IDL - Cours 3 - idl05_metrics_ML

On a vu :

- il existe plein de tâches de TAL
- il existe différentes méthodes

> **Question** : Comment évaluer un système de TAL pour une tâche X ?

❓ la métrique d'évaluation dépend-elle de la tâche ?

Donnez des exemples de tâches et de métriques.

▸ Details

Exemples de métriques

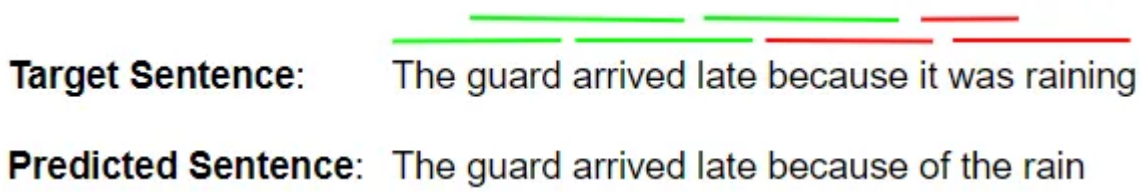
Trad : Métrique BLEU (bilingual evaluation understudy)

« plus une traduction automatique est proche d'une traduction humaine professionnelle, mieux c'est »

- fait un calcul d'exactitude sur des n-grams de mots (généralement de 1 à 4)

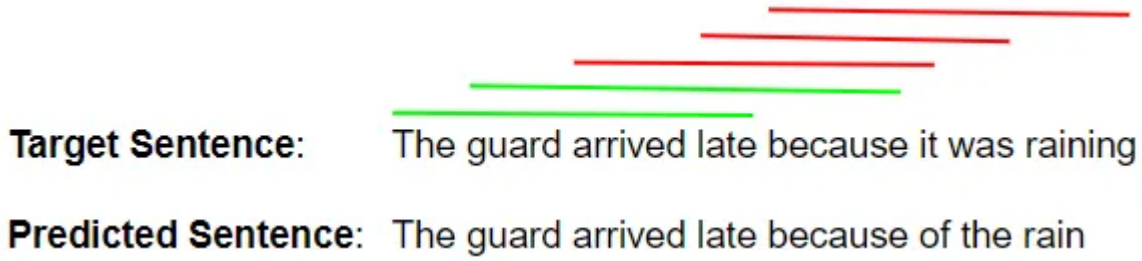
1-gram : comparaison des mots un à un

2-gram : comparaison des mots par groupe de deux



3-gram : … par groupes de 3

4-gram : ... par groupes de 4



- ne prend pas en compte l'intelligibilité
- ne prend pas en compte la correction grammaticale ou syntaxique

Utilisation :

```
from nltk.translate.bleu_score import corpus_bleu

references = [[['my', 'first', 'correct', 'sentence'], ['my', 'second',
candidates = [['my', 'sentence']]
score = corpus_bleu(references, candidates)
```

LM : la perplexité 🤔

Un modèle de langue est un modèle de probabilités sur des phrases.

Il est capable de :

- générer des phrases plausibles
- accorder une probabilité à une phrase qui lui est proposée

Si la phrase proposée est hors contexte, le modèle doit lui attribuer une faible probabilité, à l'inverse une phrase cohérente dans le contexte reçoit une forte probabilité.

La perplexité du modèle capture la capacité du modèle à être correctement "perplexe" face au phrases qui lui sont proposées.

Le calcul de perplexité repose notamment sur le calcul de probabilité d'apparition d'un certain élément sachant le contexte précédent.

-> vous voulez le calcul ?

Classif : combinaison de métriques

Annotation en parties du discours (morphosyntaxe)

Performances annoncées de Spacy : 95% (source (https://spacy.io/models/fr))

95%… de quoi ? => d'*accuracy*, c'est à dire l'exactitude.

❓ à quoi sert une évaluation ?

▸ Details

Let's check !

```
<sentence id='0'>
<mot type='NOUN'>Hymne</mot>$  ok
<mot type='ADP'>à</mot>   ok
<mot type='DET'>la</mot>   ok
<mot type='NOUN'>beauté</mot>   ok

</mot><mot type='PROPN'>Viens</mot>  nok
<mot type='PROPN'>-</mot>   nok
<mot type='VERB'>tu</mot>   nok
<mot type='ADP'>du</mot>   ok
<mot type='NOUN'>ciel</mot>   ok
<mot type='ADJ'>profond</mot>   ok
<mot type='CCONJ'>ou</mot>   ok
<mot type='ADV'>sors</mot>   nok
<mot type='NOUN'>-</mot>   nok
<mot type='VERB'>tu</mot>   nok
<mot type='ADP'>de</mot>   ok
<mot type='DET'>l'</mot>   ok
<mot type='NOUN'>abîme</mot>   ok
<mot type='PUNCT'>,</mot>   ok
</mot><mot type='NOUN'>Ô</mot>   nok
</sentence>
```

19 mots, 12 corrects, 7 incorrects.
Exactitude calculée : NB_correct / NB_total = 63%.

> => on écrit à SpaCy pour leur dire de mettre à jour leur valeur ?

Un autre cas d'étude : État de l'art automatique 🌈

L'évaluation dont vous êtes le héros / l'héroïne
Dans le cadre de votre projet tuteuré, vous utilisez un système qui sélectionne automatiquement pour vous dans une base de donnée les articles pertinents pour votre état de l'art.

- Vous lancez l'algo sur votre base de données de 100 articles
- L'algo vous renvoie 7 articles
- Vous décidez de vérifier quand même manuellement le résultat (6 sont en effet pertinents)

Quelle est la valeur/pertinence de cet outil ?

(**au tableau**) schéma de la classification / matrice de confusion / Précision / Rappel

❓ À votre avis, vaut-il mieux :

- une plus grande précision 🎯
- un plus grand rappel : 👤 👤 👤 👤 => ✷✷✷✷ ?

Plus grande précision = peu de *bruit*
Plus grand rappel = peu de *silence*

▸ Details
On utilise la F-mesure (moyenne harmonique de la précision et du rappel) comme mesure unique, adaptable selon les besoins :

$$\text{F1-score} = 2.\frac{\text{Precision}.\,\text{Recall}}{\text{Precision} + \text{Recall}}$$

Retour à l'annotation en POS

Précision = qualité NB correct / NB attribué
P = (4/6 (NOUN) + 3/3 (ADP) + 2/2 (DET) + 0/2 (PROPN) + 0/2 (VERB) + 1/1 (ADJ) + 1/1 (CCONJ) + 0/1 (ADV) + 1/1 (PUNCT) + 0/0) / 9 = 63%

Rq: ici on ne prend pas en compte la catégorie "PRON" qui aurait dû apparaître.

Lorsque tout est annoté, exactitude et précision sont confondues, car le nombre d'éléments annotés est égal au nombre d'éléments total.

Rappel = NB correct classe i / NB attendu classe i

R = (4/4 (NOUN) + 3/3 (ADP) + 2/2 (DET) + 0/2 (PROPN) + 0/2 (VERB) + 1/1 (ADJ) + 1/1 (CCONJ) + 1/2 (PUNCT) + 0/2 (PRON)) / 9 = 61%

F-mesure = (2 * P * R) / (P + R) = 0.62

❓ Toutes les erreurs se valent-elles ?

- *le chat*/VERB *mange la souris*. (NOUN attendu) vs *il est fatigué*/VERB. (ADJ attendu)
- catégories plus ou moins fréquentes

Préparation TP de demain : identification de langue

- corpus parallèle 22 langues ~ 10 000 mots chacuns

1. méthode des mots les plus fréquents
2. méthode des n-grams les plus fréquents (des 1-grams au 4-grams)
3. avec du machine learning (supervisé)

❓ quel est le type de tâche qu'on essaye de résoudre ici ?

❓ de quoi a-t-on besoin ?

▸ Details

Vectoriseur

`Sklearn Count Vectorizer` (source (https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html))

Converts a collection of text documents to a matrix of token counts.

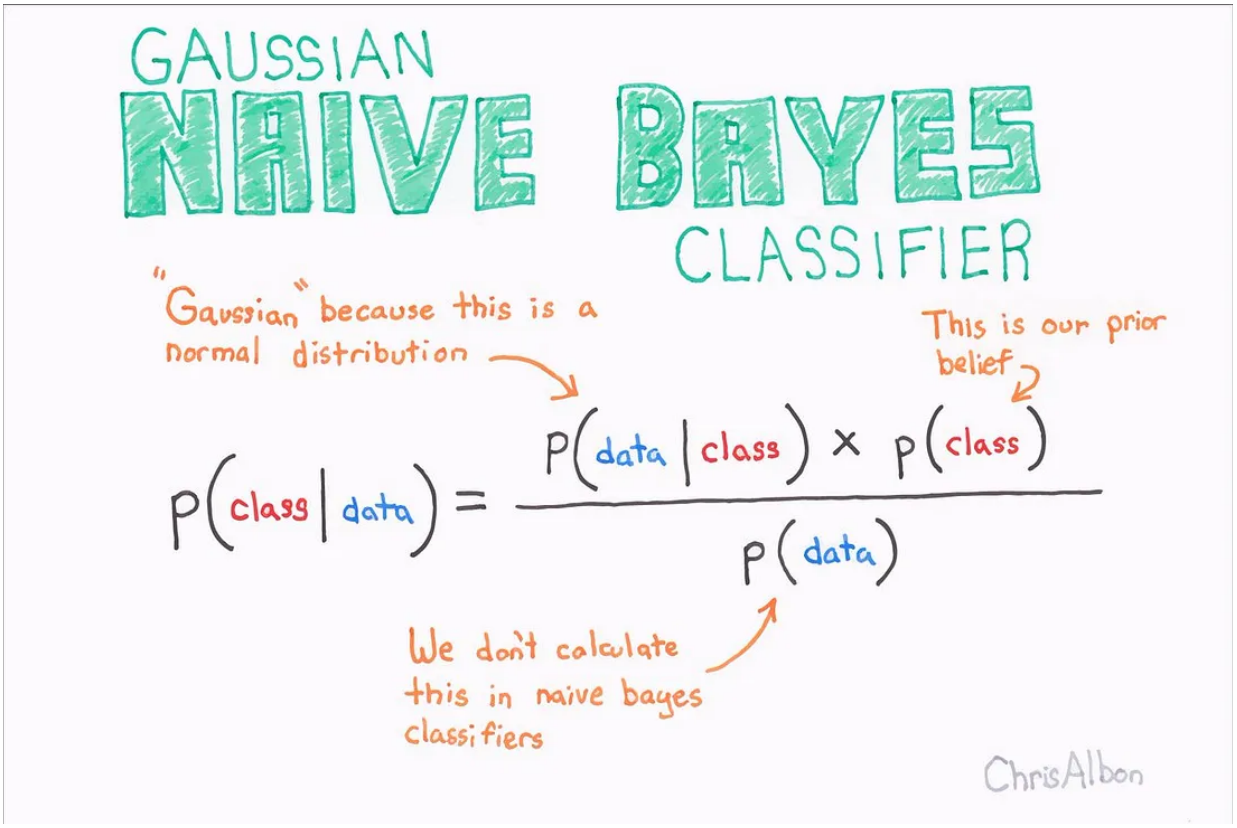
(**démo code**)

On peut ajuster le nombre de *features* ou dimensions données à nos matrices

Classifieur

`GaussianNB` (source (https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html))

Nourri avec : la réprésentation matricielle des corpus + l'output désiré !


GAUSSIAN
NAIVE BAYES
CLASSIFIER
"Gaussian" because this is a normal distribution
This is our prior belief
P(class | data) = P(data | class) × P(class) / P(data)
We don't calculate this in naive bayes classifiers
ChrisAlbon