UNIVERSITÉ PARIS DESCARTES | USPC Université Sorbonne Paris Cité

Sujet 2017-2018

MLDS – novembre 2018

Benchmarking pour évaluer la qualité de *word embeddings*

Contexte du sujet

Comme vu en cours, la technique des *word embeddings* est une technique d'IA en plein essor qui permet de représenter le sens des mots à travers des représentations vectorielles. Chaque mot est représenté par un vecteur de valeurs réelles (ce vecteur est ce qu'on appelle un *word embedding*) et des opérations sur ce vecteurs permettent de résoudre des problèmes complexes comme, par exemple, des questions d'analogie : étant donné deux paires de mots liés par une relation (par exemple, `homme:femme`, `roi:reine`) , on cache au système le quatrième mot (`reine`) et le système doit l'inférer en se basant sur les trois autres : `homme` est à `femme` ce que `roi` est à `?`

Le problème est que, pour un ensemble de mots donné, il existe plusieurs façons de construire des *word embeddings* et la qualité des vecteurs peut donc être variable. Pour mesurer la qualité d'un ensemble de vecteurs de mots, on utilise donc des métriques mesurant leur capacité à résoudre des questions sémantiques de nature variable. Voici des jeux de données pouvant servir à évaluer la capacité d'un *embedding* à répondre à des questions diverses.

similarité (au sens de quasi synonymie)

SimLex-999, SemSim, UMSRS-SIM, etc.

lien sémantique (au sens large)

Rel-122, WordSim 353, McRae, UMSRS-Rel, etc.

analogie (et autres relations)

Google Questions, etc.

Les jeux de données pour la similarité-synonymie et le lien sémantique se présentent en général sous la forme d'un ensemble de paires avec un score (de similarité-synonymie ou de lien) assigné par un humain. Ces scores induisent un rang entre les paires. Par exemple, si on a les paires (tarte, sucre) avec un score de 15, (sable, iphone) avec un score de 1, (papier, bois) avec un score de 13 on aura le classement dans l'ordre décroissant de lien :

(tarte, sucre) rang 1

(papier, bois) rang 2

(sable, iphone) rang 3

On peut alors comparer ces rangs avec ceux obtenus en calculant des similarités entre ces mots avec des embeddings et mesurer la corrélation des rangs obtenus via les humains et via la machine (ceci via des mesures comme le coefficient de Spearman ou de Kendall).

Parfois ces paires avec leurs scores sont explicitement représentés dans les jeux de données, parfois vous devrez les déduire du jeu de données (vous aurez par exemple une présentation sous forme d'une matrice de similarité mot x mot).

Pour les Google Questions, Gensim y donne accès facilement. Vous devez juste lancer les tests sur le fichier de questions.

Ce sujet met en évidence l'importance et la complexité de l'évaluation dans un projet de recherche en *text mining*.

Travail à effectuer

Ecrire un code en ligne de commande prenant en entrée un ensemble de word *embeddings* et un ensemble de jeux de données et donnant alors pour chape paire (*word embedding model*, jeu de données) le degré de corrélation de rang entre le jugement humain et celui fourni pour la machine : donc pour chaque paire (*embedding*, *dataset*) on veut en fait savoir jusqu'à quel point l'*embedding* a permis de classer les paires comme l'aurait fait l'humain.

On doit pouvoir lancer le programme soit sur tous les jeux de données, soit seulement sur ceux de similarité-synonymie, soit seulement sur ceux de lien sémantique au sens large, soit seulement sur les Google questions. Dans tous les cas,

on obtient à chaque fois la moyenne globale (corrélation moyenne) et la moyenne par couple (*embedding*, *dataset*).

La liste des jeux de données de test à prendre en compte est indiquée dans la section précédente.

Les *embeddings* à évaluer sont :

- `GoogleNews-vectors-negative300.bin`

- glove.840B.300d.zip

- text8 (qui aura été entraîné en utilisant skip-gram, avec 300 dimensions, une fenêtre de 12 et 12 exemples négatifs).

Vous pouvez aussi concevoir une interface graphique Web permettant de lancer les mêmes traitement que via la ligne de commande : widgets pour paramétrer et lancer les tests et affichage graphique des résultats. Mais ce n'est à faire que si vous êtes à l'aise avec ces technologies et si la partie calcul des valeurs a été bien finalisée. Ce sera alors un bonus.

Outils et techniques à mettre en œuvre

- Bibliothèque Python Gensim pour relire les *word embeddings*
- Bibliothèque Flask pour la partie serveur (si interface)
- Javascript pour la partie front-end (si interface)

Données (j'ai récupéré pour vous les fichiers et je les ai placés sur Moodle dans l'archive datasets_human_judgements.zip)

http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/ - wordsim.csv

https://www.cl.cam.ac.uk/~fh295/simlex.html - SimLex-999.zip

http://www.cs.ucf.edu/~seansz/rel-122/ - Rel-122 – rel122-norms.txt

https://sites.google.com/site/kenmcraelab/norms-data/cos_matrix_brm_IFR.xlsx?attredirects=0 – McRae – cos_matrix_brm_IFR.xlsx

https://nlp.cs.vcu.edu/projects/similarity.html – UMNSRS_relatedness.csv et UMNSRS_similarity.csv

http://www2.mta.ac.il/~gideon/datasets/mturk_771.html - MTURK-771.csv

RG-65 - rg.csv

Miller-Charles – mc.csv

Voici pour finir quelques indicateursde performance pour SimLex99

https://aclweb.org/aclwiki/SimLex-999_(State_of_the_art)

et pour RG-65

https://aclweb.org/aclwiki/RG-65_Test_Collection_(State_of_the_art)