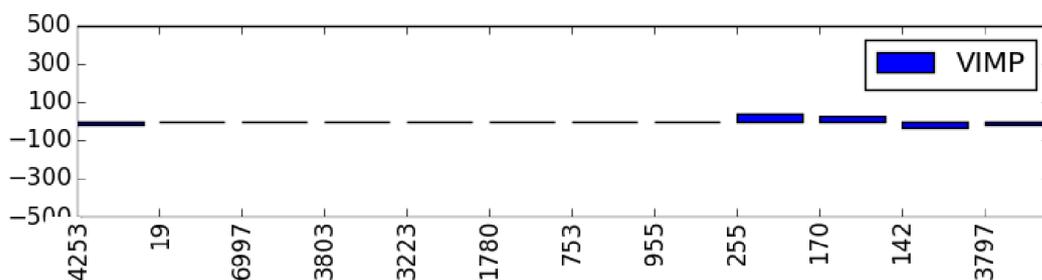
graphique 12, fréquences d'étiquettes pour le cluster 753

Les étiquettes qui sont sur représentées dans le cluster 255, sont sous représentées dans le cluster 753 et vice versa. Vous trouverez les diagrammes pour tous les clusters en annexe VII.

Nous pouvons mieux voir ces différences sous un autre angle en les regardant de plus près. L'axe des abscisses de ces diagrammes représente les clusters. Sur l'axe des ordonnées nous avons des calculs de spécificité qu'on trouve dans le tableau. Vous trouverez les diagrammes complets de toutes les étiquettes dans l'annexe VIII.

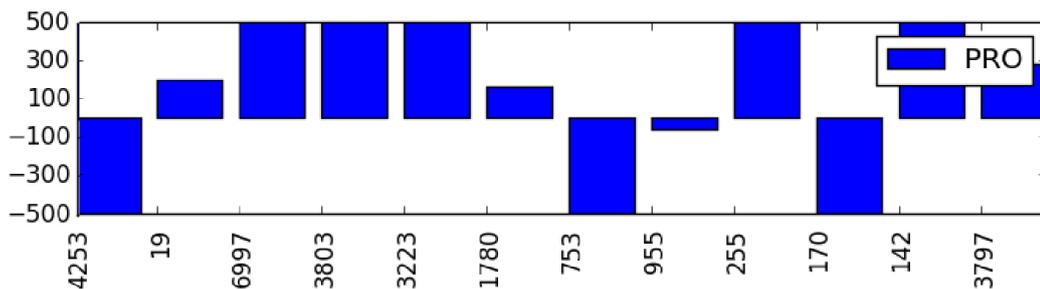


graphique 13, fréquences d'adjectifs interrogatifs par cluster

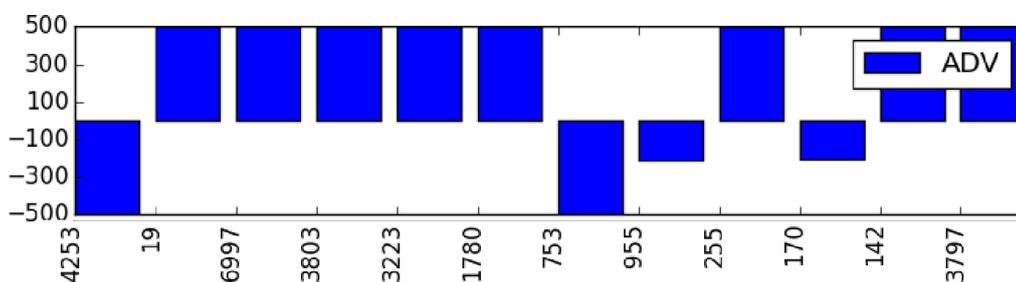


graphique 14, fréquences de verbes en impératif par cluster

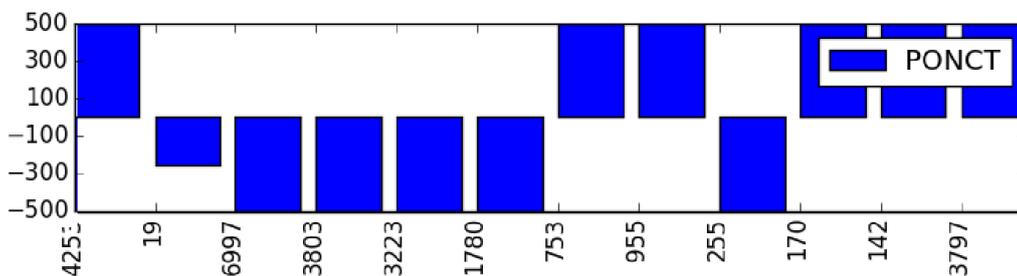
Les catégories comme ADJWH (adjectif interrogatif), DETWH (déterminant interrogatif), P+PRO (préposition+pronom amalgame), VIMP (verbe impératif), VS (subjonctif), VPR (participe présent) sont plutôt dans la norme. Tandis que nous avons plus de variété dans la distribution des adverbes, CLS (pronom clitique), NPP (les noms propres) et la ponctuation. Les deux catégories : pronom et adverbe ont des profils identiques.



graphique 15, fréquences de pronoms personnels par cluster



graphique 16, fréquences d'adverbes par cluster



graphique 17, fréquences de la ponctuation par cluster

Les catégories adverbe et ponctuation ont des profils contraires. Les clusters qui utilisent plus d'adverbes, utilisent moins de ponctuation et vice versa. Adverbe et ponctuation ont éventuellement la même utilisation. Avec la ponctuation ou les smileys nous pouvons aussi montrer une manière ou un état.

La ponctuation est sur représentée dans 9 clusters, c'est-à-dire dans presque la moitié de nos clusters étudiés. Nom propre est sous représenté dans 10 clusters. Nous pouvons en conclure que les utilisateurs utilisent beaucoup de ponctuation éventuellement pour

montrer des émotions avec les smileys. Pour analyser mieux ces catégories nous avons vérifié les mots spécifiques pour les 12 plus grands clusters.

5.4.3 Exemples des tokens spécifiques pour les 12 plus grands clusters

Nous avons extrait les tokens qui apparaissent plus de 2000 fois dans les 12 clusters. Ensuite nous avons calculé la spécificité de ces tokens avec le langage R. Nous avons extrait tous les tokens qui ont une valeur « Inf » ou qui sont sur-représentés. Dans le tableau suivant sont reportés quelques exemples des mots sur-représentés pour les 12 clusters.

Exemples des tokens sur-représentés	Numéros de cluster
réveiller, Hier, humeur, dort, dors, Wlh, mfaire, deja, kfc, embrouilles, embrouiller, aprs,	255
été, accueille, travaux, Euro, Valls, Cc, Conseiller, directement, second, immigrés, familles, #Mercato	4253
peux, X, fail, wtf, ., o, avais, @Capetlevrai, :(, :), V, Ouai, :o, :s, :O, :D, team, :P, manette, Oui, drama	142
Ivoire, YAKALA, k, Le, La, L', loool, Congo, affaire, ., AIRTEL, han, unfollowers, son, africain,	170
soir, ..., !, Limoges	3803
., :), :D, Oui, #BATB, haha, xD, Mais, concert, Hotel, ..., Bill, !, ,, Muse, regarder, mais, je, oui, ça, XD, @reisylv	3797
belgique, Le, La, L', Contrôle, #TDF, accident, ., Liège, Facebook,	753
ai, suis, mdr, anniversaire, j', je, me, trop, oua, Metz	3223
mdr, soir, ..	6997
son, http://t.co/CilGnDIYq3 , decouvrir, ce, lol, @ravineshow, lool, ..., #nowplaying, infos, #thewebradioshow, DJ, Station, En, ka,	955
..., !	19
☺	1780

Tableau 10, Mots spécifiques par cluster

Dans le cluster numéro 19, il n'y a que deux tokens sur représentés. Dans le cluster 6997, il y en a trois. Dans le cluster 3223, il existe 10 tokens sur représentés. Dans le cluster 1780, il n'y a qu'un smiley. Le cluster 3803 ne possède que 4 tokens spécifiques. Les autres clusters représentent beaucoup plus de mots sur-représentés.

Nous pouvons voir que les mots spécifiques sont différents dans chaque communauté.

5.5 Conclusion

Nous avons montré la possibilité de repérer une différence linguistique entre les différents clusters de notre corpus. L'algorithme de ce découpage en cluster pourra être modifié ou amélioré pour choisir des liens plus forts entre les utilisateurs à l'intérieur d'un cluster.

Nous avons vu qu'il y a eu des ressemblances et des différences dans les statistiques que nous avons montrées entre les clusters.

La variabilité qui existe entre les clusters peut s'exprimer à trois différents niveaux :

- Au niveau de la langue utilisée pour tweeter : nous avons vu qu'il y a une différence entre les pourcentages de tweets français. Il y a les clusters qui tweetent plus en français que les autres et ils ont moins de tweets ambigus.
- Au niveau lexical : Nous voyons que les mots spécifiques varient d'un cluster à l'autre.
- Au niveau syntaxique : Nous avons vu les différences entre les longueurs de tweets et les spécificités des parties du discours. Nous avons vu que certaines parties de discours sont sur- ou sous-représentés selon les clusters. Ainsi la longueur des tweets varie et il y a les clusters qui produisent plus de tweets longs que les autres.

Pour expliquer les raisons de ces changements et ses variations il faut examiner et vérifier plus en détail les différents clusters. Ce travail est un travail exploratoire qui avait pour but de montrer certaines variations entre les communautés à l'aide des statistiques élémentaires.

5.6 Perspectives

Ce travail ouvre le chemin vers des analyses plus détaillées et plus vastes sur les tweets comme par exemple : une analyse linguistique des tweets en prenant en considération les éléments comme l'âge des utilisateurs, leur sexe, leur niveau social ou leur provenance géographique.

Nous avons déjà observé une spatialisation géographique des clusters et nous avons vu que les variabilités linguistiques observées sont donc interprétables dialectalement.

Ce travail évoque la nécessité de développer des outils plus performants pour l'analyse des textes bruités qui réussiront à faire une bonne analyse sur les textes oralisés et courts.

Le besoin d'identificateurs plus performants qui pourront identifier la langue des tweets malgré les fautes d'orthographe ou les textes oralisés ouvre, dans ce domaine, un autre chemin pour les futurs développements d'outils.

Une autre piste de recherche pourra être la vérification de l'évolution diachronique de la langue à travers les réseaux sociaux.

Les recherches sociolinguistiques pourront porter sur l'influence des liens entre les clusters et sur la variation de la langue (comment cette langue évolue, varie et se développe au fil du temps en fonction des liens entre les individus dans un cluster ?).

Le projet SoSweet vise à effectuer, par la suite, une analyse détaillée sur la dynamique des liens entre les individus, les structures sociales sous-jacentes et leurs corrélations avec les variations linguistiques.

5.7 Bilan du stage

- **Encadrement**

Durant ces 6 mois, j'ai été suivi régulièrement par mes deux encadrants. Le fait d'avoir partagé le même bureau avec eux, m'a permis de poser mes questions au fur et à mesure de mon avancement. Nous avons un échange régulier, souvent tous les jours, pour voir mon progrès et poser mes questions éventuelles.

Leur capacité d'écoute m'a permis de progresser à ma mesure sans me sentir jugée pour mon manque d'expérience. J'ai eu l'occasion de participer aux réunions d'équipe et profiter de leurs échanges, sur leurs sujets de recherche.

Durant le mois de novembre, j'ai pu participer à une réunion de l'équipe du projet SoSweet, où tous les intervenants pouvaient discuter et donner leurs avis sur le déroulement du projet. Cela m'a appris la manière dont nous pouvons nous y prendre dans une équipe selon nos compétences et nos capacités d'écoute.

J'ai eu l'occasion de participer aux différents séminaires de linguistique du laboratoire ICAR. J'ai pu ainsi acquérir de nouvelles connaissances dans les domaines de linguistique de corpus.

- **Difficultés rencontrées et solutions apportées**

J'ai souvent rencontré des problèmes durant les phases de programmation en Python. Comme j'ai appris ce langage de programmation dans le cadre de mon stage, les premières semaines ont été consacrées à lire des manuels et faire des exercices variés pour apprendre à programmer en Python.

J'ai souvent eu des problèmes dans mes algorithmes. Pour gérer ces problèmes, j'ai essayé de découper mes algorithmes en plusieurs étapes.

Le manque d'expérience dans le domaine du TAL était une autre de mes difficultés. Je n'arrivais souvent pas à voir tous les aspects découlant d'une problématique et à trouver les solutions adaptées. Il me fallait donc du temps pour y réfléchir. Mais cela m'a permis d'avoir des échanges très intéressants avec mes encadrants et de profiter de leur expérience dans le domaine et de leurs points de vue divers afin d'élargir ma vision.

Sur le plan logistique, durant les deux premiers mois au laboratoire, nous étions souvent obligés de changer d'ordinateurs jusqu'à trouver celui qui soit le plus adapté à notre travail. Nous avons donc passé beaucoup de temps pour installer les différents logiciels sur les différents ordinateurs configurés avec le système d'exploitation Windows. Finalement, nous avons décidé d'utiliser l'environnement Linux pour gérer ce problème. Comme nous avons travaillé avec beaucoup de données, j'ai aussi été confrontée à des problèmes de gestion d'espace sur le disque dur.

- **Enrichissement personnel**

Ce stage était pour moi un premier pas vers le monde professionnel lié au TAL.

En organisant moi-même le temps consacré à chaque sujet et à chaque problématique rencontrée, j'ai pu trouver une autonomie dans mon organisation du travail.

Mon stage ayant débuté quasiment au même moment que le projet SoSweet, j'ai pu assister au plus près au développement pas à pas d'un projet.

Au niveau de la programmation, j'ai appris à programmer en Python, langage qui présente moins de contraintes au niveau de la ponctuation que les langages comme Perl.

J'ai également pu développer mes connaissances en programmation orientée objet.

J'ai pu travailler avec MongoDB, manipuler les requêtes, gérer de grandes quantités de données, en abordant des problématiques telles que le stockage des données, leur suppression et l'élaboration d'échantillonnages représentatifs.

En découvrant l'environnement Linux, j'ai appris à installer des logiciels sur cet environnement et à travailler avec.

J'ai pu connaître le site de Twitter et son fonctionnement.

Ce stage était bénéfique à tous les points de vue pour moi et j'ai pu acquérir une bonne expérience dans le domaine du TAL et j'ai pu développer mes qualités relationnelles avec l'équipe de travail.

Avec les compétences acquises pendant ce stage, je me sens prête à travailler dans le domaine du TAL. Je commencerai mon travail au début de janvier à l'Inist qui est l'unité technique de CNRS. Nous allons travailler sur le projet ISTEEX, dans une équipe constituée de linguistes et d'informaticiens et de talistes.

Le projet ISTEEX a pour but de donner l'accès aux chercheurs et aux scientifiques à une base de données d'articles scientifiques. L'équipe du projet développe les interfaces pour faciliter l'accès à ces articles grâce à des moteurs de recherches performants.

Mes connaissances en Python, mon travail avec de grandes quantités de données, l'utilisation de MongoDB, de XML etc. vont m'aider à avancer et progresser dans ce travail.

6 Bibliographie

- **Articles**

HAYTHORNANTHWAITE, and GRUZD, A.(2007), A noun phrase analysis tool for mining online conversations, *Communities and Technologies*.

NGUYEN, T. (2010), Hyper community detection in the blogosphere, *In Proceeding of second ACM SIGMM Workshop on Social Media*, Florence: ACM.

LUI, M. BALDWIN, T. (2012), Langid.Py: An Off-the-Shelf Language Identification Tool, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 25–30,.

LUI, M. BALDWIN, T. (2014), Accurate Language Identification of Twitter Messages, *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM) @ EACL*, 17–25.

- **Livres**

ZAPPAVIGNA, M. (2012). *Discourse of Twitter and social media how we use language to create affiliation on the web*, London: Continuum.

- **Mémoires et thèses**

BAMMEY,Q. (2015). Characterization of community structure and linguistic variability on Twitter, (mémoire École Normale Supérieure de Lyon).

BOMMIER-PINCEMIN, B.(1999). Diffusion ciblée automatique d'informations : conception et mise en œuvre d'une linguistique textuelle pour la caractérisation des destinataires et des documents, (thèse Université Paris IV).

- **Sitographie :**

[Consultée en décembre 2015]

<http://icar.univ-lyon2.fr/>

<https://about.twitter.com/fr/company>

http://jacquescellier.fr/histoire/site_tdh2/fichiersexemples/detection_communautes.pdf

<http://sametmax.com/lencoding-en-python-une-bonne-fois-pour-toute/>

<https://code.google.com/p/cld2/>

http://www.alyabbara.com/moteurs_recherche/utilitaires/clock_mondial.html

[https://www.rocq.inria.fr/alpage-wiki/tiki-index.php?page=MElt%20\(fr\)](https://www.rocq.inria.fr/alpage-wiki/tiki-index.php?page=MElt%20(fr))

<http://fr.slideshare.net/shuyo/short-text-language-detection-with-infinitygram-12949447>

<https://github.com/saffsd/langid.py>

7 Annexes

Annexe I

Lascript pour faire drag et drop des tweets non catégorisés après les annotations manuelles :

```
<script>
    $(function() {
        $('#sortable1, #sortable2, #sortable3').sortable({
            connectWith: '.connectedSortable',
            receive : function (event, ui){
                $.ajax({
                    type:'POST',
                    url:'reclass.php',

                    data:'id='+ui.item[0].id+'&nom_colonne='+event.target.id,
                });
            }
        });
    });
</script>
```

Annexe II

Exemples des codes insérés dans nos scripts Python pour régler le problème d'encodage des tweets :

```
from __future__ import unicode_literals

import encodings
notre_chaine =notre_chaine.decode('utf8')
```

Annexe III

La classe meltWrapper qui fait le lien entre Python et MElt :

```
class meltWrapper():
    def __init__(self, MElt_bin, MElt_options):
        self.MElt_bin=MElt_bin
        self.MElt_options=MElt_option
```

Annexe IV

Code de la fonction tag de la classe MeltWrapper :

```
def tag(self, texte):
    cmd = [self.MElt_bin, self.MElt_options]
    p = Popen(cmd, stdin=PIPE, stdout=PIPE, stderr=PIPE)
    (stdout, stderr) = p.communicate(texte.encode('utf-8'))
    stdout=stdout.decode('utf8')
```

```

texts_tagged=[]
for line in stdout.split("\n"):

    tags='ADJ|ADJWH|ADV|ADVWH|CC|CLO|CLR|CLS|CS|DET|DETWH|ET
|I|NC|NPP|P|P\+
D|P\+PRO|PONCT|PREF|PRO|PROREL|PROWH|V|VIMP|VINFIN|VPP|VPR
|VS|KK'
    exp=u'(?:\{(?P<normalisation>.*?)\})?(?P<token>\S+)?/(?P
<tag>%s)/(?P<proba>\d\.\d+)?(?      : |$|\n)'%tags
    if line!="":
        l = re.compile(exp).findall(line)
        l2=[]
        for e in l:
            if e[0]==' ':

                l2.append({'normalization':e[0], 'token':e[1], 'tag':e[2], 'pro
bability':e[3]})
            else:

                l2.append({'normalization':e[1], 'token':e[0], 'tag':e[2], 'pro
bability':e[3]})
        texts_tagged.append(l2)
    return texts_tagged

```

Annexe V

Code pour l'envoi par paquet des tweets à MELt :

```

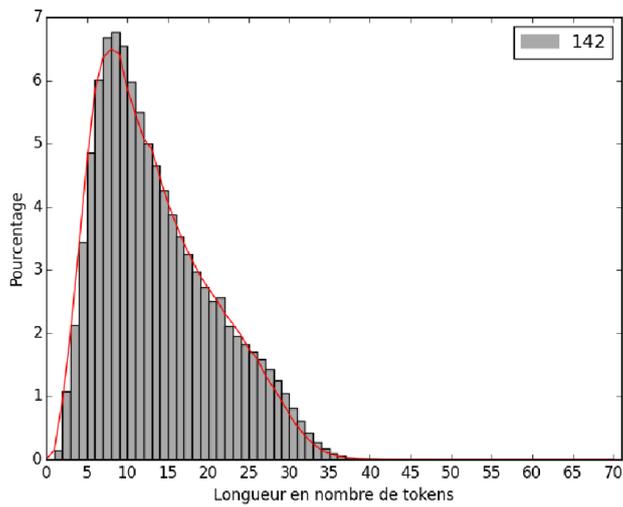
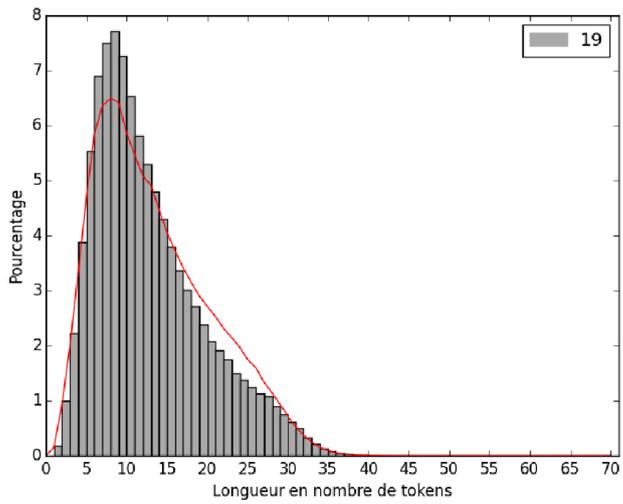
listeDeTextes=[]
listeDeId=[]
limite=20000
melt = meltWrapper.meltWrapper(MELt_bin,
MELt_options)
for tweet in
collection.find({}, {'tweet':1, 'id':1}, no_cursor_timeo
ut=True):
    tweetSansN=tweet['tweet'].replace("\n", "")
    tweetSansNR=tweetSansN.replace("\r", "")
    id_tweet=tweet['id']
    listeDeTextes.append(tweetSansNR)
    listeDeId.append(id_tweet)
    if len(listeDeTextes)>=limite:
        textes_taged =
melt.tagListOfTexts(listeDeTextes)
        for i in range(len(listeDeId)):

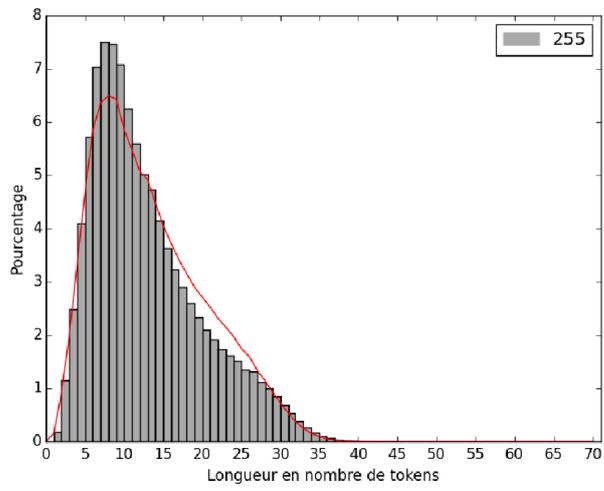
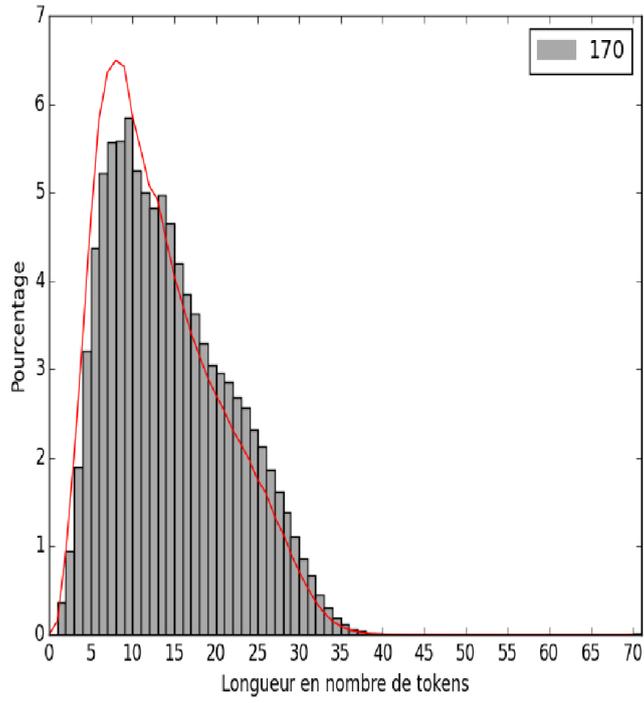
```

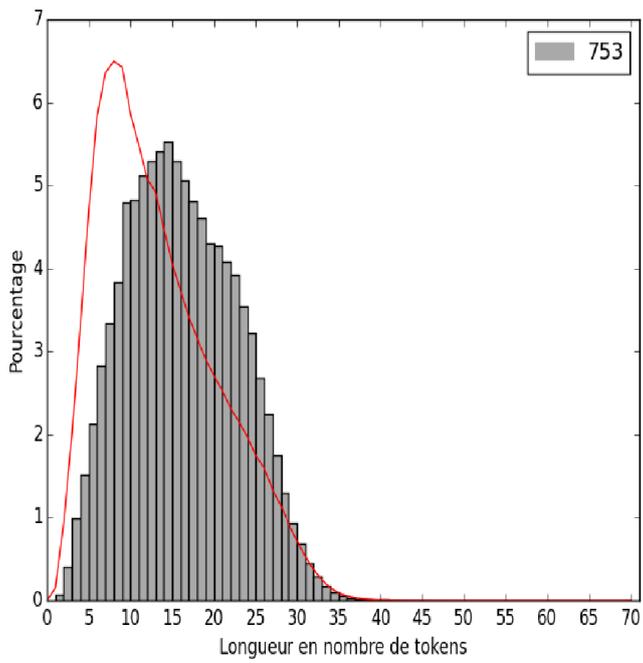
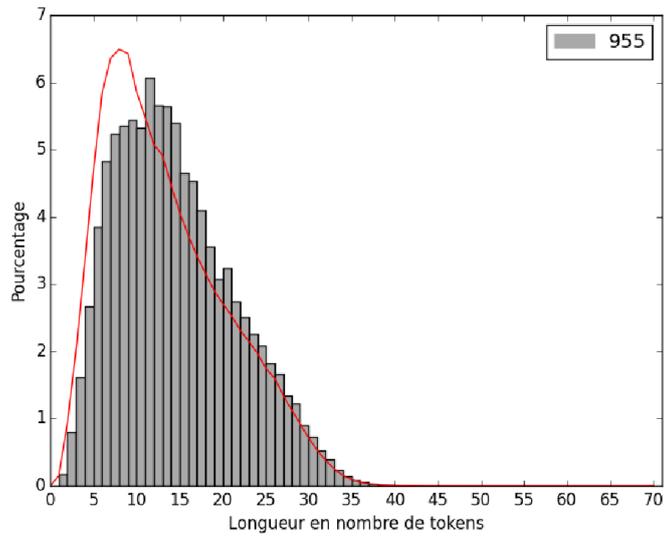
```
collection.update({'id':listeDeId[i]},{'$set':{'  
melt':textes_taged[i]}})  
listeDeTextes=[]  
listeDeId=[]
```

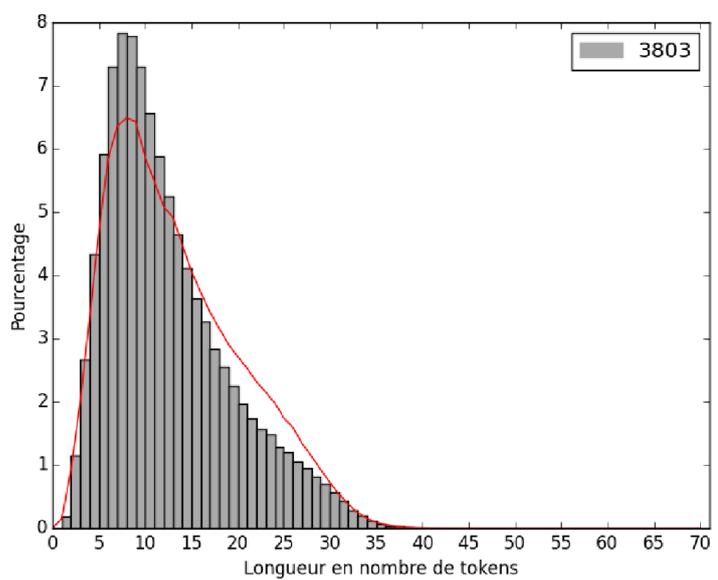
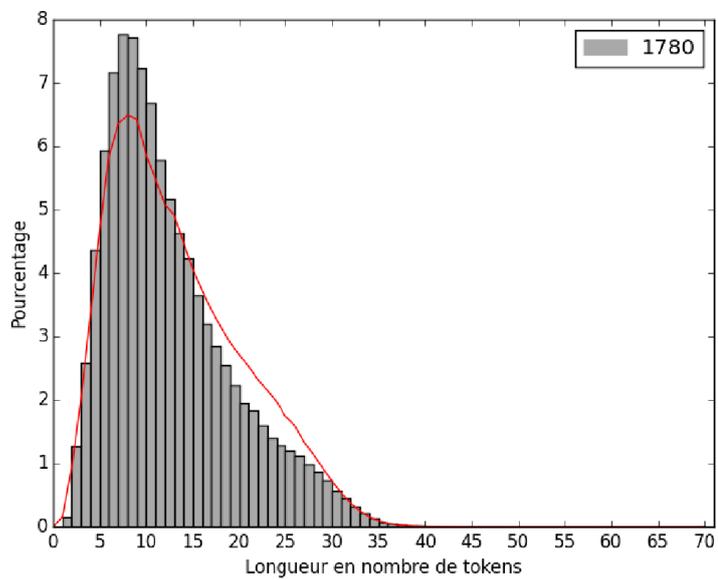
Annexe VI

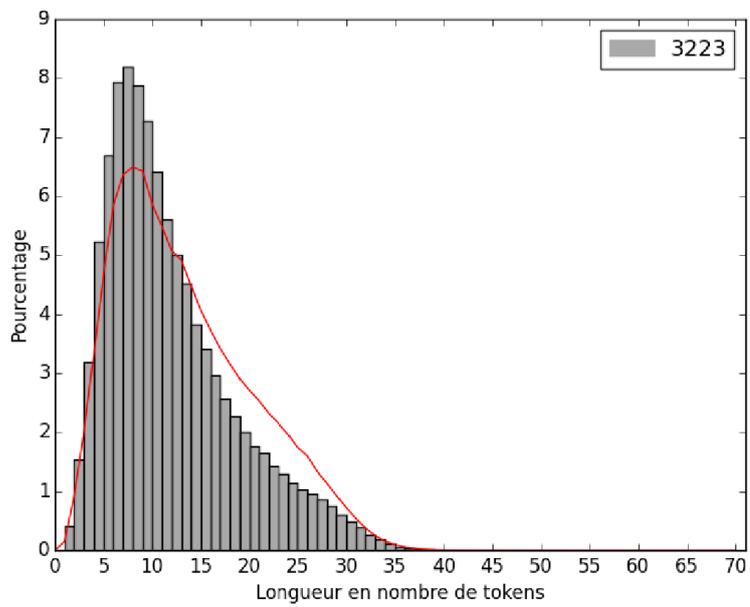
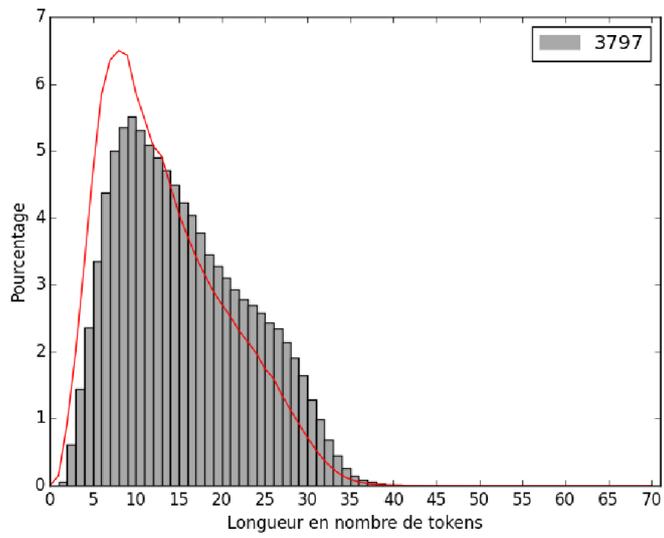
Les histogrammes des longueurs de tokens par cluster :

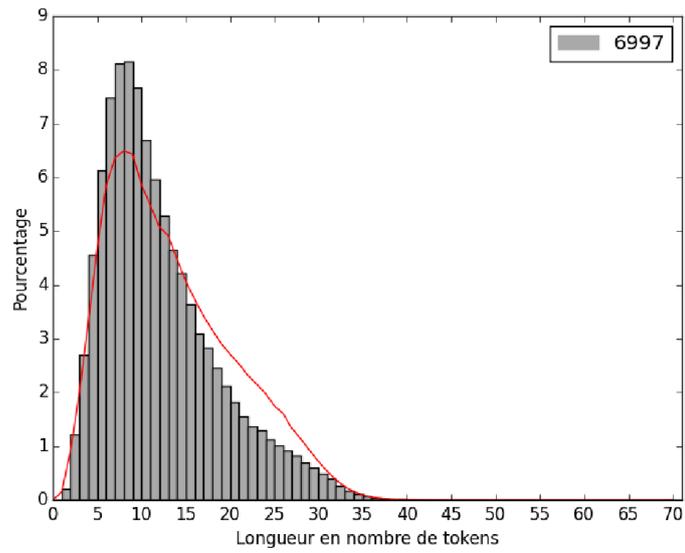
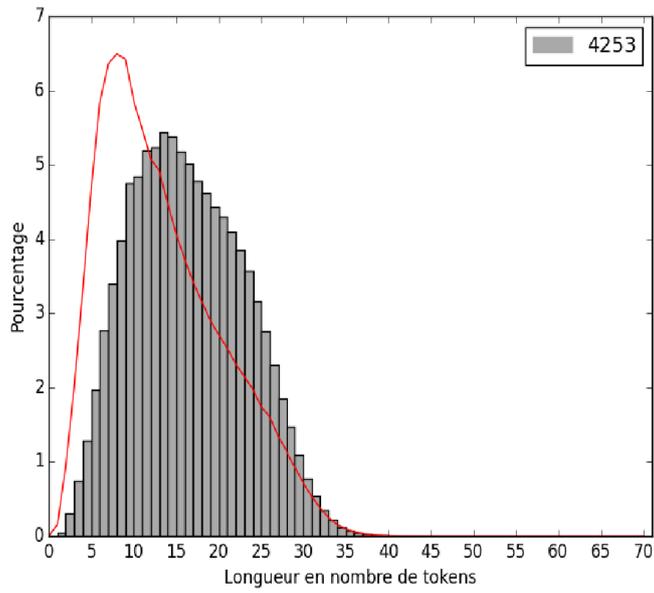






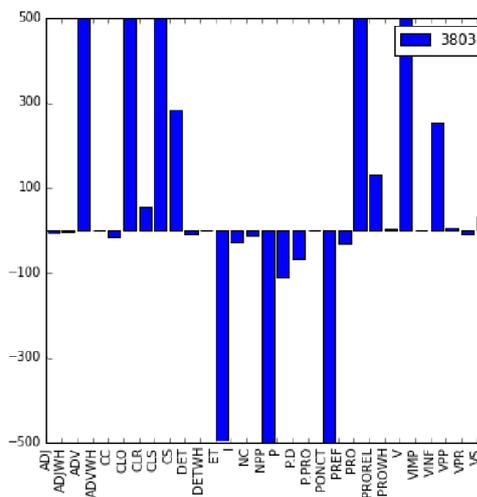
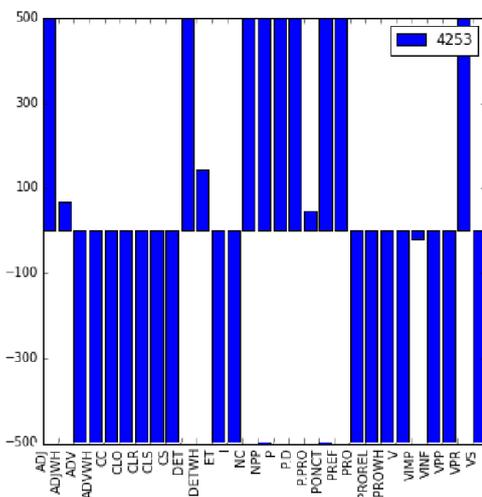
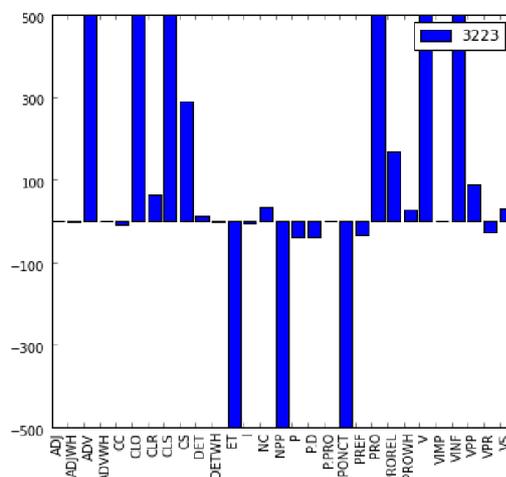
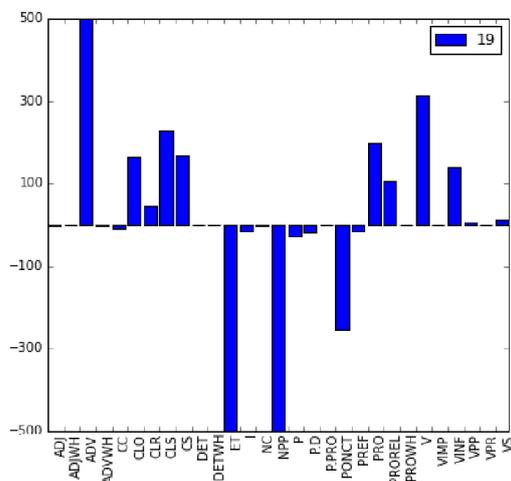


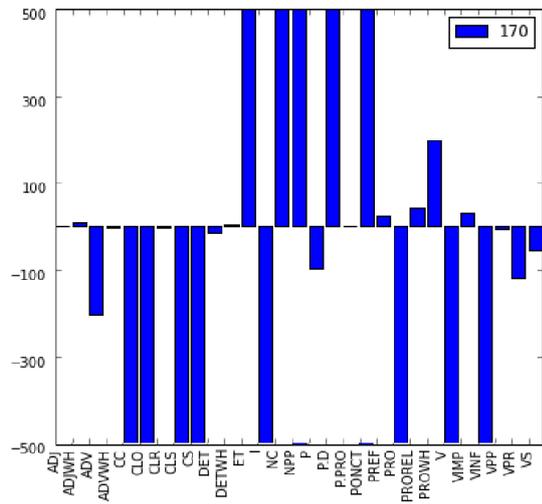
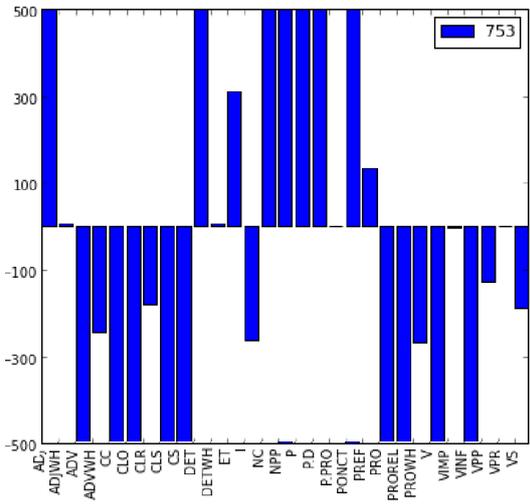
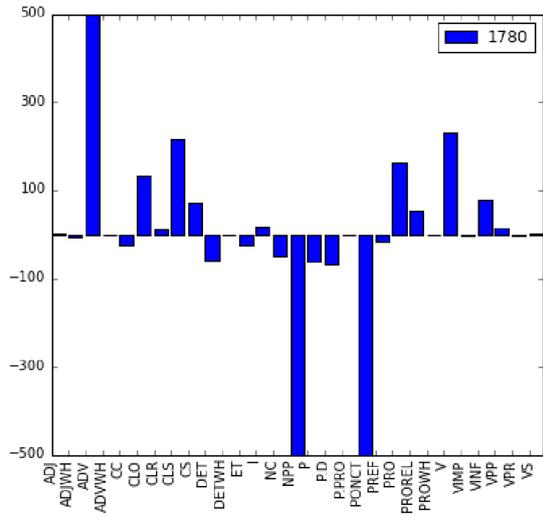
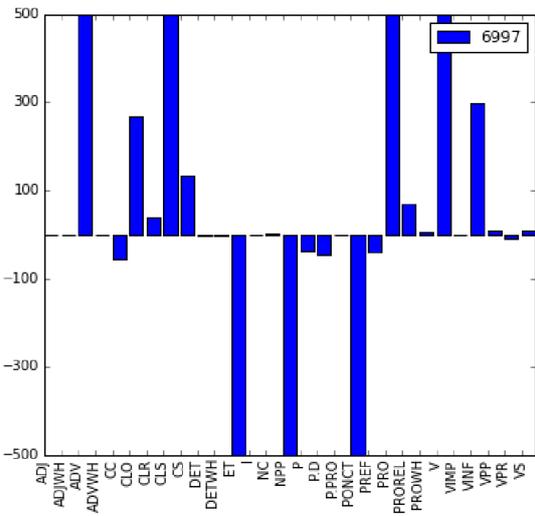


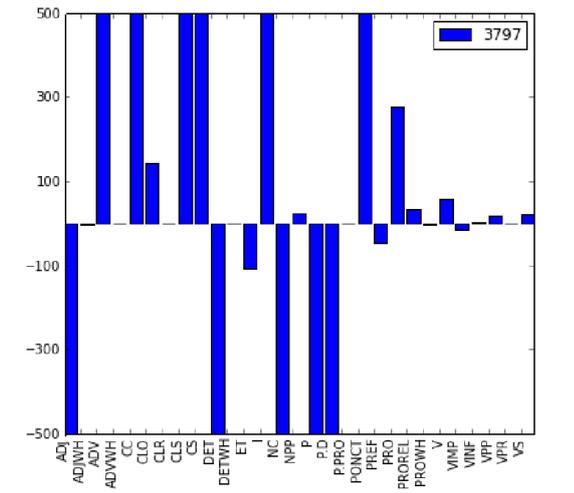
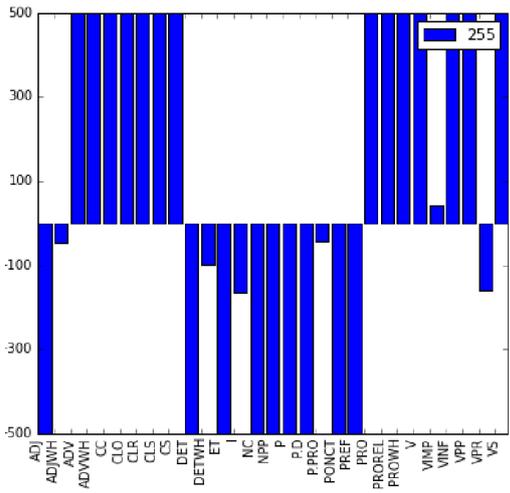
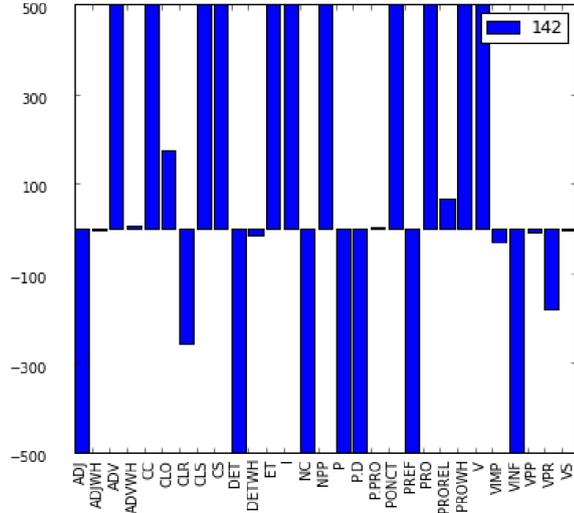
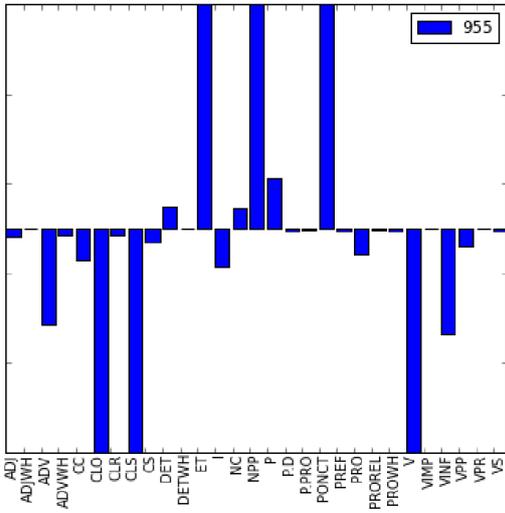


Annexe VII

Les diagrammes de la distribution des étiquettes par clusters :







Annexe VIII:

Les diagrammes de la distribution des étiquettes par clusters :

