Catégorisez automatiquement des questions

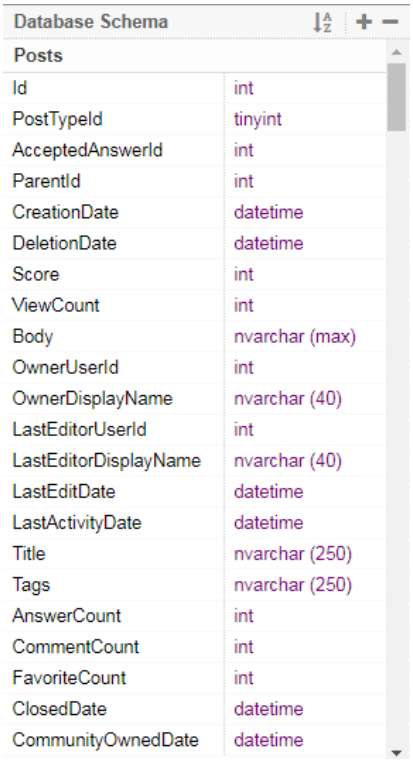
Table des matières

1. Introduction

Poser une question sur Stack Overflow est très simple, plus dur est de trouver les bons tags permettant de catégoriser votre question. Le but de ce projet est de construire un système de suggestion de tags utilisant en entrée le titre et le corps de la question que vous posez. L'idée est donc de récupérer dans un premier temps des données réelles, disponibles sur Stack Exchange, de les analyser, de construire un modèle permettant de générer des tags, et enfin de les inclure dans une IHM dédiée que l'on mettra à disposition.

2. Visualisation des données

De nombreux champs sont disponibles



Database Schema	
Posts	
Id	int
PostTypeId	tinyint
AcceptedAnswerId	int
ParentId	int
CreationDate	datetime
DeletionDate	datetime
Score	int
ViewCount	int
Body	nvarchar (max)
OwnerUserId	int
OwnerDisplayName	nvarchar (40)
LastEditorUserId	int
LastEditorDisplayName	nvarchar (40)
LastEditDate	datetime
LastActivityDate	datetime
Title	nvarchar (250)
Tags	nvarchar (250)
AnswerCount	int
CommentCount	int
FavoriteCount	int
ClosedDate	datetime
CommunityOwnedDate	datetime

	Id	Body	Title	Tags
0	4	<p>I want to use a track-bar to change a form'...	While applying opacity to a form, should we us...	<c#><winforms><type-conversion><decimal><opacity>
1	6	<p>I have an absolutely positioned <code>div</...	Percentage width child element in absolutely p...	<html><css><css3><internet-explorer-7>
2	9	<p>Given a <code>DateTime</code> representing ...	How do I calculate someone's age in C#?	<c#><.net><datetime>
3	11	<p>Given a specific <code>DateTime</code> valu...	Calculate relative time in C#	<c#><datetime><time><datediff><relative-time-s...
4	13	<p>Is there any standard way for a Web Server ...	Determine a User's Timezone	<javascript><html><browser><timezone><timezone...

3. Nettoyage

3.1. Title & Body

Regardons un exemple

```
print(dataraw.Title[0])
print(dataraw.Body[0])
```

```
While applying opacity to a form, should we use a decimal or a double value
?
<p>I want to use a track-bar to change a form's opacity.</p>
<p>This is my code:</p>
<pre><code>decimal trans = trackBar1.Value / 5000;
this.Opacity = trans;
</code></pre>
<p>When I build the application, it gives the following error:</p>
<blockquote>
  <p>Cannot implicitly convert type <code>'decimal'</code> to <code>'double
'</code>.</p>
</blockquote>
<p>I tried using <code>trans</code> and <code>double</code> but then the co
ntrol doesn't work. This code worked fine in a past VB.NET project.</p>
```

D'après l'exemple ci-dessus on voit qu'il va falloir :

-enlever les balises HTML

-enlever les caractères spéciaux, mais pas tous car certains sont liés à une information imp ortante comme par exemple le # de c#

-convertir le texte en minuscules

-enlever les stopwords

-extraire la racine des mots

```
appli opac form use decim doubl valu want use track bar chang form opac
code decim tran trackbar1 valu 5000 opac tran build applic give follow e
rror cannot implicitli convert type decim doubl tri use tran doubl contr
ol work code work fine past vb net project
```

3.2. Tags

Dans un premier temps on peut :

-enlever les balises ‘<’ et ‘>’

-Garder les caractères spéciaux car ils font partie des tags

-convertir le texte en minuscules

Pour les stop words je voudrais vérifier l’intérêt, j’affiche donc la liste de ceux qu’on trouve dans les tags :

```
['against',
 'any',
 'between',
 'can',
 'd',
 'each',
 'having',
 'll',
 'm',
 'out',
 'this',
 'was',
 'where']
```

```
<perl><hash><iteration><each>
<php><sql><where>
<d>
<d><popularity>
<editor><d>
<input><d><tango>
<java><constructor><methods><call><this>
<serial-port><microcontroller><can><can-bus>
<windows><installation><d>
<java><c++><c><d>
<linux><d><powerpc><tango>
<sql><sql-server-2005><foreign-keys><between>
<arrays><d>
<jquery><radio-button><this>
<d>
<mysql><full-text-search><against>
<perl><hash><each><while-loop>
<c#><generics><attributes><constraints><where>
<c#><.net><casting><ref><out>
```

On peut donc enlever les stopwords, mais attention il y a des occurrences qui se retrouverons s ans tag et qu'on va devoir faire disparaitre de nos tests, on peut considérer que ce sont des outl iers, et comme le nombre et la nature des échantillons à notre disposition sont largement suffis ants il n'y a aucun souci à se débarrasser de quelques données pour entrainer notre modèle.

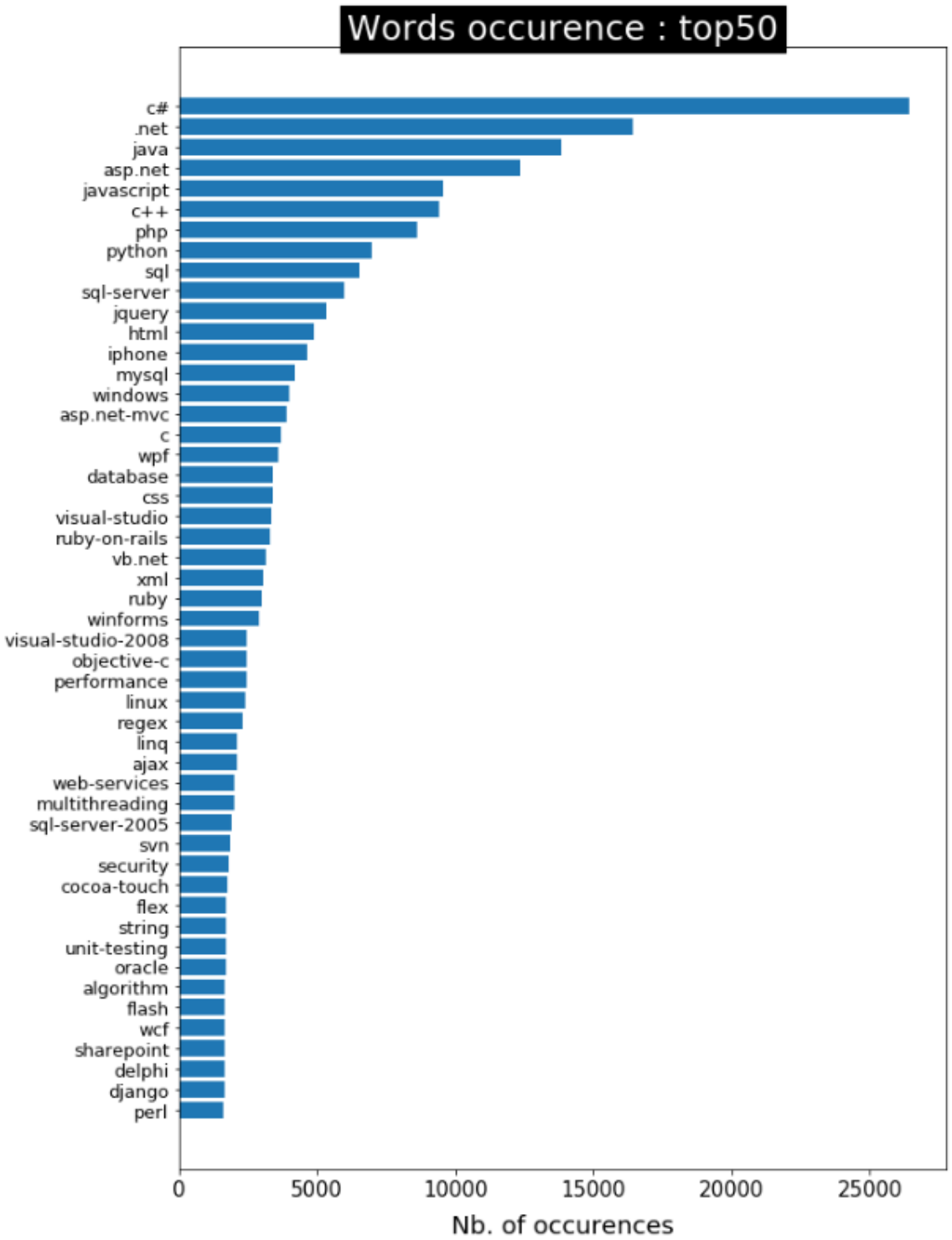
```
Il y a 15090 tags différents, voilà les 20 premiers classés alphabétiquement :

['.bash-profile',
 '.doc',
 '.emf',
 '.htaccess',
 '.htpasswd',
 '.net',
 '.net-1.0',
 '.net-1.1',
 '.net-2.0',
 '.net-3.0',
 '.net-3.5',
 '.net-4.0',
 '.net-assembly',
 '.net-attributes',
 '.net-client-profile',
 '.net-core',
 '.net-framework-source',
 '.net-framework-version',
 '.net-internals',
 '.net-micro-framework']
```

Cohérence entre les tags et la question posée (titre et corps)

En analysant la répartition des tags dans les questions on voit que pour environ 90% des cas au moins un des tags se retrouve dans la question, ce qui permet d'être optimiste quant à la possibilité qu'un algorithme obtienne de bons résultats sur ce jeu.

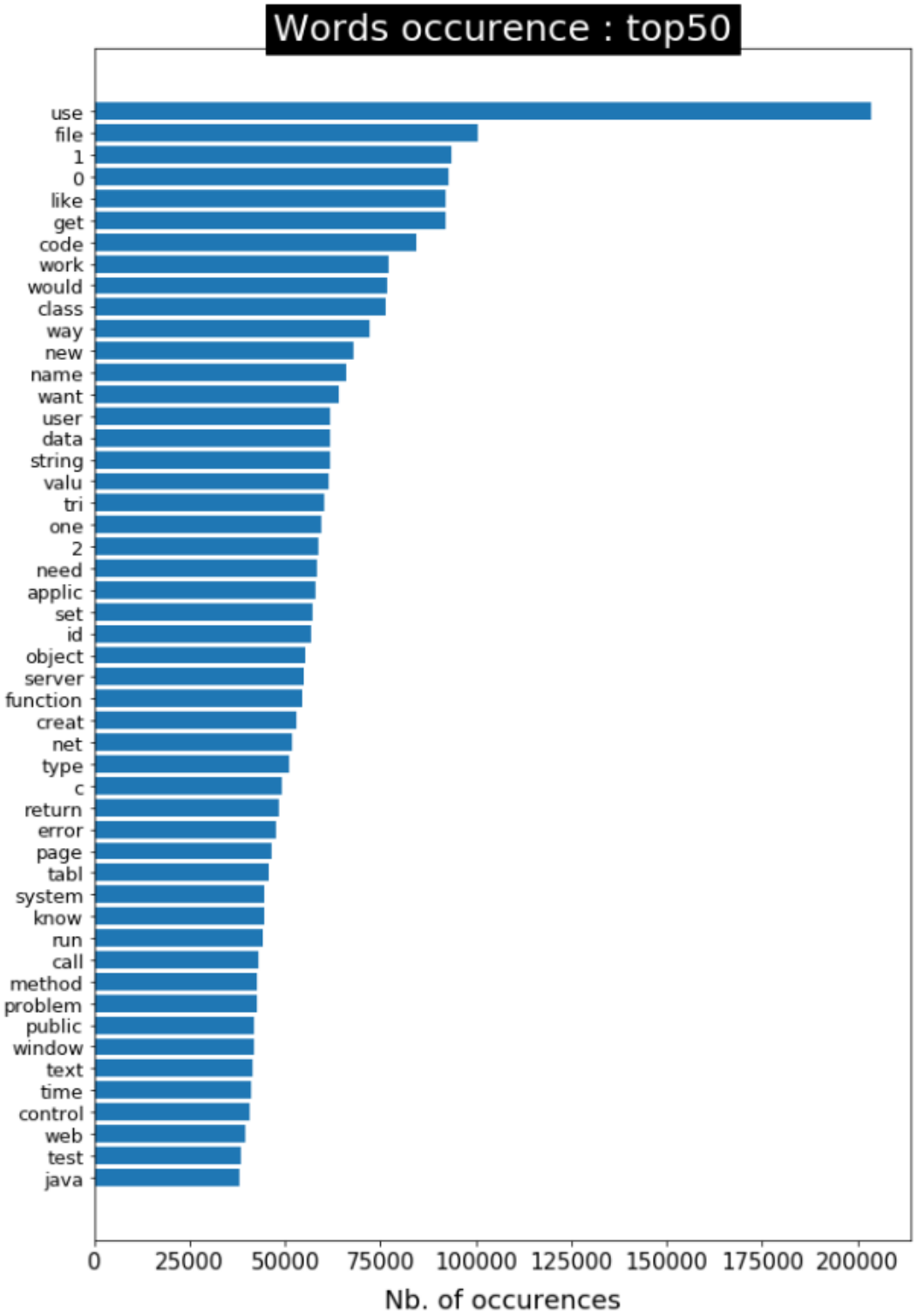
4. Word occurrence

4.1. Tags



Le point important maintenant est qu'il va falloir choisir un nombre de tags à garder pour notre modèle supervisé. En effet, on va faire de la classification multi labels, et la liste des labels possibles doit être connue à l'avance.

4.2. Title & Body



On voit que les mots les plus utilisés ne sont pas forcément pertinents, on pourrait faire un nettoyage à ce niveau ici mais le paramètre max_df de la vectorisation que l'on va utiliser par la suite devrait s'en charger.

5. Méthode non supervisée

5.1. LDA + TF

CountVectorizer

LatentDirichletAllocation

5.2. NMF + TF IDF

TfidfVectorizer

NMF

5.3. Comparaison

Nos deux modèles, LDA et NMF peuvent être comparés à travers certains paramètres communs entre les deux vectorizers :

P_min : 5, 10, 20

P_max : 0.5, 0.8, 0.9

Max_features :1000, 10000, 100000

Et pour NMF on peut aussi tester différentes fonctions de perte: frobenius et kullback-leibler

Topic=15, Top words=10 → LDA meilleur

```
MIN
Note=0.51 (77 matches sur 150 possibles - LDA max=0.8 - min=5 - feat=10000)
Note=0.47 (70 matches sur 150 possibles - NMF max=0.8 - min=5 - feat=10000)
Note=0.44 (66 matches sur 150 possibles - LDA max=0.8 - min=10 - feat=10000)
Note=0.48 (72 matches sur 150 possibles - NMF max=0.8 - min=10 - feat=10000)
Note=0.47 (70 matches sur 150 possibles - LDA max=0.8 - min=20 - feat=10000)
Note=0.48 (72 matches sur 150 possibles - NMF max=0.8 - min=20 - feat=10000)
MAX
Note=0.42 (63 matches sur 150 possibles - LDA max=0.5 - min=5 - feat=10000)
Note=0.49 (74 matches sur 150 possibles - NMF max=0.5 - min=5 - feat=10000)
Note=0.51 (77 matches sur 150 possibles - LDA max=0.8 - min=5 - feat=10000)
Note=0.47 (70 matches sur 150 possibles - NMF max=0.8 - min=5 - feat=10000)
Note=0.51 (77 matches sur 150 possibles - LDA max=0.9 - min=5 - feat=10000)
Note=0.47 (70 matches sur 150 possibles - NMF max=0.9 - min=5 - feat=10000)
FEATURES
Note=0.45 (68 matches sur 150 possibles - LDA max=0.8 - min=5 - feat=1000)
Note=0.45 (67 matches sur 150 possibles - NMF max=0.8 - min=5 - feat=1000)
Note=0.51 (77 matches sur 150 possibles - LDA max=0.8 - min=5 - feat=10000)
Note=0.47 (70 matches sur 150 possibles - NMF max=0.8 - min=5 - feat=10000)
Note=0.47 (71 matches sur 150 possibles - LDA max=0.8 - min=5 - feat=100000)
Note=0.50 (75 matches sur 150 possibles - NMF max=0.8 - min=5 - feat=100000)
NMF loss
Note=0.47 (70 matches sur 150 possibles - NMF max=0.8 - min=5 - feat=10000)
Note=0.39 (58 matches sur 150 possibles - NMF max=0.8 - min=5 - feat=10000)
done in 5288.812s.
```

Meilleur modèle : LDA, 0.8, 5, 10000

```
done in 736.365s.
Topic 0  (7 matchs) : file line use text string data print read output format
Topic 1  (3 matchs) : test date 10 key 00 2009 time unit 12 datetim
Topic 2  (5 matchs) : page html form asp control javascript text id function click
Topic 3  (5 matchs) : file window use run project applic instal work visual version
Topic 4  (4 matchs) : tabl sql databas queri data row column id select use
Topic 5  (6 matchs) : list user view item id model field key custom select
Topic 6  (2 matchs) : error messag log event tri session connect rubi assembl bar
Topic 7  (5 matchs) : imag width div style color height text bind li background
Topic 8  (5 matchs) : thread product report process program start time excel year day
Topic 9  (5 matchs) : java xml org eclips apach class com xsl jar hibern
Topic 10  (1 matchs) : use like way need code know applic look want net
Topic 11  (6 matchs) : number match express point doubl regex anim draw frame algorithm
Topic 12  (7 matchs) : librari load memori flash bit function json flex video world
Topic 13  (7 matchs) : class object string public return new int method function valu
Topic 14  (9 matchs) : server servic web http php com request client url use
Note=0.51 (77 matches sur 150 possibles - LDA max=0.8 - min=5 - feat=10000)

doc 0, topic 3, unabl start django server comput export path djang...
doc 1, topic 10, commerci use googl api question mayb better fit bu...
doc 2, topic 1, anoth linq pivot problem convert sql script linq q...
doc 3, topic 4, limit global memori resourc oracl 10g databas serv...
doc 4, topic 13, visual studio shortcut automat creat constructor i...
doc 5, topic 3, asp net search subdirectori search sub directori a...
doc 6, topic 1, get row start end time start end time anoth row sq...
doc 7, topic 10, free python debugg watchpoint pdb winpdb seem miss...
doc 8, topic 14, wcf servic activ directori authent write wcf servi...
doc 9, topic 7, authent activ directori use python ldap authent ad...
```

Topic=50, Top words=5 → NMF meilleur

```
MIN
Note=0.48 (119 matches sur 250 possibles - LDA max=0.8 - min=5 - feat=10000)
Note=0.52 (129 matches sur 250 possibles - NMF max=0.8 - min=5 - feat=10000)
Note=0.44 (109 matches sur 250 possibles - LDA max=0.8 - min=10 - feat=10000)
Note=0.47 (118 matches sur 250 possibles - NMF max=0.8 - min=10 - feat=10000)
Note=0.47 (118 matches sur 250 possibles - LDA max=0.8 - min=20 - feat=10000)
Note=0.49 (123 matches sur 250 possibles - NMF max=0.8 - min=20 - feat=10000)
MAX
Note=0.46 (114 matches sur 250 possibles - LDA max=0.5 - min=5 - feat=10000)
Note=0.52 (130 matches sur 250 possibles - NMF max=0.5 - min=5 - feat=10000)
Note=0.48 (119 matches sur 250 possibles - LDA max=0.8 - min=5 - feat=10000)
Note=0.52 (129 matches sur 250 possibles - NMF max=0.8 - min=5 - feat=10000)
Note=0.48 (119 matches sur 250 possibles - LDA max=0.9 - min=5 - feat=10000)
Note=0.52 (129 matches sur 250 possibles - NMF max=0.9 - min=5 - feat=10000)
FEATURES
Note=0.46 (115 matches sur 250 possibles - LDA max=0.8 - min=5 - feat=1000)
Note=0.49 (122 matches sur 250 possibles - NMF max=0.8 - min=5 - feat=1000)
Note=0.48 (119 matches sur 250 possibles - LDA max=0.8 - min=5 - feat=10000)
Note=0.52 (129 matches sur 250 possibles - NMF max=0.8 - min=5 - feat=10000)
Note=0.43 (107 matches sur 250 possibles - LDA max=0.8 - min=5 - feat=100000)
Note=0.48 (119 matches sur 250 possibles - NMF max=0.8 - min=5 - feat=100000)
NMF loss
Note=0.52 (129 matches sur 250 possibles - NMF max=0.8 - min=5 - feat=10000)
Note=0.36 (89 matches sur 250 possibles - NMF max=0.8 - min=5 - feat=10000)
done in 8723.960s.
```

6. Méthode supervisée

6.1. Sparses matrices

6.2. Multi output – multi label

6.3. Predict vs predict proba

7. Comparaison des résultats

SVC avec 40k d'échantillons :

Unigram : accuracy=0.194 en 700s pour 34Mo

Bigram : accuracy=0.205 en 1000s pour 270Mo

La taille étant importante pour moi étant donnée la mise en ligne, et le gain faible je vais rester en unigram pour augmenter la taille de l'échantillon.

SVC avec 100k d'échantillons en unigram : accuracy=0.213 en 2100s pour 65Mo

8. Evaluation

9. Remerciements