

Brève introduction à la recherche d'information

Julien Velcin

<https://velcin.github.io>

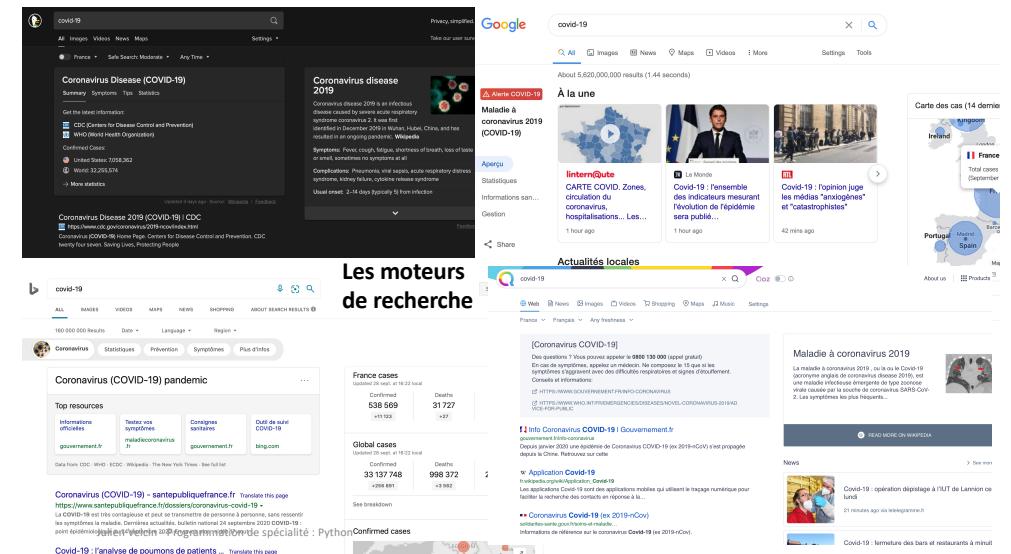


Julien Velcin - Programmation de spécialité : Python

Un déluge d'information

- Big data :
 - V de Volume
 - V de Vélocité
 - V de Variété (**texte**, image, vidéo, son, tags...)
 - etc.
 - Le WWW est une source phénoménale de données, en particulier textuelle, mais il existe beaucoup d'autres sources : mémoire d'entreprise, données du patrimoine (ex. BnF, INA)

Julien Velcin - Programmation de spécialité : Python



sign in subscribe search

dating more International

the guardian

UK world sport football opinion culture business lifestyle fashion environment tech travel

headlines

Now 4°C Lyon 12:00 15:00 18:00 21:00 9:00 13:00 9:00 6:00

Climate change / February breaks global temperature records by 'shocking' amount

Great Barrier Reef Severe coral bleaching worsens

Japan US sailor arrested in Okinawa on suspicion of rape

German elections Anti-refugee AfD party makes dramatic gains

US elections 2016 Clinton and Sanders attack 'pathological liar' Trump

Ivory Coast Gunmen open fire on tourist resort, killing 16

Thailand Eight die in bank after chemical fire extinguisher leak

Egypt Justice minister sacked for saying he would arrest prophet Muhammad

Turkey President vows to defeat terror as bomb kills 34 in Ankara

United Arab Emirates Plane reported missing in Yemen

+ More headlines

highlights

Julien Velcin - Programmation de spécialité : Python

100 Best Nonfiction Books of All Time No. 7 - The Right

#journéedelalanguefrançaise

Top Direct Comptes Photos Vidéos Autres options

Suggestions · Actualiser · Tout afficher

- Khalil (pilgrim) @sehnaoui Suivre Sponsored
- Tom Kenter @TomKenter Suivi par Shiri Dor-Hacohen ... Suivre
- Alberto Lumberas @alberto... Suivi par Bertrand Jouve Suivre

Trouver des amis

Tendances · Modifier

- #JournéeDeLaLangueFrançaise #BoudinDirect #SRFCOL #SOSPascal #BrunoFunRadio Lacazette Troyes Olivier Bourdeau Albert Einstein Dany Laferrière Julien Velcin - Programmation de spécialité : Python
- POUAHHAHAH #JournéeDeLaLangueFrançaise
- ben&jerry&ana @oggordan - 1 min Pourquoi faire une journée pour cette langue si c'est pour la massacrer avec une réforme par la suite? #JournéeDeLaLangueFrançaise
- Moins gentil ligné @ParathorO - 2 min #JournéeDeLaLangueFrançaise zig
- ben&jerry&ana @oggordan - 3 min Si vous voulez honorer la langue française alors s'il vous plaît pas de "ognon" #JournéeDeLaLangueFrançaise

4 nouveaux résultats

amazon.fr

Toutes nos boutiques

Amazon.fr Ventes Flash Meilleures ventes Offres reconditionnées Nos idées cadeaux Services Amazon Amazon Assistant

Star Wars : Battlefront - édition limitée > Commentaires client

Commentaires client

★★★★★ 59 3,2 sur 5 étoiles

Hidden for obvious reasons

5 étoiles 17 4 étoiles 14 3 étoiles 7 2 étoiles 8 1 étoile 13

Évaluez cet article Écrire un commentaire

Meilleur commentaire positif

Voir les 31 commentaires positifs >

★★★★☆ Pas parfait mais un Star Wars

Par Client d'Amazon le 21 décembre 2015

Le titre pourrait être plus riche en terme de contenu, surtout en solo qui fait seulement guider l'introduction aux bases, mais l'immersion est tellement réussie que les fans de l'univers Star Wars seront conquis.

L'ambiance sonore et visuelle est magistrale, et incarner un stormtrooper en pleine bataille d'Endor ou sur Hoth est un réel plaisir !

A éviter si vous ne jouez pas en ligne.

Julien Velcin - Programmation de spécialité : Python

THE NEW REDDIT JOURNAL OF SCIENCE

hot new rising controversial top

4389 15 hours ago by dreepoope 3720 comments share

Humans have triggered the last 16 record-breaking hot years experienced on Earth (up to 2014), with the new research tracing our impact on the global climate as far back as 1937. The findings suggest that without human-induced climate change, recent hot summers and years would not have occurred. + 1999c

Top 200 Comments show 500

sorted by: best (suggested)

XiiCuded 1957 points 13 hours ago * Switch to nuclear energy.

old-table 685 points 13 hours ago edit: thanks for the gold nuclear energy fwiw

So what can we actually do to combat this? Aside from colonizing space and getting humans off this planet?

XiiCuded 1957 points 13 hours ago * Switch to nuclear energy.

Mr_Industrial 839 points 13 hours ago Good luck convincing several million people that nuclear energy is safer than most other forms of energy. It's not about the facts, it's about perception of the facts.

climbre 828 points 12 hours ago You don't have to. The public rarely has input into power plant construction etc. Once they're up and running no-one cares about it anymore.

If you ask people if they'd like a change, 90% will say no, 95% if you say it might involve danger. If you make the change and ask how happy people are most are just as happy.

This is a good point. The thing you have to remember though is that the people in charge who have the power to decide what type of

Julien Velcin - Programmation de spécialité : Python



L.Balthasar - L.Mondada
Julien Velcin - Programmation de spécialité : Python

```

174 <token rang="8">>on</token>
175 <token rang="9">>salt</token>
176 <token generique="que" rang="10">>que:</token>
177 </productionVerbale>
178
179 <productionVerbale pseudo="M" rang="5">
180 <espace longueur="10" rang="1"/>
181 <chevauchement type="fin" position="Externe" rang="2">[</chevauchement>
182 <token rang="3">>oui</token>
183 <espace longueur="3" rang="4"/>
184 <chevauchement type="fin" position="Externe" rang="5">]</chevauchement>
185 <espace longueur="8" rang="6"/>
186 </productionVerbale>
187
188 <productionVerbale pseudo="O" rang="6">
189 <token rang="7">>on</token>
190 <token rang="2">></token>
191 <token rang="3">>positionné</token>
192 <token rang="5">>les</token>
193 <token rang="6">>les</token>
194 <tokens rang="7">>d</token>
195 <token generique="attente" rang="8">>atTENt</token>
196 <token generique="zone" rang="9">>zone</token>
197 <token generique="de" rang="10">>de</token>
198 <token rang="11">>desert</token>
199 <token generique="déserte" rang="12">>deSERRt</token>
200 <token generique="accès" rang="13">>acces</token>
201 <token generique="principale" rang="14">>princiPAL</token>
202 <token generique="l'" rang="15">>l</token>
203 <token generique="accès" rang="16">>l</token>
204 <token generique="l'" rang="17">>'accès</token>
205 <token generique="l'" rang="18">>d</token>
206 <token rang="19">>service</token>
207 <token type="courte" duree=".1" rang="20">>(.).</pause>
208 <token generique="l'" rang="21">>h::</token>
209 <token generique="l'" rang="22">>la</token>
210 <token generique="l'" rang="23">>seule</token>
211 <token rang="24">>chose</token>
212 <token type="courte" duree="0.2" rang="25">>(0.2)</pause>
213 <token generique="qui" rang="26">>qui</token>
214 <token rang="27">>serial</token>
215 <token rang="28">>a</token>
216 <token rang="29">>modifier</token>
217 <token generique="l'" rang="30">>l</token>
218 <token rang="31">>l</token>
219 <token generique="instant" rang="32">>instant</token>
220 <token rang="33">>h</token>
221 <token generique="l'" rang="34">>l</token>
222 <token rang="35">>est</token>
223 <token rang="36">>on</token>
224 <token rang="37">>salt</token>
225 <token rang="37">>salt</token>
226

```

Julien Velcin - Programmation de spécialité : Python

Gaussian Embedding of Linked Documents from a Pretrained Semantic Space

Antoine Gourru^{1*}, Julien Velcin¹ and Julien Jacques¹

¹Université de Lyon, Lyon 2, ERIC UR3083

{antoine.gourru, julien.velcin, julien.jacques}@univ-lyon2.fr

Abstract

Gaussian Embedding of Linked Documents (GELD) is a new method that embeds linked documents (e.g., citation networks) onto a pretrained semantic space (e.g., a set of word embeddings). We formulate the problem in such a way that we model each document as a Gaussian distribution in the word vector space. We design a generative model that combines both words and links in a consistent way. Leveraging the variance of a document allows us to model the uncertainty related to word and link generation. In most cases, our method outperforms state-of-the-art methods when using our document vectors as features for usual downstream tasks. In particular, GELD achieves better accuracy in classification and link prediction on Cora and Dblp. In addition, it derives quantitatively the consistency of several properties of our method. We provide the implementation of GELD and the evaluation datasets to the community (<https://github.com/AntoineGourru/DNEembedding>).

1 Introduction

Linked documents are everywhere, from web pages to bibliographic networks (e.g., scientific articles with citations) and social networks (e.g., tweets in a Followee/Follower network). The corpus structure provides rich additional semantic information. For example, in a citation article, cross links

ods are mainly based on matrix factorization [Brochier *et al.*, 2019; Huang *et al.*, 2017] and deep architectures [Liu *et al.*, 2018; Tu *et al.*, 2017; Kipf and Welling, 2016]. Most of these techniques learn documents as points in the embedding space. However, considering a measure of dispersion around those vectors brings useful information, as shown on corpus with no link between documents [Nikolentzos *et al.*, 2017]. In Graph2Gauss [Bojchevski and Günnemann, 2018], each document is associated with a measure of uncertainty along with its vector representation. However, the objective function optimizes the uncertainty using the network information and it does not model the dispersion at the word level. Additionally, variational methods such as [Kipf and Welling, 2016; Meng *et al.*, 2019] introduce gaussian posteriors, but the generative process uses the dot product between documents' mean only to model the adjacency and attribute matrix entries. Hence it is not clear what the uncertainty obtained from the variational variance captures.

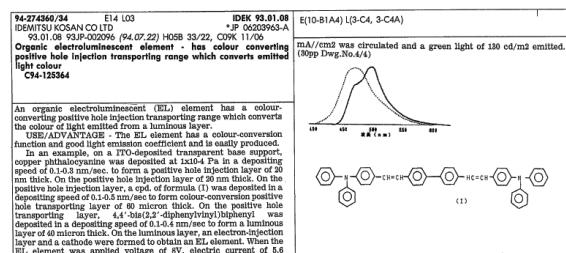
Finally, none of earlier methods represents both documents and words in the same semantic space, as opposed to text-based methods such as [Le and Mikolov, 2014]. LDE [Wang *et al.*, 2016] and RLE [Gourru *et al.*, 2020] both build a joint space for embedding words and linked documents. However, these approaches do not take the uncertainty into account.

In this paper, we propose an original model that learns both documents and words in a single semantic space, with a vector representation and a vector of uncertainty for each document, named GELD for Gaussian Embedding of Linked Document.

References

- [Barkan, 2017] Oren Barkan. Bayesian neural word embedding. In *Thirty-First AAAI Conference on Artificial Intelligence*, pages 3135–3143, 2017.
- [Bojchevski and Günnemann, 2018] Aleksandar Bojchev and Stephan Günnemann. Deep gaussian embedding graphs: Unsupervised inductive learning via ranking. *Proceeding of the International Conference on Learn Representations*, ICLR, 2018.
- [Brochier *et al.*, 2019] Robin Brochier, Adrien Guille, & Julien Velcin. Global vectors for node representations. *Proceedings of the World Wide Web Conference*, WWW, pages 2587–2593, 2019.
- [Das *et al.*, 2015] Rajarshi Das, Manzil Zaheer, and CI Dyer. Gaussian lda for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 795–804, 2015.
- [Deerwester *et al.*, 1990] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science* 41(6):391–407, 1990.
- [Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume (Long and Short Papers)*, pages 4171–4186, 2019.
- [Gourru *et al.*, 2020] Antoine Gourru, Adrien Guille, Jul

Brevets



Julien Velcin - Programmation de spécialité : Python

Science des données

- Il faut automatiser la manipulation de ces grands volumes :
 - ⇒ Systèmes d'information (*information systems*)
 - ⇒ Recherche d'information (*information retrieval*)
 - ⇒ Fouille de données (*data mining*)
- Cela nécessite le recours à :
 - ⇒ analyse des données (*data analysis*)
 - ⇒ traitement automatique de la langue (*NLP*)
 - ⇒ apprentissage automatique (*machine learning*)

Science des données
(*data science*)

Chaîne de traitement des données

- Extraction, stockage des données :
 - ⇒ Comment gérer l'hétérogénéité des formats ?
 - ⇒ Quelle structure de stockage ?
- Représentation, indexation :
 - ⇒ Quelle est la meilleure représentation ?
 - ⇒ Comment indexer les données de manière efficace ?
- Analyse des données :
 - ⇒ Comment comparer des données textuelles ?
 - ⇒ Quels algorithmes choisir ?

Difficultés spécifiques au texte

- Volume important, vocabulaire très vaste (erreurs, abréviations, argot, néologismes, noms propres...)
- Ecart entre la surface des mots et leur sens
- Relations implicites entre les mots : synonymie, polysémie, liens de subordination, co-références, etc.
- Ambiguité sémantique : « Il voit le garçon avec ses lunettes » (qui possède les lunettes ?)
- Selon la tâche, la représentation est différente
- Similarité entre deux textes (à partir de quels éléments, malédiction de la dimension)

Analyser les données textuelles

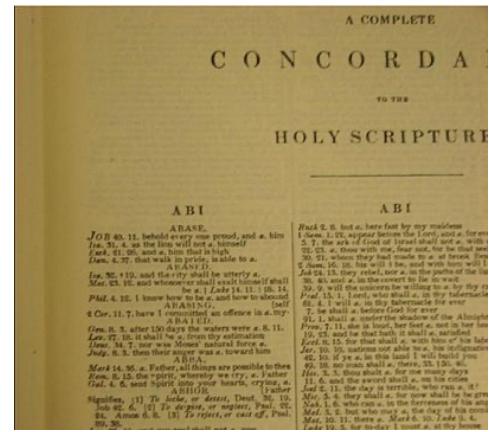
Expressions régulières

- Librairie `re` :

```
texte = "Lyon fut la capitale des Gaules en -20 avant J.C."
s = re.search("^Lyon", texte)
s = re.search("capitale.*Gaules", texte)
print(s.span())
```

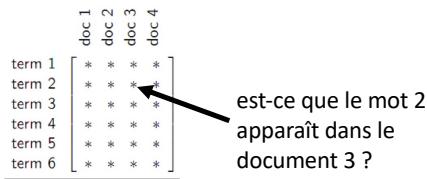
- `findall` permet de trouver toutes les occurrences
- `sub` remplace une sous-chaîne par une autre

Application : le concordancier

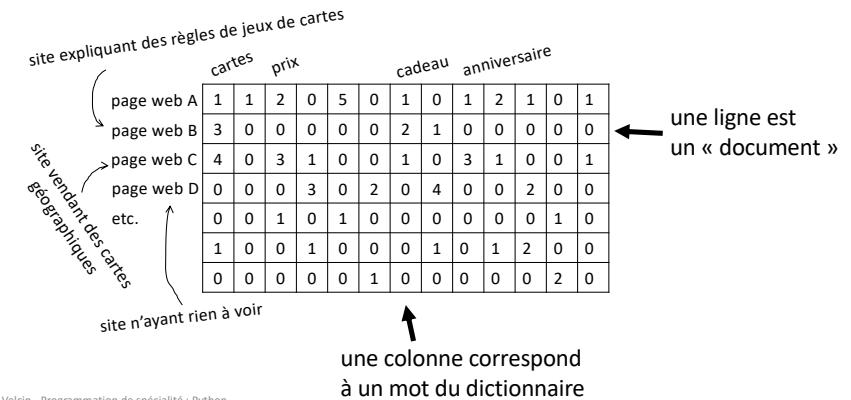


Aller plus loin avec la construction d'un index

- L'approche des expressions régulières a plusieurs limitations :
 - le motif cherché doit être contigu
 - l'algorithme de correspondance (*matching*) est coûteux
- L'objectif est d'encoder directement les mots présents dans un document en suivant l'hypothèse du « sac de mots » (*bag of words*)



Requête « carte » sur l'index des sites Web



Principales étapes

- Construction du dictionnaire de mots (vocabulaire)
- Pour chaque document, construire une représentation basée sur ce dictionnaire :
 - binaire : 0 si le mot est absent, 1 s'il est présent
 - nombre d'occurrences (*term frequency*)
(il y a d'autres schémas de pondération)
- A partir de ce tableau (matrice), on peut :
 - calculer l'importance de chaque mot (nuage de mots clefs)
 - trouver les documents les pertinents pour une requête
 - comparer les documents entre eux (classification, clustering)

Julien Velcin - Programmation de spécialité : Python

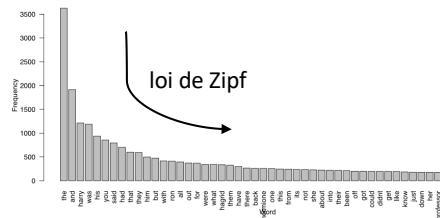
Segmentation du texte en mots

- Cette étape de *tokenization* est assez simple dans les langages occidentaux, comme le français ou l'anglais
- Elle consiste généralement en :
 - définir ce qui constitue la frontière entre deux « mots »
 - écrire l'expression régulière correspondante
 - découper la chaîne en une liste de sous-chaînes (les mots)(en Python avec la fonction `split()`)
- Construire le vocabulaire consiste à faire l'union de tous les mots trouvés et dédoublonner (avec un ensemble en Python par exemple)

Julien Velcin - Programmation de spécialité : Python

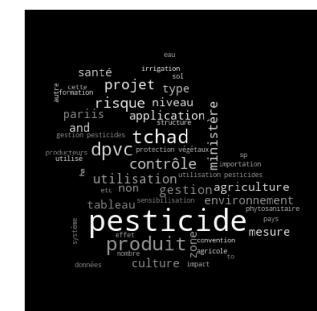
Quelques prétraitements usuels

- Mettre en minuscule les mots du document (cf. fonction `lower()`)
- Protéger certaines expressions (ex. « H5N1 » ou « Covid-19 »)
- Supprimer les chiffres, les ponctuations
- Supprimer les mots trop peu fréquents
- Supprimer les mots outils



Julien Velcin - Programmation de spécialité : Python

Visualiser le corpus : le nuage de mots



Julien Velcin - Programmation de spécialité : Python

Réaliser son propre moteur de recherche

- Requête de l'utilisateur
- Récupérer les listes des documents
- Agréger les listes :
 -> union des ensembles
 -> pondérer les documents puis les trier