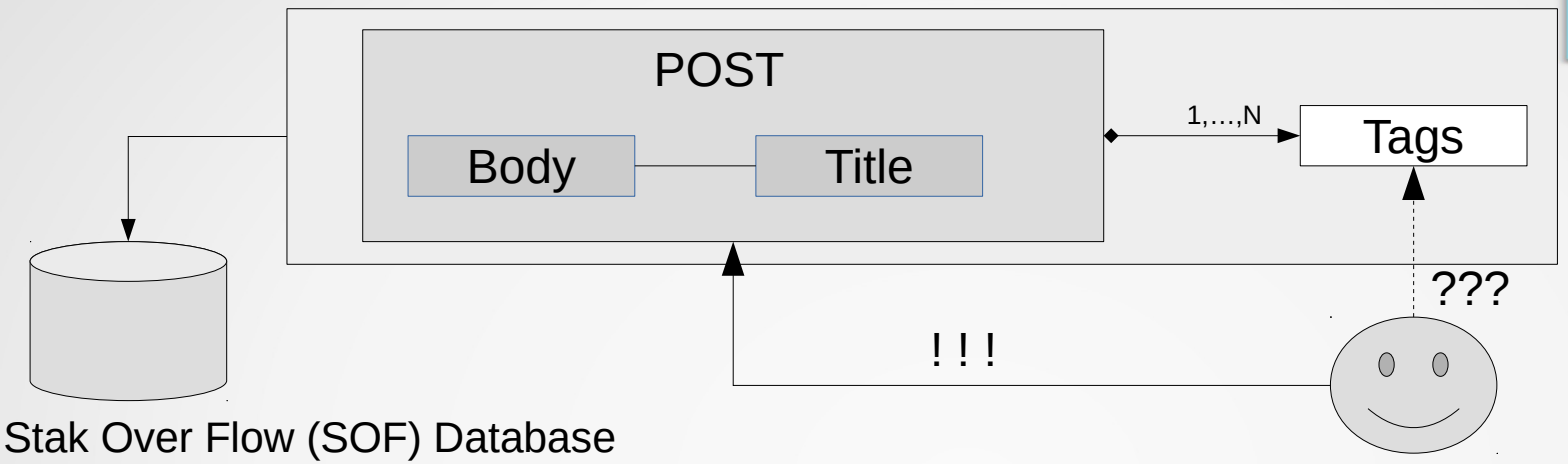
Parcours Datascientist : projet 6

Catégorisation automatique de questions

issues de la plateforme Stack Overflow

Francois BANGUI

Mission / study asumptions

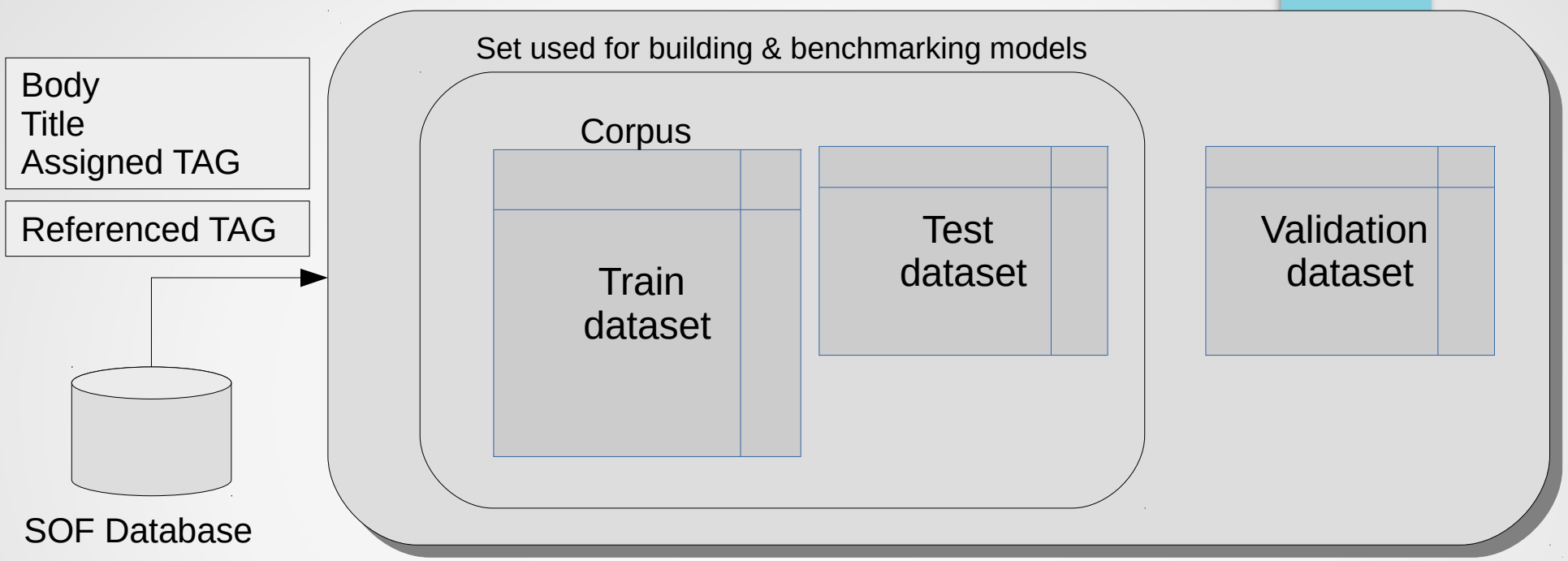


MISSION: to suggest TAG from a POST

- Context : IT
- Problem class : NLP
- Problem type : Multi-label classification (1 POST / N TAGs)

To suggest appropriate TAG thanks to a classification algorithm

Data-sets organization

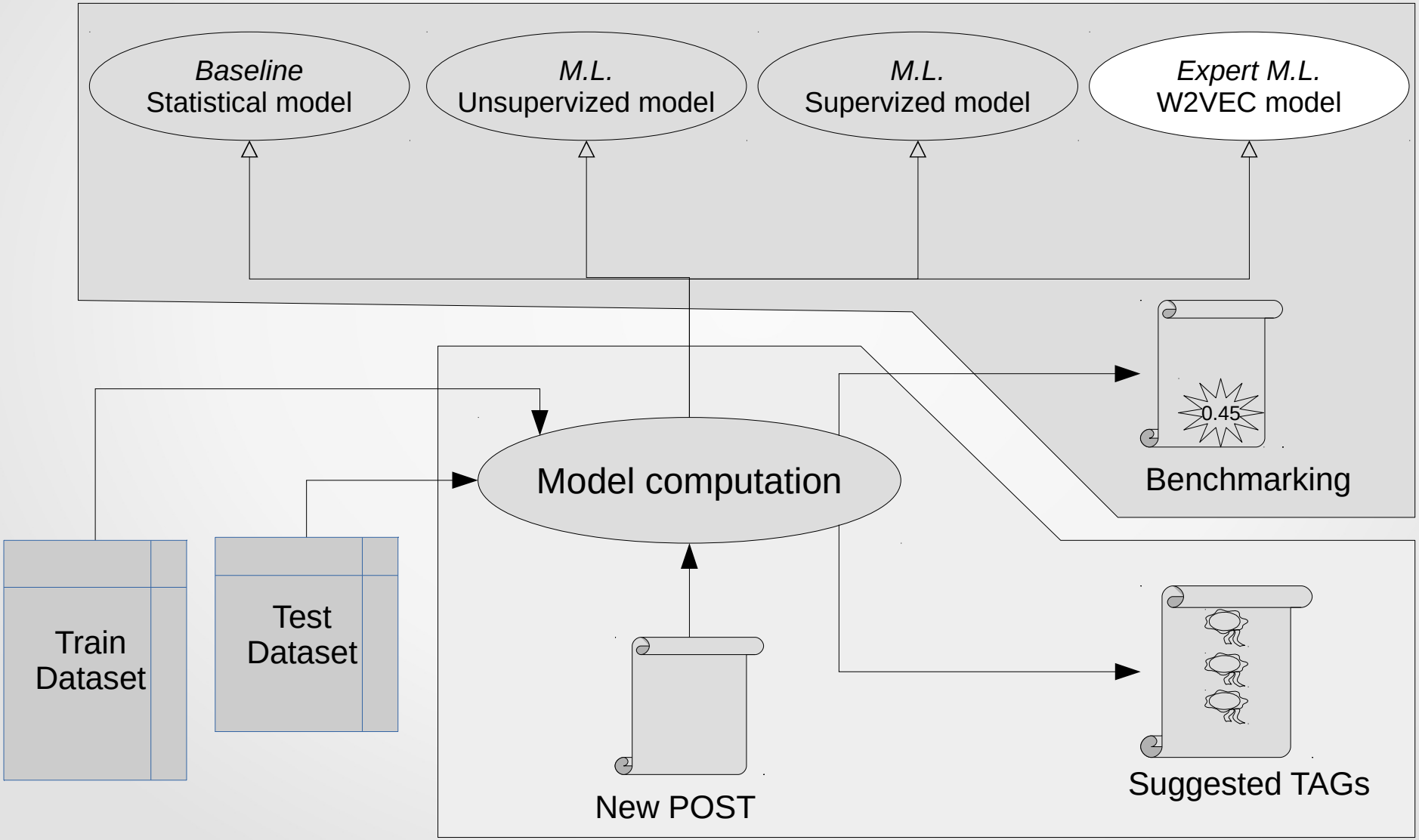


- Train dataset :
  - 25K POST
  - 6K Features
- Test dataset :
  - 15K POST
- Referenced TAG
  - 50K

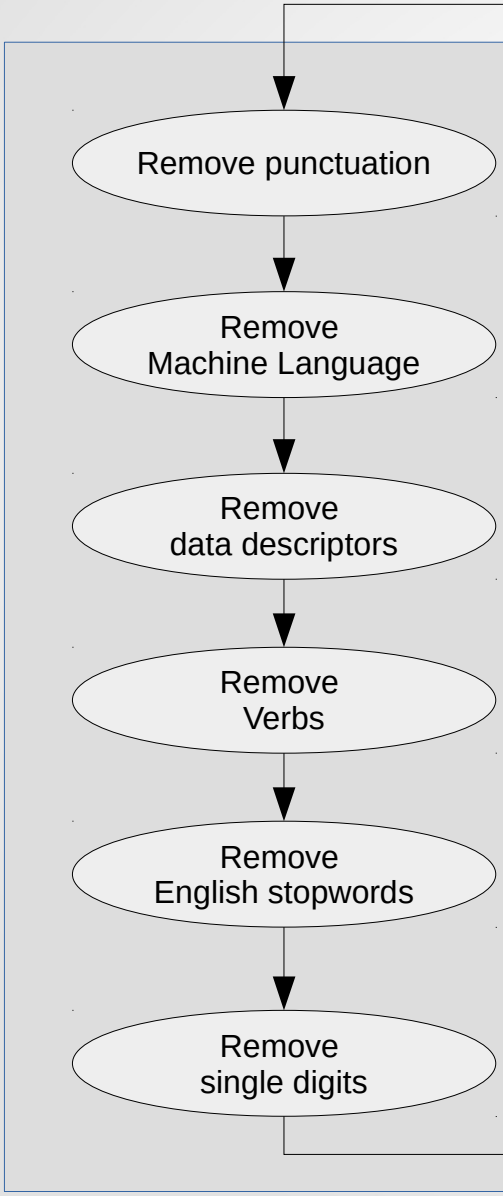
Models evaluation
- Same dataset
- Same measurement

Benchmarking

Benchmark global sheme



Standardization : POST --> list of TOKENS



how to use the c socket api in c++ on z/os

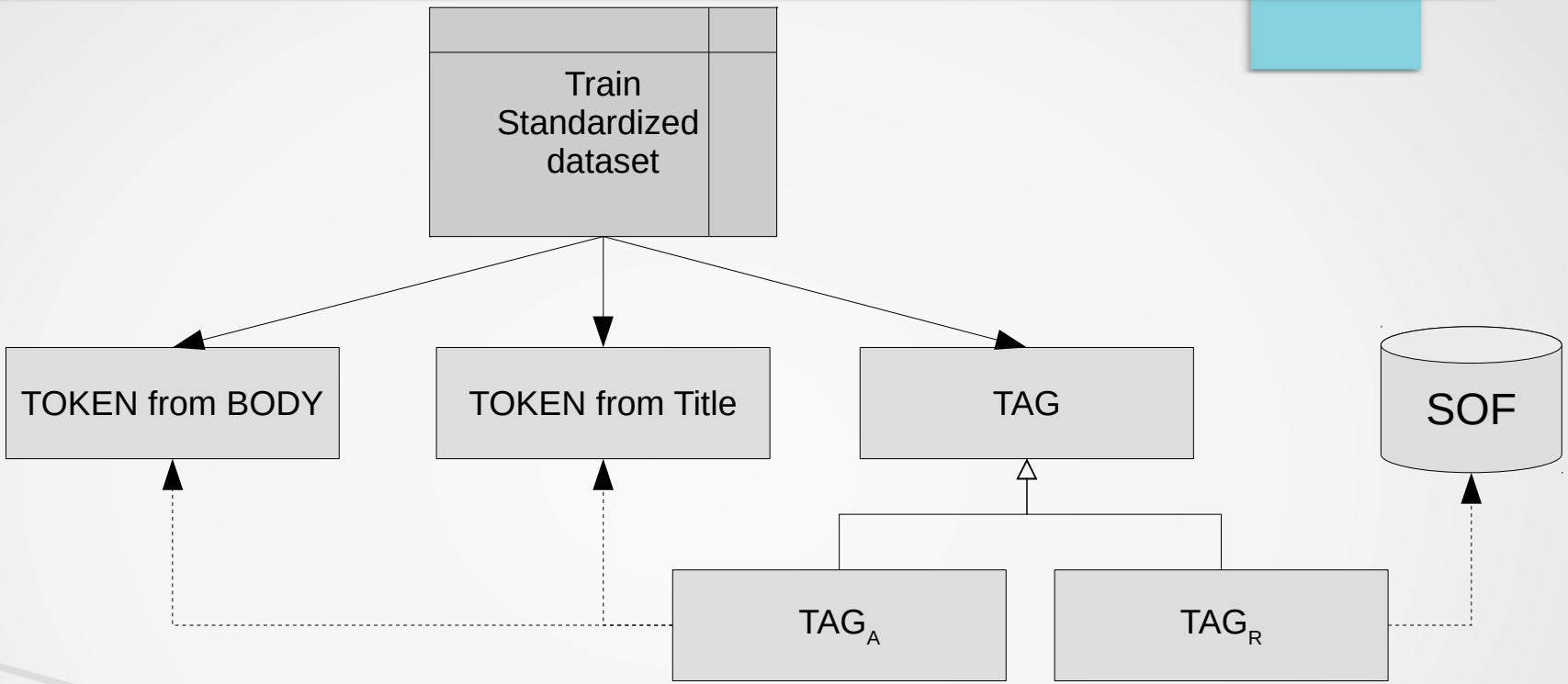
```
<p>I'm having issues getting the C sockets API to work properly in C++ on
<code>z/OS</code>.</p>
<p>Although I am including <code>sys/socket.h</code>, I still get compile
time errors telling me that <code>AF_INET</code> is not defined.</p>
<p>Am I missing something obvious, or is this related to the fact that
being on <code>z/OS</code> makes my problems much more complicated?</p>
<hr>
<p><strong>Update</strong>: Upon further investigation, I discovered that
there is an <code>#ifdef</code> that I'm hitting. Apparently
<code>z/OS</code> isn't happy unless I define which "type" of sockets I'm
using with:</p>
<pre><code>#define _OE_SOCKETS
</code></pre>
<p>Now, I personally have no idea what this <code>_OE_SOCKETS</code> is
actually for, so if any <code>z/OS</code> sockets programmers are out
there (all 3 of you), perhaps you could give me a rundown of how this all
works?</p>
```

TAG_A : ['c++', 'c', 'sockets', 'mainframe', 'zos']

['c', 'socket', 'api', 'c++', 'z', 'os']

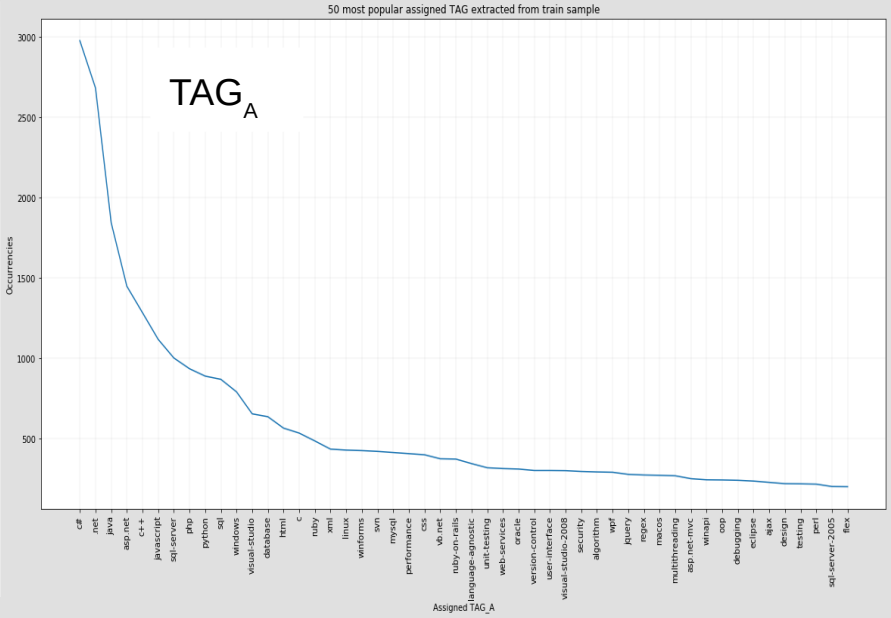
['m', 'issues', 'c', 'sockets', 'api', 'properly', 'c++', 'which', 'basically', 'whole', 'file', 'sure']

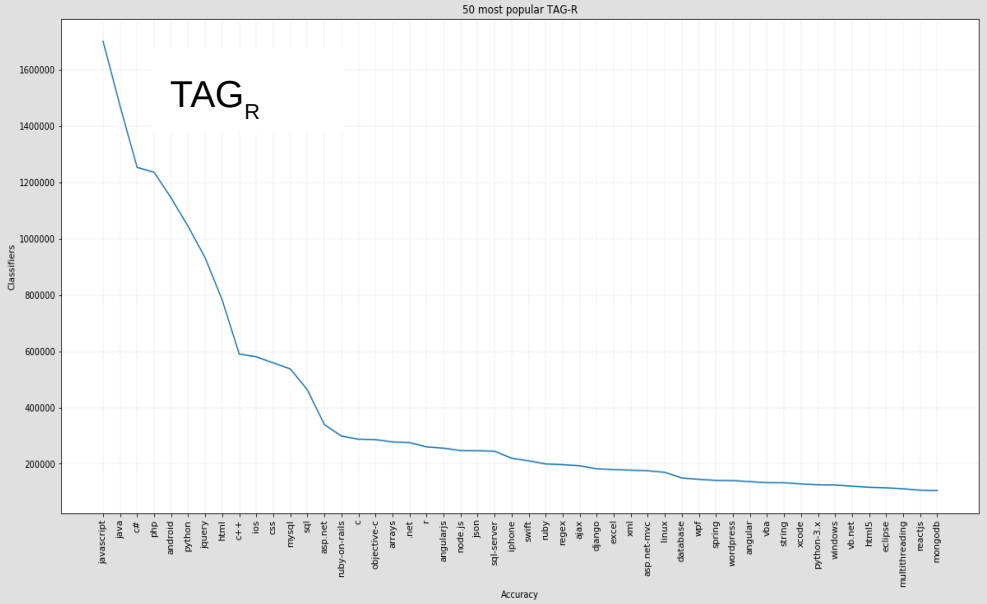
Data analysis over train Tokenized data-set

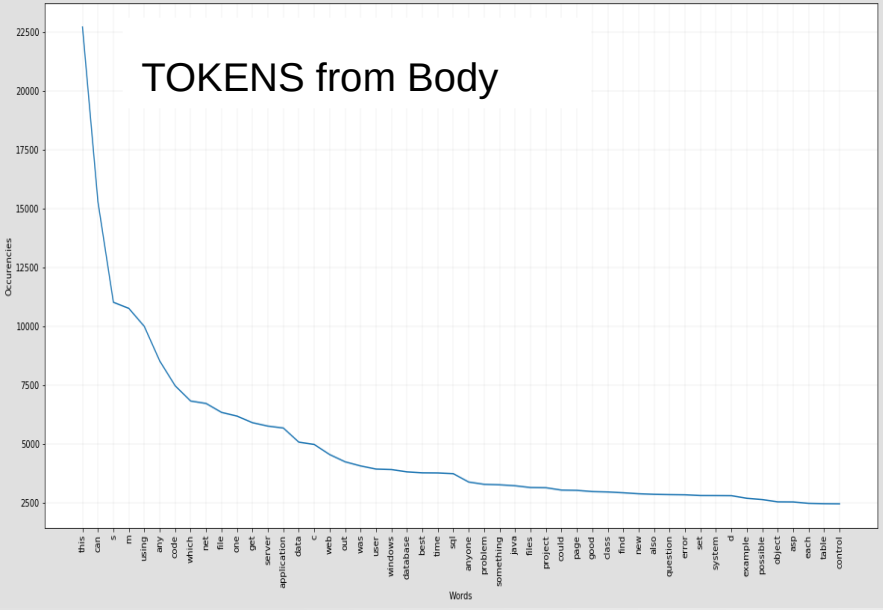


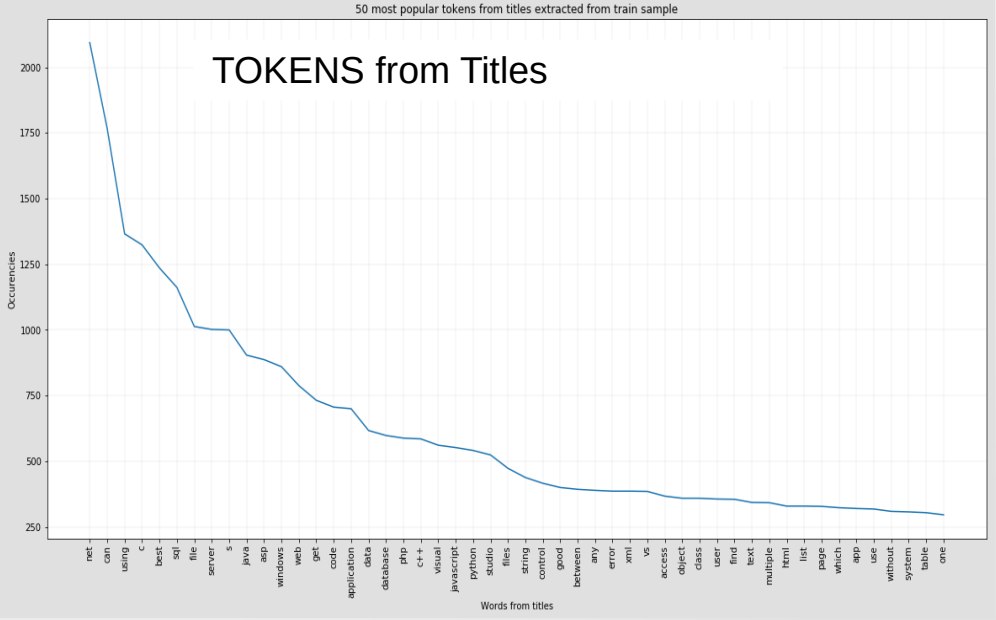
- TAG_R : Referenced TAG
- TAG_A : Assigned TAG issued from users
- TAG_S : Suggested TAG issued from benchmarked models
- TAG_E : Assigned TAG issued from expert process
- TOKENS : Words issued from standardization process
- N<X> : Number of elements referenced by <X>

TAG and TOKEN occurencies

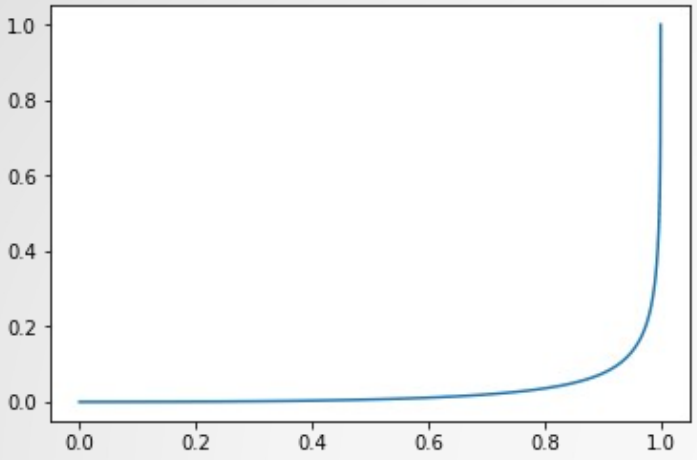




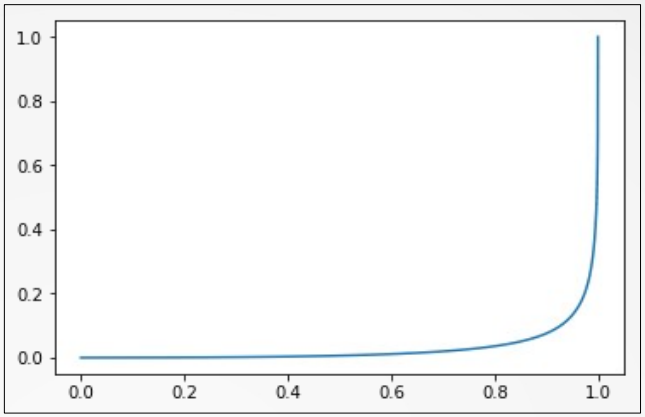




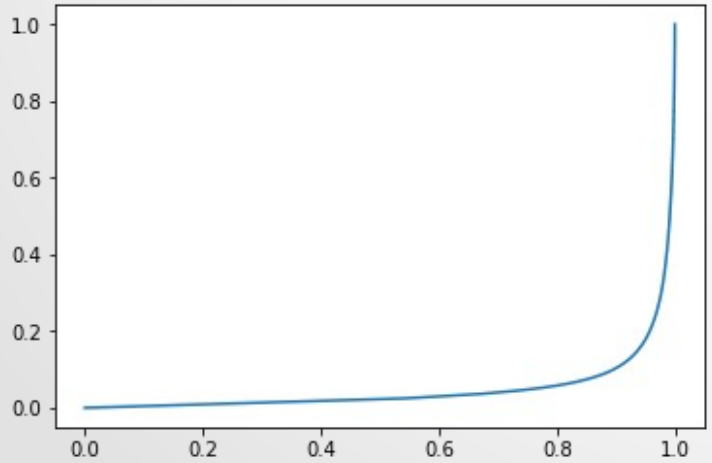
Lorenz curves / GINI indexes



TAG_R GINI index: 0.94



TAG_A GINI index: 0.91



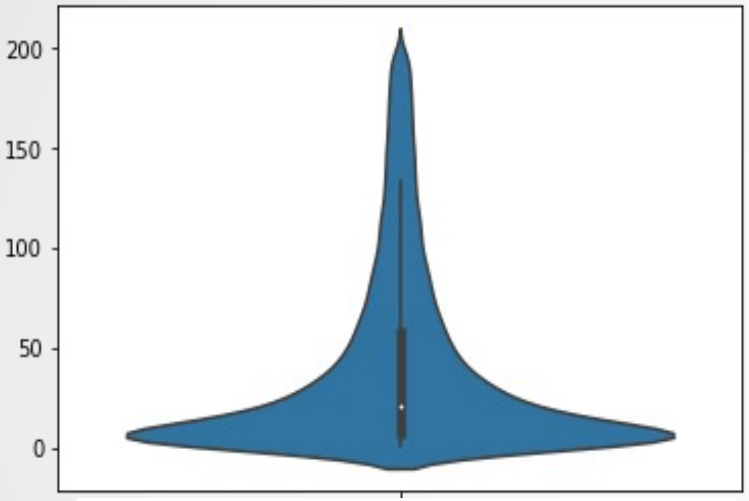
Tokens GINI index: 0.90

Less then 15 % of TAG or TOKEN
leads to ~ 90 % of POST representation

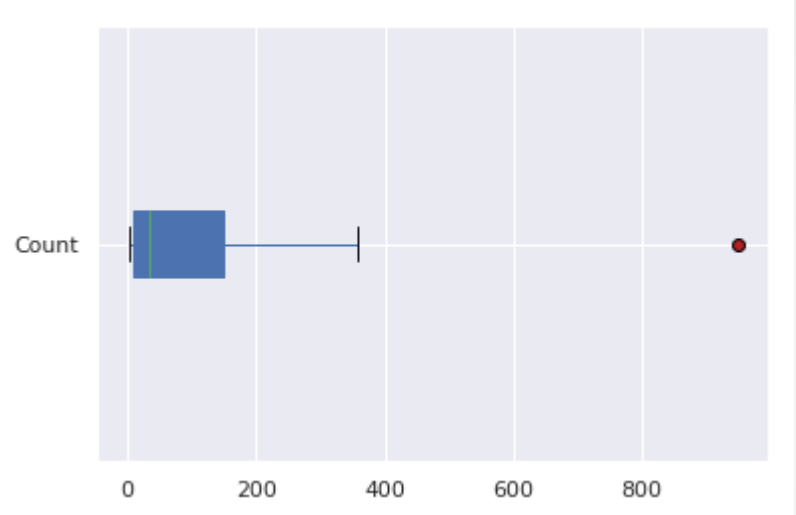
$$P(TAG_R|TAG_A)=0.93 \quad \Rightarrow Echantillon\ représentatif$$

$$P(TAG_R|TOKEN_{Body})=0.18$$
$$P(TAG_R|TOKEN_{Title})=0.40$$
$$\Rightarrow \{POST\}=\{BODY\}\cup\{TITLE\}$$

Hypothesis about TAG_A statistics law



TAG_A distribution & density from 25K POST



- Shapiro / Wilk : Statistics= 0.014 / p-value ~ 0.0
- Kolmogorov / Smirnov : Statistics= 0.93 / p-value~ 0.0

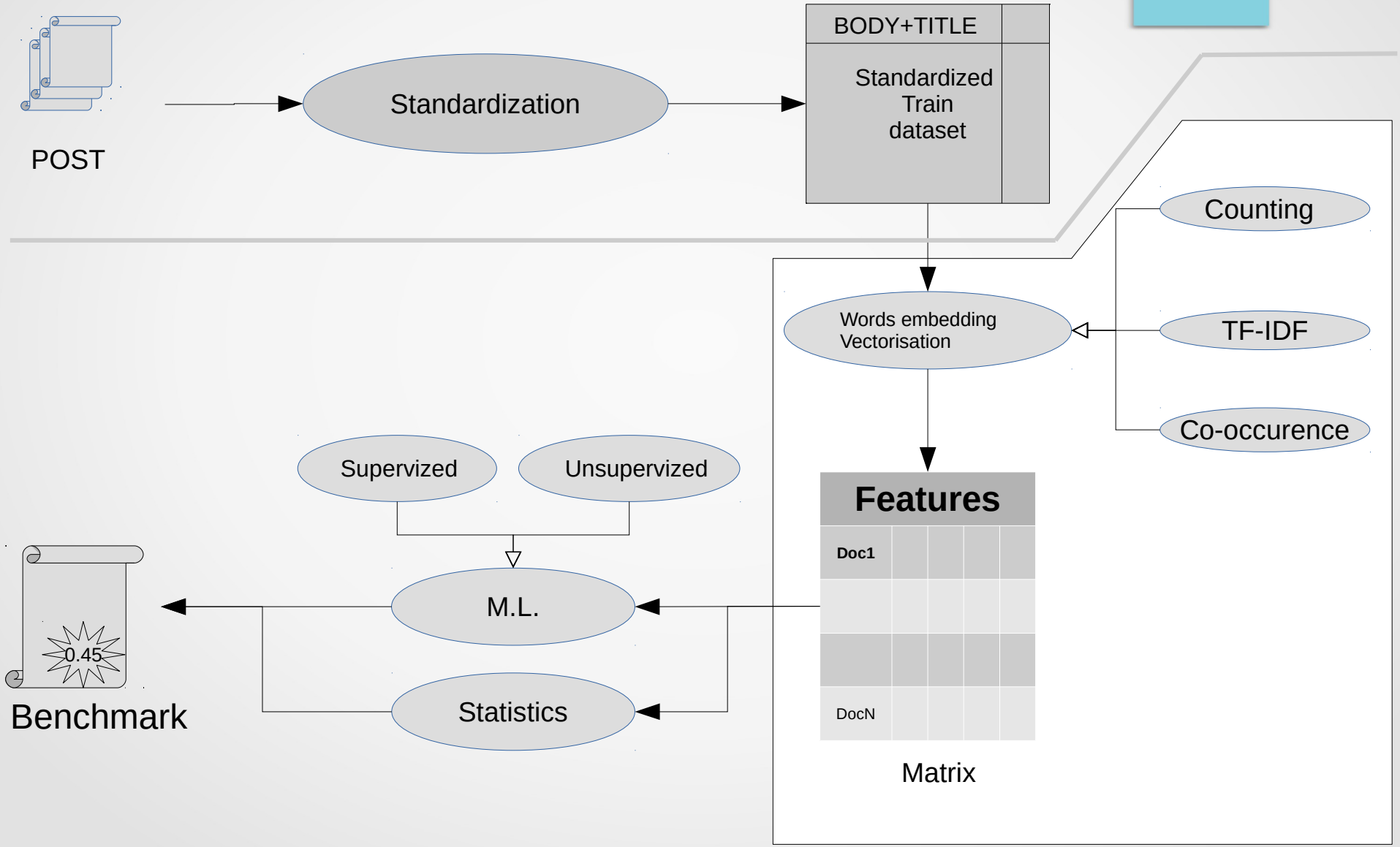
Null hypothesis is not very likely : $P(x|_{H0}) \sim 0$

Study asumptions

- ‘Document’ used for ‘POST’
  - POST = Body + Title
- No normal probabilistic law

- Bayesian asumptions :
  - Documents are independantly generated
  - Documents are identicaly distributed (same probability law)

Data-set vectorization : NLP --> Words Emb.



Vectorization process

TF-IDF : Token Frequency / Inverse Documents Frequency

- TF : The most a TOKEN is frequent in a POST,
  the most this Token represents the POST
- IDF : The most this TOKEN is frequent in other POST, less TF is valuated

Higher TOKEN values of TF-IDF from a POST discriminate this POST against others.

Co-occurency : (1,1) → (N,M):

- « Expression Reguliere» ↔ TAG_R
  - Measure of likehood (2,2) : $$P(Reguliere|Expression)=\frac{P(Expression,Reguliere)}{P(Expression)*P(Reguliere)}$$
  - Generalization : $$P(MOT_M|MOT_{1,\dots},MOT_{M-1})=\prod_{I=1}^{I=M-1}\frac{P(MOT_M,MOT_I)}{P(MOT_M)*P(MOT_I)}$$

POST are regarded independantly from each-others

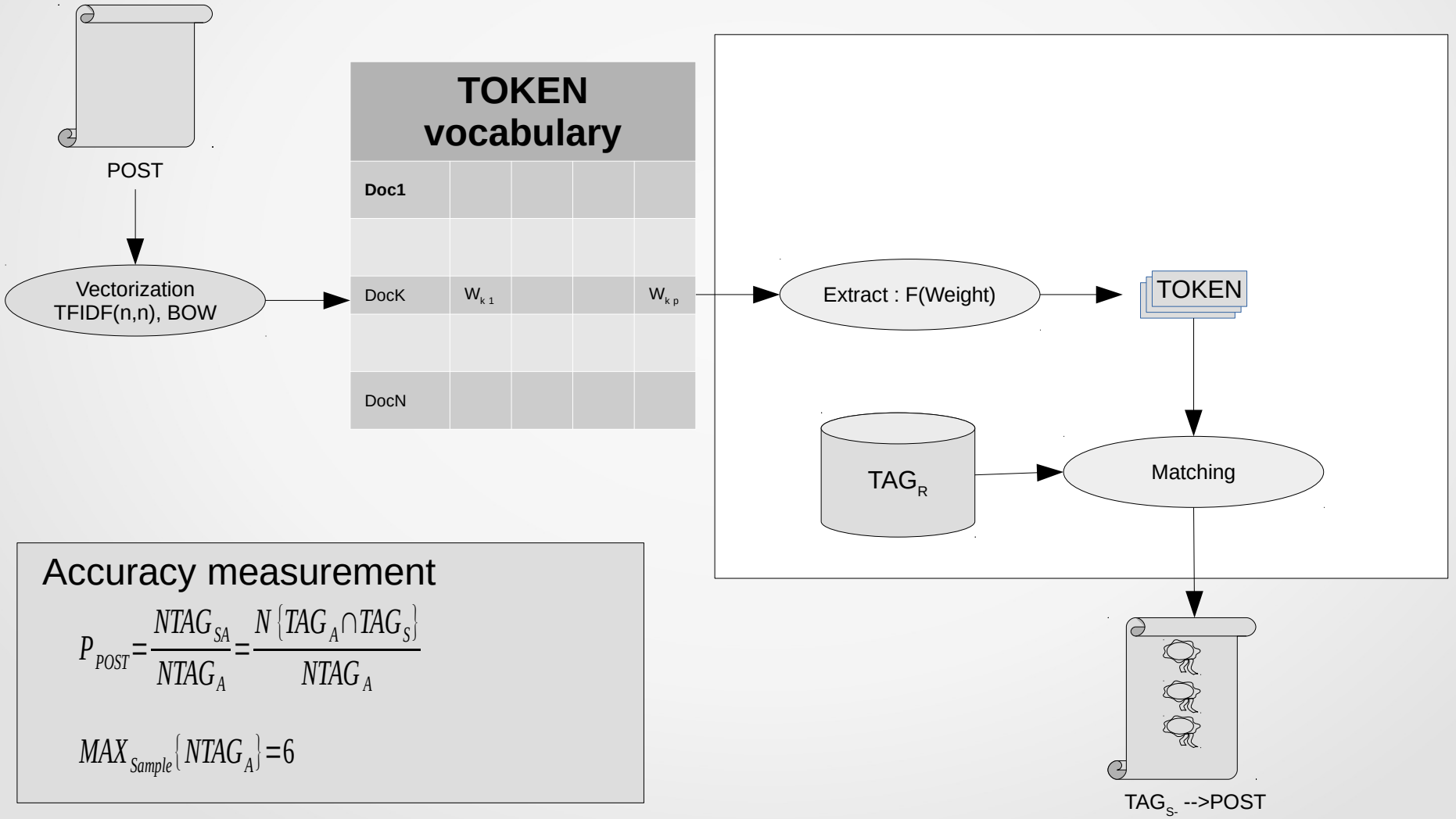
POST are digitalized independantly of TOKEN order

Benchmarking

STATISTICAL METHODS

Statistical methods

Prediction is based on statistics of features over the whole corpus.

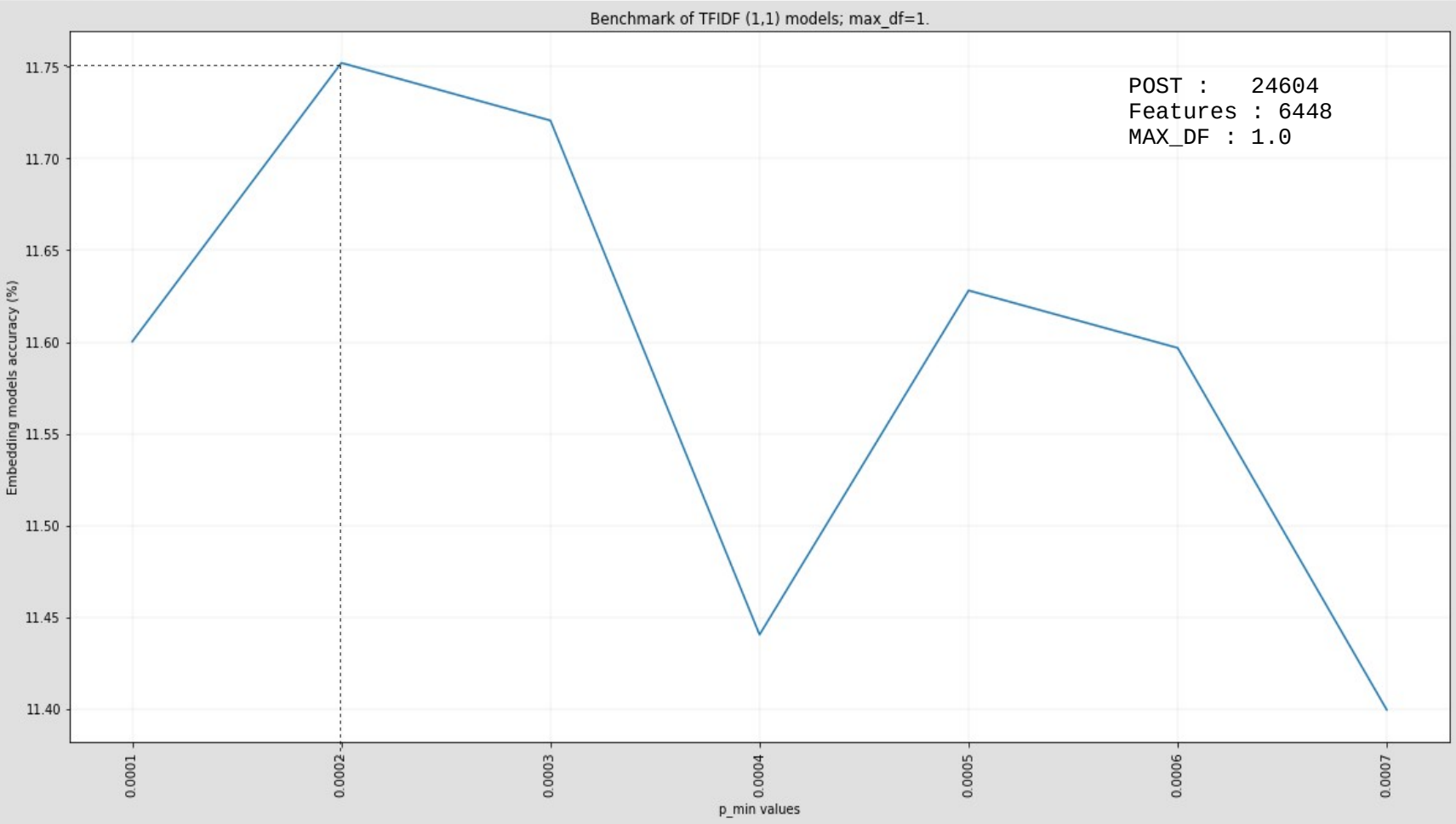


Accuracy measurement

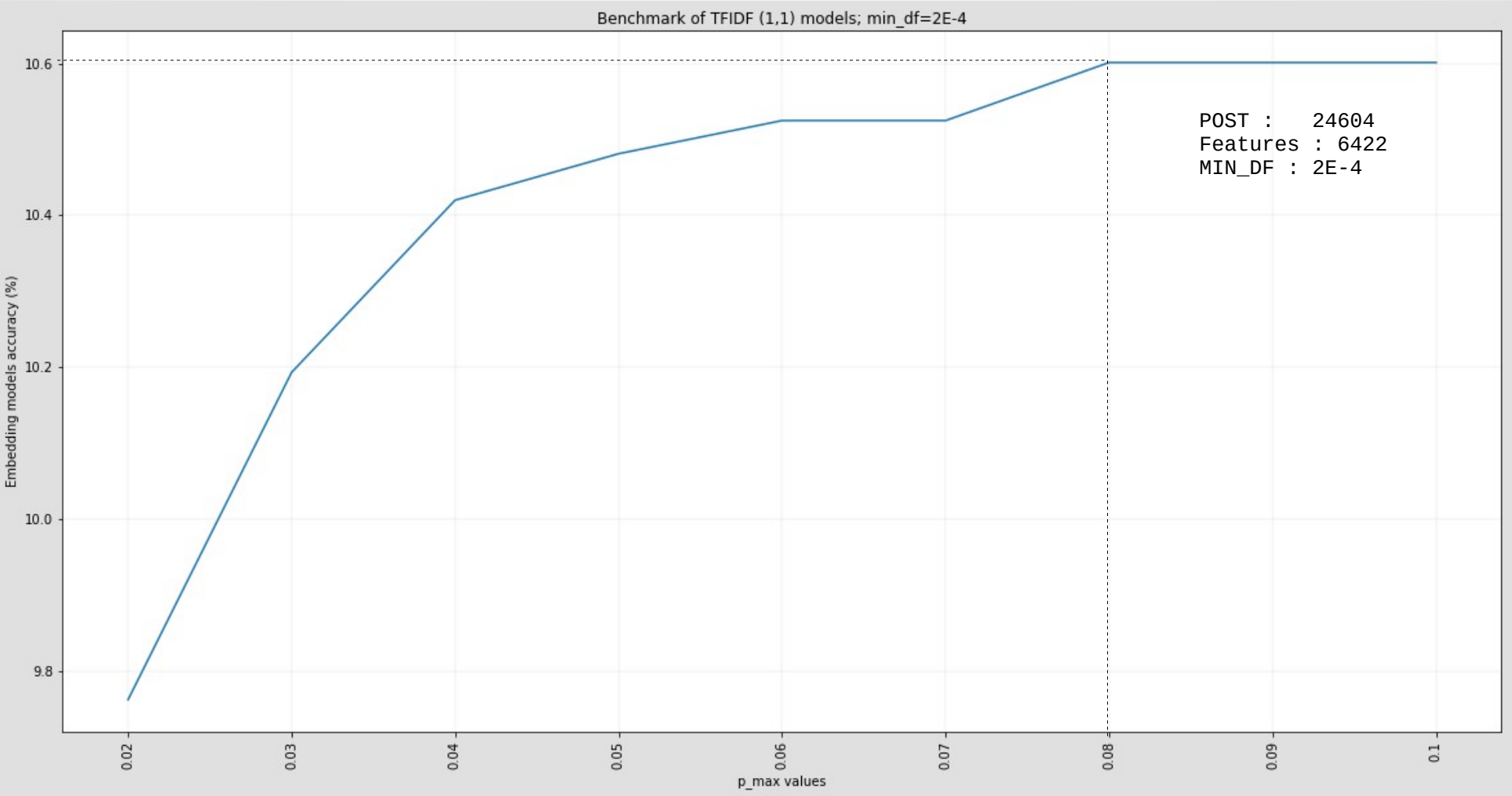
$$P_{POST} = \frac{NTAG_{SA}}{NTAG_A} = \frac{N\{TAG_A \cap TAG_S\}}{NTAG_A}$$

$$MAX_{Sample}\{NTAG_A\} = 6$$

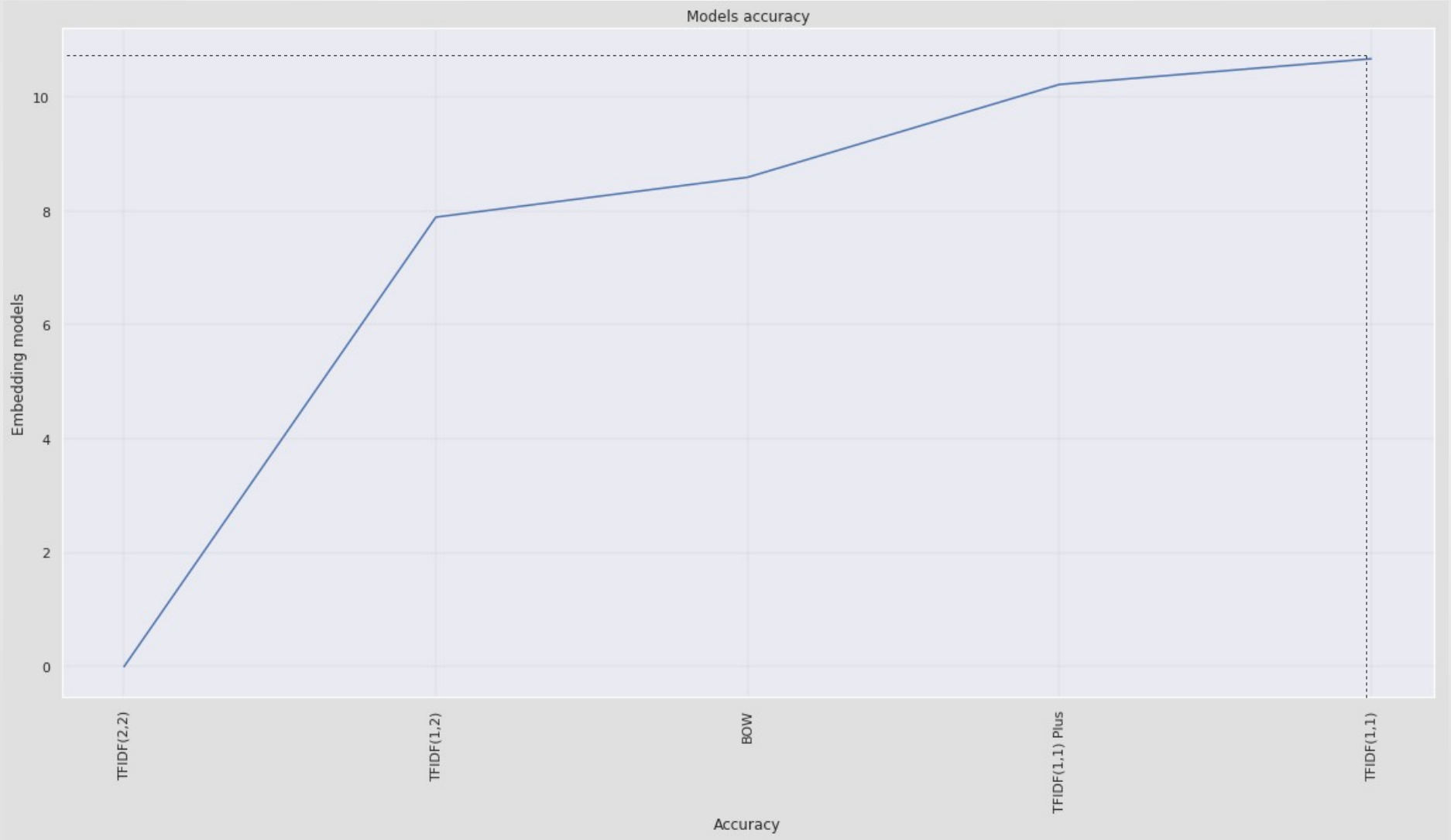
TF-IDF(1,1) : model selection % p_min



TF-IDF(1,1) : model selection % p_max



Statistical methods: 10 % posts evaluated

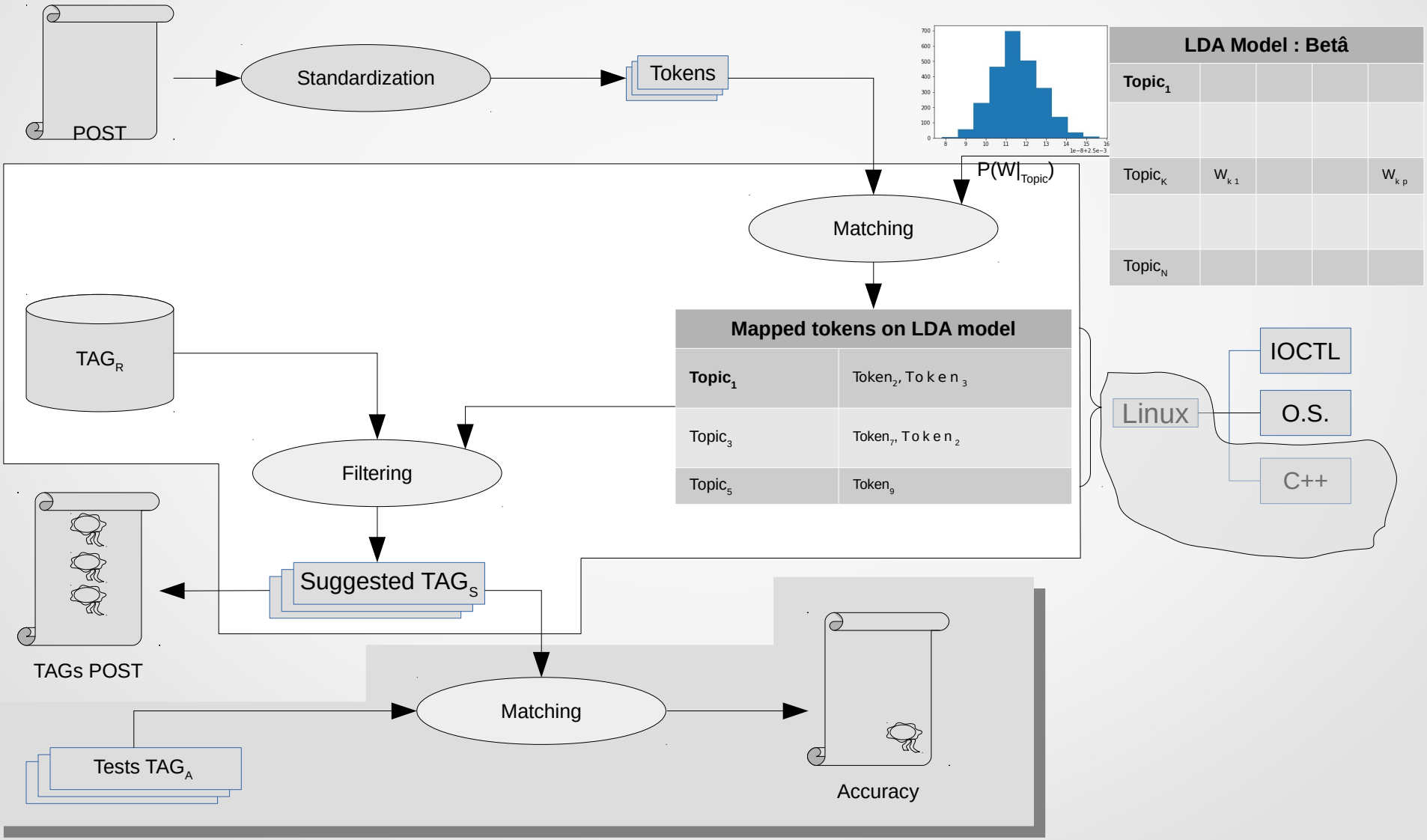


Benchmarking

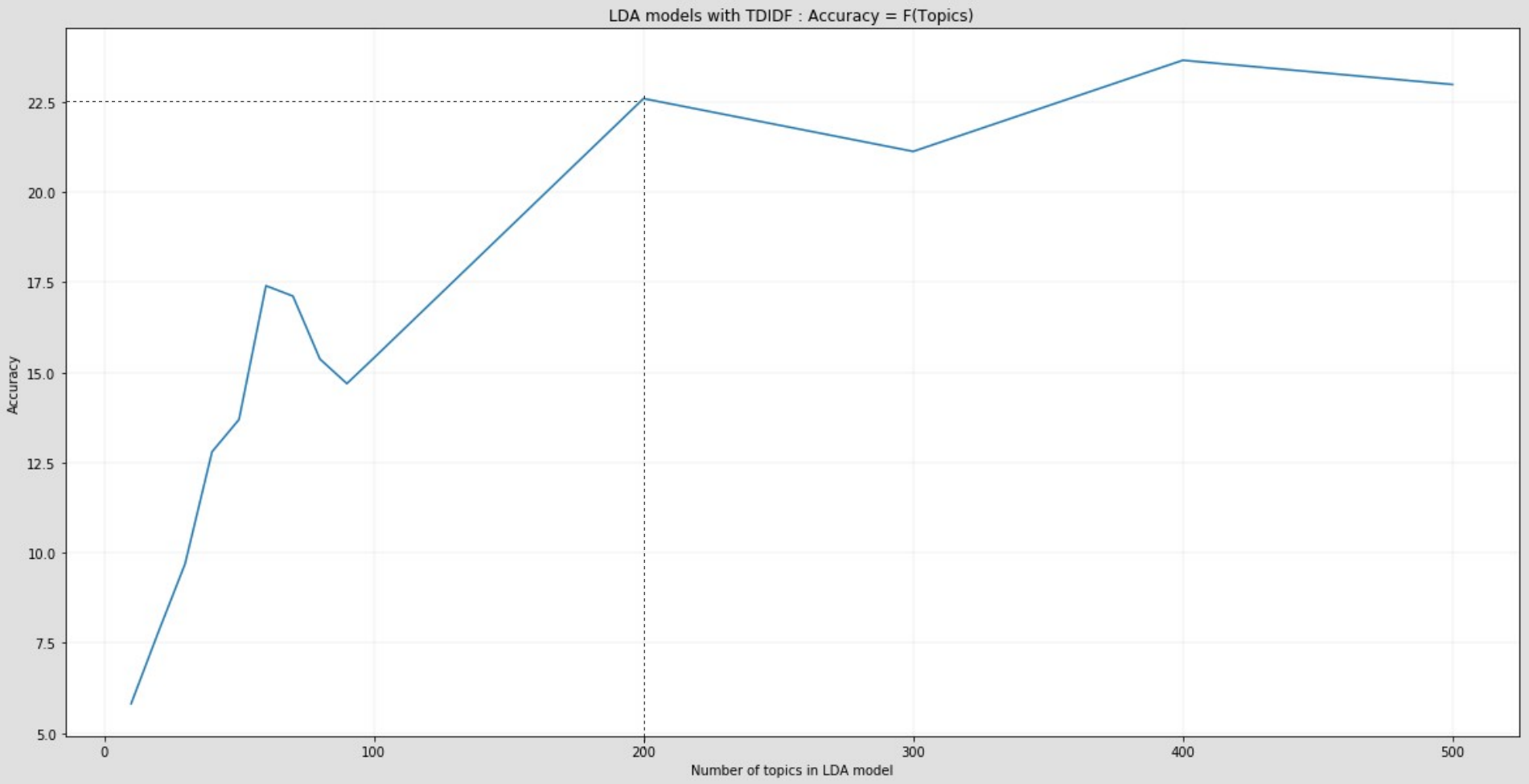
UNSUPERVIZED METHODS

Unsupervized method based on LDA

Prediction is based on words issued from topics

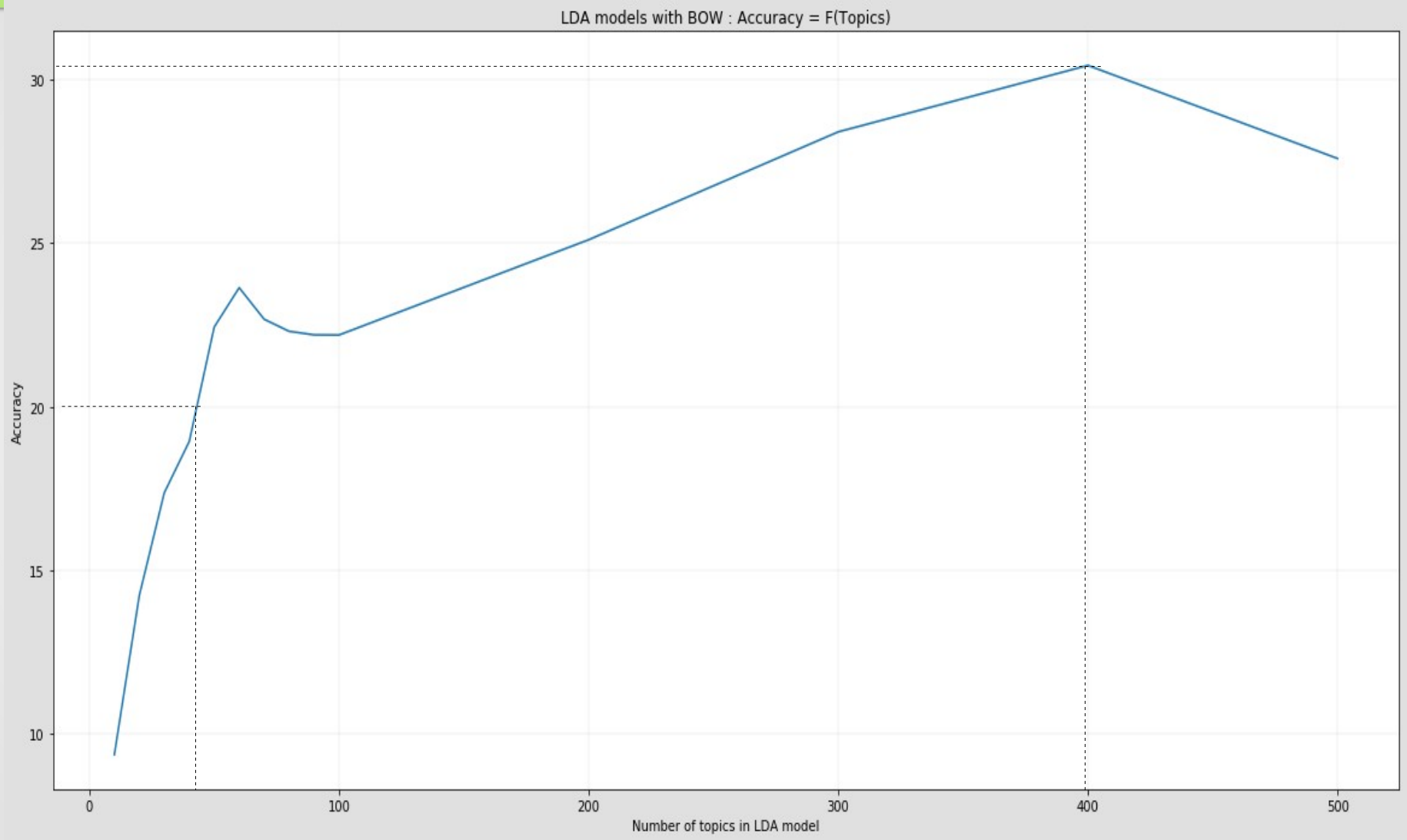


Mean accuracy of LDA with TF-IDF



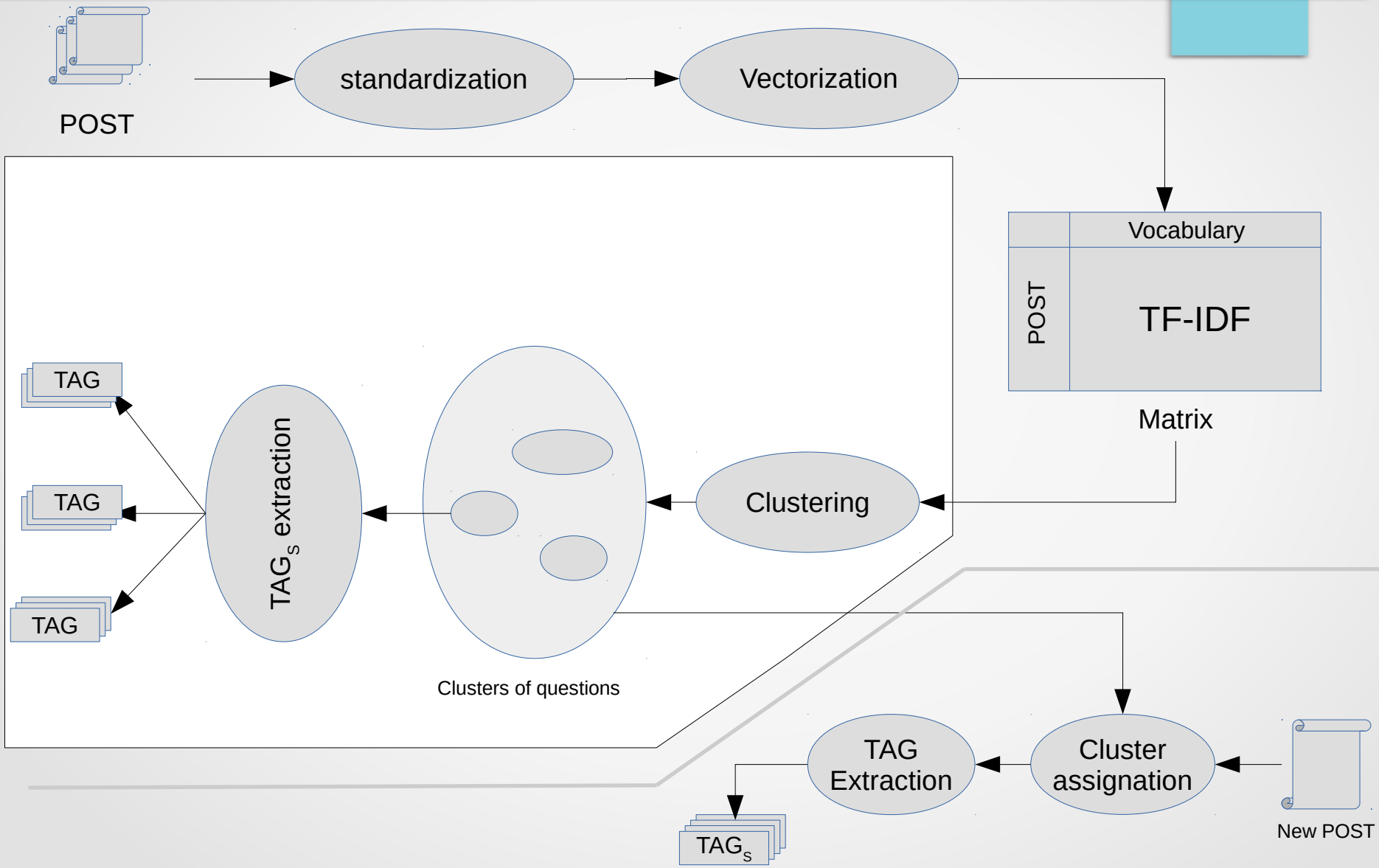
Best LDA model : 200 topics / 1000 POSTs per model : 20 % accuracy

Mean accuracy of LDA with BOW

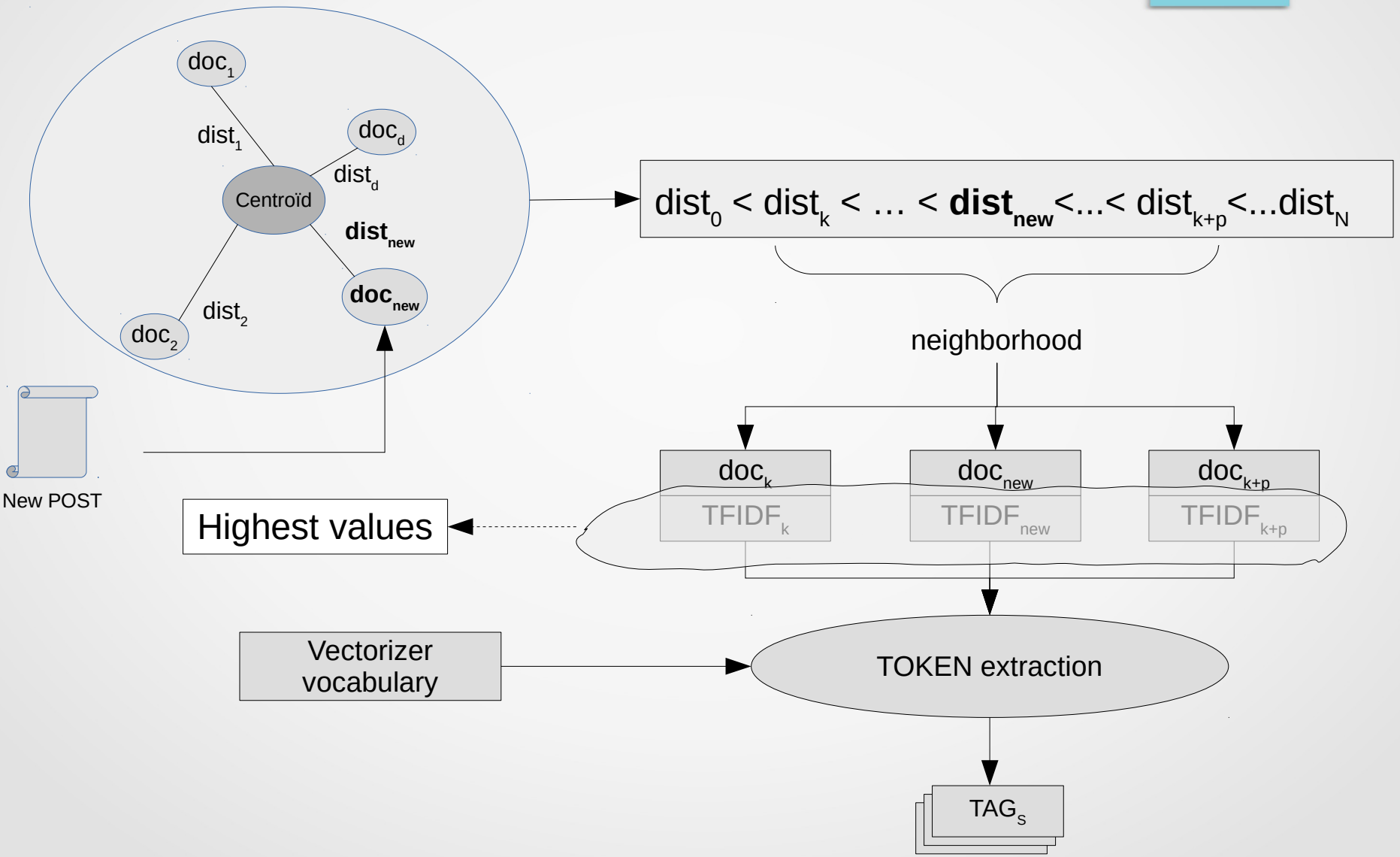


Best LDA model : 400 / 1000 POSTs per model : 30 % accuracy

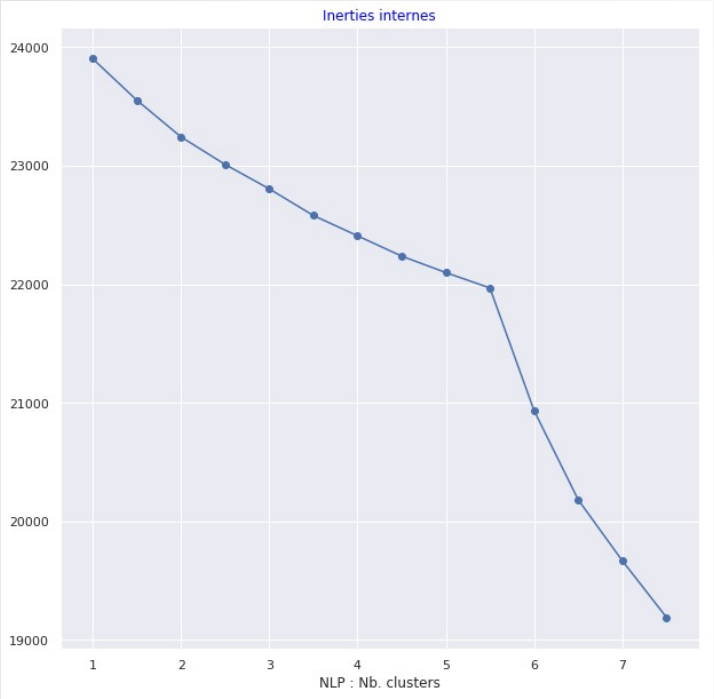
Unsupervized methods based on clustering

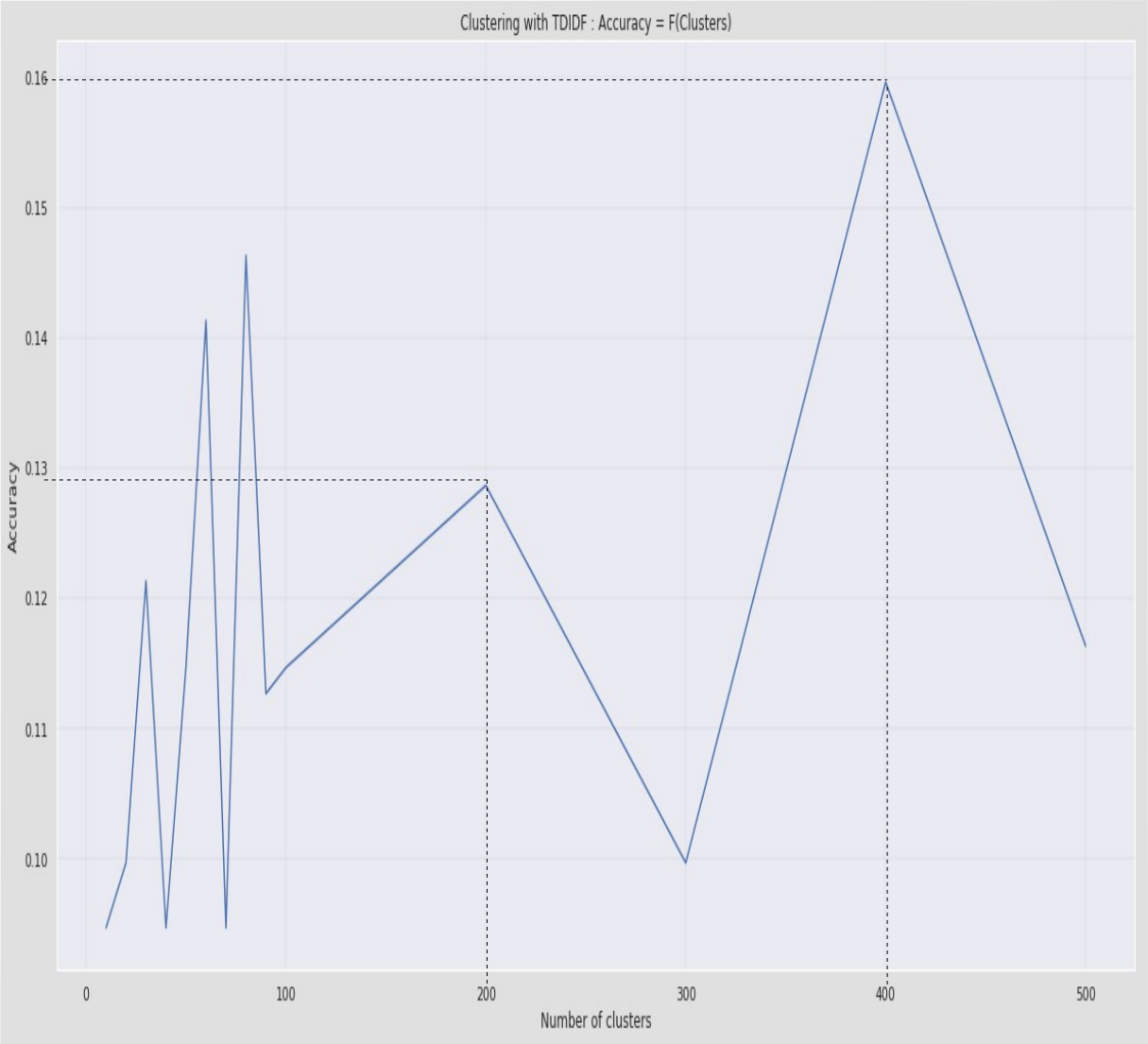


TAG extraction from a Kmeans cluster



Inter-cluster inertia: 10 to 500 clusters



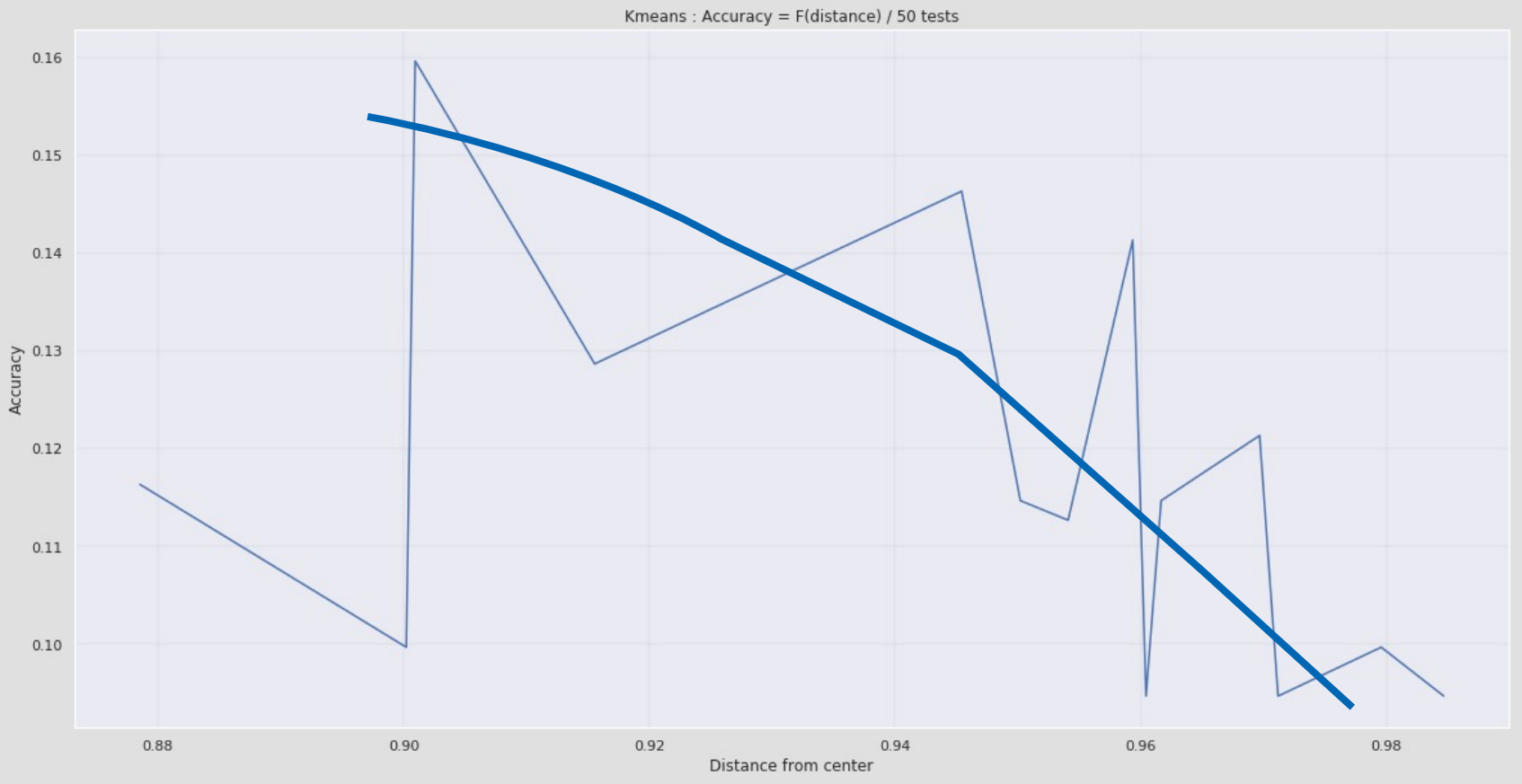


Kmeans / TF-IDF:

TF-IDF : min=2e-4, max=1.0
Dimension : ~4000

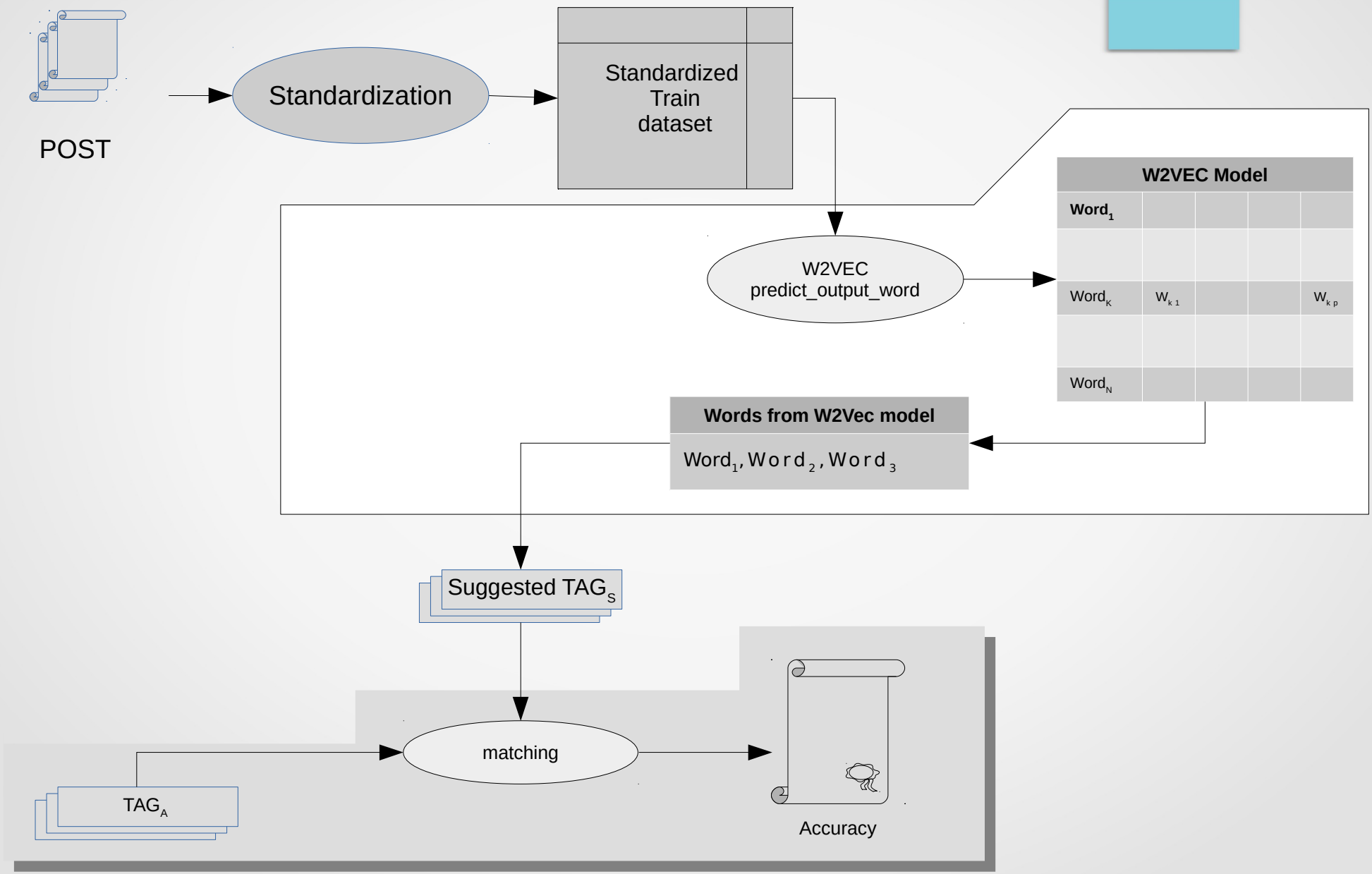
400 clusters / 15 %

Kmeans : Accuracy = F(Distance)

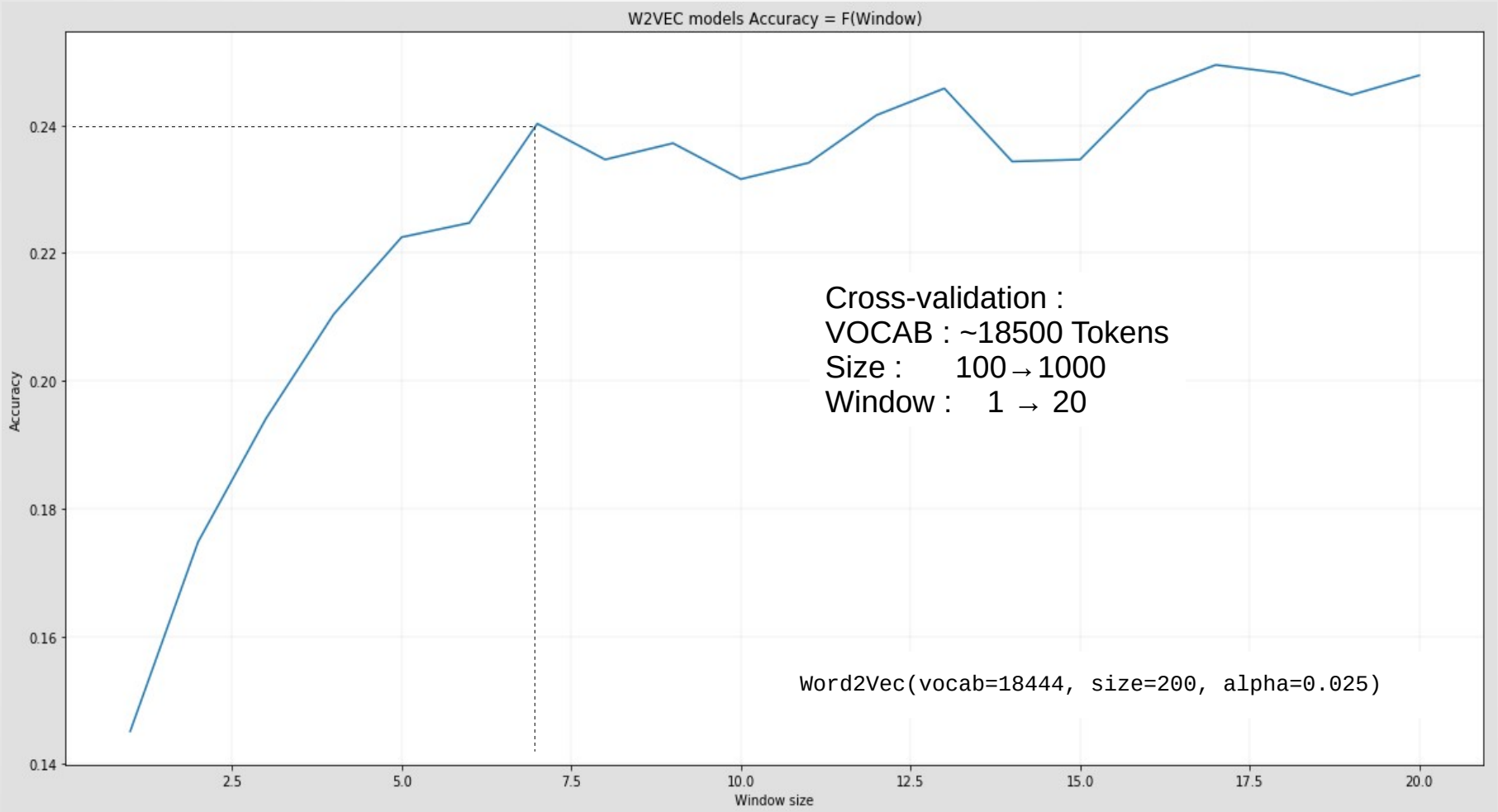


Globaly : accuracy decreases when distance from cluster center increases

Unsupervized method : W2VEC : skipgram

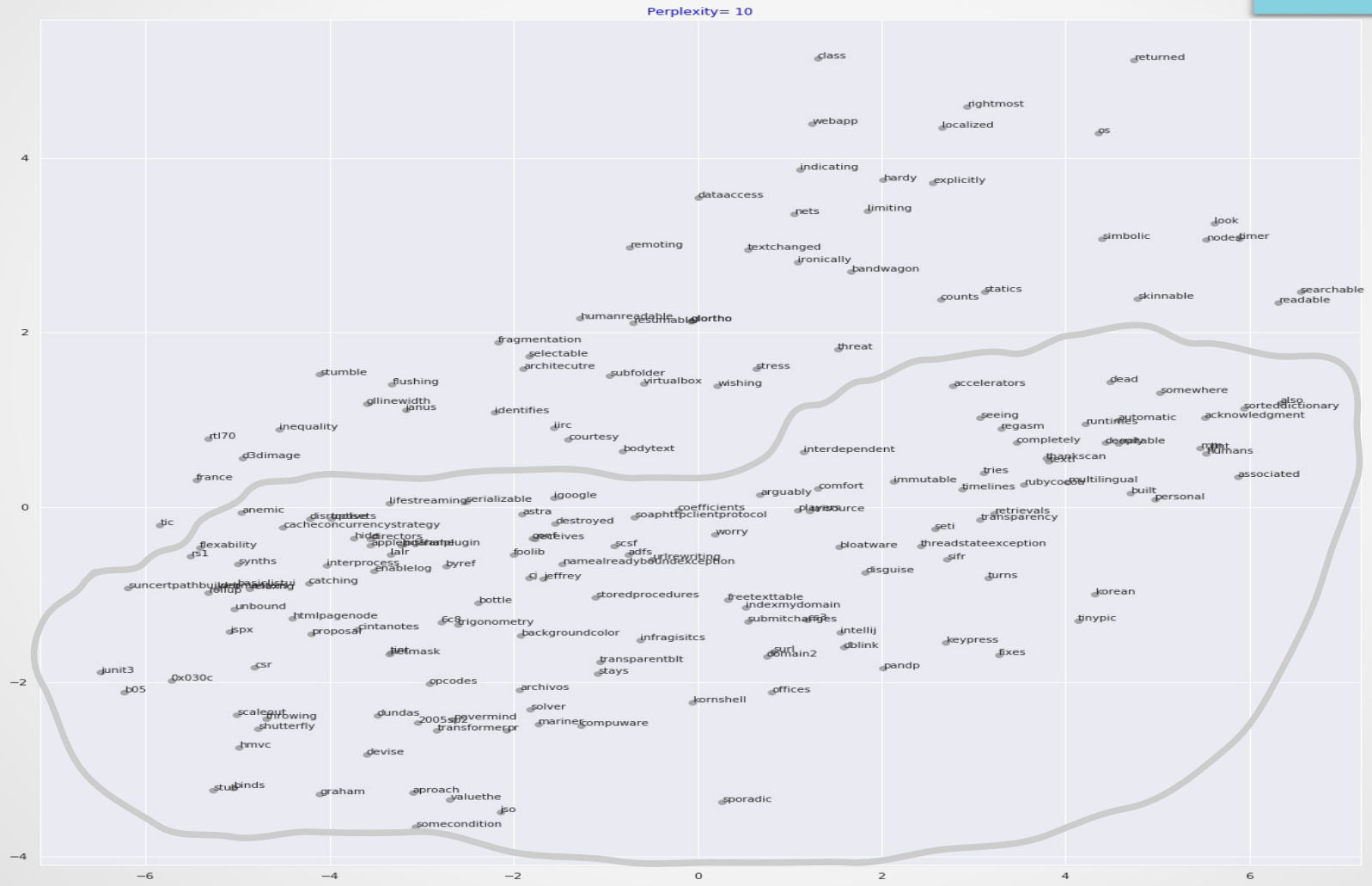


Mean accuracy of W2VEC / skip gram



Best W2VEC model : window=7 / 200 POSTs per model : 25% accuracy

t-SNE : 2D over W2VEC



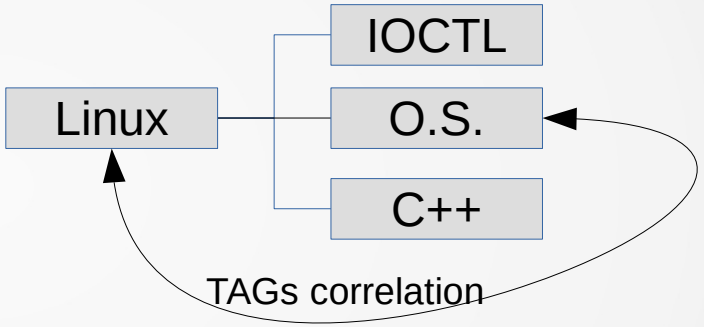
Benchmarking

SUPERVIZED METHODS

Supervized models : classification type

Multiple tags are assigned to one POST ← What we see

1 POST is crossed by multiple topics:



Multi-label classification issue

1 Tag ↔ 1 classifier

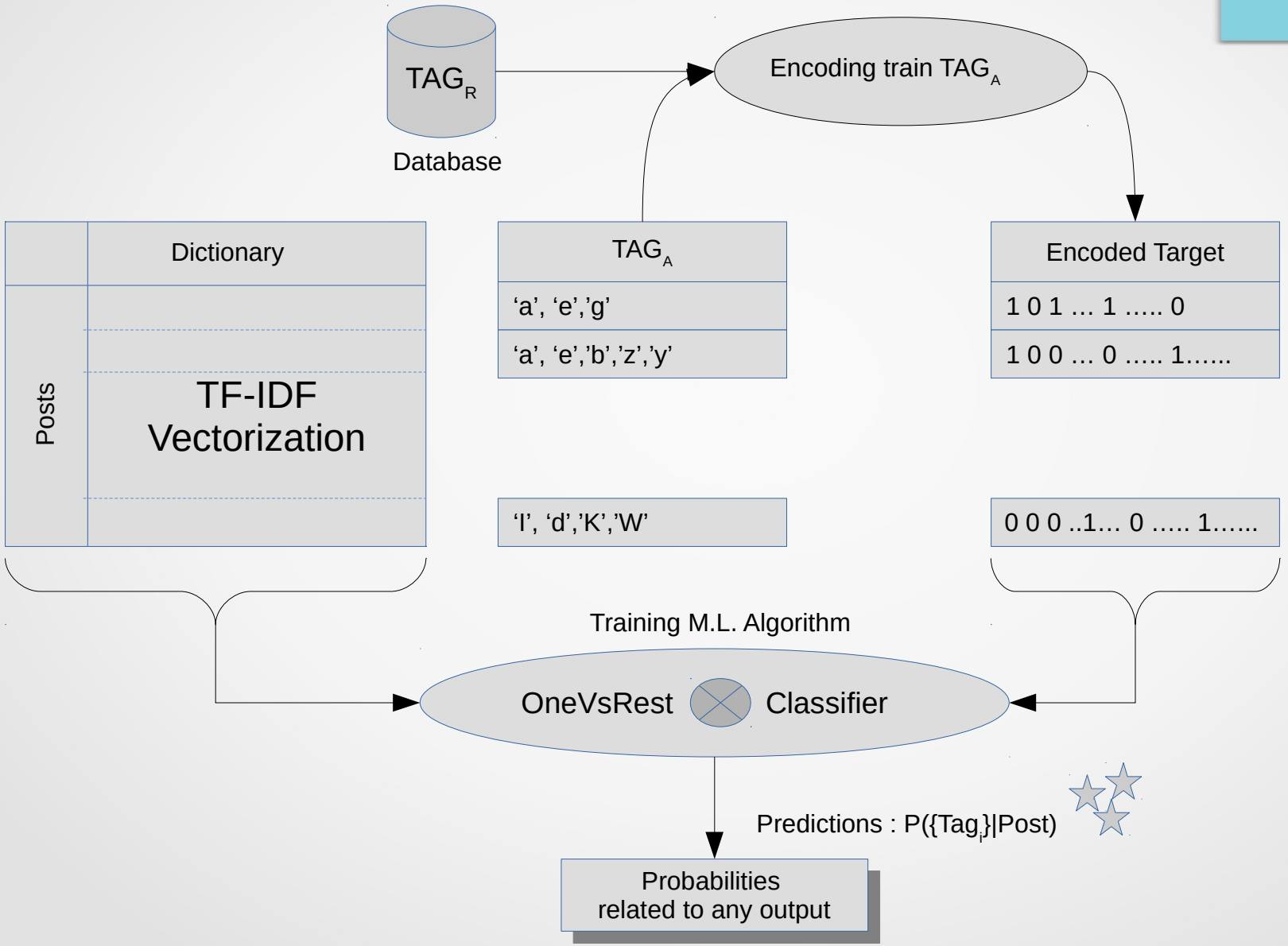
Results :

For each one of the classifiers :

- If 1 : Tag is assigned
- Else : Tag is ignored

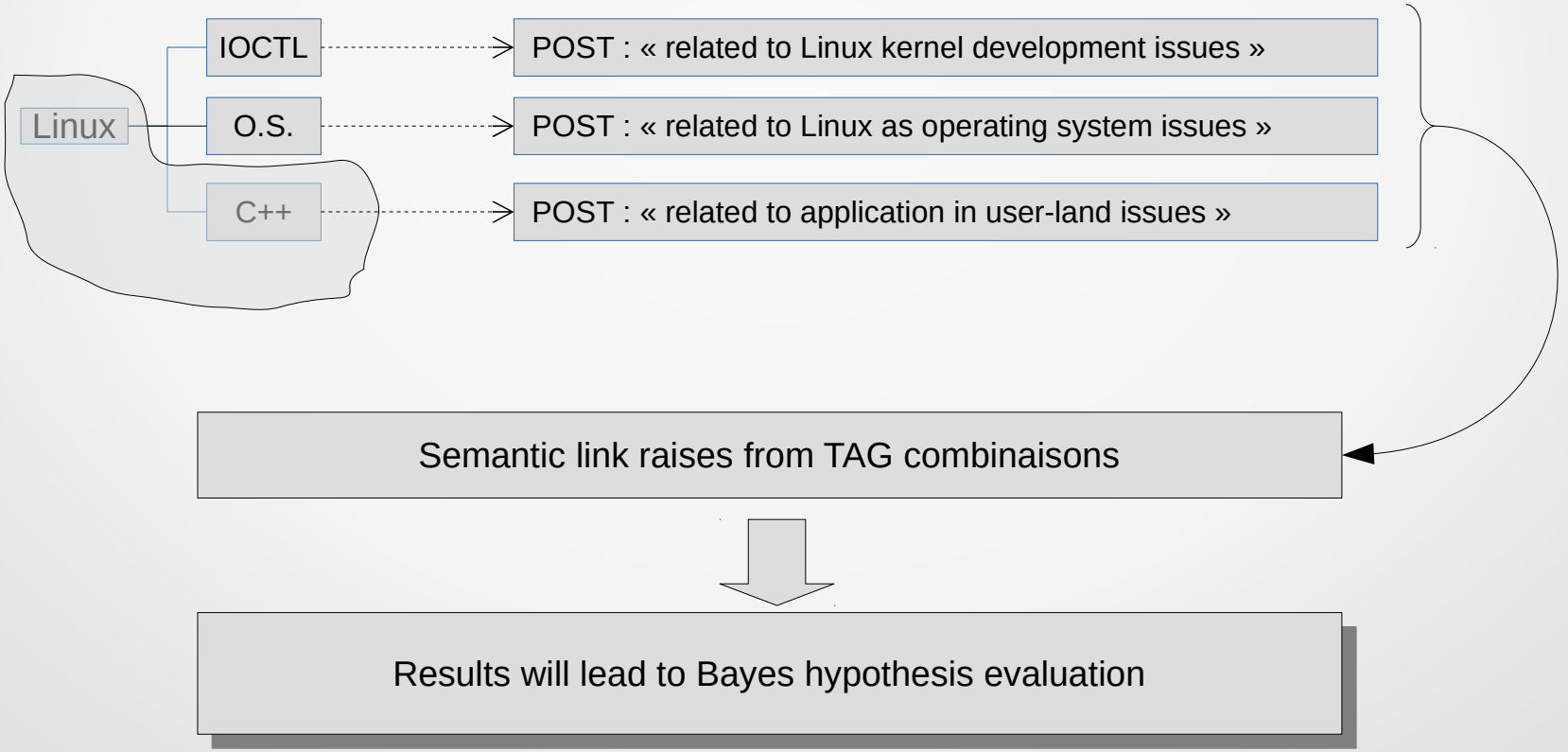
!! Binary approach : No correlation betweens Tags !!

Supervized algorithm : multinomial classifiers

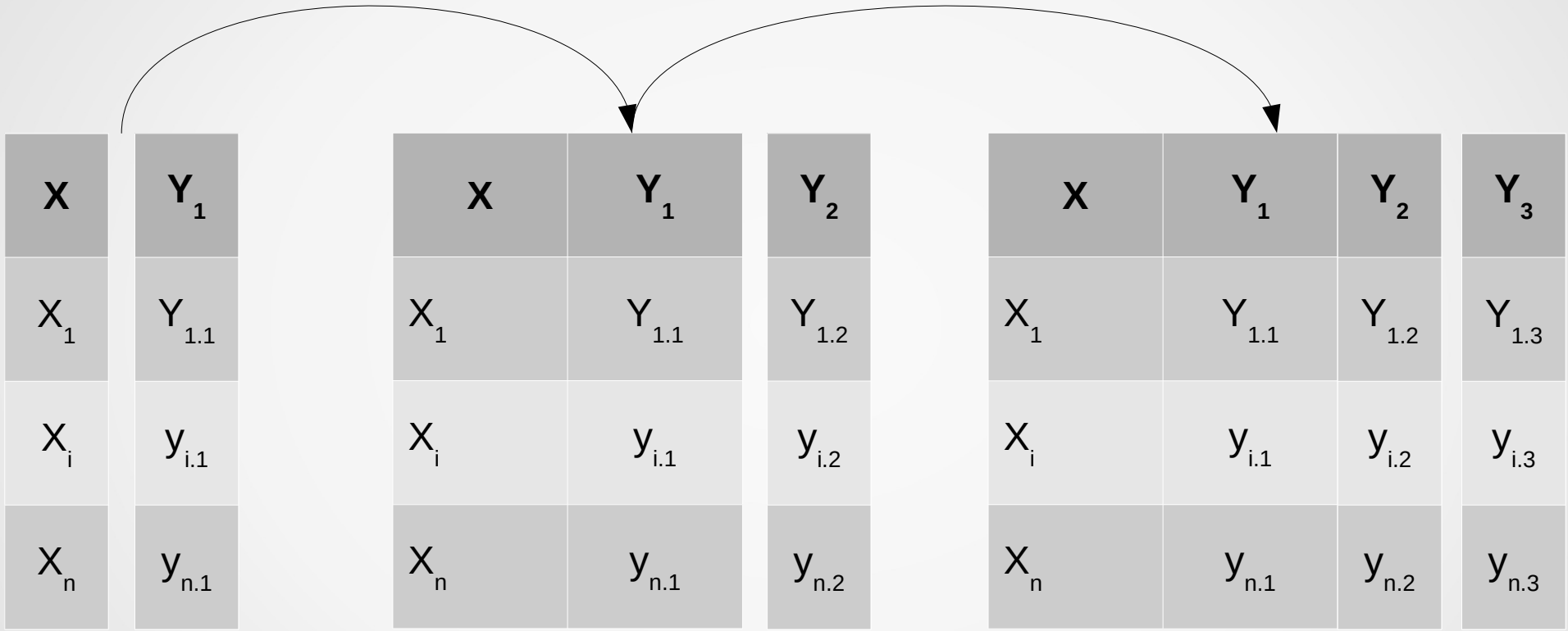


Bayes hypothesis

I.I.D : TAG are supposed to be independant from each other



Chained classifier



X	Y_1
X_1	$Y_{1.1}$
X_i	$y_{i.1}$
X_n	$y_{n.1}$

Classifier 1

X	Y_1	Y_2
X_1	$Y_{1.1}$	$Y_{1.2}$
X_i	$y_{i.1}$	$y_{i.2}$
X_n	$y_{n.1}$	$y_{n.2}$

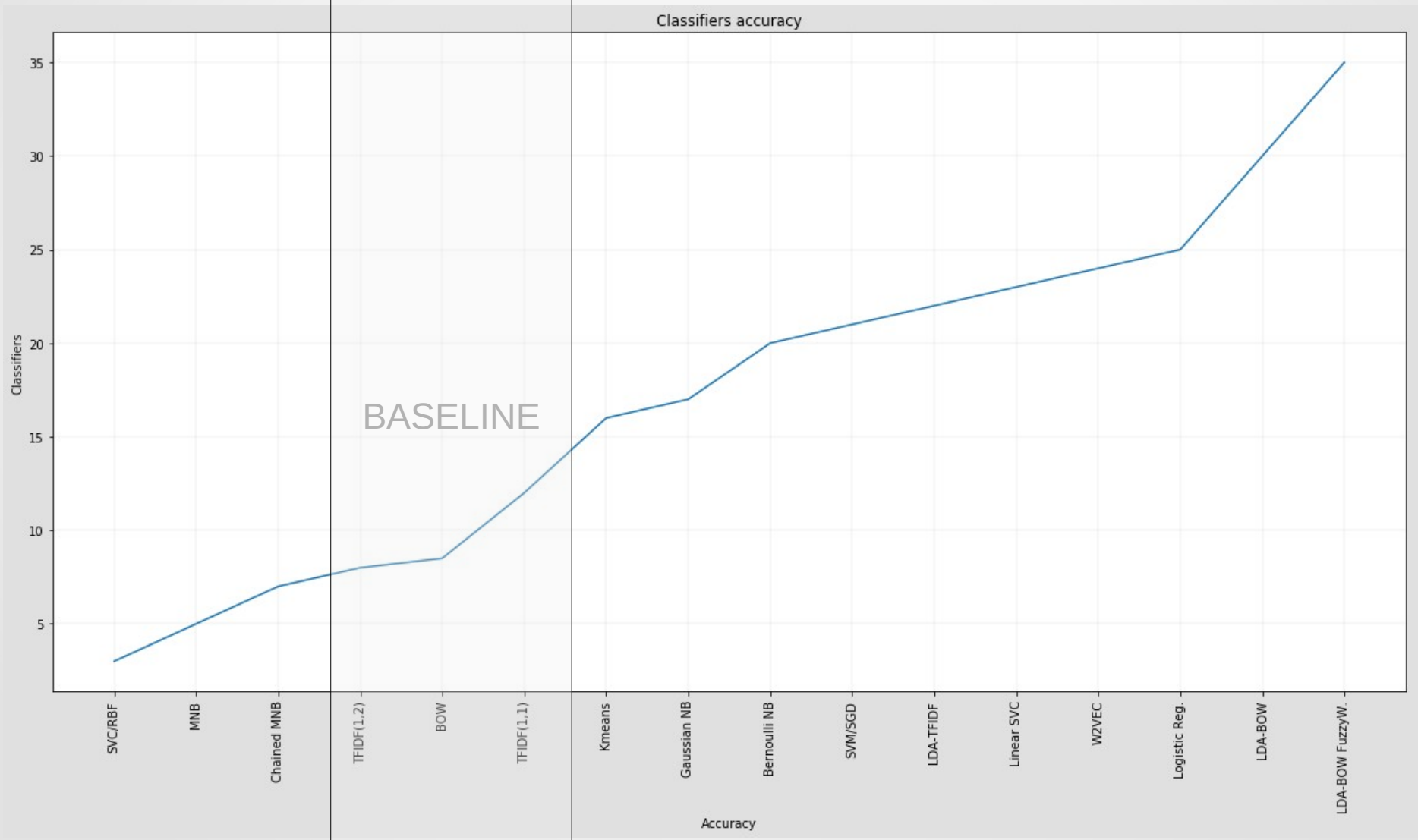
Classifier 2

X	Y_1	Y_2	Y_3
X_1	$Y_{1.1}$	$Y_{1.2}$	$Y_{1.3}$
X_i	$y_{i.1}$	$y_{i.2}$	$y_{i.3}$
X_n	$y_{n.1}$	$y_{n.2}$	$y_{n.3}$

Classifier 3

For any row : Labels correlation is saved

Global benchmark results



Conclusions & perspectives

- Résultats dépendants de la préparation des données (matching)
- Résultats dépendent de la mesure de précision
  - Alternative Extraction de TAG_S : Fuzzy-wuzzy
- Modèles supervisés : résultats compatibles avec la genèse des données.
  - Données improprement générées ⇒ classification idoine
- Mesures de similarité dans la SVM : kernel basé sur fuzzy-wuzzy
- Modèles non supervisés: insensibles aux TAG_A
  - W2VEC ou GLOVES : modèles experts
  - Qualification du benchmarking : calcul du recall, de la F-mesure
- Phases de data-preparation : la structure du texte non prise en compte (ponctuation)
- NLP : Quantité de POST utilisés limite la précision du modèle.

Annexe 1 : source files organization

Notebooks : notebook

- P6_DataAnalysis.ipynb : data analysis
- P6_Standardization.ipynb : data standardization
- P6_StatisticalMethod.ipynb : baseline
- P6_SupervizedMethods.ipynb : supervized methods
- P6_UnsupervizedMethods.ipynb : unsupervized meto=hods
- P6_ModelBuilder.ipynb : classifier builder and test

Python source files: src

- P6_PostClassifier.py : classifier implementation
- p6_util.py : functions for P6 project
- p6_util_plot.py : functions for plot
- p5_util.py : functions from P5 project used in P6 project
- p3_util.py : functions from P3 project used in P6 project

Report: report

- Openclassrooms_ParcoursDatascientist_P6-V1.pdf : presentation
- Rapport-P6.pdf : technical report

Annexe 2 : deployment

Classifier object :
- oP6_PostClassifier.dump : dumped formated classifier

Application directory
- tag_suggested

Launching Flask server :
- cd .../<projet>; python3.6 views.py;

API :
- http://127.0.0.1:5000?post_id=<number>
- http://127.0.0.1:5000?* : random ID for POST