

Catégorisation des questions/réponses

XU Yizhou JIANG Chunyang



10 avril 2019

Plan

- ➊ Introduction
- ➋ Données
- ➌ Ingénierie de caractéristiques
- ➍ Sélection de classifieurs
- ➎ Paramétrage de classifieurs
- ➏ Évaluation
- ➐ Module de prédiction

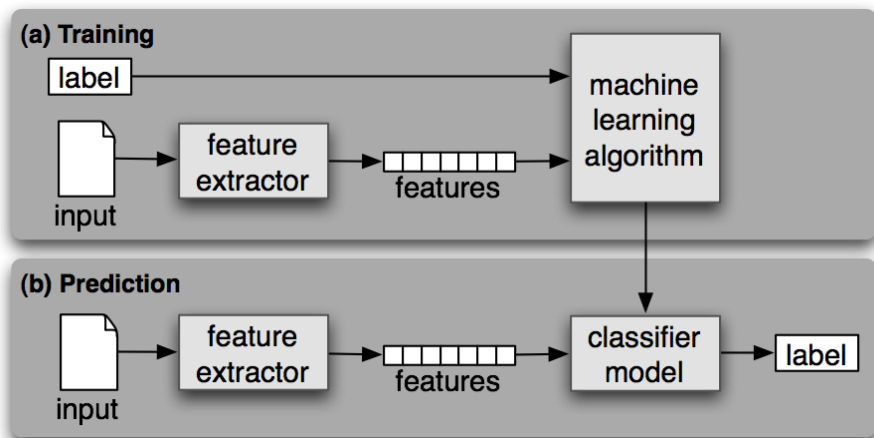


FIGURE – classification supervisée

(source : <http://www.nltk.org/>)

Plan

- ➊ Introduction
- ➋ Données
- ➌ Ingénierie de caractéristiques
- ➍ Sélection de classifieurs
- ➎ Paramétrage de classifieurs
- ➏ Évaluation
- ➐ Module de prédiction

Plan

- ① Introduction
- ② Données
- ③ Ingénierie de caractéristiques
- ④ Sélection de classifieurs
- ⑤ Paramétrage de classifieurs
- ⑥ Évaluation
- ⑦ Module de prédiction

Jeu de données

- **multi-classe (8)**

⇒ *immobilier (imm)*

⇒ *travail (trv)*

⇒ *entreprise (ent)*

⇒ *personne et famille (per)*

⇒ *finances, fiscalité et assurance (fin)*

⇒ *société (soc)*

⇒ *justice (jus)*

⇒ *internet, téléphonie et prop. intellectuelle (int)*

Jeu de données

- déséquilibre

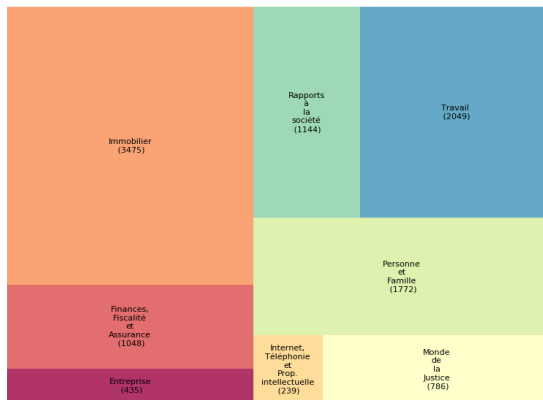


FIGURE – Distribution de classes

Jeu de données

- **Taille** : 12308 doc(questions/réponses)
- **Répartition** :
 - ▷ train 80%(9848)
 - ▷ test 20%(2460)

Plan

- ① Introduction
- ② Données
- ③ Ingénierie de caractéristiques
- ④ Sélection de classifieurs
- ⑤ Paramétrage de classifieurs
- ⑥ Évaluation
- ⑦ Module de prédiction

Caractéristiques

- Caractéristiques lexicales (sac de mots)
- Caractéristiques sémantiques (plongement)
- Caractéristiques d'apprentissage en profondeur
- Caractéristiques à base de règles

Caractéristiques

Sac de X

Mot (Token ou Unigram)

« notre », « syndic », « indique », « cette »

Lemma + POS

« notre/DET :pos », « syndic/NOM »,
« indiquer/VER :pres », « ce/DET :dem »

N-Gram+Lemma (Bigram, Trigram)

« notre syndic », « syndic indiquer », « indiquer ce »

Représentation

- présence/absence
- nombres d'occurrences (fréquence)
- **tf-idf**
- χ^2
- information mutuelle
- ...

Filtrage et Nettoyage

Outil : *sklearn.feature_extraction.text.TfidfVectorizer*

- mots vides et mots fréquents (*corpus-specific stop words*)

⇒ `max_df=0.7`

- mots rares (*cut-off*)

⇒ `min_df=5`

- accents

⇒ `strip_accents='ascii'`

Premature optimization is the root of all evil.

— Donald Knuth

Plan

- ① Introduction
- ② Données
- ③ Ingénierie de caractéristiques
- ④ Sélection de classifieurs**
- ⑤ Paramétrage de classifieurs
- ⑥ Évaluation
- ⑦ Module de prédiction

Classifieurs

- Baseline (BL)
- Random Forest (RF)
- Gradient Boosting (GB)
- Logistic Regression (LR)
- Naive Bayes (NB)
- Support Vector Machines (SVM)

scikit-learn

- ⇒ DummyClassifier
- ⇒ RandomForestClassifier
- ⇒ GradientBoostingClassifier
- ⇒ LogisticRegression
- ⇒ ComplementNB
- ⇒ LinearSVC

Plan

- ① Introduction
- ② Données
- ③ Ingénierie de caractéristiques
- ④ Sélection de classifieurs
- ⑤ Paramétrage de classifieurs
- ⑥ Évaluation
- ⑦ Module de prédiction

Paramétrage

- Outil : *sklearn.model_selection.GridSearchCV*
- Exemple :

Random Forest

- * `n_estimators` : 20,30,40,...,200 \Rightarrow **130**
- * `max_depth` : 10,20,30,...,100 \Rightarrow **60**
- * `min_samples_split` : 2, 5, 10 \Rightarrow **10**
- * `min_samples_leaf` : 1, 2, 4 \Rightarrow **2**

Plan

- ① Introduction
- ② Données
- ③ Ingénierie de caractéristiques
- ④ Sélection de classifieurs
- ⑤ Paramétrage de classifieurs
- ⑥ Évaluation**
- ⑦ Module de prédiction

Évaluation

	BL	RF	GB	LR	NB	SVM
Token	0.18	0.72	0.77	0.77	0.76	0.78
Lemma+POS	0.18	0.72	0.76	0.78	0.76	0.78
N-Gram	0.19	0.72	0.75	0.76	0.69	0.79

TABLE – Micro F1

- **Qu'est-ce un bon modèle (dans le cadre du projet) ?**
 - * Léger & rapide
 - ⇒ économie de mémoire RAM
 - ⇒ prévention de surcharge serveur
 - * Bonne F1-mesure

Modèle choisi

- * **Caractéristiques** ⇒ lemma + POS
- * **Classifieur** ⇒ SVM
- * **Taille** ⇒ 1,8 Mo
- * **F1-mesure (micro)** ⇒ 78%

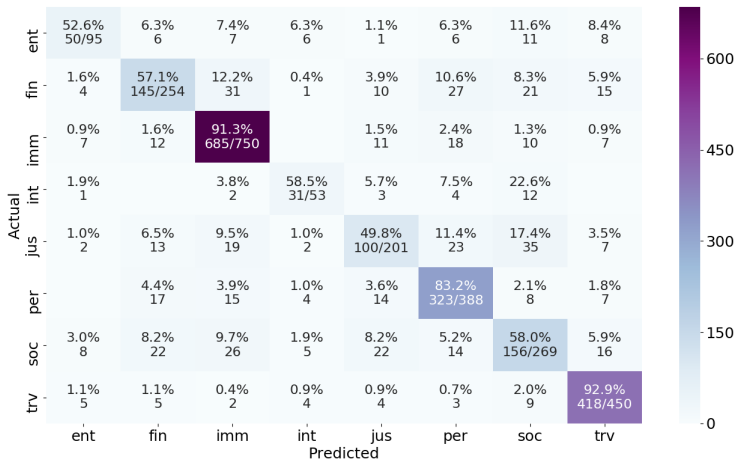


FIGURE – SVM - matrice de confusion

Plan

- ① Introduction
- ② Données
- ③ Ingénierie de caractéristiques
- ④ Sélection de classifieurs
- ⑤ Paramétrage de classifieurs
- ⑥ Évaluation
- ⑦ Module de prédiction

Module de prédiction

Scripts : *make_prediction.py* *test_make_prediction.py*

Fonctionnalités :

load_model

- * Chargement d'un modèle en mémoire
⇒ « **une fois pour toutes** »

make_prediction

- * Prétraitement de phrase en entrée
⇒ dépend des caractéristiques du modèle chargé
- * *Fit* et prédiction