

Analyse de tendance



*BROUTE Nicolas, DIALLO Aïcha, DOUGNAC Jeanne,
FALL Elhadji Fallou, GODE Valentin, PAILLER Nicolas,
PRADIER Jérémy, REJOUANY Meryem, ROLLET Fanny,
WADE Samba Diallo*

GROUPE n°8
Projet inter-promotion
Année universitaire : 2017/2018

SOMMAIRE

Table des figures	3
1. Objectif du groupe	5
2. Organisation	6
2.1. Planning	6
2.1.1. Planning Prévisionnel	6
2.1.2. Planning Réel	7
2.2. Gestion de projet	8
2.2.1. Organigramme	8
2.2.2. Méthode de développement	11
2.2.3. Gestion des versions et plateforme de collaboration	13
3. Déroulement des missions principales	14
3.1 Calcul des n mots les plus représentatifs d'une période	14
3.2 Analyse de tendance par jour	15
3.3 Analyse de tendance par période via des séries chronologiques	17
3.3.1 Analyse de la série par décomposition :	17
3.3.2 Analyse de la série via une droite de régression :	18
3.3.3 Analyse de la série par moyenne mobile :	20
3.3.4 Analyse de tendance par période des mots associés à un mot pertinent	21
3.4 API REST	23
4. Missions complémentaires	26
4.1 Intégration des concepts sémantiques dans les analyses de tendance	26
4.2 Prédiction de la tendance de la période suivante	27
5. Difficultés rencontrées et solutions apportées	28
6. Conclusion	29

Table des figures

Figure 1 : Planning prévisionnel	6
Figure 2 : Planning réel	7
Figure 3 : Organigramme	8
Figure 4 : Trello.....	11
Figure 5 : exemple de tâche.....	12
Figure 6 : Formule TD-IDF	14
Figure 7 : Données d'entrées	14
Figure 8 : TF-IDF pour 2 jours.....	16
Figure 9 : Résultat tendance par jour	16
Figure 10 : Exemple jeu de test semaine	17
Figure 11 : Données utilisées pour la régression.....	18
Figure 12 : Exemple de sortie graphique.....	19
Figure 14 : Sortie droite de régression	19
Figure 13 : R^2 obtenu	19
Figure 15 : Données moyenne mobile.....	20
Figure 16 : Résultat moyenne mobile.....	21
Figure 17 : Figure bigram	21
Figure 18 : Exemple bigrams	22
Figure 19 : Exemple résultat	22
Figure 20 : Schéma méthodologie des fonctions.....	23
Figure 21 : Schéma API REST.....	24
Figure 22 : Résultat top mots pour un thème donné.....	25
Figure 23 : Top adjectifs.....	25
Figure 24 : Exemple données positivités	26
Figure 25 : Résultat concept	27



1. Objectif du groupe

Ce rapport rentre dans le cadre du projet « Watch News » réalisé par la totalité de la formation SID, du 8 Janvier 2018 au 19 Janvier 2018.

Au cours de ce projet, l'objectif de notre groupe est de fournir des analyses statistiques pour alimenter le site web. Les données « disponibles » pour réaliser ces analyses sont relatives aux contenus des articles issus de différentes sources (Le Figaro, Le Monde, etc.). Des informations provenant des groupes sémantique, filtrage et prédiction sont également à disposition.

Les activités principales de notre groupe sont de fournir les n mots les plus représentatifs d'une période définie (jour, mois, semaine) et les analyses de tendances associées. La tendance d'un mot peut être définie comme l'évolution dans le temps de l'importance d'un mot dans un corpus d'articles. Nous distinguons ainsi deux types d'analyses :

- Statique : Analyse effectuée périodiquement de manière automatique
- Dynamique : Analyse effectuée via le site, à la demande de l'utilisateur, selon des besoins spécifiques

Pour mener à bien ces tâches, nous avons communiqué avec le groupe « BD QUERY ». L'une des responsabilités de ce groupe est de nous fournir les données nécessaires pour réaliser nos analyses et de transmettre les résultats au site web. Cette communication s'effectue au travers d'« API REST » (1 pour notre groupe et 1 pour le groupe « BD QUERY »).

2. Organisation

2.1. Planning

2.1.1. Planning Prévisionnel

Missions	Semaine 1					Semaine 2				
	Jour 1	Jour 2	Jour 3	Jour 4	Jour 5	Jour 6	Jour 7	Jour 8	Jour 9	Jour 10
Version 1										
1- Répartition du travail										
2- Traitements sur les données disponibles										
4- Analyse de tendance globale										
5- Récupération des JSON définit avec les groupes 2 et 3										
6- Adaptation des fonctions (génériques) réalisées aux JSON										
7- Vérification de l'interaction avec les groupes BD et mise en place des jobs										
8- Analyse de tendance à la demande										
Version 2										
9- Analyse de tendance par source et par catégorie										
10- Prédiction de la tendance d'un mot pour la semaine suivante										
Qualité et analyse du code										
1 - Préparation et mise en forme du GIT										
2 - Revue de code										
3 - Rédaction rapport										
Présentation et soutenance										

Figure 1 : Planning prévisionnel

2.1.2. Planning Réel

Missions	Semaine 1					Semaine 2				
	Jour 1	Jour 2	Jour 3	Jour 4	Jour 5	Jour 6	Jour 7	Jour 8	Jour 9	Jour 10
Version 1										
1- Répartition du travail										
2- Traitements sur les données disponibles										
4- Analyse de tendance globale										
5- Création de JSON fictif pour tester les fonctions										
6- Adaptation des fonctions (génériques) réalisées aux JSON										
7- Vérification de l'interaction avec les groupes BD										
8- Analyse de tendance à la demande										
9- Intégration des fonctions dans le serveur										
Version 2										
10- Analyse de tendance par source, par catégorie, et des mots associés										
11- Analyse de tendance en fonction des concepts sémantiques										
12- Prédiction de la tendance d'un mot pour la semaine suivante										
13- Intégration des fonctions dans le serveur										
Qualité et analyse du code										
1 - Préparation et mise en forme du GIT										
2 - Revue de code										
3 - Rédaction rapport										
4- Présentation et soutenance										

Figure 2 : Planning réel

Le diagramme prévisionnel pour la V1 a été respecté au niveau des délais ainsi que des tâches. A noter que nous avons dû créer des fichiers de données JSON afin de tester nos fonctions.

Pour le diagramme prévisionnel V2 nous avons eu une vision trop globale des fonctions à réaliser. Cependant grâce au bilan à la fin de chaque journée, nous avons vite défini les besoins nécessaires des fonctions à concevoir par la suite.

2.2. Gestion de projet

2.2.1. Organigramme

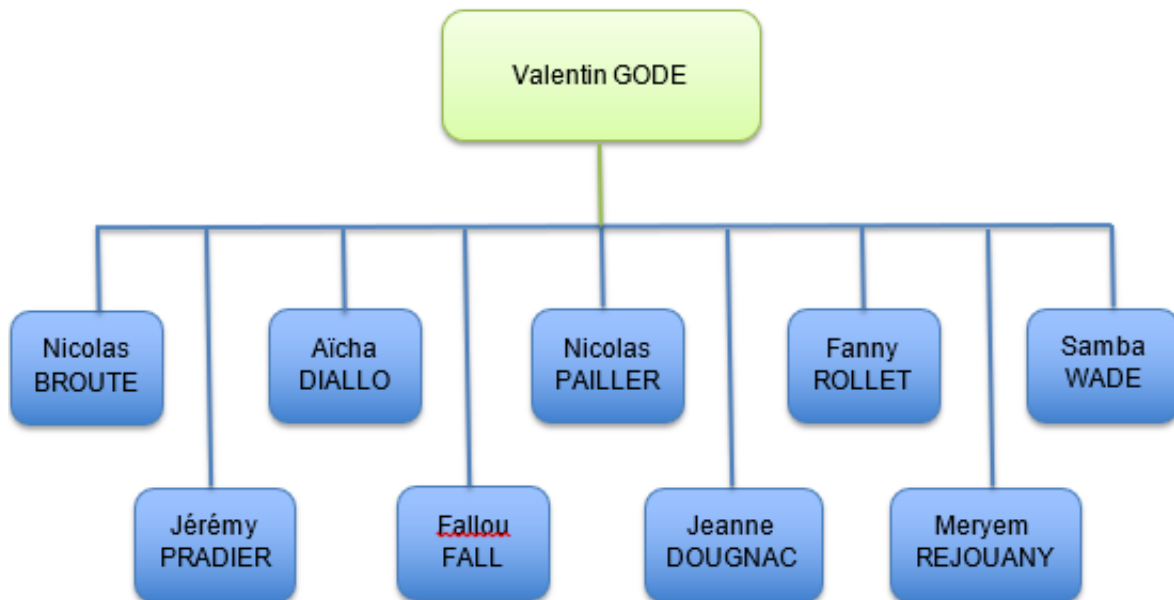


Figure 3 : Organigramme

Nom	Tâches
Valentin GODE	<ul style="list-style-type: none"> - API - Séries chronologiques - Git - Supervision - Planning - Rédaction du rapport - Analyse de tendance - Mise des fonctions sur le serveur
Nicolas BROUTE	<ul style="list-style-type: none"> - Top des mots pour la semaine (prise en compte de la nature du mot pour la V2) - Top des mots par jour (prise en compte de la nature du mot pour la V2) - Analyse des bigrams - Mise des fonctions sur le serveur
Nicolas PAILLER	<ul style="list-style-type: none"> - Régression linéaire sur séries chronologiques - Recherche sur les tests statistiques (Kendall, Durbin-Watson) - Moyennes mobiles - Rédaction du rapport
Jérémy PRADIER	<ul style="list-style-type: none"> - Polarité des mots - Recherche sur le test de Kendall - Top des mots pour la semaine
Meryem REJOUANY	<ul style="list-style-type: none"> - API - Revue de code - Régression linéaire sur séries chronologiques - Rédaction du rapport
Fanny ROLLET	<ul style="list-style-type: none"> - Analyse de tendance - Revue de code - Polarité des mots - Mise en forme des fichiers json avec le groupe BD requête

Jeanne DOUGNAC	<ul style="list-style-type: none"> - Top des mots pour la semaine (prise en compte de la nature du mot pour la V2) - Top des mots par jour (prise en compte de la nature du mot pour la V2) - Revue de code - Rédaction du rapport
Fallou Elhadji FALL	<ul style="list-style-type: none"> - Analyse de tendance - Régression linéaire sur séries chronologiques - API - Rédaction du rapport
Aïcha DIALLO	<ul style="list-style-type: none"> - Polarité des mots - Analyse de tendance - Régression linéaire sur séries chronologiques - Fonction mot par semaine - Revue de code
Samba Diallo WADE	<ul style="list-style-type: none"> - Analyse de tendance - Polarité des mots - Régression linéaire sur séries chronologiques - Recherche sur Kendall - Revue de code

2.2.2. Méthode de développement

Au sein de notre groupe, un développement selon une méthode agile a été privilégié. Les méthodes agiles reposent sur une structure itérative, incrémentale et adaptative.

Chaque itération représente un mini projet qui mène à la livraison d'un code exécutable. Cette méthode de développement permet de produire un système opérationnel et stable rapidement.

Elle permet également de prendre en compte les évolutions et/ou les améliorations en cours de projet.

L'outil central de suivi de l'avancement du projet utilisé est Trello. Il contient toutes les users stories (fonctionnalités) ainsi que les tâches associées à celles-ci.

Voici un aperçu de l'outil utilisé :

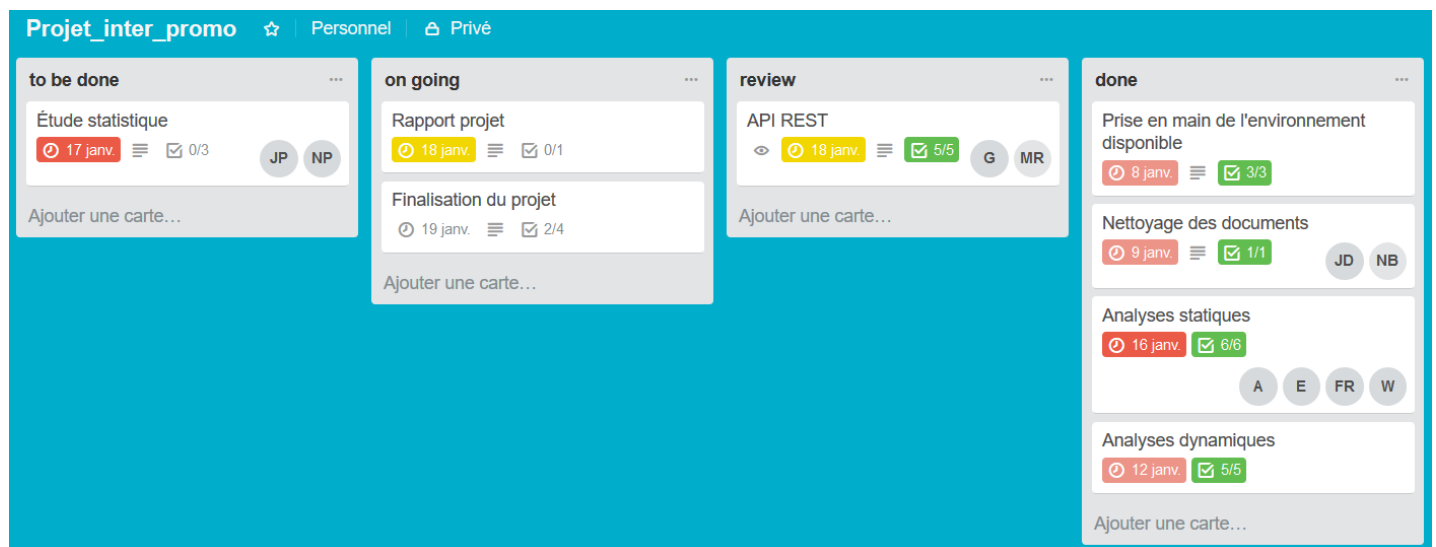


Figure 4 : Trello

Pour chaque fonctionnalité, l'équipe estime les points de complexité relatifs à celle-ci. Une fois la user storie réalisée par un ou plusieurs membres du groupe (selon l'assignation), celle-ci passe en « review ». La revue de code est réalisée par un membre du groupe n'ayant pas participé aux développements de la fonctionnalité concernée. Une fois le code validé, celui-ci passe au statut « Done ».

Une user storie est découpée en tâches. Voici un exemple de tâches associées à la fonctionnalité « Prise en main de l'environnement disponible » :

 **Prise en main de l'environnement disponible**

Dans la liste [done](#)

Échéance

☐ 8 janv. à 12:00 (échéance passée)

Description [Éditer](#)

Prise en main des outils, logiciels et environnement de test

☒ **Checklist** [Cacher les éléments complétés](#) [Supprimer...](#)

100%

☒ *Serveur*

☒ *GIT*

☒ *Python*

Ajouter un élément...

Figure 5 : exemple de tâche

2.2.3. Gestion des versions et plateforme de collaboration

Afin de garder un historique des anciennes versions et de pouvoir effectuer un suivi des changements, nous avons utilisé GitHub, un logiciel collaboratif de dépôt Git.

Les logiciels de gestion de versions sont largement utilisés pour gérer un projet informatique, notamment lorsque celui-ci fait intervenir plusieurs collaborateurs. En effet, ces logiciels présentent deux avantages importants :

- L'historisation du code source
- La collaboration

Ils permettent de :

- Stocker un ensemble de fichiers en conservant les informations relatives à toutes les modifications effectuées
- Revenir en arrière lorsque des anomalies sont constatées
- Travailler à plusieurs sur un même fichier, sans risquer d'effacer le travail des autres
- Faciliter l'interactivité entre les développeurs

Pour la communication inter et intra-groupe, nous avons utilisé Slack. Cet outil est une plateforme de communication collaborative ainsi qu'un logiciel de gestion de projet. Muni d'un chat, il facilite la communication intra-groupe et inter-groupes, d'autant plus que tous les échanges sont conservés. Il permet également le partage de fichier.

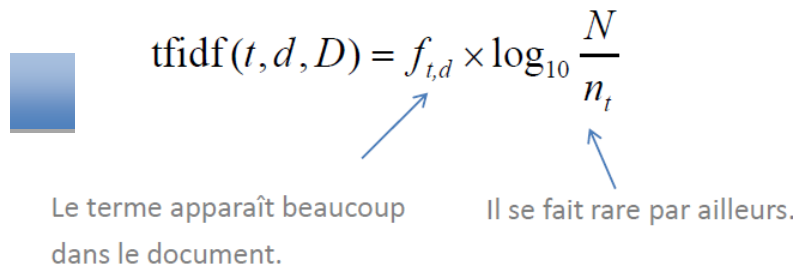
3. Déroulement des missions principales

Note : Toutes les études décrites dans ce rapport ont été réalisées sur des jeux de données fictifs. En effet, durant ce projet, nous n'avons pas eu accès aux données issues des articles.

3.1 Calcul des n mots les plus représentatifs d'une période

Le but de cette étude est de retourner au site web les n mots les plus importants d'une période (jour, mois, semaine). Dans le cadre de ce projet, seules les analyses par semaine ont été intégrées au site web.

Les données utilisées sont les TF-IDF calculés par article et par jour par mot.



$$\text{tfidf}(t, d, D) = f_{t,d} \times \log_{10} \frac{N}{n_t}$$

Le terme apparaît beaucoup dans le document. Il se fait rare par ailleurs.

Figure 6 : Formule TD-IDF

Voici un exemple de fichier JSON à fournir par le groupe BD QUERY :

```
{
  "period": "02/08/2017_09/08/2017",
  "Trump" : [[0.5, 0.6], [0.5, 0.8, 0.5], [], [0.5, 0.6], [0.5, 0.6], [0.5], [0.3, 0.8, 0.9]],
  "foot" : [[0.5, 0.6], [0.5, 0.8, 0.5], [], [0.5, 0.6], [0.5, 0.6], [0.5], [0.3, 0.8, 0.9]]
}
```

Figure 7 : Données d'entrées

Pour chaque mot (Trump et foot sur l'exemple), une liste de TF-IDF par jour et par mot est disponible. Chaque liste, disposée par ordre chronologique, contient les TF-IDF observés pour le mot observé par article par jour.

Les n mots retournés au site web sont ceux qui ont la moyenne des moyennes de TF-IDF la plus élevée sur la semaine courante.

Les données retournées au site web sont les 10 mots les plus représentatifs de la semaine étudiée ainsi que la moyenne de moyennes obtenue pour qu'il puisse réaliser un nuage de mots.

La fonction réalisée pour faire ce calcul est générique et peut être utilisée sur plusieurs problématiques (les n mots les plus représentatifs par thème, source, etc.). Une extension de cette fonction a également été réalisée pour retourner les n mots les plus représentatifs par type (adjectif, verbe, etc.). Les n mots retournés par cette fonction ainsi que leur liste de TF-IDF agrégée par jour (somme des TF-IDF), servent d'entrées aux fonctions d'analyses de tendance.

3.2 Analyse de tendance par jour

Les données disponibles pour réaliser cette étape sont les valeurs des TF-IDF observées pour le jour en cours et le jour précédent.

Pour détecter une tendance, un test de comparaison de moyenne (T de Student) a été utilisé. Ce test nous procure un résultat relatif aux hypothèses suivantes :

- $H_0 : \mu_{j-1} = \mu_j$
- $H_1 : \mu_{j-1} \neq \mu_j$

Où μ_{j-1} et μ_j correspondent respectivement à la moyenne observée pour la journée précédente et la journée d'intérêt.

Les critères de détection de tendance mis en place sont les suivants :

- Si la p-value est comprise entre 0 et 0.001 la tendance est qualifiée de très significative, si la statistique de test est positive la tendance est fortement en hausse sinon fortement en baisse.
- Si le p-value est supérieure à 0.05, aucune tendance n'est détectée.

Voici un exemple de données d'entrées et de résultats obtenus via cette méthode :

- Données d'entrées :

```
{ "period": "02/08/2017",
  "Macron" : [[0.6,0.5,0.7],[0.9,0.9,0.9]],
  "Tempete" : [[0.1,0.2,0.3],[0.25,0.8,0.9]],
  "Enfant" : [[0.5,0.6,0.9],[0.1,0.1,0.2]],
  "Fleur" : [[0.9,0.6],[0.5]],
  "Jardin" : [[0.3,0.6],[0.7,0.7]],
  "Jouet" : [[0.1,0.1,0.2],[0.5,0.9,0.9]],
  "Jeux" : [[0.6],[0.15]],
  "Magazine" : [[0.3,0.65],[0.1,0.5]],
  "Noel" : [[0.5,0.6],[0.9,0.9,0.9,0.9]],
  "Jour" : [[0.5,0.1],[0.9,0.9]],
  "Test" : [[0.5,0.6],[0.5]]
}
```

Figure 8 : TF-IDF pour 2 jours

- Résultats obtenus :

```
{ 'Enfant': 'Decreasing_trend',
  'Fleur': 'No_trend',
  'Jardin': 'No_trend',
  'Jeux': 'No_trend',
  'Jouet': 'Increasing_trend',
  'Jour': 'No_trend',
  'Macron': 'Increasing_trend',
  'Magazine': 'No_trend',
  'Noel': 'Strongly_increasing_trend',
  'Tempete': 'No_trend',
  'Test': 'No_trend',
  'period': '02/08/2017' }
```

Figure 9 : Résultat tendance par jour

Ci-dessus un exemple des résultats obtenus en exécutant un test d'égalité des moyennes pour une journée (on compare ici la moyenne du jour avec la moyenne de la veille). Nous pouvons voir que cette méthode, qui bien que très simple, semble fournir des résultats cohérents sur notre jeu de test.

Cette méthode n'a pas été approfondie car la priorité des analyses de tendances concernent les semaines et les mois, via des séries chronologiques.

3.3 Analyse de tendance par période via des séries chronologiques

Note : Les analyses décrites dans cette partie fonctionnent également par mois, année... La seule condition est la présence des données de la période précédente.

Voici un exemple de données utilisées pour cette partie, qui représentent la somme des TF-IDF observée par jour :

```
data_mult = {"test": [0.5, 0.3, 0.2, 0.1, 0, 0.1, 0.2, 0.7, 0.9, 0.8, 0.8, 0.9, 1, 0.6]}
```

Figure 10 : Exemple jeu de test semaine

Dans cette partie, plusieurs méthodes, ont été implémentées :

3.3.1 Analyse de la série par décomposition :

Pour cette méthode, nous avons utilisé la fonction python `seasonal_decompose` qui permet d'obtenir pour chaque série chronologique, selon une période spécifiée (7 pour la semaine), les valeurs de la tendance par filtre de convolution et les résidus associés. Le pré-requis pour utiliser cette fonction est d'indiquer si le modèle est additif ou multiplicatif.

Pour identifier le type de modèle, nous nous sommes intéressés à l'auto-corrélation des résidus de la manière suivante :

- Si la somme des carrés des auto-corrélations du modèle additif est supérieure à celle du modèle multiplicatif, alors celui-ci est additif. Dans le cas inverse, il est multiplicatif.

Une fois le type de modèle identifié, nous avons récupéré les valeurs de tendance obtenues par filtre de convolution pour chaque semaine et appliqué le test de comparaison de moyennes expliqué précédemment.

Les résultats obtenus sur des séries chronologiques de tests qui, bien que satisfaisants, ne correspondent pas à nos attentes. En effet, cette méthode repose sur l'identification du type de modèle à utiliser. Or, dans le cas d'une valeur nulle dans la série, il est impossible d'utiliser cette fonction (impossible de détecter un modèle multiplicatif). Nous avons donc choisi de tester d'autres méthodes, présentées ci-dessous.

3.3.2 Analyse de la série via une droite de régression :

Ce programme se compose d'une fonction nommée `linear_regression` permettant d'appliquer une régression linéaire sur les données issues d'un fichier JSON ; cette fonction retourne un dictionnaire contenant, pour chaque mot, la valeur du coefficient directeur de la droite de régression, ainsi que la valeur de l'intercept. Elle nécessite pour cela l'utilisation des packages `statsmodels.api` et `numpy` ; le premier permettant de réaliser la régression linéaire (commande `ols`).

Afin d'illustrer le principe de cette fonction, nous nous appuyons sur les données suivantes :

```
In [56]: data_v2
Out[56]:
{'airport': [8, 8, 7, 6, 6, 4, 5, 3, 3, 2, 1, 1, 0, 0],
 'avalanche': [6, 6, 7, 6, 6, 6, 5, 6, 6, 7, 6, 6, 6, 6],
 'lactalis': [5, 3, 3, 2, 2, 1, 2, 2, 3, 3, 4, 5, 5, 6],
 'parcoursup': [0, 1, 1, 3, 3, 4, 5, 7, 7, 8, 10, 9, 11, 12]}
```

Figure 11 : Données utilisées pour la régression

Initialement, cette fonction faisait apparaître la représentation graphique (*line plot*) ; ainsi que la droite de régression linéaire. Cette visualisation ne figurera pas sur

le site web ; l'exemple ci-dessous est associé au mot « *parcoursup* » (issu du jeu de données test).

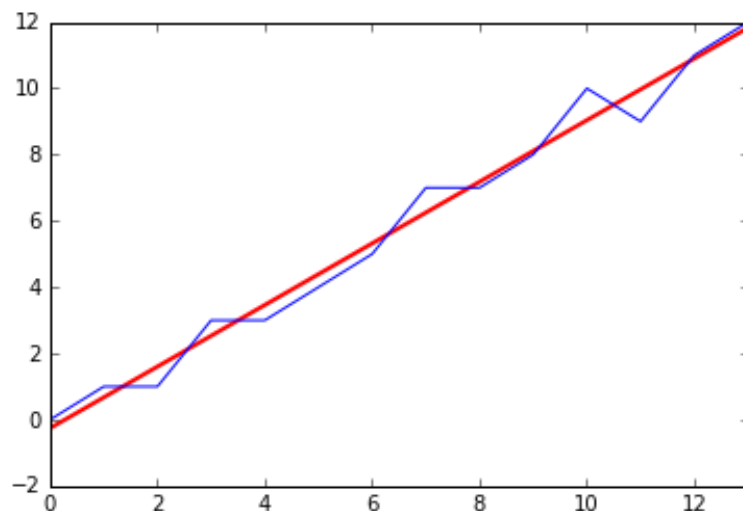


Figure 12 : Exemple de sortie graphique

Elle affiche également la valeur du r^2 ; qui témoigne de la qualité de la régression linéaire ; plus elle est proche de 1, plus cela signifie que la courbe bleue se rapproche de la droite rouge (ie. la droite de régression « colle » aux données).

```
word :
parcoursup
r2 :
0.981373920197
```

Figure 13 : R^2 obtenu

Cette fonction stocke les valeurs des coefficients directeurs associées à chacune des droites au sein d'une liste, ainsi que les valeurs de l'*intercept* ; dans la version finale, ces informations sont référencées au sein d'un dictionnaire, retournées à l'issue l'exécution de la fonction *linear_regression*.

```
In [59]: lin_reg
Out[59]:
{'airport': [-0.67252747252747258, 8.2285714285714313],
 'avalanche': [-0.0065934065934062508, 6.1142857142857148],
 'lactalis': [0.17142857142857165, 2.1714285714285708],
 'parcoursup': [0.92967032967033014, -0.25714285714285778]}
```

Figure 14 : Sortie droite de régression

En conclusion, cette fonction utilise une méthode statistique élémentaire, très efficace lorsque les données suivent une même tendance (hausse ou baisse sensible) ; elle reste cependant peu robuste quand la série présente plusieurs

variations successives. En raison de la complexité d'utilisation de cette méthode pour des séries ne présentant pas de tendance évidente, celle-ci n'a pas été intégrée au serveur.

3.3.3 Analyse de la série par moyenne mobile :

Après avoir calculé le score, représenté par la somme des tf de chacun des mots, et déterminé la liste des dix mots les plus fréquents, nous étudierons leur tendance respective (hausse ou baisse) sur la période de temps considérée. A terme, le programme devra permettre de mener une étude relative aux séries chronologiques, et d'évaluer la saisonnalité de la série.

Ce programme lit le fichier *JSON* contenant les sommes des TF associées aux dix premiers mots, et commence par en extraire les mots (*keys*) et les sommes de TF (*values*).

```
In [30]: data_v2
Out[30]:
{'airport': [8, 8, 7, 6, 6, 4, 5, 3, 3, 2, 1, 1, 0, 0],
 'avalanche': [6, 6, 7, 6, 6, 6, 5, 6, 6, 7, 6, 6, 6, 6],
 'lactalis': [5, 3, 3, 2, 2, 1, 2, 2, 3, 3, 4, 5, 5, 6],
 'parcoursup': [0, 1, 1, 3, 3, 4, 5, 7, 7, 8, 10, 9, 11, 12]}
```

Figure 15 : Données moyenne mobile

Il calcule ensuite, pour chacun des mots, les moyennes mobiles d'ordre 3 ; la série de données (en paramètre d'entrée) contient, dans le cas d'une analyse sur la semaine, quatorze valeurs. A l'issue de la fonction, nous obtiendrons une liste contenant $14 - 3 + 1 = 12$ valeurs. En réalité, nous devrions choisir l'ordre en fonction de la période ; au vu des données, nous avons décidé de fixer la valeur de l'ordre à 3 (d'un point de vue pertinence). La pertinence a été définie de la manière suivante : calcul des moyennes mobiles d'ordre 2, 3 et 4 et test de comparaison de moyennes pour définir la tendance. Les moyennes mobiles d'ordre 3 ont été retenues, car moins sensibles pour les données ne comportant pas de tendance.

La moyenne mobile permet de « lisser » une série de valeurs exprimée en fonction du temps ; et ainsi d'éliminer les fluctuations les moins significatives. Cette liste de moyennes mobiles doit par la suite être divisée en deux parties égales, de taille 6 chacune, afin de mener un test de comparaison de moyennes, qui, en fonction de la *p*-valeur et du signe de la statistique de test, retourne la tendance :

hausse ou baisse (parfois significative), absence de tendance. En définitive, nous obtenons le résultat suivant :

```
In [27]: trend_with_moving_average(data_v2)
Out[27]:
{'airport': 'strongly_decreasing_trend',
 'avalanche': 'no_trend',
 'lactalis': 'no_trend',
 'parcoursup': 'strongly_increasing_trend'}
```

Figure 16 : Résultat moyenne mobile

3.3.4 Analyse de tendance par période des mots associés à un mot pertinent

Le but de cette étude est de retourner la tendance des mots les plus associés à ceux-ci (en global puis par type de mot, par exemple par verbe, adjectif). Les données relatives au type des mots sont issues de pos tagging réalisé par le groupe filtrage.

Cette étude a comme données d'entrées les listes de listes de TF-IDF pour les n mots les plus pertinents.

Pour retrouver les n mots les plus associés à un mot important, nous utilisons les bigrams :

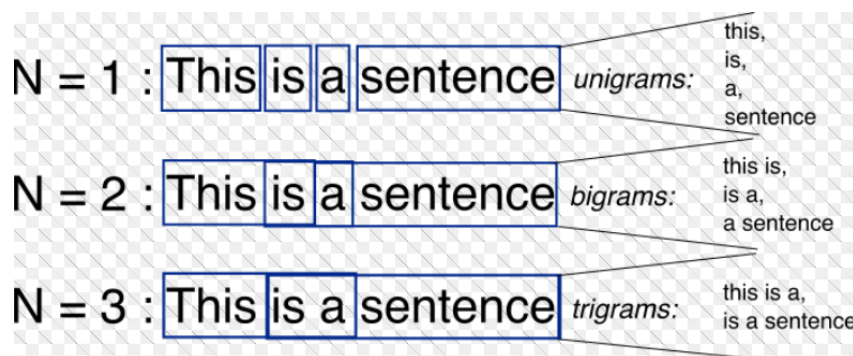


Figure 17 : Figure bigram

Voici le type de JSON attendu dans le cadre d'une analyse dynamique. Pour l'exemple, nous considérons que l'utilisateur souhaite obtenir la tendance des noms communs les plus associés au mot « Macron » selon une semaine définie :

```
{
  "Macron_president" : [[0.6],[0.3,0.4,0.5],[0.3,0.4,0.8,0.4]],
  "Macron_president_type" : ["COMMON_NOUN"],
  "Macron_polemique" : [[0.1],[0.1,0.4,0.5],[0.4,0.8,0.4]],
  "Macron_polemique_type" : ["COMMON_NOUN"],
  "Macron_France" : [[0.1],[0.1,0.1],[0.4,0.8,0.4]],
  "Macron_France_type" : ["PROPER_NOUN"]
}
```

Figure 18 : Exemple bigrams

La première étape consiste à retourner les n noms communs associés les plus pertinents ainsi que leurs poids ;

```
{'Macron_president': 0.50416666666666654, 'Macron_polemique': 0.37886904761904755}
```

Figure 19 : Exemple résultat

Une fois ces mots identifiés, nous effectuons une analyse de tendance sur les listes de valeurs TF-IDF disponibles pour ces bigrams. Cette analyse de tendance est effectuée via les moyennes mobiles et le test de comparaison de moyennes expliqué précédemment.

Cette méthodologie fonctionne également par source, par thème, par source et par thème. La période définie par l'utilisateur peut également être variable (1 semaine, 1 mois, etc). La seule condition à respecter est la suivante : les données d'entrées doivent obligatoirement contenir les données de la période précédente.

3.4 API REST

Afin de pouvoir répondre aux besoins du site web (analyse statique et/ou dynamique), une API REST a été développée. Cette API nous permet de communiquer avec l'API du groupe « BD QUERY » (fournisseur de données), via des requêtes HTTP.

Le schéma ci-dessous illustre la démarche globale (décrite précédemment) utilisée pour effectuer nos analyses :

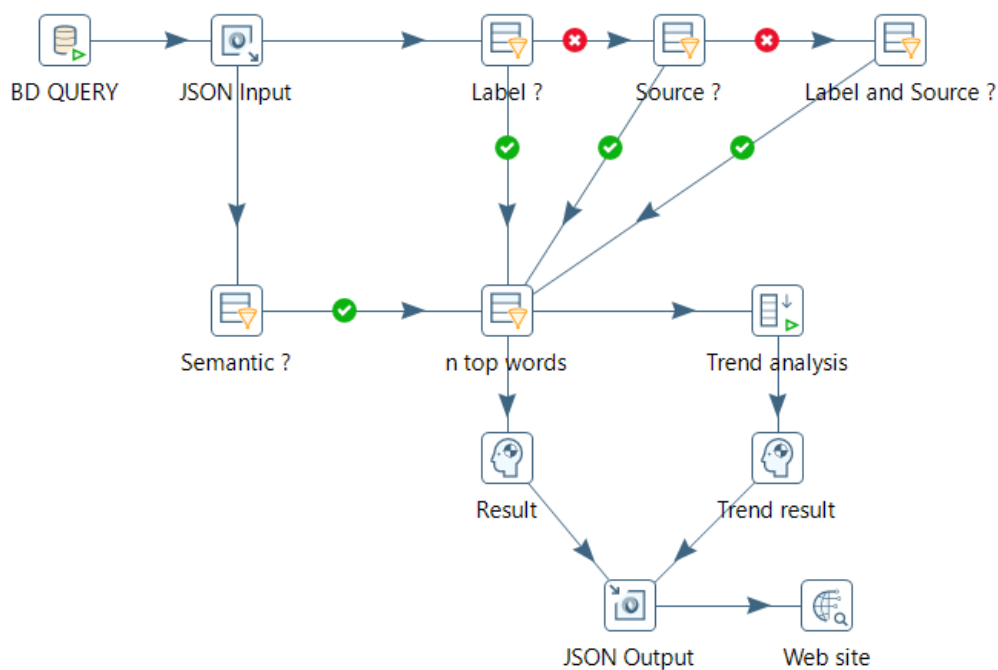


Figure 20 : Schéma méthodologie des fonctions

Cette démarche nous permet d'optimiser le temps d'exécution de nos traitements. En effet, les données du JSON contiendront des centaines voire des milliers de mots selon la période d'intérêt. Il est donc nécessaire d'effectuer nos calculs de tendance sur les n mots les plus représentatifs. En effet, cette étape est la plus coûteuse en termes de temps de calculs et d'exécution.

Le schéma suivant décrit les communications initialement retenues dans le cadre de ce projet :

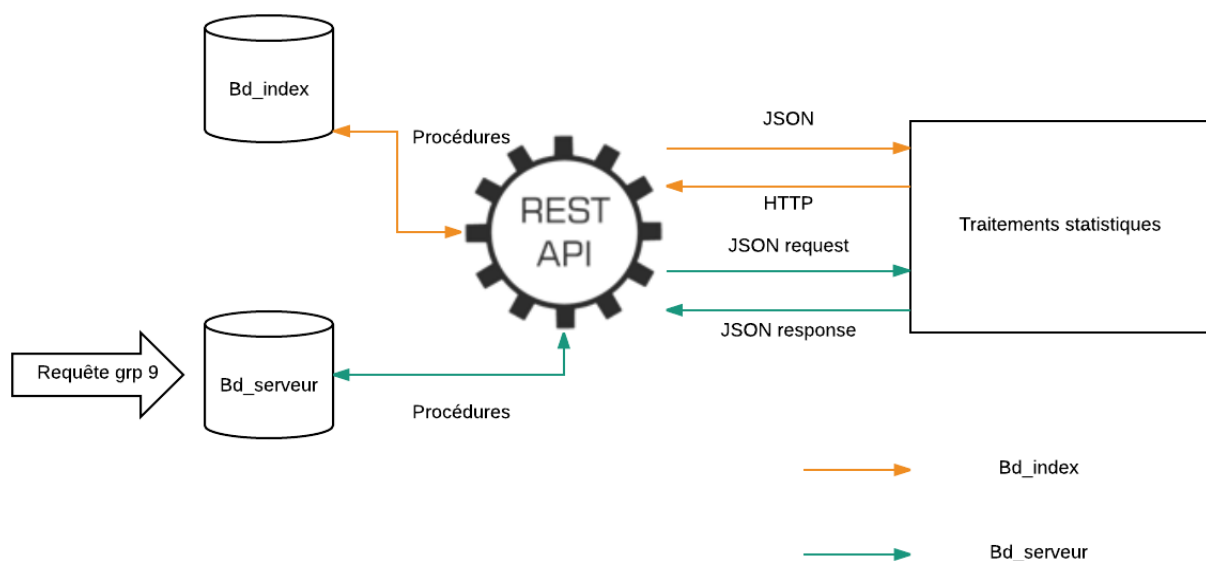


Figure 21 : Schéma API REST

Au terme de ces deux semaines de projet, seule la partie communication avec le groupe BD_serveur a été implémentée. Nous étions en revanche opérationnel pour fournir au groupe BD_index les données à stocker dans la base de données.

Selon les besoins du site web, seul les fonctionnalités suivantes ont été intégrées dans le serveur :

- Retour des 10 mots les plus pertinents de la semaine et leur tendance associée
- Retour des 10 mots les plus représentatifs de la semaine et de leur tendance, ainsi que le verbe, le nom propre et l'adjectif (avec les tendances) les plus représentatifs selon un thème choisi par l'utilisateur pour une semaine.

Pour répondre à ces besoins nous avons créés deux « routes » dans notre API et deux fonctions (une par route) réalisant les différents traitements nécessaires. Les routes permettent au groupe BD QUERY d'exécuter les traitements adéquats selon la fonctionnalité à exécuter.

Voici les deux fichiers JSON retournés au site web pour la deuxième fonctionnalité décrite page précédente :

- Top mot de la semaine pour un thème défini par l'utilisateur :

```
{0: {'text': '10', 'trend': 'Pas_de_tendance', 'weigh': 1.0},
1: {'text': '12', 'trend': 'Pas_de_tendance', 'weigh': 0.53259285714285709},
2: {'text': '13', 'trend': 'Pas_de_tendance', 'weigh': 0.37679047619047618},
3: {'text': '15',
'trend': 'Tendance_en_baisse',
'weigh': 0.29888928571428569},
4: {'text': '17',
'trend': 'Tendance_en_baisse',
'weigh': 0.25214857142857144},
5: {'text': '21', 'trend': 'Pas_de_tendance', 'weigh': 0.22098809523809523}}
```

Figure 22 : Résultat top mots pour un thème donné

- Tendance de l'adjectif, du verbe et du nom propre les plus représentatifs de la période :

```
{0: {'text': '10', 'trend': 'Pas_de_tendance', 'type': 'ADJ'},
1: {'text': '15', 'trend': 'Tendance_en_baisse', 'type': 'ADJ'},
2: {'text': '12', 'trend': 'Pas_de_tendance', 'type': 'ADJ'}}
```

Figure 23 : Top adjectifs

Note : Ces formats ont été définis par le site web. Le champ 'text' contient l'identifiant du mot dans la base de données.

4. Missions complémentaires

4.1 Intégration des concepts sémantiques dans les analyses de tendance

Le but de cette partie est d'affiner les analyses de tendance via les données fournies par le groupe sémantique. Les données d'intérêt sont la positivité et la polarité. Pour l'exemple, nous allons décrire la démarche mise en place pour intégrer la positivité dans les analyses de tendance (même principe pour la polarité).

Cette information nous permet entre autres de connaître la tendance d'un mot représentatif appartenant à des articles jugés « positifs » et « négatifs » sur deux semaines. Nous faisons donc l'hypothèse que le mot appartenant à un article positif est également positif (en fonction de son contexte).

La méthodologie utilisée est similaire à celle décrite dans la partie 3 :

- Récupération des n mots les plus représentatifs sur la période
- Analyse de tendance via les moyennes mobiles et le test de comparaison de moyennes

La différence est la suivante : nous retournons deux analyses de tendance par mot représentatif, à savoir zéro pour les articles positifs et un pour les négatifs.

Voici un exemple de JSON nécessaire pour faire cette analyse :

```
data={"0":{"Trump": [0.2,0.5,0.3,0.1,0.7,0.9,0.8,0.9,0.2,0.9,0.1,0.9,0.3,0.9],
  "Macron": [0.7,0.9,0.8,0.9,0.2,0.5,0.3,0.1,0.2,0.9,0.5,0.9,0.3,0.9]
},
"1":{"Trump": [0.5,0.1,0.2,0.9,0.2,0.5,0.3,0.1,0.8,0.1,0.9,0.8,0.9,0.9]
  "Macron": [0.9,0.9,0.7,0.9,0.2,0.5,0.3,0.1,0.1,0.1,0.1,0.1,0.1,0.1]
}}
```

Figure 24 : Exemple données positivités

La clé « 0 » correspond aux articles positifs et « 1 » pour les articles négatifs. Les valeurs correspondent aux TF-IDF des mots appartenant à ces articles.

Voici un exemple de résultat obtenu :

```
{'0': {'Macron': 'Pas_de_tendance', 'Trump': 'Pas_de_tendance'},  
'1': {'Macron': 'Tendance_fortement_en_baisse',  
      'Trump': 'Tendance_en_hausse'}}
```

Figure 25 : Résultat concept

4.2 Prédiction de la tendance de la période suivante

Cette partie n'a pas été intégrée dans les livrables en raison de la difficulté d'évaluer la qualité et la pertinence de la méthode mise en place.

Le but de cette partie est donc de présenter la démarche réfléchie pour réaliser cette problématique.

L'idée est de prédire la tendance d'un mot pour la période suivante. Pour cela, nous récupérons toutes les données disponibles sur le mot pour avoir un jeu de test et d'apprentissage le plus complet possible. Afin de prendre en compte la temporalité dans le modèle, nous réalisons un « cercle temporel ». Par exemple, pour la journée nous réalisons un modulo 24.

Ensuite, nous apprenons un modèle (SVR, Random Forest,...) sur les données et estimons les paramètres optimaux du modèle (gridsearch en python) et l'évaluons par validation croisée. Pour estimer la tendance, nous estimons les n valeurs correspondantes à la période à prédire, recalculons les valeurs des moyennes mobiles sur la série contenant les valeurs initiales et les valeurs prédites. Ensuite, nous effectuons le test de comparaison de moyennes pour détecter une éventuelle tendance.

Cette proposition n'a pas été approfondie en raison de l'indisponibilité des données. Réaliser cette méthode sur des données fictives ne nous aurait pas permis d'avoir une information interprétable sur la qualité de la méthode, car elles sont trop différentes des données réelles.

5. Difficultés rencontrées et solutions apportées

La principale difficulté rencontrée tout au long de ce projet est l'indisponibilité totale de données. En effet, il est très compliqué de réaliser et d'évaluer la qualité des analyses sans donnée. Pour pallier ce problème, nous avons réalisé nos propres jeux de test et implémenté des méthodes relativement basiques. Il était en effet inutile de faire des analyses plus poussées sans connaître la pertinence de ces méthodes sur les données réelles.

Nous avons ainsi axé notre travail sur l'élaboration de fonctions les plus génériques possibles pour pouvoir intégrer facilement le maximum de fonctionnalités dans le logiciel. De plus, cela nous permettait d'être en mesure de répondre rapidement à un besoin spécifique dans le cas de données disponibles. Ces fonctions permettent également d'éviter la duplication de code dans le projet et ainsi de respecter des normes qualité de génie logiciel de base.

La seconde difficulté rencontrée est due à un manque de communication entre les groupes et à la dépendance de notre groupe avec les autres. Les différents travaux réalisés par notre groupe ont en effet été difficilement compris par les autres groupes. Cela a impliqué trois points :

- Changement régulier des travaux réalisés
- Remise en cause d'une grande partie du cahier des charges
- Gestion par le site web de seulement deux fonctionnalités réalisées par notre groupe

6. Conclusion

Malgré les diverses difficultés rencontrées, ce projet est de notre point de vue une expérience enrichissante. Celui-ci nous a permis de travailler en équipe et de travailler selon des normes qualité définies. De plus, pour certains, ce projet a permis de découvrir et de manipuler de nouveaux outils (GIT) et langage de programmation (PYTHON).

De plus, les divers problèmes relevés tout au long de ce rapport nous préparent également au monde du travail. En effet, les problèmes de communication dans une entreprise sont relativement courants.

Les points positifs sont surtout le gain d'autonomie et le travail par anticipation. Ceux-ci se sont installés pour les membres du groupe tout au long du projet. Ces qualités sont très appréciées et essentielles pour un ingénieur dans le monde du travail.