Anticiper le retard des vols

Contenu

[Synthèse 2](#_Toc502434178)

[Introduction 3](#_Toc502434179)

[Préparation des données 3](#_Toc502434180)

[Récupération des dataset 3](#_Toc502434181)

[Pré-traitement du dataset 3](#_Toc502434182)

[Exploration 4](#_Toc502434183)

[Préparations des Matrices 5](#_Toc502434184)

[La matrice Term Frequency 5](#_Toc502434185)

[La matrice Term Frequency-Inverse Document Frequency 5](#_Toc502434186)

[Réduction de dimensions – LSA 6](#_Toc502434187)

[Modèle Non Supervisé 6](#_Toc502434188)

[Latent Dirichlet Allocation 6](#_Toc502434189)

[Non-Negative Matrix Factorization 8](#_Toc502434190)

[Proposition de Tags par la méthode non supervisée 8](#_Toc502434191)

[Modèle Supervisé 8](#_Toc502434192)

[API 9](#_Toc502434193)

[Pistes d'évolutions 9](#_Toc502434194)

# Synthèse

#### Contexte :

Air Data, une nouvelle compagnie aérienne désire optimiser sa logistique et anticiper les retards possibles de sa flotte. A partir de données existantes sur d'autres compagnies aériennes, celle-ci nous demande de mettre en place un modèle de régression afin de prédire les possibles retards/avances.

#### Problème :

Ce problème est un problème de régressions. A l'aide de différents modèles, une évaluation de la prédiction sera faite. Par la suite, une optimisation sera faite sur les hyperparamètres des modèles ainsi qu'avec du Boosting. L'objectif majeur étant de prédire les courts retards fréquents.

#### Données :

Les données fournies sont des données issues d'une base publique gouvernementale (www.transtats.bts.gov). Ce site regroupe les données de chaque vol intérieur aux USA par mois sur les 30 dernières années. Dans notre étude, nous avons à notre disposition les données de l'année 2016.

#### Approche :

Après un nettoyage des données inutiles dans les différents datasets, différents modèles vont être testés sur différentes configurations des features. Les modèles seront ensuite évalués sur le MAE et MSE en cas d'égalité. Une recherche des meilleurs hyperparamètres sera faite à l'aide de Grid Search pour chacun d'entre eux.

#### Performances des modèles :

Lors de ce projet, un des problèmes majeurs a été la performance. Beaucoup de modèles ne passent pas en mémoire sur les Notebook et ont donc été découpés dans des scripts. Concernant l'évaluation, le MAE a été le critère principal. L'objectif étant de prévoir majoritairement les petits retards facilement anticipables plutôt que les gros retards potentiellement dus à des problèmes imprévisibles (sécurité, météo, panne, …).

#### Résultats :

Le 1er modèle souffre fortement du bruit et ne permet pas de prédire une tendance particulière. Quant au modèle 2, l'agrégation par heure du retard moyen permet de prédire tout de même une tendance globale correcte. Cependant les 2 modèles ne sont peut-être pas les plus pertinents pour ce type de prédiction car il y a une souffre d'underfitting. Des pistes d'évolutions, sont donc présentées à la fin du rapport.

# Introduction

A partir d'une API du site Stack Overflow, l'objectif de ce projet est de mettre en place un modèle de prédiction de tags pour la question posée. Cela dans le but d'aider les nouveaux membres sur le site Stack Overflow à mieux classifier leurs questions et ainsi avoir des réponses plus pertinentes.

Dans un 1er temps, nous allons récupérer des datasets, explorer un peu leur contenu et faire un peu de nettoyage. Par la suite une approche non supervisée sera faite afin de trouver les sujets principaux de la question et ainsi essayer de prédire les tags censés.

Dans un second temps, une approche supervisée sera faite avec un fine tuning du meilleur modèle.

Pour finir, l'API vous sera présentée avec des critiques sur la prédiction. Des ouvertures à l'amélioration seront proposées

# Préparation des données

## Récupération des dataset

L'API de Stack Overflow nous permet via une requête SQL de récupérer diverses données publiques (sur les post, utilisateurs, tags, etc.). Dans notre cas, nous sommes interessé par le titre, le contenu de la question ainsi que les tags. Pour avoir un dataset d'entrainement et de test pour la phase supervisée, on va prendre les questions au hasard. Les 50 000 premiers sujets seront pour l'entrainement et les 50 000 suivant pour les tests. L'Id de la question est donc aussi téléchargé pour s'assurer qu'il n'y ait pas de doublons. Les requêtes sont donc :

|  |  |
| --- | --- |
| **SELECT** Id**,** Title**,** Tags**,** **Body**  **FROM** Posts  **WHERE** PostTypeId **=** 1  **AND** Score **>** 3  **ORDER** **BY** **RAND()**  OFFSET 0 **ROWS** **FETCH** **NEXT** 50000 **ROWS** **ONLY** | **SELECT** Id**,** Title**,** Tags**,** **Body**  **FROM** Posts  **WHERE** PostTypeId **=** 1  **AND** Score **>** 3  **ORDER** **BY** **RAND()**  OFFSET 50000 **ROWS** **FETCH** **NEXT** 50000 **ROWS** **ONLY** |

PostTypeId est mis à 1 pour n'avoir que les questions. Pour s'assurer de la qualité du dataset, seul les questions avec un score supérieur à 3 est pris. Cela permet de s'assurer que l'auteur a fait des efforts sur le contenu et les tags.

## Pré-traitement du dataset

Après avoir vérifié qu'il n'y a pas d'ID en doublons dans les 2 datasets. Une exploration des features a été fait.

#### Titre

Concernant le titre, il n'y a pas besoin de traitements particuliers. Il a juste été fusionné avec le body pour avec un corpus constitué que d'une seul feature

#### Tags

Pour les tags, ils sont comme des balises HTML (<tag>). Une regexp a permis de convertir cette feature en une liste de liste. Cette liste a ensuite été stockée avec pickle. Une seconde matrice est aussi génératé pour la partie supervisée avec un MultiLabelBinarizer (similaire au One Hot Encoder mais avec 1 pour chauqe tags).

#### Body

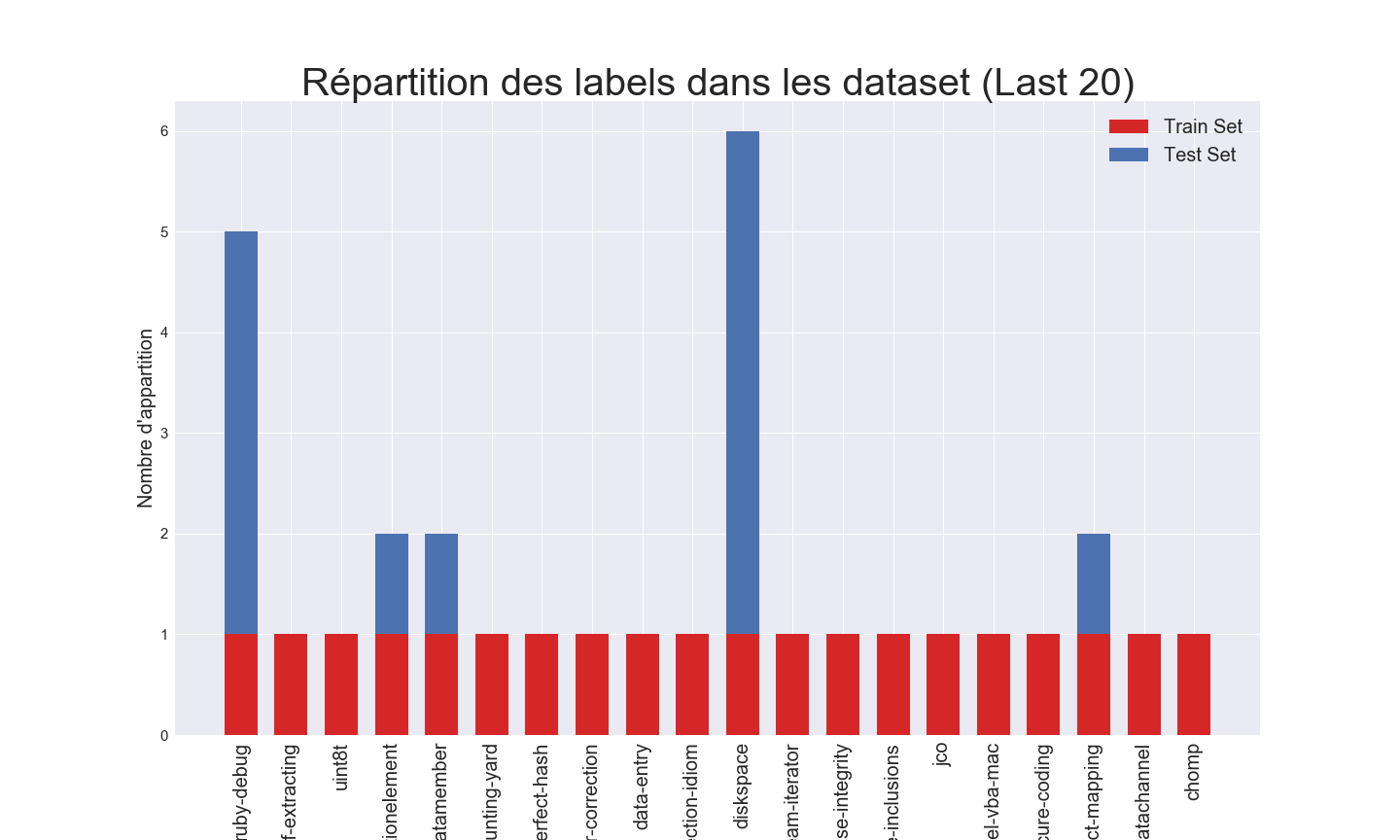
Pour le body, il y a plus plus de travail. Le texte entre les balises code on été supprimés, les balises aussi. Cela a permis d'avoir un corpus avec moins de sens mais ne conservant que du "vrai texte". Les balises ne sont là que pour le rendu sur le site. Concernant le code, j'ai pris la décision de la supprimer car la majorité des mots sont uniques car ce sont des variables. On perd certes de l'information sur le langage avec des mots clé. Cependant beaucoup de ces mots clés sont les même dans tous les langages (for, while, break, try, return, …).

Un compte des mots courant a été fait ensuite et le top 100 a été rajouté aux StopWords (English) de nltk plus spécifiques aux Sujets de Stack Overflow.

# Exploration

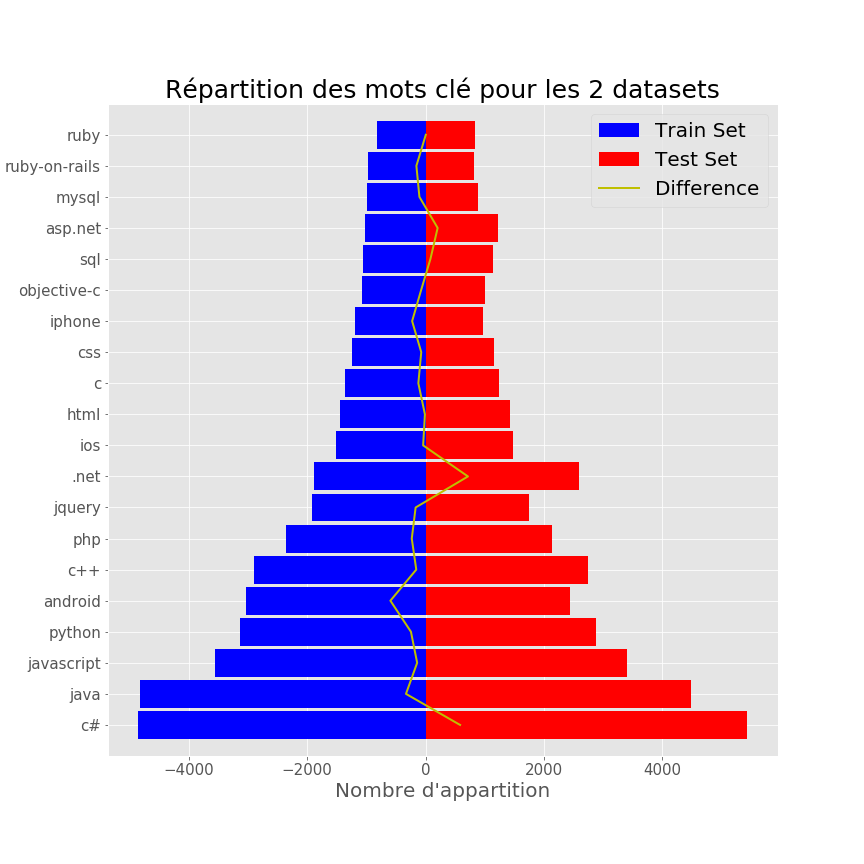
Après la phase de nettoyage, une exploration des tags a été fait. Sur 50 000 posts, plusieurs milliers de tags ont été utilisés. Afin de pouvoir faire un classifieur par la suite, j'ai gardé que les tags avec plus de 25 apparitions dans les trains set. Tous les autres tags ont été supprimé.

En effet si on regarde les moins courants, ils n'ont qu'une apparition dans le train set et entre 0 et 3 apparitions dans le test set :



Au final on conserve 773 tags. Par la suite, j'ai supprimé tous les posts n'ayant plus de tags. Cela représente que très peu de posts (< 1000 sur le train set et environ 3000 sur le test set).

Afin de s'assurer de la balance des 2 datasets, on peut regarder la différence entre chaque avec une pyramide



On a quelques différences entre les 2 datasets mais hormis avec .net, le dataset est plutôt balance sur les 20 principaux tags.

## Préparations des Matrices

### La matrice Term Frequency

Le titre et la question étant groupés, on a maintenant un seul corpus de presque 50000 posts. Celui-ci été utilisés pour générer la matrice de Term-Frequency. Cette matrice est très sparse avec seulement 1.72 millions d'entiers stockés dans une matrice de 48357 x 91349 (soit un remplissage de 1 élément pour 4000). Une fois générée, elle a été sauvegardée ainsi que le modèle pour l'API. Comparé à la matrice TF-IDF que l'on verra par la suite, aucun lemmatisation a été mise en place car la différence en terme de dimension est peu importante.

### La matrice Term Frequency-Inverse Document Frequency

De la même manière, une matrice TF-IDF a été générée. Celle-ci a bien moins de dimensions car tous les mots ayant des petits scores ont été supprimés. De ce fait, la matrice finale est de 48357 x 2764 remplis avec 1.55millions de float. Cette matrice est donc aussi moins creuse avec 1 élément sur 86. Celle-ci aussi a été sauvegardée ainsi que le modèle.

### Réduction de dimensions – LSA

Pour pouvoir tester l'entrainement du modèle supervisé aussi sur la matrice TF (c'est actuellement impossible avec autant de dimensions car il y aura le fléau de la dimension). J'ai voulu tester le Latent Semantic Analysis qui, comme le PCA, permet de réduire les dimensions en ne gardant que les mots participant le plus à la variance. Le résultat est donc similaire au TFIDF car les mots trop fréquents ne participent pas à la variance et seront donc supprimés.

Malheureusement, le résultat est une matrice très lourde (1.3go) et un modèle aussi très important (2go). De plus l'entrainement est très lent (plusieurs minutes). Il devient donc peut interessants car il faut aussi penser au fait qu'il doit servir en production.

On a donc toutes les matrices nécessaires aux différents modèles.

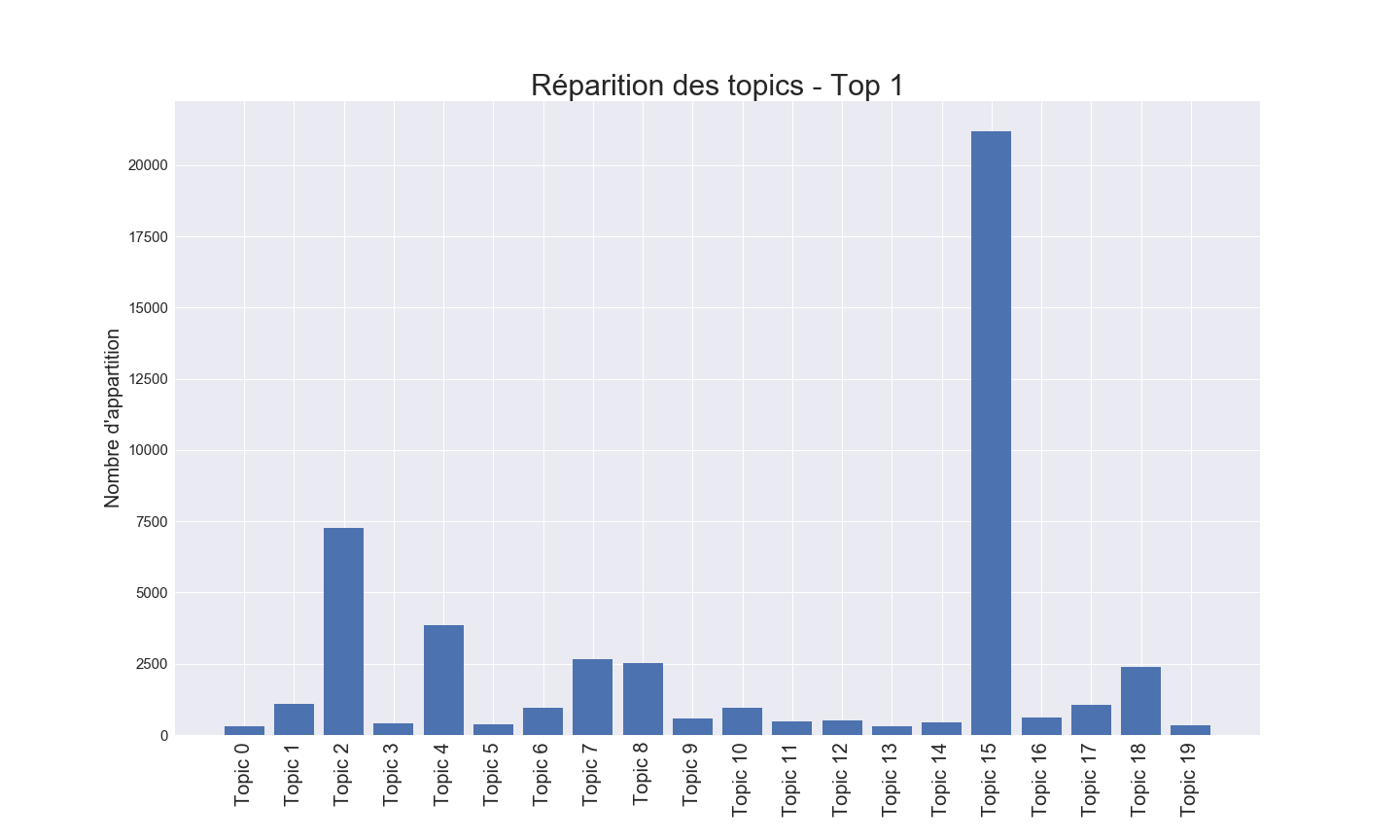
# Modèle Non Supervisé

### Latent Dirichlet Allocation

Basé sur la matrice TF, le LDA a été entrainé avec plusieurs taille de choix de Topics. Si l'on met trop de topics, beaucoup sont identiques et ont les mêmes principaux mots. Si on choisit 20 topics, on peut voir ci-dessous les mots clé et ainsi faire un e analyse du sujet:

|  |  |  |
| --- | --- | --- |
| Topics | Mots clés | Analyse |
| 1 | javascript event js events node tag component tags form control | Gestion de formulaires HTML |
| 2 | string value number java memory values list variable two array | Types de données |
| 4 | text jquery element css html json button change set click | Mise en page site (CSS, js, forms) |
| 7 | project files android build version directory folder git studio eclipse | Gestion de projets/applications |
| 8 | table database sql query key mysql field column array id | Base de données |
| 10 | image images android size map points video draw plot matlab | Graphiques/images |
| 18 | page view net web http asp url controller request mvc | Fonctionnement site web |

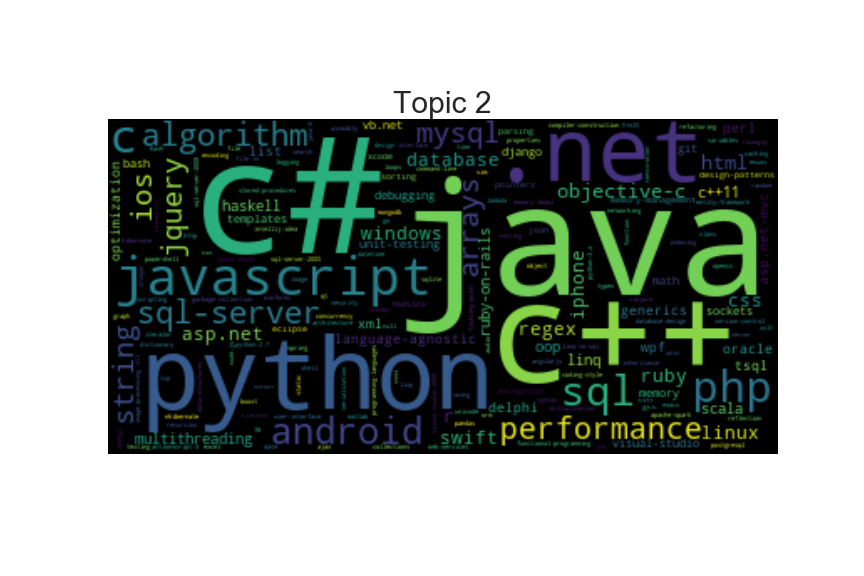
Un des problème de ce modèle sur ce type de données c'est que si l'on prendre le topic prépondérant du sujet, on remarque qu'il sont majoritairement dans un seul topic (15) car celui-ci regroupe des mots très courants (tests, run, try, server, …)



Peut-être que les StopWords étendu lors de la préparation n'ont pas suffi.

#### Analyse des tags par Topics

Basé sur le top 3 des topics de chaque sujet, les mots clé ont été compté. Si on fait un nuage de Mots sur ceux-ci on trouve :



On a majoritairement les languages en tête. Cela s'explique car ils sont en tête des tags. Si on veut des mots clé un peu moins courant, on peut diviser leur nombre d'apparition par le nombre d'apparition dans le corpus et on trouve :





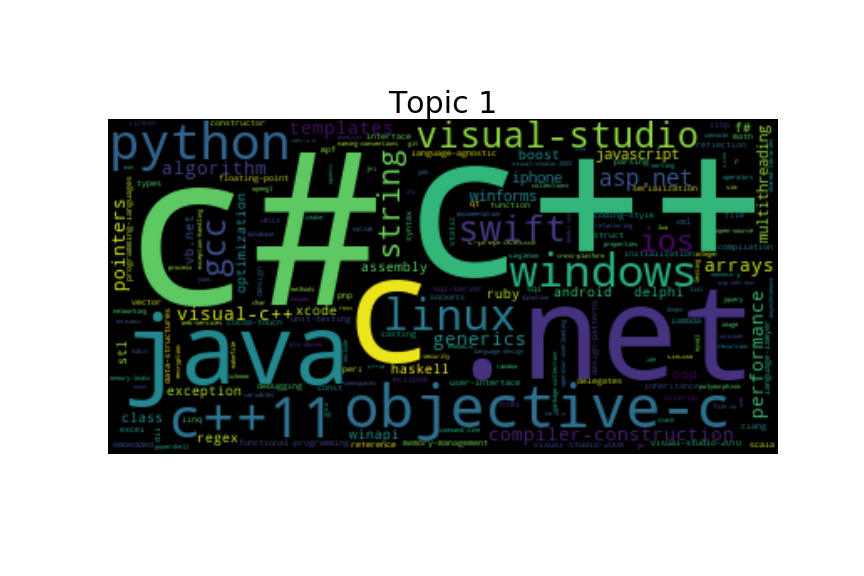
On a donc maintenant des tags plus parlant si on le compare aux topics présenté précédemment.

### Non-Negative Matrix Factorization

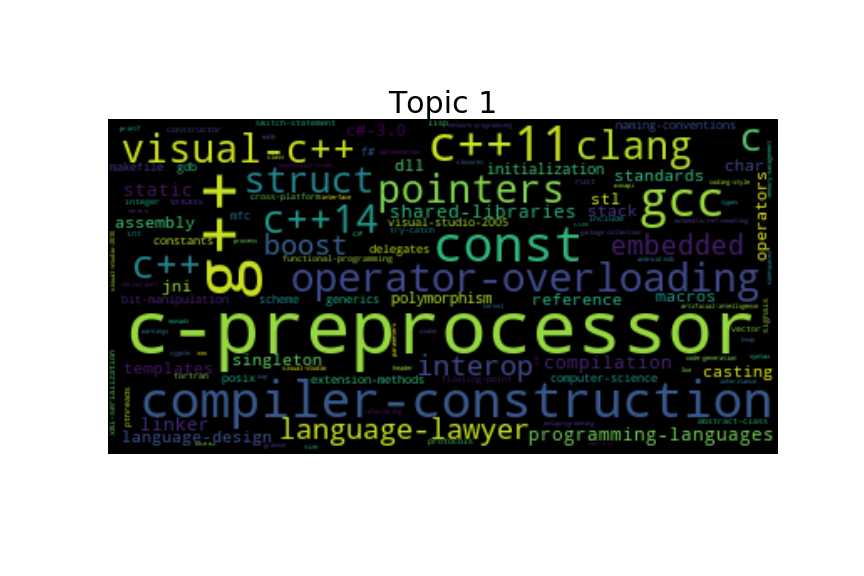
Basé sur la matrice TF-IDF, la même analyse a été faite avec le NMF. Au niveau des topics on trouve :

|  |  |  |
| --- | --- | --- |
| Topics | Mots clés (lemmatisés) | Analyse |
| 1 | c compil program librari languag pointer b gcc declar dll | Compilation, librairie et C |
| 2 | tabl column queri row sql databas mysql select index field | Base de Données |
| 3 | server sql connect client databas servic web request send http | Requete Server |
| 10 | array element byte loop index numpi sort pointer size number | Type de données et Structures |
| 13 | php script mysql variabl 5 upload session page ini email | PHP |
| 15 | page jqueri html element javascript button click text event div | Mise en page web |
| 18 | valu return variabl key set null properti default field type | Type de données |

Si on regarde du coté des tags sans normalisation on trouve :



Et avec normalistion on a des tags moins courants mais possiblement hors sujet.



### Proposition de Tags par la méthode non supervisée

# Modèle Supervisé

# API

# Pistes d'évolutions