Algorithmes de classification, data mining et text mining

- M1 SID
- 2018-2019
- J. G. Moreno et Y. Pitarch

Pourquoi le prétraitement est-il important ?

Prime Minister **Theresa May** has jetted out to Davos for crunch talks with top bankers following a landmark speech this week in which she set out her vision for Brexit.

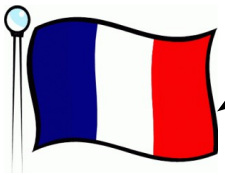
NNP,MD,VB ou MD,MD,VB if minuscules

May will meet chief executives from banks such as Goldman Sachs and JP Morgan a day after two other bosses – HSBC's Stuart Gulliver and UBS investment chief Andrea Orcel – confirmed that between them they would shift up to 2,000 jobs out of London when Britain leaves the EU.

Brexit will be top of the agenda when May hosts a roundtable including Goldman's Lloyd Blankfein and JP Morgan's Jamie Dimon, both of whom also landed in Switzerland yesterday. However, City experts do not believe there will be an exodus of banking jobs from London, despite the HSBC and UBS plans.

Représentation Granularité ?

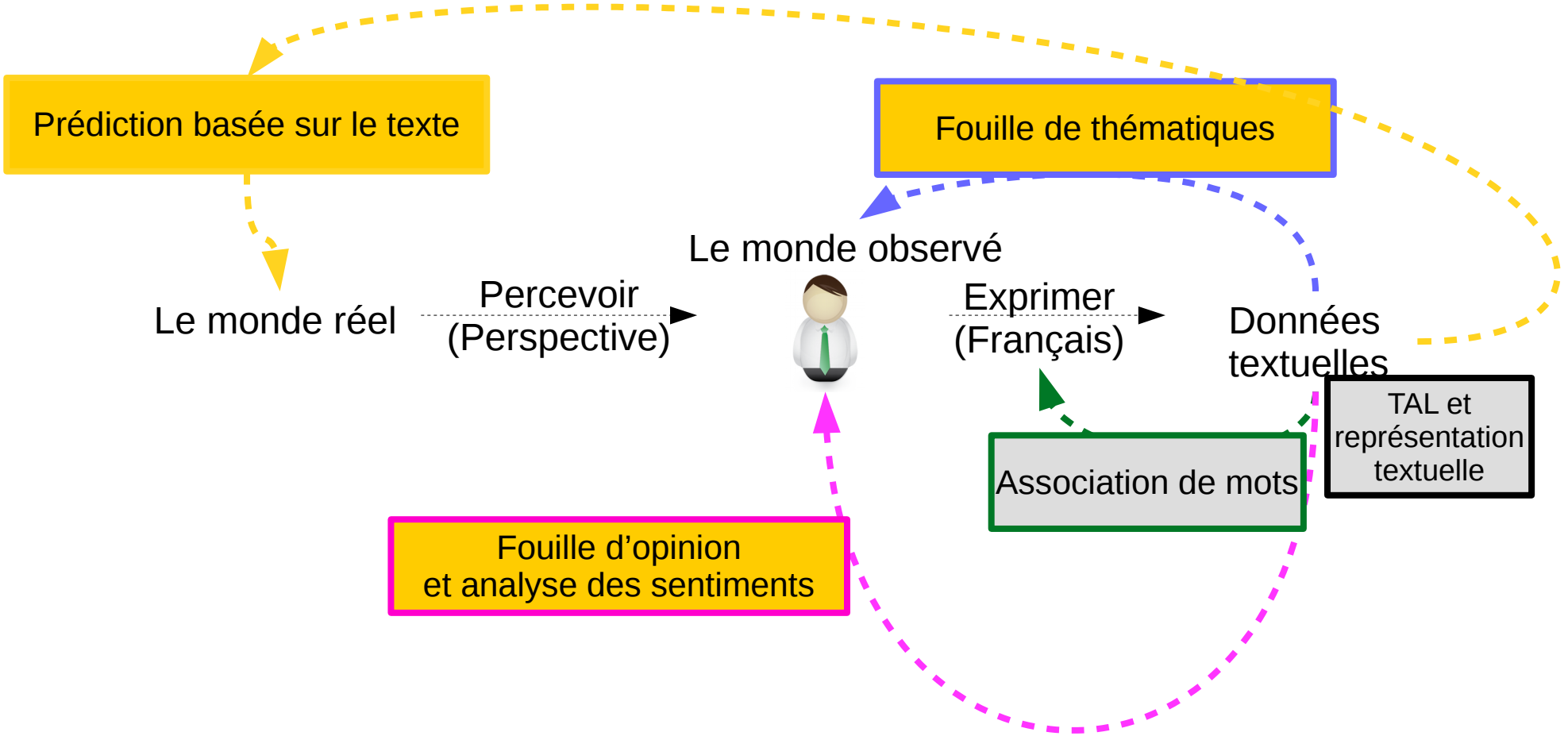
"It's all positioning, it's all playing to the gallery," added **Simon French**, chief economist at Panmure Gordon.



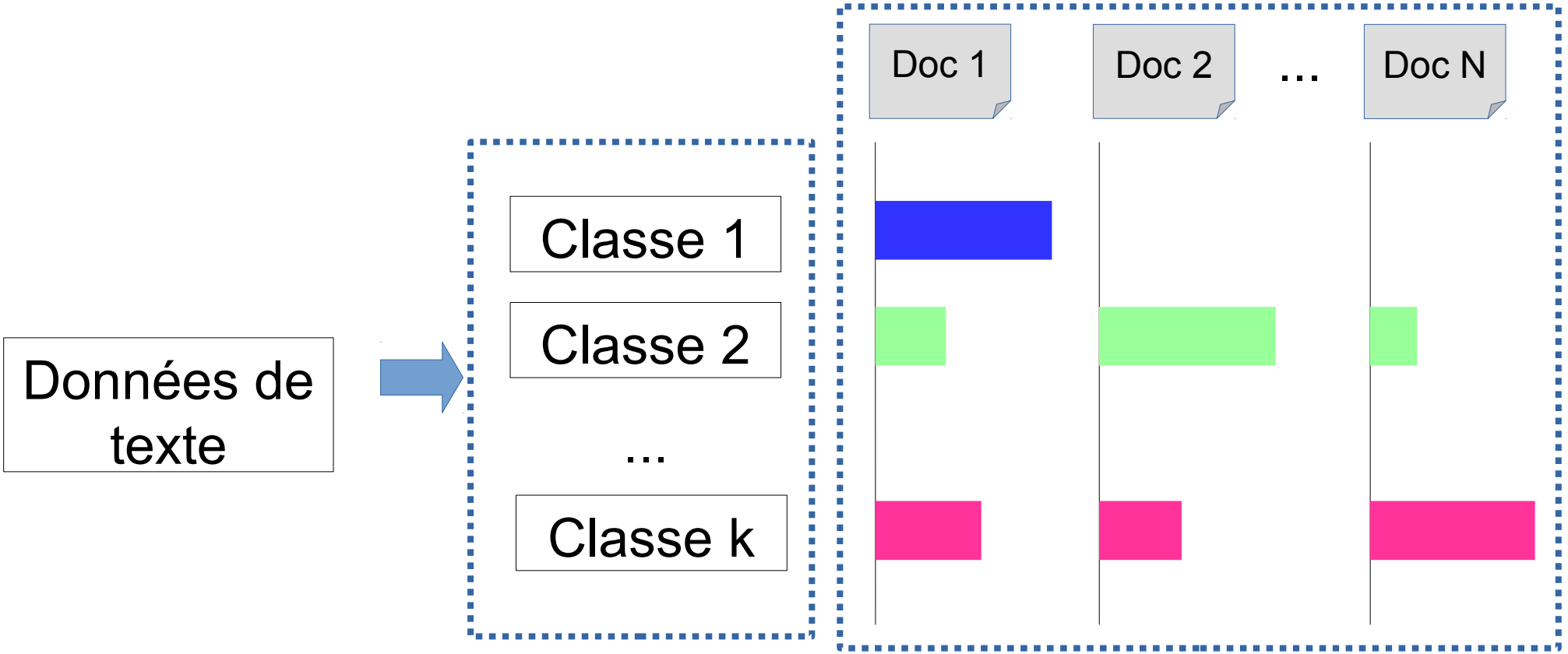
?

French said other banks are unlikely to make similar announcements – partly because they risk prompting staff to switch to rivals that guarantee permanent jobs in London.

Paysage de la fouille et de l'analyse de textes



Tâche de l'extraction et de l'analyse du sujet



Problème supervisé (Classification)

Exploration et analyse du sujet

- Groupe de documents (non supervisé)
  - Regroupement
- **Groupe de documents mais en classes (supervisé)**
  - **Classification ou catégorisation**
- ? Groupe de mots - réduction de la dimensionnalité (Nous allons les voir dans la partie non supervisée)
  - NNMF, LDA, PLSA, etc.
  -

Classification ou catégorisation du texte

- Compte tenu de
  - Un ensemble de catégories prédéfinies, formant éventuellement une hiérarchie
  - Un ensemble d'apprentissage des texte étiquetés
- Tâche : Classer un objet texte dans une ou plusieurs catégories

Exemples de catégorisation de texte

- Les objets texte peuvent varier (p. ex. documents, passages textes, phrases, etc.).
- Les catégories peuvent également varier
  - Catégories internes qui caractérisent un objet texte (p. ex. catégories thématiques, catégories de sentiments)
  - Catégories externes qui caractérisent une entité associée à l'objet (p. ex., attribution d'auteur ou toute autre catégorie significative associée aux données textuelles)
- Quelques exemples d'applications
  - Catégorisation des nouvelles, catégorisation de la littérature (p. ex., annotations MeSH)
  - Détection/filtrage des courriels non sollicités
  - Catégorisation du sentiment des commentaires sur les produits ou des tweets
  - Tri/routage automatique des courriels
  - Attribution de l'auteur

Variantes de la formulation du problème

- Catégorisation binaire : seulement deux catégories
  - Recherche d'information (pertinente ou non)
  - Filtrage du spam (spam ou non)
  - Opinion (négative ou positive)
- Catégorisation en K catégories : plus de deux catégories
  - Catégorisation des sujets (sports, sciences, voyages, affaires, etc.)
  - Routage du courrier électronique (dossier 1, dossier 2, etc.)
- Catégorisation hiérarchique : catégories pour une hiérarchie
- Catégorisation conjointe : plusieurs tâches de catégorisation connexes exécutées conjointement.

Pourquoi la catégorisation du texte ?

- Enrichir la représentation textuelle (meilleure compréhension du texte)
  - Le texte peut maintenant être représenté en plusieurs niveaux (mots-clés+catégories)
  - Les catégories sémantiques attribuées peuvent être directement ou indirectement utiles pour une application.
  - Les catégories sémantiques facilitent l'agrégation du contenu textuel (p. ex. agrégation de toutes les opinions positives/négatives sur un produit).
- Permettre de déduire les propriétés des entités associées aux données textuelles (découverte de la connaissance du monde)
  - Tant qu'une entité peut être associée à des données texte, nous pouvons toujours utiliser les données texte pour aider à catégoriser les entités associées.
  - Par exemple, découverte de locuteurs non natifs d'une langue ; prédiction de l'appartenance à un parti en fonction d'un discours politique.

Méthodes de catégorisation : manuelle

- Déterminer la catégorie en fonction de règles soigneusement conçues pour refléter la connaissance du domaine sur le problème de catégorisation.
- Fonctionne bien quand
  - Les catégories sont très bien définies
  - Les catégories se distinguent facilement en fonction des caractéristiques de la surface du texte (p. ex., le vocabulaire spécial n'est connu que dans une catégorie particulière).
  - Une connaissance suffisante du domaine est disponible pour suggérer de nombreuses règles efficaces
- Problèmes
  - Intensité de main-d'œuvre -> ne passe pas à l'échelle
  - Impossible de gérer l'incertitude des règles ; les règles peuvent être incohérentes -> pas ou peu robustes
- Les deux problèmes peuvent être résolus/atténués en utilisant l'apprentissage automatique.

Méthodes de catégorisation : automatique

- Faire appel à des experts humains pour
  - Annoter les ensembles de données avec des étiquettes de catégorie -> données de formation
  - Fournir un ensemble de fonctions pour représenter chaque objet texte qui peut potentiellement fournir un "indice" sur la catégorie.
- Utiliser l'apprentissage machine
  - pour apprendre les "règles" permettant de séparer les différentes catégories.
  - Déterminer quelles caractéristiques sont les plus utiles pour séparer les différentes catégories.
  - Combiner de façon optimale les fonctionnalités pour minimiser les erreurs de catégorisation sur les données d'entraînement
  - Le classificateur entraîné peut ensuite être appliqué à un nouveau texte pour prédire la catégorie la plus probable (qu'un expert humain lui affecterait).

Apprentissage machine pour la catégorisation des textes

- Configuration générale : apprendre un classificateur f : X->Y
  - Entrée : X= tous les objets ; Sortie : Y toutes les catégories
  - Apprendre une fonction de classificateur, f : X->Y, de sorte que f(x)=y donne la catégorie correcte pour x (correcte - basée sur les données d'entraînement)
- Toutes les méthodes
  - S'appuyent sur les caractéristiques discriminantes des objets texte pour distinguer les catégories
  - Combinent plusieurs caractéristiques de manière pondérée
  - Ajustent les poids sur les caractéristiques pour minimiser les erreurs sur les données d'entraînement
- Les différentes méthodes ont tendance à varier dans les domaines suivants
  - Leur façon de mesurer les erreurs sur les données de formation (peut optimiser une autre fonction objectif/perte/coût)
  - Leur façon de combiner les caractéristiques (p. ex. linéaires ou non linéaires)

Classificateurs génératifs et discriminants

- Les classificateurs génératifs (apprener à quoi ressemblent les données dans chaque catégorie)
  - Tenter de modéliser p(X,Y)=p(Y)P(X|Y) et calculer p(Y|X) à partir de p(X|Y) et p(Y) en utilisant la règle de Bayes
  - La fonction objective est la probabilité, ce qui permet de mesurer indirectement les erreurs de formation
  - Naive Bayes
- Les classificateurs discriminants (apprenez quelles sont les caractéristiques des catégories distinctes)
  - Tentative de modélisation directe de p(Y|X)
  - La fonction objective mesure directement les erreurs de catégorisation sur les données de formation
  - Régression logistique, machines à vecteurs de support (SVM), k-plus proches voisins (k-NN), réseaux de neurones, etc.

• Catégorisation de texte avec Naïve Bayes

- Considérer chaque catégorie indépendamment comme une classe *c* (pour les classes multiples).
  - Exemple d - document
  - Caractéristique w - mot ou terme

$$score(c) = \log \frac{P(c|d)}{P(\sim c|d)} = \sum_{w \in d} \log \frac{P(w|c)}{P(w|\sim c)} + \log \frac{P(C)}{P(\sim C)}$$

  - Classez *c* si *score(c)> θ*
    - Généralement un seuil spécifique pour chaque classe, en raison de l'inexactitude de l'estimation probabiliste de *P(d|c)* avec des statistiques d'entraînement et une hypothèse d'indépendance données,
    - ... mais une estimation biaisée de la probabilité pour *c* peut tout de même bien correspondre à la décision de classification.

Algorithme de l'apprentissage par les plus proches voisins

- L'apprentissage consiste simplement à stocker les représentations des exemples de formation dans l'ensemble de données D
- Pour la collection de test :
  - Calculer la similarité entre x et tous les exemples en D
  - Attribuer à x la catégorie des exemples les plus similaires
- Ne calcule pas explicitement une généralisation ou une catégorie de prototypes (c.-à-d. pas de "modélisation").
- Aussi appelé :
  - Basé sur des cas
  - Basé sur la mémoire
  - Apprentissage paresseux

Les K plus proches voisins pour le texte

- Entraînement :
  - Pour chaque exemple de formation <x, c(x)> Є D
    - Calculer le vecteur TF-IDF correspondant, dx, pour le document x

- Test pour y :
  - Calculer le vecteur TF-IDF, dy, pour le document y
  - Pour chaque <x, c(x)> Є D
    - Calculer sx = cos(dy, dx)
  - Trier les exemples, x, en D de la plus petite valeur à la plus grande
  - Utiliser les premiers k exemples en D (obtenir les voisins les plus proches)
  - Retourne la classe majoritaire d'exemples dans N

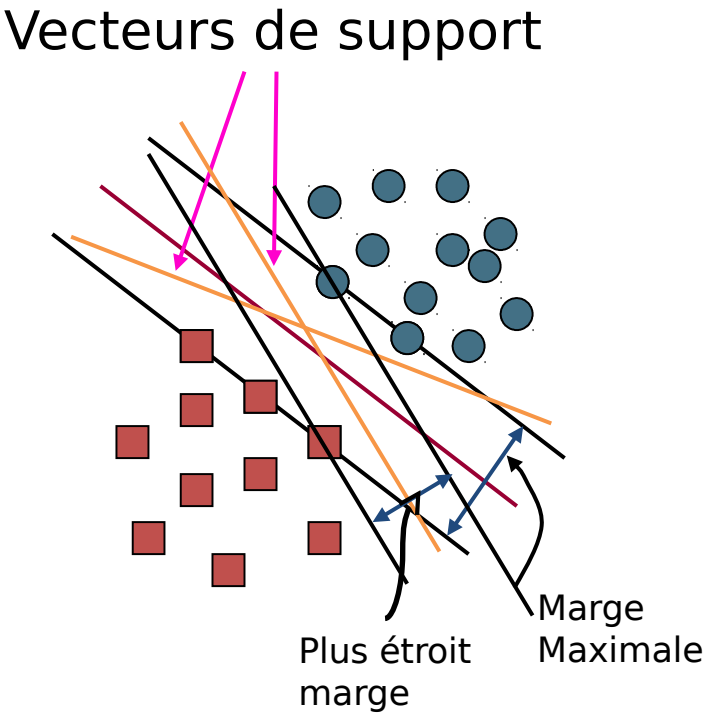
Simple mais puissant dans de très grandes collections !

Discussion KNN

- Aucune sélection de caractéristiques n'est nécessaire
- Aucune “entraînement” nécessaire
- S'adapte bien à un grand nombre de classes/documents
- Pas besoin d’avoir n classificateurs pour n classes
- Les classes peuvent s'influencer mutuellement
- De petits changements à une classe peuvent avoir un effet d'entraînement
- Fait naïvement, il peut être très cher au moment du test

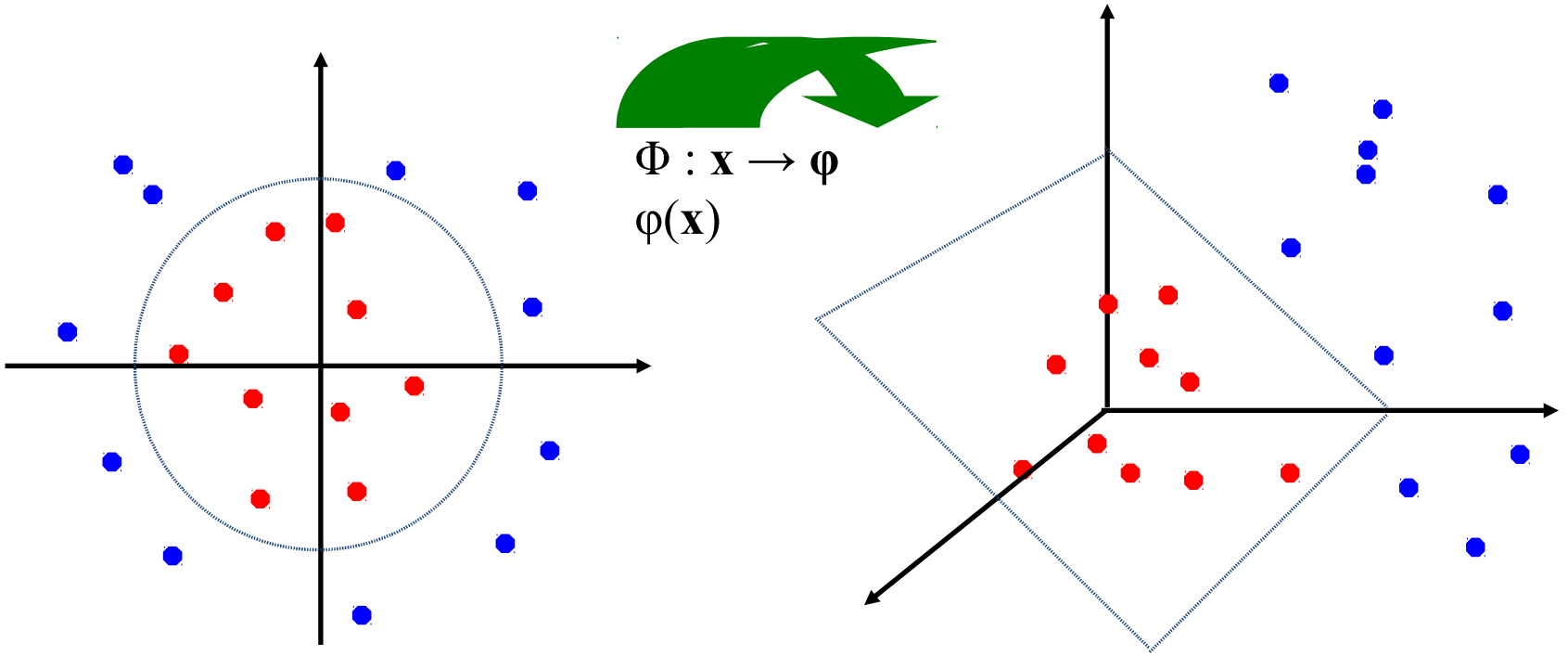
Machine à vecteurs de support (SVM)

- Les SVM maximisent la marge autour de l'hyperplan de séparation.
  - alias classificateurs de grosses marges
- La fonction de décision, les vecteurs de support, est entièrement spécifiée par un sous-ensemble d'échantillons d'entraînement.
- Résoudre les SVMs est un problème de programmation quadratique
- Considéré il n'y a pas longtemps comme la méthode de classification de texte la plus performante



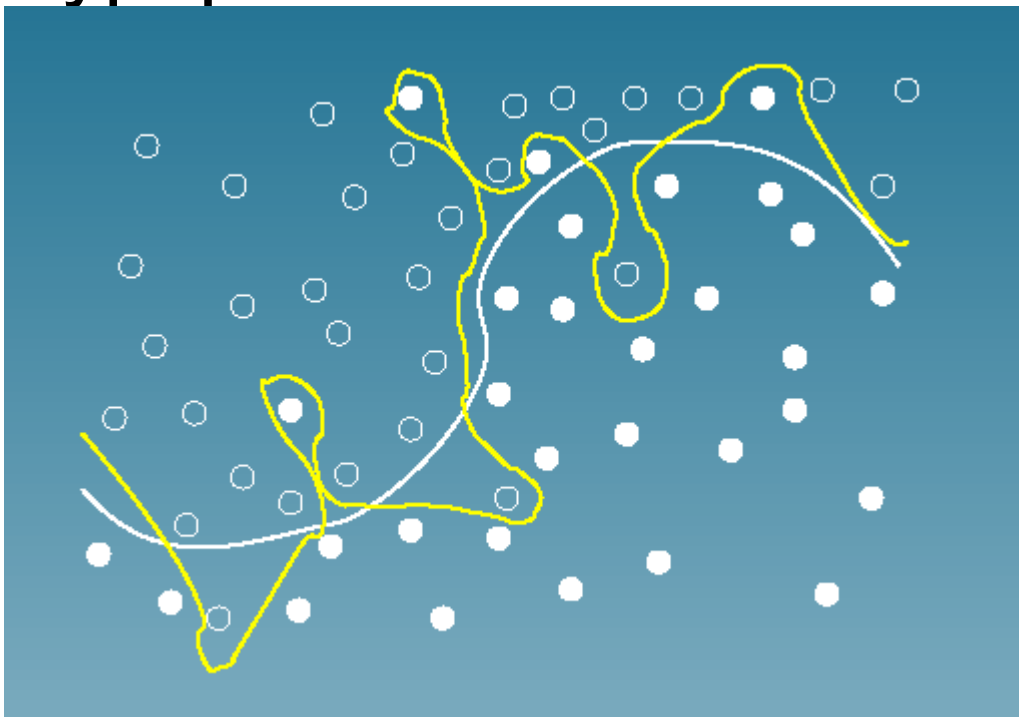
SVM non linéaires : Espaces des caractéristiques

- Idée générale : l'espace original peut toujours être mappé à un espace plus grand où l'ensemble d'apprentissage est plus facilement séparable :



Le tour des paramètres ?

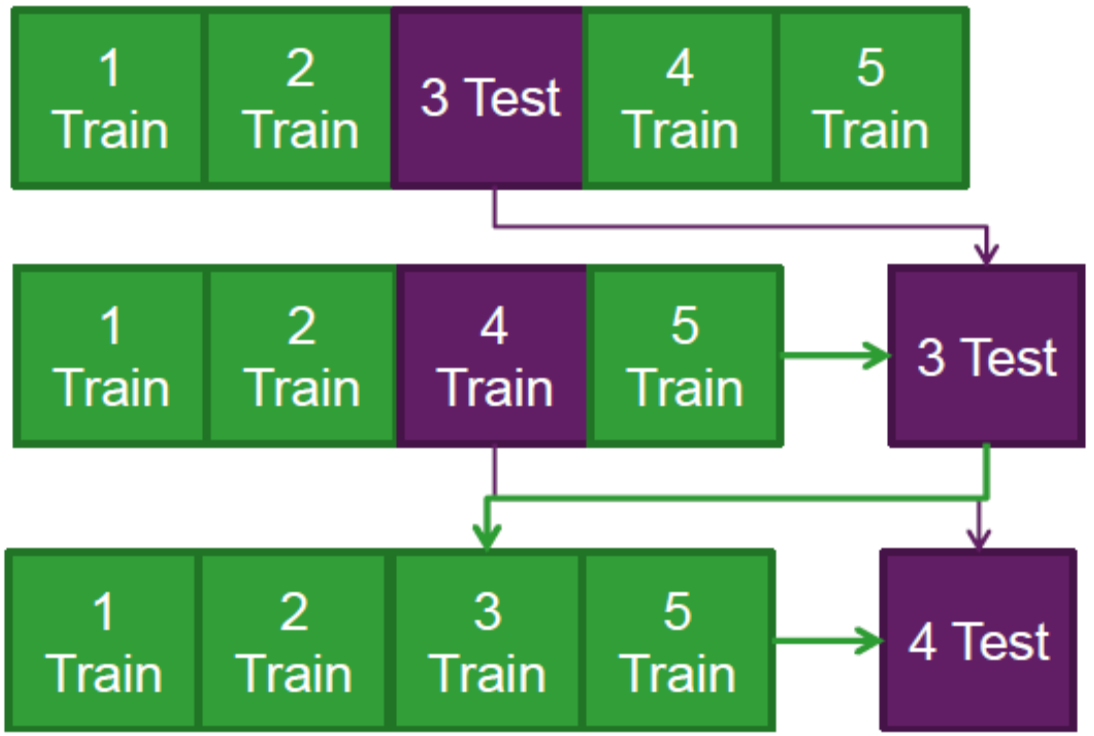
- Le problème de sur-apprentissage est un problème typique



Évaluation

- Rappel : Fraction des documents de la classe i correctement classés
- Précision : Fraction des documents assignés à la classe i qui concernent en fait la classe i
- Exactitude : (1 - taux d'erreur) Fraction des documents classés correctement
- D'autres métriques, mais encore une fois le problème définit les métriques appropriées

Exemple de validation croisée



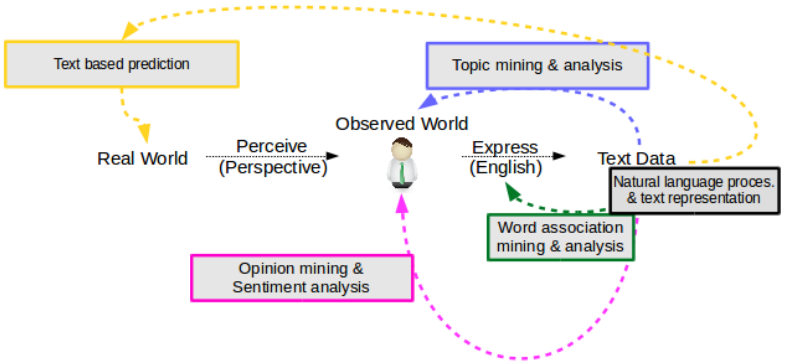
- Diviser les données en 5 échantillons
- Ajuster un modèle aux échantillons de formation et utiliser l'échantillon d'essai pour calculer une mesure du CV.
- Répétez le processus pour l'échantillon suivant, jusqu'à ce que tous les échantillons aient été utilisés pour former ou tester le modèle.

Exemples dans le paysage de l'exploration et analyse de textes

http://scikit-learn.org/stable/auto_exemples/applications/topics_extraction_with_nmf_lda.html

http://scikit-learn.org/stable/auto_examples/text/document_clustering.html

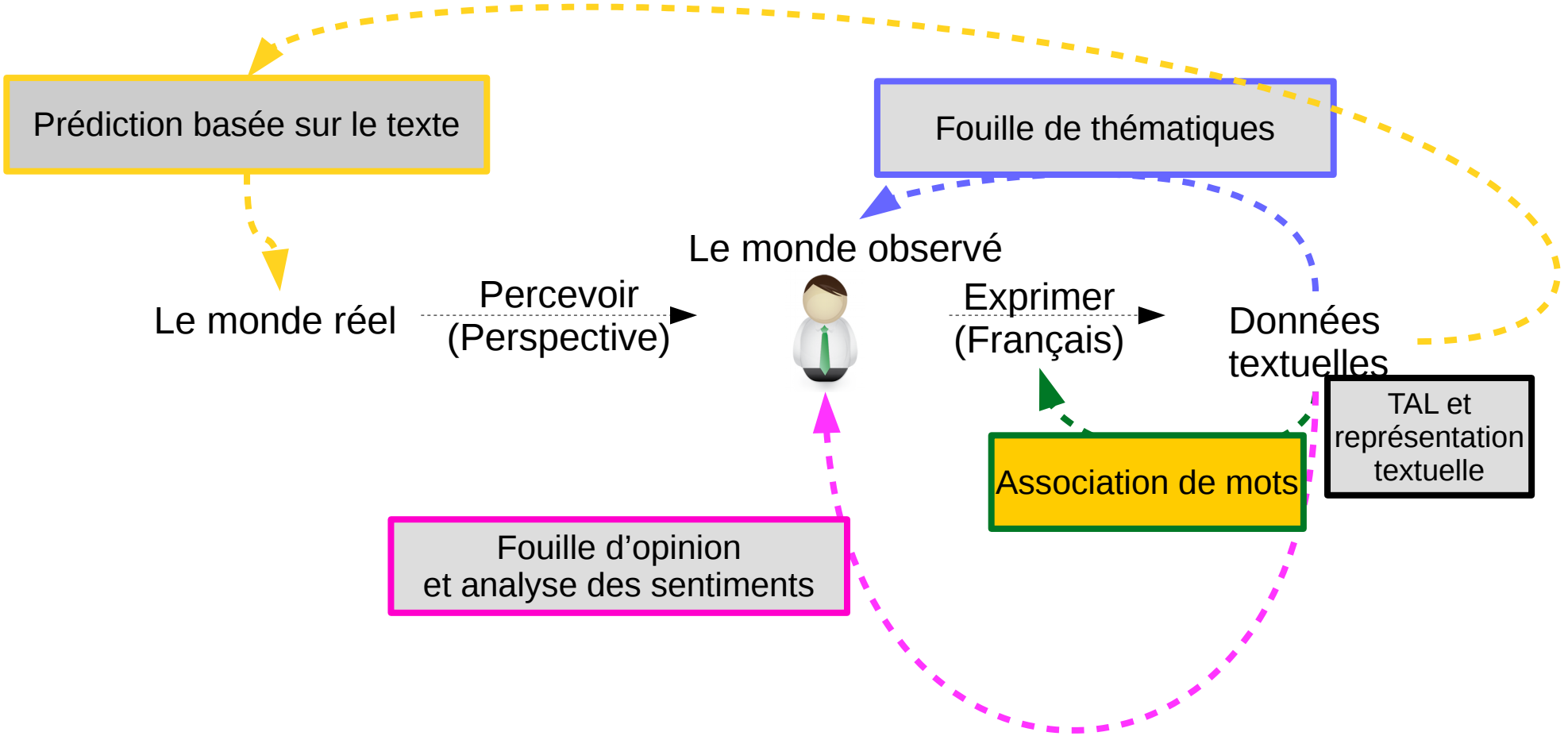
http://scikit-learn.org/stable/auto_examples/text/document_classification_20newsgroups.html



Corpus (Wikipédia)

https://sites.google.com/site/rmyeid/projects/polyglot

Paysage de la fouille et de l'analyse de textes



```
import nltk.collocations
import collections
```

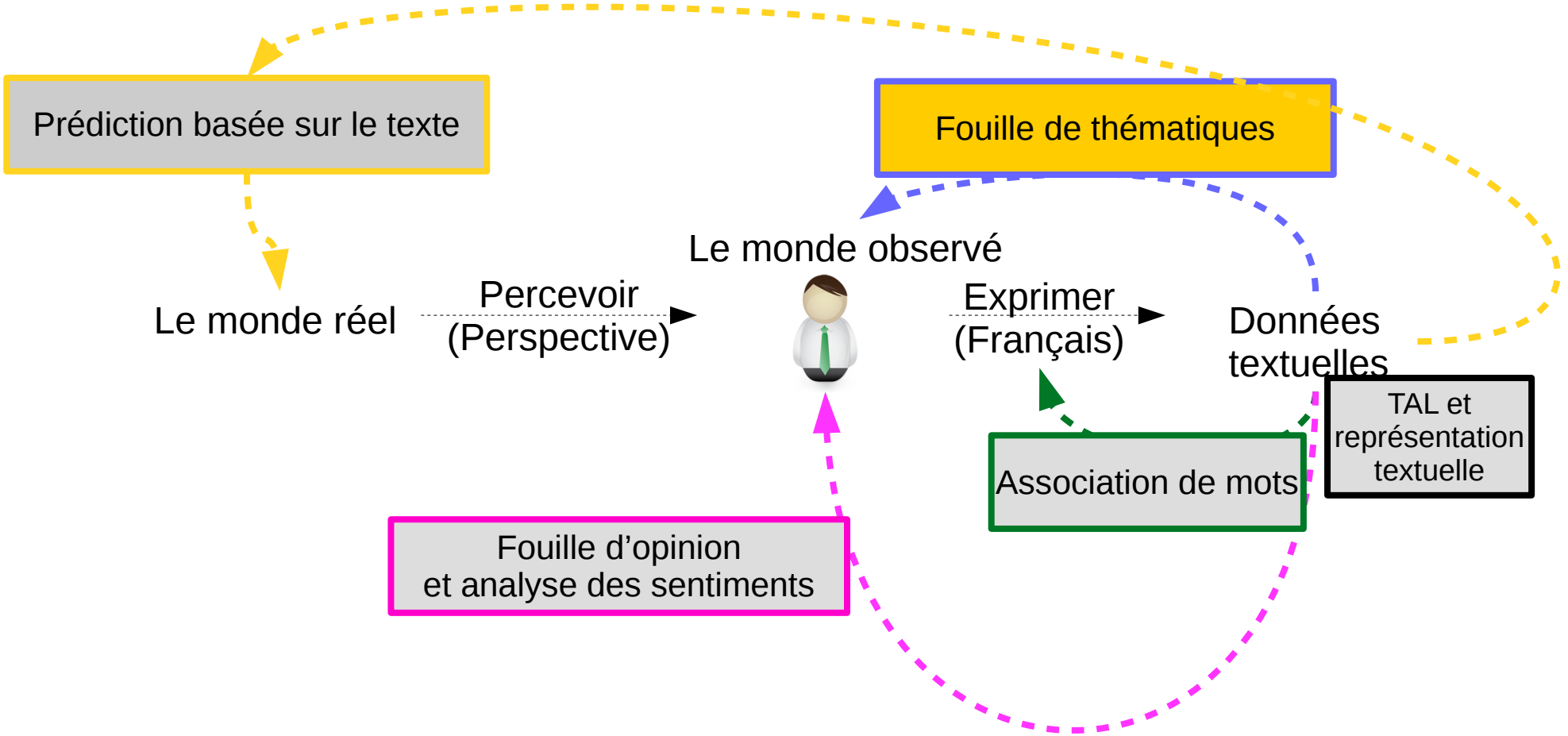
```
f = open("full.txt")
documents = " ".join([line for line in f.readlines()])
f.close()
```

```
bgm    = nltk.collocations.BigramAssocMeasures()
finder = nltk.collocations.BigramCollocationFinder.from_words(documents.split())
ignored_words = nltk.corpus.stopwords.words('english')
finder.apply_word_filter(lambda w: len(w) < 3 or w.lower() in ignored_words)
```

```
finder.nbest(bgm.raw_freq, 10)
finder.nbest(bgm.pmi, 10)
finder.nbest(bgm.likelihood_ratio, 10)
finder.nbest(bgm.chi_sq, 10)
finder.nbest(bgm.dice, 10)
finder.nbest(bgm.fisher, 10)
finder.nbest(bgm.jaccard, 10)
finder.nbest(bgm.mi_like, 10)
finder.nbest(bgm.poisson_stirling, 10)
finder.nbest(bgm.student_t, 10)
```

```
scored = finder.score_ngrams(bgm.pmi)
```

Paysage de la fouille et de l'analyse de textes



```
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.decomposition import NMF, LatentDirichletAllocation
from sklearn.cluster import KMeans
```

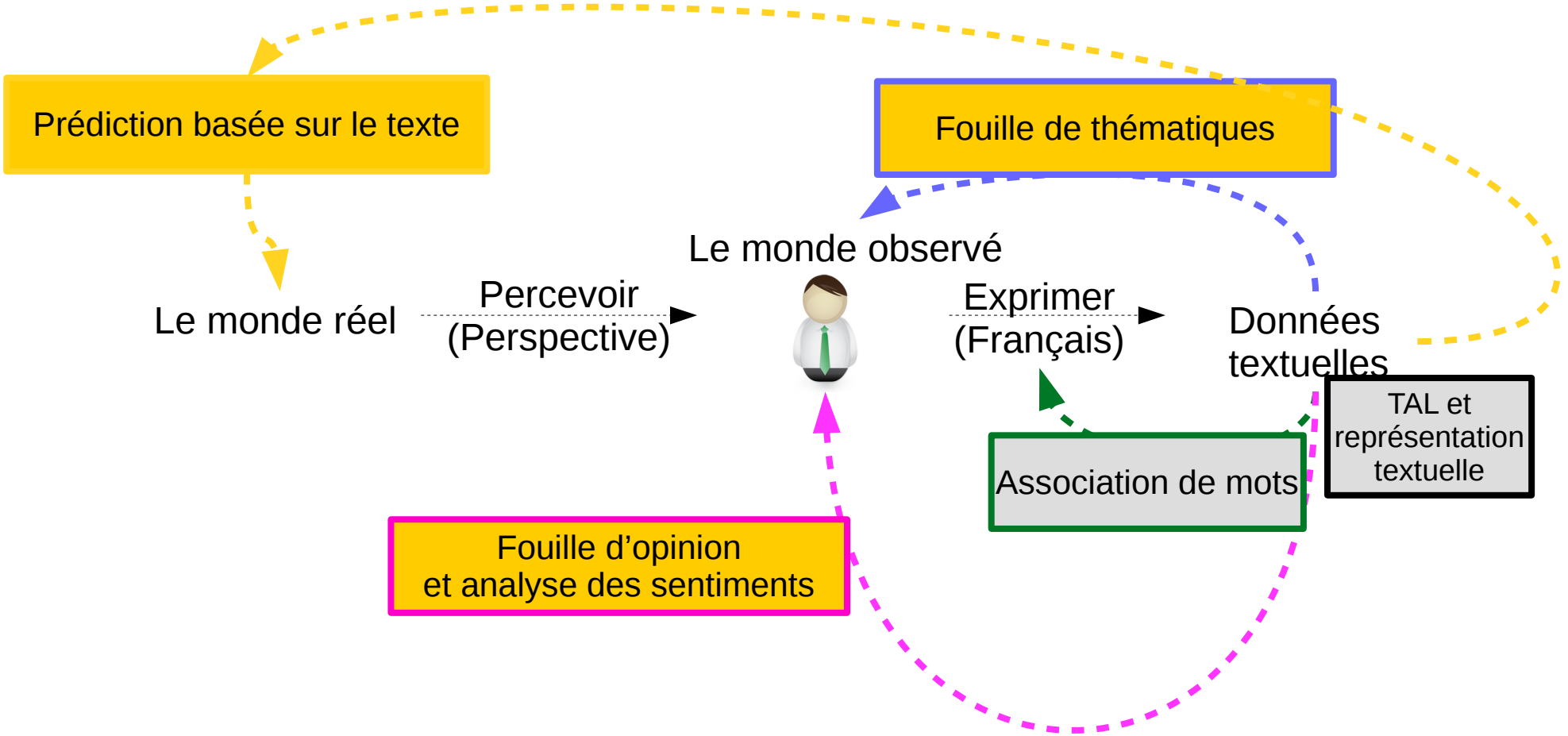
```
f = open("full.txt.head")
documents = [line for line in f.readlines()]
f.close()
```

```
tf_v =    CountVectorizer(max_df=0.95, min_df=2, max_features=100000, stop_words='english')
tfidf_v = TfidfVectorizer(max_df=0.95, min_df=2, max_features=100000, stop_words='english')

tf =      tf_v.fit_transform(documents)
tfidf = tfidf_v.fit_transform(documents)
```

```
nmf = NMF(n_components=10, random_state=1,alpha=.1, l1_ratio=.5).fit(tfidf)
km = KMeans(n_clusters=10, init='k-means++', max_iter=100, n_init=1).fit(tfidf)
lda = LatentDirichletAllocation(n_topics=10, max_iter=5, learning_method='online', learning_offset=50., random_state=0).fit(tf)
```

Paysage de la fouille et de l'analyse de textes



```
from sklearn.svm import SVC
from sklearn.feature_extraction.text import TfidfVectorizer
import random
```

```
from nltk.corpus import movie_reviews
docs = [(list(movie_reviews.words(fileid)), category)
            for category in movie_reviews.categories()
            for fileid in movie_reviews.fileids(category)]
random.shuffle(docs)
X,y= [" ".join(w[0]) for w in docs],[1 if w[1] =='pos' else 0 for w in docs]
```

```
tfidf_v = TfidfVectorizer(max_df=0.95, min_df=2, max_features=100000, stop_words='english')

tfidf = tfidf_v.fit_transform(X)
```

```
clf = SVC()
clf.fit(tfidf, y)
```

```
tfidf_test = tfidf_v.transform("One of my all time favorites. Shawshank Redemption is a very moving story about hope and the power of friendship. The cast is first rate with everyone giving a great performance. Tim Robbins and Morgan Freeman carry the movie, but Bob Gunton and Clancy Brown are perfect as the Warden Norton and prison guard captain Hadley respectively. And James Whitmore's portrail of an elderly inmate Brooks is moving. The screenplay gives almost every actor at least one or more memorable lines through out the film. As well as a very surprising twist near the end that almost knocked me out of my chair. If you have not seen this movie rent it or better yet buy it. As I bet you'll want to see this one more than once.".split())
```

```
print(clf.predict(tfidf_test))
```