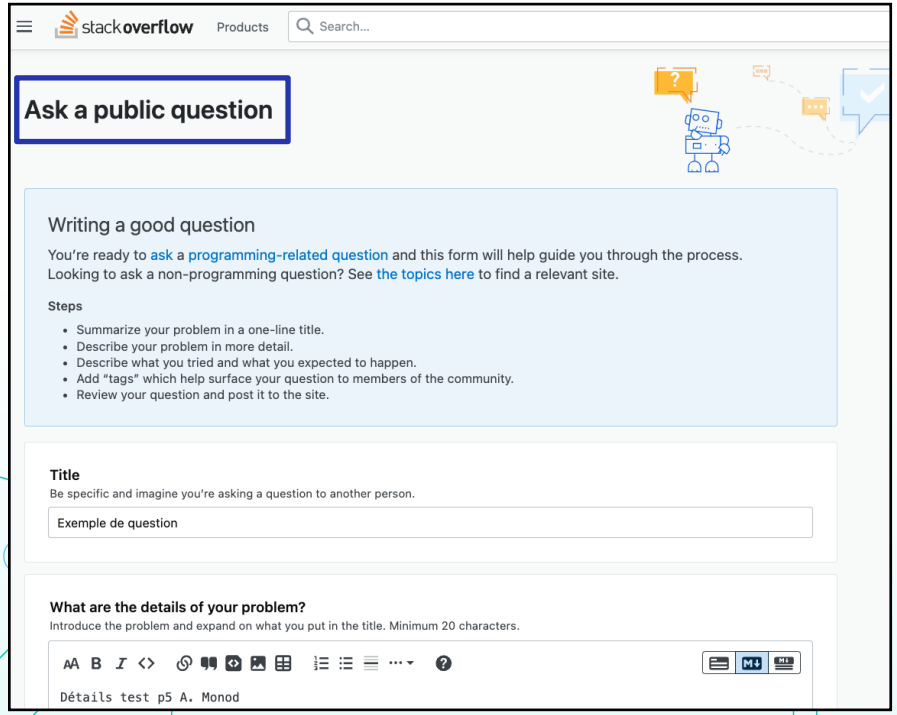
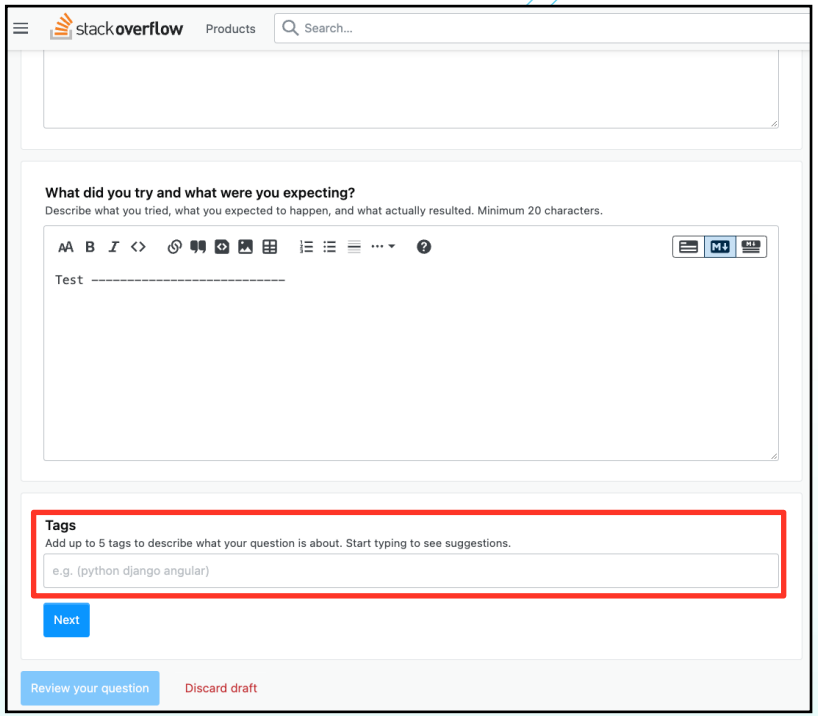
Catégorisez automatiquement des questions

A. Monod - Parcours ML - p5

Contexte

Contexte


stackoverflow
Products
Search...
Ask a public question
Writing a good question
You're ready to ask a programming-related question and this form will help guide you through the process.
Looking to ask a non-programming question? See the topics here to find a relevant site.
Steps
Summarize your problem in a one-line title.
Describe your problem in more detail.
Describe what you tried and what you expected to happen.
Add "tags" which help surface your question to members of the community.
Review your question and post it to the site.
Title
Be specific and imagine you're asking a question to another person.
Exemple de question
What are the details of your problem?
Introduce the problem and expand on what you put in the title. Minimum 20 characters.
Détails test p5 A. Monod


stackoverflow
Products
Search...
What did you try and what were you expecting?
Describe what you tried, what you expected to happen, and what actually resulted. Minimum 20 characters.
Test ----------
Tags
Add up to 5 tags to describe what your question is about. Start typing to see suggestions.
e.g. (python django angular)
Next
Review your question
Discard draft

Contexte

Problématique

Nouveaux utilisateurs

Proposition

Suggestion de tags.

Solution technique

Algorithme NLP - Python

SOMMAIRE

1

Preprocessing NLP

Nettoyage, Tokenization, Lemmatization

2

BOW et Modèles non supervisés

CV, Tf-idf, LDA

3

Modèles supervisés

W2V, BERT, USE

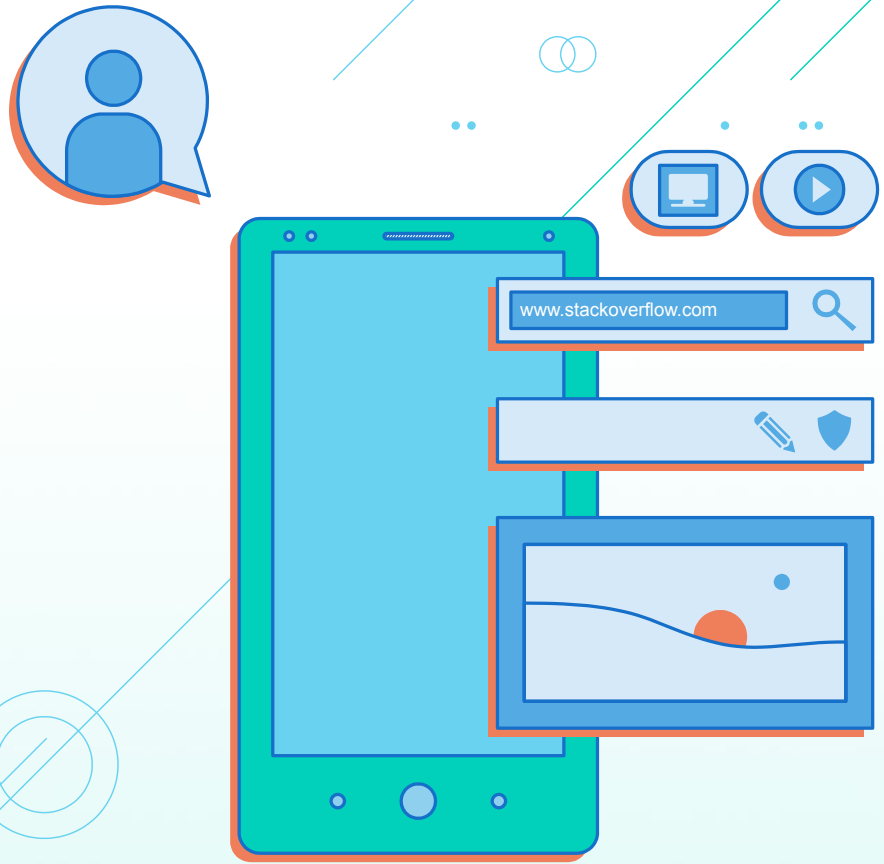
4

Résultats et déploiement

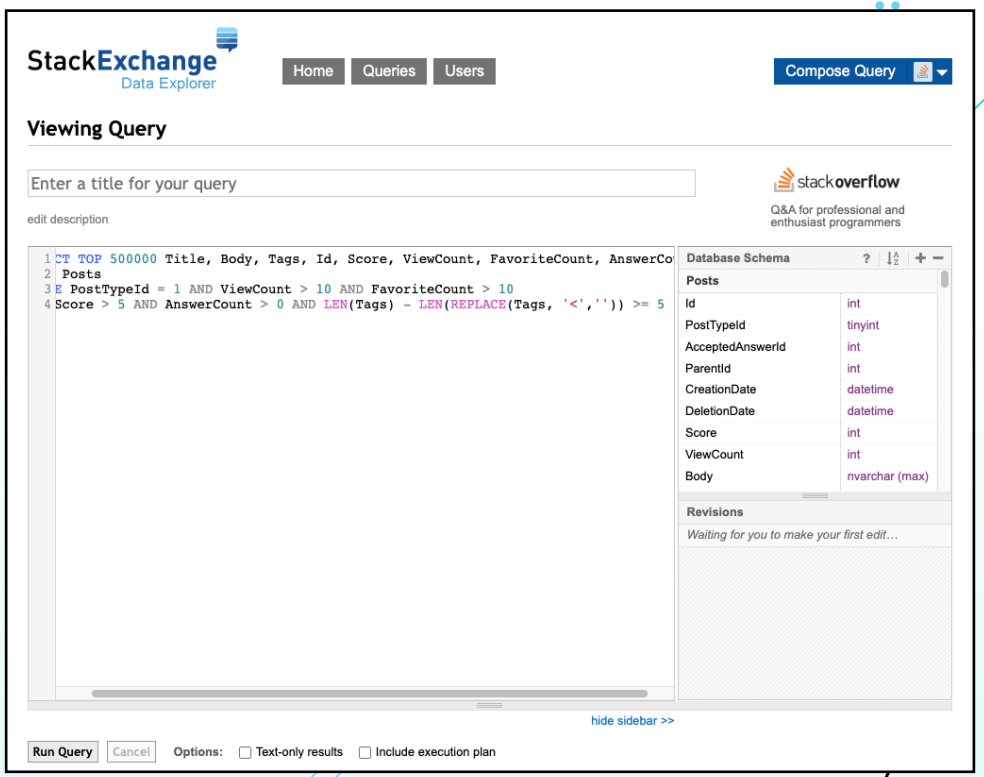
Tableau, API

1

Preprocessing



1 - Requête SQL



2 - Nettoyage tags

BeautifulSoup

```
1  # Show before suppr
2  print(query_results.Body[0])
```

```
<p>Is it possible to define a class in C# such that</p>

<pre><code>class GenericCollection&lt;T&gt; : SomeBaseCollection&lt;T&gt; where T : Delegate
</code></pre>

<p>I couldn't for the life of me accomplish this last night in .NET 3.5. I tried using</p>

<p><code>delegate, Delegate, Action&lt;T&gt; and Func&lt;T, T&gt;</code></p>

<p>It seems to me that this should be allowable in some way. I'm trying to implement my own EventQueue.</p>

<p>I ended up just doing this [primitive approximation mind you].</p>

<pre><code>internal delegate void DWork();

class EventQueue {
    private Queue&lt;DWork&gt; eventq;
}
</code></pre>

<p>But then I lose the ability to reuse the same definition for different types of functions.</p>

<p>Thoughts?</p>
```

3 - Tokenization

```
"Is it possible to define a class in C# such that class GenericCollection<T> :
SomeBaseCollection<T> where T : Delegate I couldn't for the life of me accomplish this last night
in .NET 3.5. I tried using delegate, Delegate, Action<T> and Func<T, T> It seems to me that this
should be allowable in some way. I'm trying to implement my own EventQueue. I ended up just doing
this [primitive approximation mind you]. internal delegate void DWork();  class EventQueue {
private Queue<DWork> eventq; } But then I lose the ability to reuse the same definition for
different types of functions. Thoughts?"
```

NLTK

```
[ is', it', 'possible', 'to', 'define', 'a', 'class', 'in', 'c', 'such', 'that', 'class', 'genericcollection', 't', 'somebasecollection', 't',
'where', 't', 'delegate', 'i', 'couldn', 't', 'for', 'the', 'life', 'of', 'me', 'accomplish', 'this', 'last', 'night', 'in', 'net', '3', '5',
'i', 'tried', 'using', 'delegate', 'delegate', 'action', 't', 'and', 'func', 't', 't', 'it', 'seems', 'to', 'me', 'that', 'this', 'should',
'be', 'allowable', 'in', 'some', 'way', 'i', 'm', 'trying', 'to', 'implement', 'my', 'own', 'eventqueue', 'i', 'ended', 'up', 'just', 'doing',
'this', 'primitive', 'approximation', 'mind', 'you', 'internal', 'delegate', 'void', 'dwork', 'class', 'eventqueue', 'private', 'queue',
'dwork', 'eventq', 'but', 'then', 'i', 'lose', 'the', 'ability', 'to', 'reuse', 'the', 'same', 'definition', 'for', 'different', 'types', 'of',
'functions', 'thoughts', 'dwork', 't', 't', 't', 't']
```

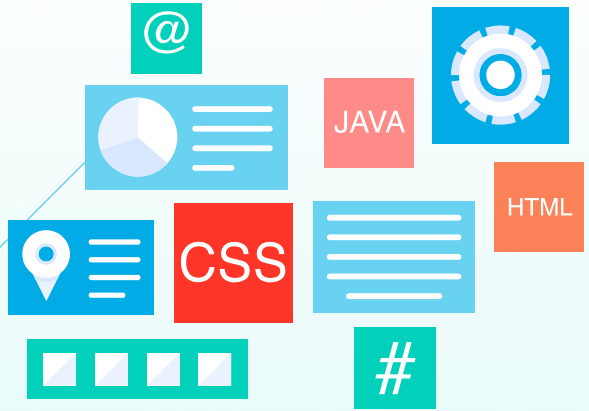
```
['possible', 'define', 'class', 'class', 'genericcollection', 'somebasecollection', 'delegate',
'life', 'accomplish', 'last', 'night', 'net', 'tried', 'using', 'delegate', 'delegate', 'action',
'func', 'seems', 'allowable', 'way', 'trying', 'implement', 'eventqueue', 'ended', 'primitive',
'approximation', 'mind', 'internal', 'delegate', 'void', 'dwork', 'class', 'eventqueue',
'private', 'queue', 'dwork', 'eventq', 'lose', 'ability', 'reuse', 'definition', 'different',
'types', 'functions', 'thoughts', 'dwork']
```

4 - Lemmatization

```
['possible', 'define', 'class', 'genericcollection', 'somebasecollection', 'delegate', 'life', 'accomplish', 'last', 'night', 'net', 'try',
'use', 'action', 'func', 'seem', 'allowable', 'way', 'try', 'implement', 'eventqueue', 'end', 'primitive', 'approximation', 'mind', 'internal',
'void', 'dwork', 'private', 'queue', 'eventq', 'lose', 'ability', 'reuse', 'definition', 'different', 'type', 'function', 'thought']
```

```
1  query_results["body_cleaned"]
✓ 0.5s

0        [define , class , genericcollection , somebase...
1        [scan , directory , folder , file , cross , pl...
2        [wcf , service , return , know , topic , retur...
3        [text , file , hdfs , convert , data , frame ,...
4        [place , split , use , mysqli , stuff , statem...
                                ...
27729    [stephan , lavavej , talk , cppcon , class , t...
27730    [spring , boot , application , command , mvn ,...
27731    [response , url , want , page , item , nextpag...
27732    [ohlc , sample , time , series , data , pandas...
27733    [design , support , library , bottomsheetbehav...
Name: body_cleaned, Length: 27734, dtype: object
```



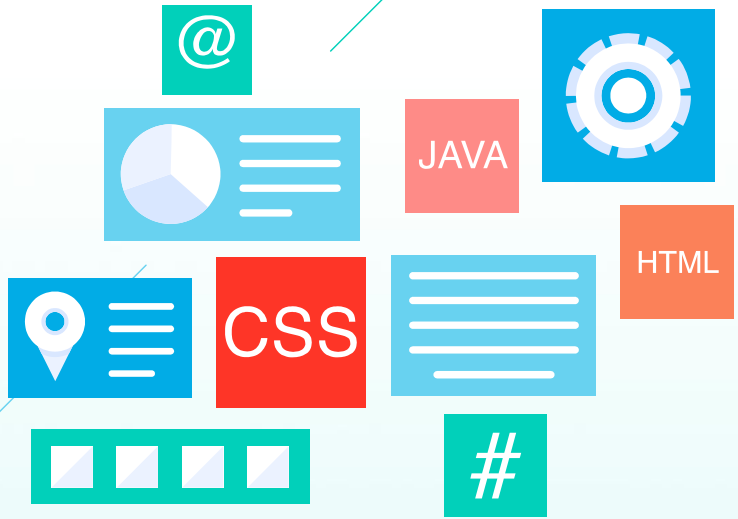
5 - Sélection des noms

{'possible': 'a', 'define': 'a', 'class': 'n', 'genericcollection': 'n', 'somebasecollection': 'n', 'delegate': 'n', 'life': 'n', 'accomplish': 'a', 'last': 'a', 'night': 'n', 'net': 'n', 'tried': 'v', 'using': 'v', 'action': 'n', 'func': 'n', 'seems': 'v', 'allowable': 'a', 'way': 'n', 'trying': 'v', 'implement': 'a', 'eventqueue': 'n', 'ended': 'v', 'primitive': 'a', 'approximation': 'n', 'mind': None, 'internal': 'a', 'void': 'n', 'dwork': 'n', 'private': 'a', 'queue': 'n', 'eventq': 'v', 'lose': 'a', 'ability': 'n', 'reuse': 'v', 'definition': 'n', 'different': 'a', 'types': 'n', 'functions': 'n', 'thoughts': 'n'}

NLTK —> WordNet

6 - Tags

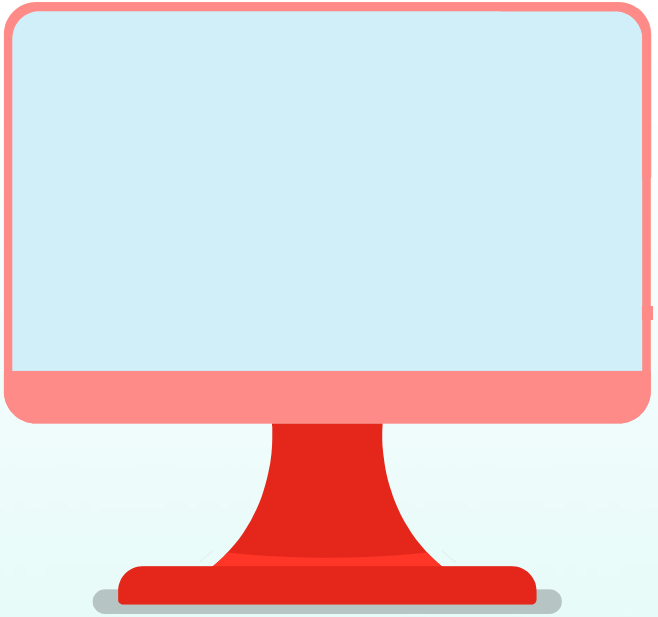
- Nettoyage
- Sélection des 100 tags les plus courants



2

BOW et Modèles non supervisés

CV - Tf-Idf - LDA



1 - CountVectorizer

	0px	100ms	100px	10k	10px	16dp	1px	1st	200px	20px	...	zeros	zip	zipcode	zipfile	zlib	zombie	zone
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
26150	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
26151	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
26152	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
26153	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
26154	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0

26155 rows × 6086 columns

2 - Tf-Idf

	0px	100ms	100px	10k	10px	16dp	1px	1st	200px	20px	...	zeros	zip	zipcode	zipfile	zlib	zombie	zone
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
26150	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
26151	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
26152	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
26153	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
26154	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0

26155 rows × 6086 columns

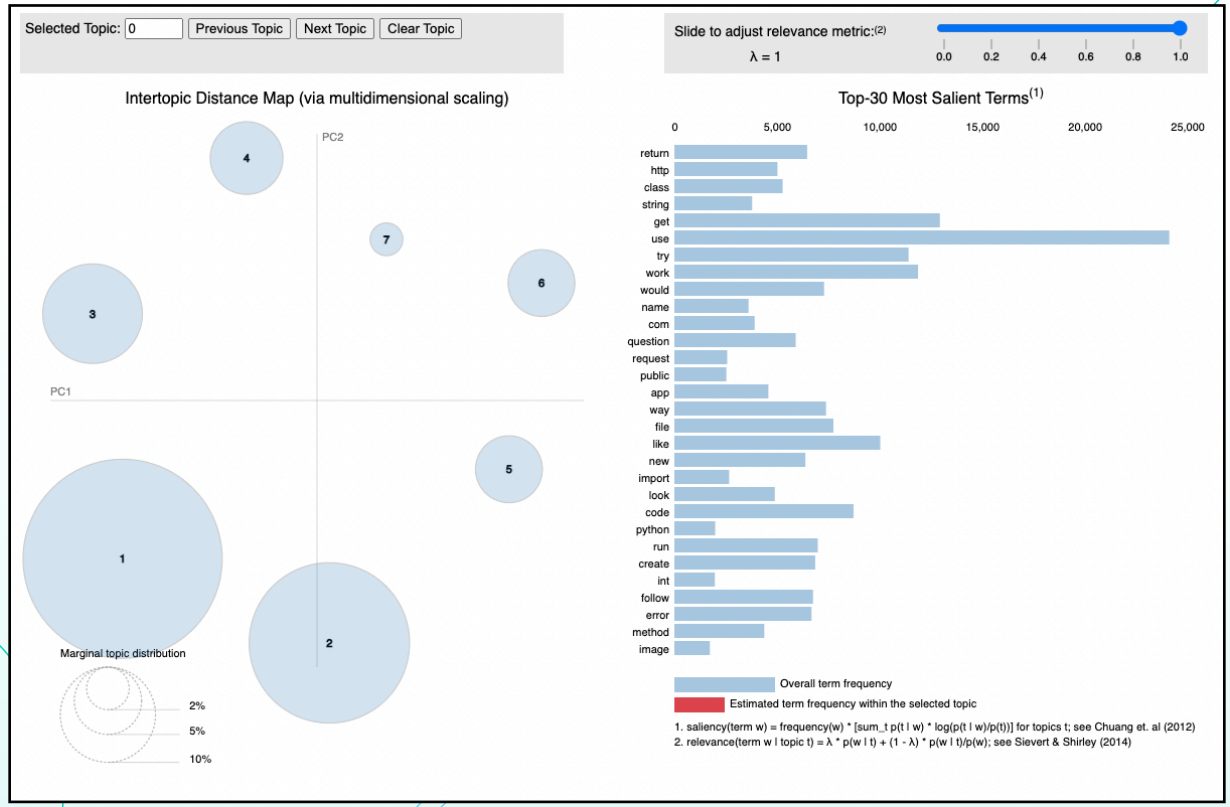
3 - LDA

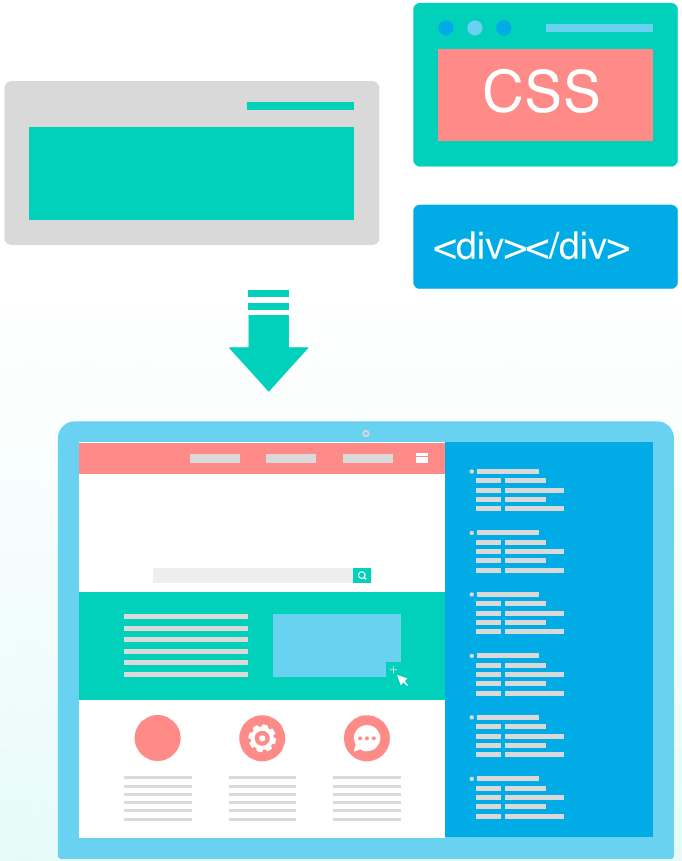
- P(word/topics) ; P(topics/documents)

GENSIM
topic modelling for humans

```
[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1),
(6, 1), (7, 1), (8, 1), (9, 1), (10, 1), (11, 1),
(12, 1), (13, 1), (14, 1), (15, 1), (16, 1), (17,
1), (18, 1), (19, 1)]
```

3 - LDA


Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
5
6
7
Marginal topic distribution
2%
5%
10%
Top-30 Most Salient Terms(1)
0
5,000
10,000
15,000
20,000
25,000
return
http
class
string
get
use
try
work
would
name
com
question
request
public
app
way
file
like
new
import
look
code
python
run
create
int
follow
error
method
image
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)



3

Modèles supervisés

W2Vec - BERT - USE

1 - W2Vec

Principe général

Transformer chaque mot en vecteur.

Fonctionnement

Transformation via matrice d'embedding.
Combinaison des différents embeddings des mots qui le composent.

Implémentation

Choix d'un vecteur de taille 300

2 - BERT

Principe général

Transfer Learning
Modèle bidirectionnel.

Fonctionnement

Non supervisé
Tient compte de l'ordre des mots.

Implémentation

Huggingface
Bert hub Tensorflow

3 - USE

Principe général

Transfer Learning

Fonctionnement

Matrice d'embedding avec vecteurs de grande taille
Tient compte de l'ordre des mots.

1 - RL

Principe général

Classification

Variable cible : qualitative

2 - SGDC

Principe général

Amélioration du SGD

Méthode itérative.

3 - RandomForrest

Principe général

Ensembles aléatoires.

Forêt aléatoire d’arbres de décision.

Implémentation

GridSearch pour optimisation des hyperparamètres

4

Résultats et déploiement

Scores

	W2V - RL	W2V - SGDC	W2V - RF	CV - RL	CV - SGDC	CV - RF	TF - RL	TF - SGDC	TF - RF
Jaccard	0.160442	0.129794	0.060781	0.422330	0.010088	0.277058	0.379594	0.298339	0.396181
Accuracy	0.108347	0.106544	0.076798	0.100775	0.061114	0.170723	0.224626	0.193258	0.180818
F1	0.265168	0.217026	0.108966	0.577192	0.019411	0.393203	0.526503	0.430747	0.544787
Precision	0.530518	0.535871	0.437806	0.482195	0.318617	0.660717	0.708699	0.716595	0.592142
Recall	0.182031	0.144221	0.066168	0.755437	0.010124	0.319366	0.434787	0.328803	0.551340
mean	0.249301	0.226691	0.150104	0.467586	0.083871	0.364213	0.454842	0.393548	0.453054

	BERT HF - RL	BERT HF - SGDC	BERT HF - RF	BERT TF - RL	BERT TF - SGDC	BERT TF - RF	USE - RL	USE - SGDC	USE - RF
Jaccard	0.677180	0.680855	0.730316	0.691499	0.706645	0.737376	0.646127	0.609971	0.771231
Accuracy	0.000000	0.000000	0.001000	0.000000	0.000000	0.000000	0.000000	0.000000	0.001147
F1	0.764567	0.771975	0.795186	0.781285	0.773366	0.805502	0.688463	0.662655	0.841262
Precision	0.792883	0.781816	0.791167	0.807121	0.831813	0.801630	0.688033	0.678238	0.849049
Recall	0.745856	0.763430	0.835764	0.762161	0.775387	0.841109	0.702215	0.661764	0.868541
mean	0.596097	0.599615	0.630687	0.608413	0.617442	0.637123	0.544968	0.522525	0.666246

API

https://fierce-oasis-92155.herokuapp.com/requete

Conclusion

<u>Apprentissage</u>

- Processus NLP
- Classification multi-classes

<u>Blocages</u>

- Pbs versioning TensorFlow
- Modèles BERT - USE —> Google Colab

<u>Améliorations</u>

- optimiser l’entrainement avec BERT
- Moins de 100 tags sélectionnés