Gensim

Topic Modelling for Humans

Corpus, ressources et linguistique outillée
Sorbonne Université
2022 - 2023

Présenté par :
SACI Thiziri - FAURY Anne - QIAN Yuyan

C'est quoi Gensim ?

- Une librairie Python de **topic modeling** open source pour le traitement automatique du langage naturel
- Créé par l'équipe RaRe Technologies (spécialisé en ML & NLP consulting)
- Plusieurs algorithmes de NLP : Word2Vec, fasttext, LSI, LDA...
- Approuvé et adopté par plusieurs entreprises telles que Amazon, Channel 4, CapitalOne...

Fonctionnalités :

- **Pré-traitement** de corpus (indexation de documents, tokenization, lemmatisation, découpage en phrases, suppression de mots outils)
- Recherche de thématique
- Récapitulation d'informations
- **Word embedding** (vectorisation)
- Calcul de **similarité** et du **TfIdf**
- Classification de textes

Latent Dirichlet Allocation (LDA)

- Paramètres : LdaModel(corpus=corpus, id2word=dictionary, num_topics=2, passes=10)
- Document : combinaison de thèmes
- Thème : combinaison de mots
- Entrée : **corpus + nombre de thèmes**
- Calcul de la distribution des mots pour chaque thème
- Calcul de la distribution des thèmes pour chaque document
- Sortie : il associe **un thème à chaque document**

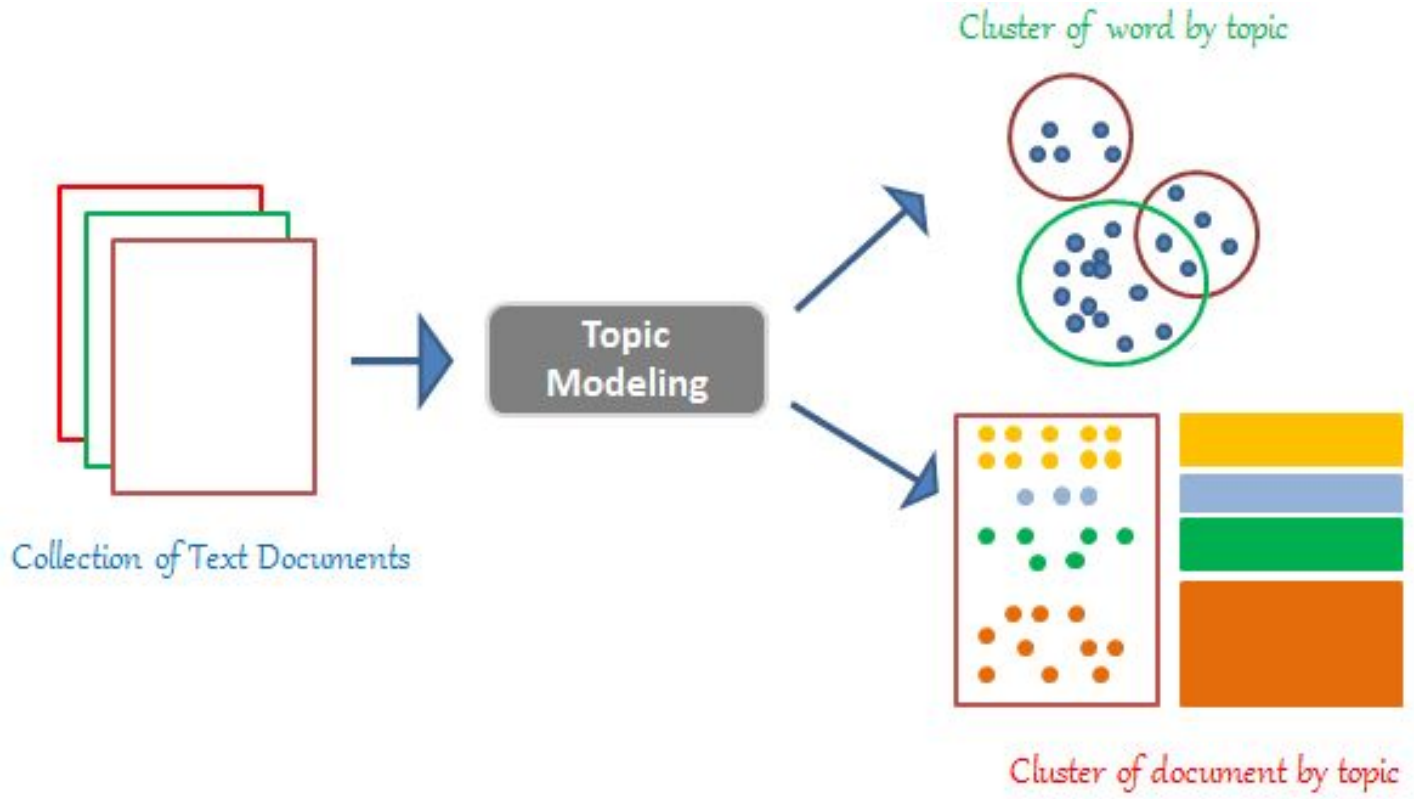

Figure 01 : illustration du LDA

(Source : Latent Dirichlet Allocation : Topic Modeling en Python - La revue IA)

Word2Vec

- Un algorithme probabiliste basé sur des réseaux de neurones
- Établit automatiquement des **relations sémantiques** entre les mots grâce à leur contexte
- Permet d'établir des similarités entre les mots
- Word Mover's Distance évalue la distance entre deux documents

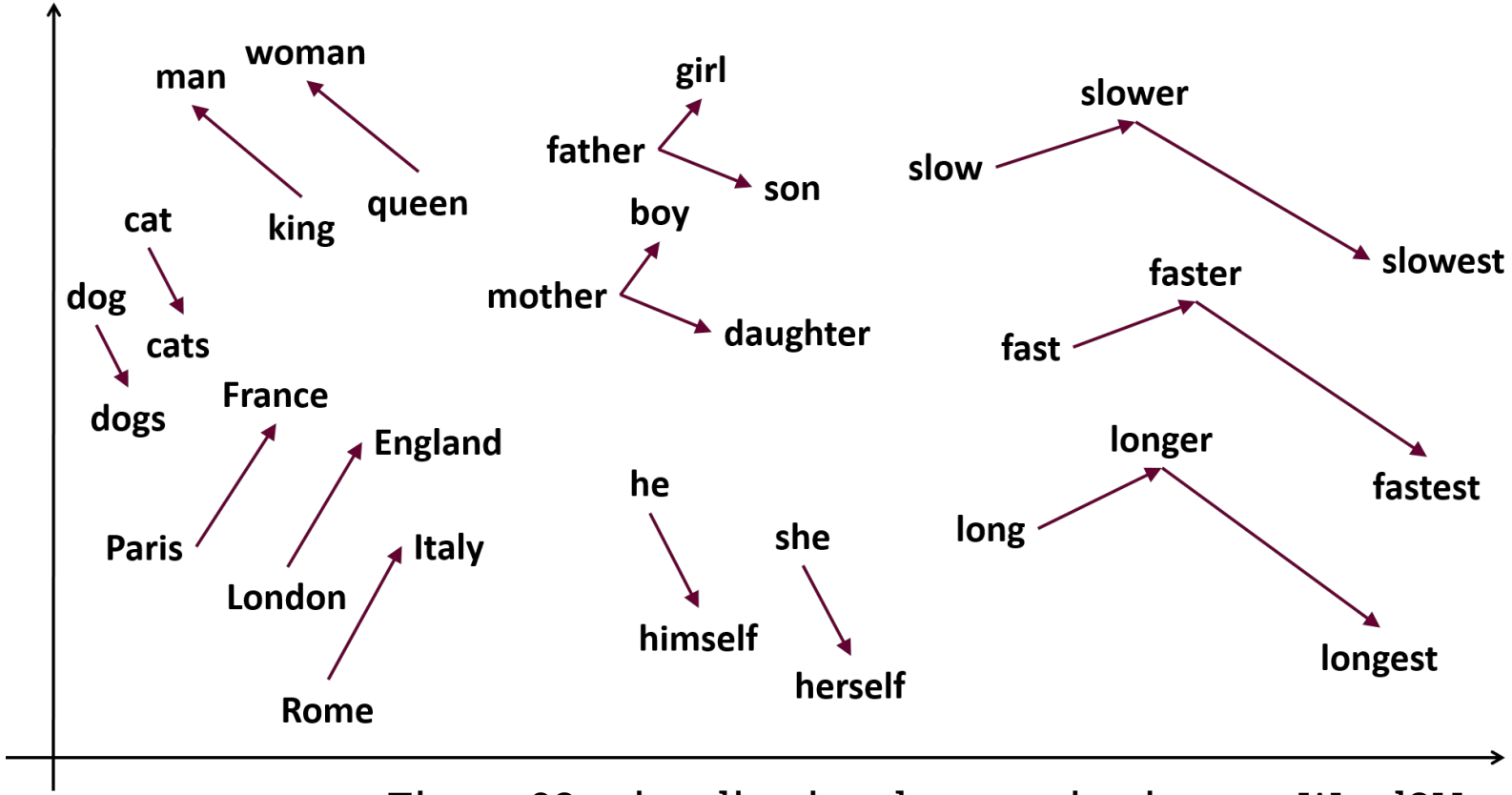

Figure 02 : visualisation de vectorisation par Word2Vec

(Source : NLP with gensim (word2vec))

FastTextModel

- FastText est une librairie pour entraîner des modèles de word embedding
- Prend en compte la structure morphologique des mots
- Traite chaque mot comme l'agrégation de ses n-grams de caractères
- Obtient de meilleurs résultats pour les **tâches syntaxiques** sur un petit corpus
- Peut établir des similarités avec des mots hors-vocabulaire

Gensim VS NLTK:

	NLTK	Gensim
Grand corpus	–	+
Diversité Langues	+	–
Reconnaissance de langue	+	–
Prétraitement : lemmatisation, détectionNER, tokenisation, POS tagging	+	–
Modélisation de sujets	–	+
Orienté	Sémantique	Statistique
Public visé	Débutant	Expérimenté

Avantages :

- **Rapide** et **efficace** sur les grands corpus
- Utilise moins de mémoire : “Corpus Streaming – One Document at a Time”
- Il dispose de corpus tout prêts
- Il traite **plusieurs langues** (essai sur le Français et l’Arabe)

Inconvénients :

- Fonctionnalités supprimées sans mettre d'autres alternatives (gensim.summarization, gensim.utils.lemmatize()...)
- Efficacité **relative** à la langue du corpus étudié
- Fonctionnement par approche **non supervisée seulement**
- Installation de NumPy et Scipy **requise** avant l'installation de Gensim

Sitographie :

- API — gensim : https://tedboy.github.io/nlps/api_gensim.html
- Documentation — gensim : https://radimrehurek.com/gensim
- GitHub - RaRe-Technologies/gensim: Topic Modelling for Humans : https://github.com/RaRe-Technologies/gensim
- Latent Dirichlet Allocation : Topic Modeling en Python - La revue IA : https://larevueia.fr
- Natural Language ToolKit (NLTK) : https://www.nltk.org
- NLP with gensim (word2vec) : https://www.samyzaf.com
- Topic Modeling in Python: Latent Dirichlet Allocation (LDA) | by Shashank Kapadia | Towards Data Science : https://towardsdatascience.com

Démonstration sur Python…