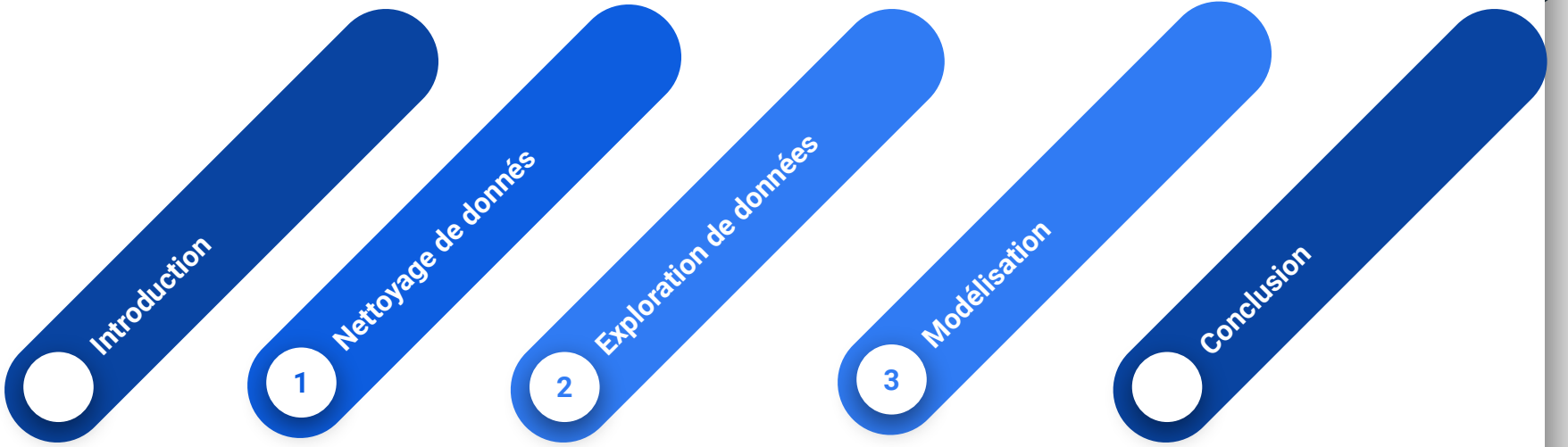
Projet 6 : Catégoriser automatiquement des questions

Réalisé par : Taher Haggui

Plan:


Introduction
1
Nettoyage de données
2
Exploration de données
3
Modélisation
Conclusion

Problématique:

Développer un système de suggestion de tags, permettant d'assigner plusieurs tags pertinent à une question.

Comment ?:

1.Recenser une base de données de milliers de questions.

2.Prétraitrement des textes de questions .

3. Tester les approches supervisées et non supervisées .

4.Développer une API permettant d’assigner plusieurs tags pertinents à une question .

Résultats attendus:

Une API capable d’assigner des tags pertinents à une question.

Nettoyage de données : Nbre de questions extraites

Requéte : **Select Id,Body, Tags from Posts where id <50000**

Récupérer 27182 questions

Nettoyage de données : Décomposition des textes

- ➔ Tokenization : le découpage en mots des différents textes.

- ➔ Normalisation: ne pas tenir compte des détails importants (ponctuation, majuscules, etc.)

Décomposition des textes: Illustration

Texte avant tokenization :

```
"<p>I want to use a track-bar to change a form's opacity.</p>\n\n<p>This is my code:</p>\n\n<pre><code>decimal tran
s = trackBar1.Value / 5000;\nthis.Opacity = trans;\n</code></pre>\n\n<p>When I build the application, it gives the
following error:</p>\n\n<blockquote>\n  <p>Cannot implicitly convert type <code>'decimal'</code> to <code>'doubl
e'</code>.</p>\n</blockquote>\n\n<p>I tried using <code>trans</code> and <code>double</code> but then the control d
oesn't work. This code worked fine in a past VB.NET project.</p>\n"
```

Texte après tokenization:

```
'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,valu
e,5000,this,opacity,trans,code,pre,p,when,i,build,the,application,it,gives,the,following,error,p,blockquote,p,canno
t,implicitly,convert,type,code,decimal,code,to,code,double,code,p,blockquote,p,i,tried,using,code,trans,code,and,co
de,double,code,but,then,the,control,doesn,t,work,this,code,worked,fine,in,a,past,vb,net,project,p'
```

Texte après Stemming:

```
'p,i,want,to,use,a,track,bar,to,chang,a,form,s,opac,p,p,this,is,my,code,p,pre,code,decim,tran,trackbar1,valu,5000,t
his,opac,tran,code,pre,p,when,i,build,the,applic,it,give,the,follow,error,p,blockquot,p,cannot,implicit,convert,typ
e,code,decim,code,to,code,doubl,code,p,blockquot,p,i,tri,use,code,tran,code,and,code,doubl,code,but,then,the,contro
l,doesn,t,work,this,code,work,fine,in,a,past,vb,net,project,p'
```

Nettoyage de données : Stopwords

- ➔ Les 100 mots les plus utilisés dans nos textes .

- ➔ Les stopwords fournis par NLTK pour la langue anglaise.

Stopwords :Illustration

Texte avec Stopwords :

```
'p,i,want,to,use,a,track,bar,to,chang,a,form,s,opac,p,p,this,is,my,code,p,pre,code,decim,tran,trackbar1,valu,5000,t
his,opac,tran,code,pre,p,when,i,build,the,applic,it,give,the,follow,error,p,blockquot,p,cannot,implicit,convert,typ
e,code,decim,code,to,code,doubl,code,p,blockquot,p,i,tri,use,code,tran,code,and,code,doubl,code,but,then,the,contro
l,doesn,t,work,this,code,work,fine,in,a,past,vb,net,project,p'
```

Texte sans Stopwords:

```
'track,bar,chang,form,opac,decim,tran,trackbar1,valu,5000,opac,tran,build,give,follow,error,blockquot,cannot,implic
it,convert,type,decim,doubl,blockquot,tri,tran,doubl,control,fine,past,vb,project'
```

Nettoyage de données: Bilan de nettoyage

	Avant nettoyage	Apres nettoyage
Vocabulaires	116108	53966

Exploration des données :Bag of words

Objet: représenter chaque texte par la fréquence d'apparition de ses mots.

➡ En pratique, nous avons établi la matrice composé de N textes qui forment notre corpus.

➡ Nous avons utilisé La fonction **CountVectorizer** de la librairie sklearn pour mettre en oeuvre cette matrice.

Exploration des données :TF-IDF

Objet: Mesurer l'importance de chaque terme contenu dans chaque texte relativement au corpus.

Accorder un poids à chaque terme de chaque texte.

Exploration des données :TF-IDF

poids= fréquence du terme x indicateur de similarité

indicateur de similarité= l'inverse de proportion des textes qui utilise le terme, à l'échelle logarthmique.

Exploration des données :Tags pertinents

Nbre total des tags= 2425

Choix: sélectionner les tags les plus fréquent et qui sont présent dans 80% des questions.

➡ 100 tags

➡ 4481 questions

Modélisation : méthode non supervisée

- La matrice bags of words .
- La méthode LDA

Méthode non supervisée: choix du nombre de topics

↳ Choix du nombre de topics: GridSearchCV

Meilleur modèle:

```
Best Model's Params : {'n_components': 10}
Best LOg likelihood score : -756287.2624386012
Model perplexity : 2192.6454846669
```

Méthode non supervisée: Illustration des topics

```
Topic 0 :
format wiki wikipedia blog number thread librari msdn microsoft en
Topic 1 :
request site session mail content header password email secur user
Topic 2 :
insert return end array queri index select list databas sql
Topic 3 :
wcf currentpag stackoverflow pattern os https regex python rubi book
Topic 4 :
user know chang thing control question think page report project
Topic 5 :
js jqueri javascript text id element tag div amp event
Topic 6 :
2gb testantlr blobout cc myapp getupperbound orig applet callback matrix
Topic 7 :
properti count null type void method valu static return public
Topic 8 :
acm directcast jdk lang stub javas union row sup sun
Topic 9 :
visual plugin eclips commit dim color subvers repositori merg branch
```

Méthode non supervisée: Choix des tags

Choix: Retourner les 2 mots les plus utilisés de trois sujets les plus représentatifs de chaque texte.

Démarche:

- ➔ représenter la distribution des textes par les topics.
- ➔ choisir les trois topics les plus représentatifs de chaque texte.
- ➔ représenter la fréquence d'utilisation des termes par chaque sujet.
- ➔ choisir les deux termes les plus utilisés par chaque topic.

méthode non supervisée: Performances

True Tags:

Approche: Cette approche consiste à déterminer le pourcentage de nouveaux mots clés retournés par la méthode non supervisée, correspond au vrais tags du jeu de données initial.

Méthode: Comparer manuellement, chaque mot clés retourné avec les différents tags réels de texte en question.

Résultats: Score de performance de 2.5%.

Interprétation: 2.5% des nouveaux mots clés correspond parfaitement aux tags réels.

méthode non supervisée: Performances

Pertinence des Tags:

Approche: **Cette approche consiste à évaluer la pertinence des nouveaux mots clés retournés par la méthode non supervisée. Un nouveau mot clés,est jugé pertinent s'il faisait partie de l'ensemble des tags du jeu de données initial.**

Méthode: Tester manuellement, si chaque mot clés fait partie de l'ensemble des tags du jeu de données initial.

Résultats: Score de performance de 30%.

Interprétation: 30% des nouveaux mots clés sont jugés pertinent tout en se basant sur cette approche.

Modélisation: Approche supervisée

- TF-IDF
- Vrais tags de chaque texte
- OneVsRestClassifier

Approche supervisée : Implémentation

- ➔ **Séparation des données en jeu d'entraînement et jeu de test (40% en jeu d'entraînement et 60% en jeu de test)**
- ➔ **Faire l'apprentissage de l'algorithme machine learning sur le jeu de données.**
- ➔ **Prédictions des labels sur les données de test.**

Approche supervisée: Performance

True Tags:

Approche: Cette approche consiste à déterminer le pourcentage de mots clés retournés par la méthode supervisée, correspond au vrais tags du jeu de données initial.

Méthode: Comparer manuellement, chaque mot clés de chaque question par les vrais tags de celle-ci.

Résultats: Score de performance de 6.7%.

Interprétation: 6.7% des nouveaux mots clés correspond parfaitement aux tags réels.

Comparaison entre les deux approches

	Tags réels	Pertinence des tags
Approche non supervisée	2.5%	30%
Approche supervisée	6.7%	-

➔ L'approche non supervisée, peut sembler efficace dans la génération des nouveaux mot clés pertinent.

➔ L'approche supervisée , se montre inefficaces dans la prédiction des vrais tags et elle est très gourmande en termes de ressources.

API

Finalité: Assigner plusieurs tags pertinents à une question.

Approche adopté: Méthode non supervisée.

Outils: Flask, html, Pythonanywhere.

http://taher.pythonanywhere.com/

Conclusion

- ★ L’approche non supervisée , est très efficace pour retourner des nouveaux mots clés associés à une question.
- ★ L’approche supervisé, ne montre pas une efficacité significative dans la définition des tags pertinents.
- ★ Nous avons pu développer une API, capable de retourner 6 tags pertinents à une question , basée sur l’approche non supervisée.