Arbres, Random Forests et XGBoost

Ensembling, Bagging, Boosting

COURS PRÉCÉDENT

QUESTIONS ?

I: Text Mining - NLP

APPLICATIONS

Simple et directe

- prediction, classification, identification
  - Exemple: spam
  - multiclass: sujet du document

Non supervisée

- topic modeling

Avancée: productive

- Résumé
- Traduction automatique
- Chatbots

voir les nouvelles fonctionnalités de gmail

Interpretative

- Sentiment analysis

CORPUS

TEXTE BRUT

- forums, réseaux sociaux (peu structuré)
- plus structuré: discours, news, articles, emails, ...
- plus ou moins long: livres, articles scientifiques, abstracts, ...

CLOUD

- AWS
- Comprehend
- Google NLP
- Speech to text

LIBRAIRIES PYTHON

- spacy.io
- nltk
- gensim

Nombreuses librairies open source en R, Java, ...

ARABIC

- Stanford NLP: https://nlp.stanford.edu/projects/arabic.shtml
- Deep learning for Arabic NLP https://www.sciencedirect.com/science/article/pii/S1877750317303974

RESOURCES

- Livre: Speech and Language Processing https://web.stanford.edu/~jurafsky/slp3/

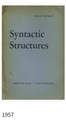
FERDINAND DE SAUSSURE

1916

BENVENISTE

1966

CHOMSKY



1957

TEXT GENERATION

- https://ml5js.org/docs/lstm-interactive-example

NUMERISER LE TEXTE

Comment passer d'un texte libre a une matrice numérique ?

Approche **Bags of words**

TF-IDF

Pour un mot donné dans un corpus de plusieurs documents

- Fréquence dans un document / frequence des mots dans les autres documents
- tf-idf means term-frequency times inverse document-frequency

TRANSFORMATIONS

- lemmatization,
  - la voiture est grande
  - Je suis sur un grand bateau
  - est, suis => etre
  - grande, grand => grand
- tokens, bi-grams
- stopwords: je, tu, il, et, me, sa, son, mais, donc, par, ...

WORD2VEC ET GLOVE

- Approche très recente qui associe un vecteur de grande dimension (128, 256, ...) a des milliers de mots
- Comme on a des vecteurs on a une distance entre les mots. Cosine distance
- Corpus original: Wikipedia
- Capture du sens du mot
  - Reine - femme = Roi - homme
  - Rabat - capitale = Paris - capitale

Predictive models learn their vectors in order to improve their predictive ability of Loss(target word | context words; Vectors), i.e. the loss of predicting the target words from the context words given the vector representations. In word2vec, this is cast as a feed-forward neural network and optimized as such using SGD,

COSINE DISTANCE

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

SPACY

https://spacy.io/usage/vectors-similarity

```
from gensim.models import Word2Vec

#loading the downloaded model
model = Word2Vec.load_word2vec_format('GoogleNews-vectors-negative

#the model is loaded. It can be used to perform all of the tasks m

# getting word vectors of a word
dog = model['dog']

#getting king queen equal
print(model.most_similar(positive=['woman', 'king'], negative=['ma

#getting odd one out
print(model.doesnt_match("breakfast cereal dinner lunch".split()))

#getting similarity index
print(model.similarity('woman', 'man'))
print(model.similarity('car', 'minivan'))
```

see word2vec_demo.py

WORD2VEC

Word2vec is not a single algorithm but a combination of two techniques – CBOW(Continuous bag of words) and Skip-gram model.

- Skip - gram : to predict the context given a word
- CBOW tends to predict the probability of a word given context words
- word2vec is a "predictive" model, predict word / context + context / word

http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

http://mccormickml.com/assets/word2vec/Alex_Minnaar_Word-Vectors_Model.pdf

SEE ALSO GLOVE

Glovle is a "count-based" model .Dimensionality reduction on the co-occurrence counts matrix.

- http://scikit-learn.org/stable/tutorial/text_analytics/working_with_te

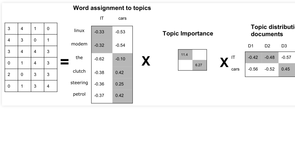
TF-IDF - SKLEARN

The most intuitive way to do so is to use a bags of words representation:

Assign a fixed integer id to each word occurring in any document of the training set (for instance by building a dictionary from words to integer indices).

For each document #i, count the number of occurrences of each word w and store it in X[i, j] as the value of feature #j where j is the index of word w in the dictionary.

TOPIC MODELING



II: Lab : text classification sur bbc dataset