AMELIORATION DE LA RECHERCHE ELATICSEARCH

A L'AIDE DES TRANSFORMERS NETWORKS ET L'INDEXATION PAR DENSE VECTORS

Florian Laborde
AI Lab Intern - Telecom Paris
Etalab
DINUM
florian.laborde@data.gouv.fr

Pavel Soriano-Morales
Lab IA Datascientist
Etalab
DINUM
pavel.soriano@data.gouv.fr

Tam Kien Duong
Open data Team
Etalab
DINUM
tamkien.duong@data.gouv.fr

August 18, 2020

ABSTRACT

Le moteur de recherche actuel des jeux de données Etalab est basé sur une architecture Elasticsearch. Cela permet une recherche par mot-clé et permet de gérer manuellement l'indexation de certains jeux de données. Cependant, les résultats effectifs pour des recherches qui n'ont pas été manuellement indexées sont passables. L'objectif est de conserver un contrôle sur l'affichage de certains résultats et améliorer les résultats de recherches moins courantes.
Les évolutions récentes en Traitement Automatique du Langage permettent d'avoir une meilleure compréhension de la recherche, en particulier quand elle n'est pas par mots-clés et une meilleure cohérence dans les jeux de données proposés.
Nous présentons ici une solution possible pour combiner le framework stable et robuste d'ElasticSearch avec une représentation vectorielle des jeux de données de data.gouv.fr dans l'espoir qu'à l'avenir la compréhension sémantique de la phrase de requête et du contenu textuel de contexte des jeux de données puisse améliorer les recherches des utilisateurs.
Ce rapport propose une piste de solution technique et donne une intuition sur la valeur ajoutée de la représentation vectorielle proposée.

1 Un mot rapide sur les deux algorithmes

1.1 Elasticsearch

Elasticsearch est un moteur de recherche tout-en-un qui gère la recherche l'indexation est plusieurs problématiques de data-engineering dans un package complet maintenu et open-source. La brique élémentaire de la recherche de document est basée sur l'ordre et la fréquence d'apparition de mots-clés. En NLP cela se rapproche de TF-IDF, une méthode efficace et bien connue mais qui ne constitue plus l'état de l'art. A cela s'ajoute un schéma d'indexation et une structure permettant une bonne rapidité de recherche et d'éviter qu'une requête échoue. A l'indexation par fréquence de mot s'ajoute aussi beaucoup d'options de recherches sur les métadonnées, dates, localisations etc.. quand les champs sont renseignés de manière structuré.

1.1.1 Avantages

- Rapide
- Robuste
- Paramétrable manuellement
- Package logiciel

1.2 SBERT

Pour des détails sur les transformer networks, BERT et SBERT vous pouvez lire la présentation "Paper Talk - Transformers NLP" et le pdf correspondant "Support à la présentation". SBERT est basé sur les réseaux de neurones, il permet une compréhension sémantique de la phrase et classifie toute la recherche, contrairement au fonctionnement par mots-clés. Il est capable de comprendre des phrases et des questions. Pour l'aspect indexation, SBERT calcule un vecteur pour chaque jeu de donnée et pour chaque requête. Il cherche ensuite les vecteurs les plus proches et donc les jeux de données les plus proches. De cette manière les paraphrases, synonymes et contexte sont mieux traités que par ElasticSearch. Cependant, deux inconvénients majeurs demeurent dans l'utilisation de SBERT. Il est impossible de paramétrer à la main certain résultats. Les mots-clés seuls sans contexte ne donnent pas des résultats très pertinents. Les exemples sont sous-forme de question car l'ordre peut y être inversé et c'est l'un des moyens les plus bruités au sens des mots-clés de convertir une recherche. On se rapproche avec ce format de l'idée de recherche par chatbot ou encore de ce que l'on appelle 'Conversational Search Engine', des moteurs de recherches conversationnels.

1.2.1 Avantages

- Sens sémantique
- Traitement des phrases et questions
- Indice de similarité des jeux de données
- Traitement Multilingue (Anglais, Allemand, Italien, Français)

2 Benchmark qualitatifs et quantitatifs des différents modules de recherche

2.1 Exemples qualitatifs de différentes recherches

On présente ici des exemples de requêtes haut niveau, sous forme de phrases ou de questions. La colonne de gauche est la requête 'associée' Elasticsearch avec les mots-clés directs correspondant au nom du dataset.

Figure 1: Exemples sémantiques 'Piscines'

L'idée est de montrer que SBERT peut paraphraser les mots clés dans une structure de phrase compliquée et obtenir les mêmes résultats. On ne montre pas le résultat de ElasticSearch avec une requête sous forme de phrase car il n'y a jamais de résultat. On compare également la performance de SBERT avec des entrées de type mots-clés. Dans ce cas on obtient des résultats acceptables mais qui ne sont pas toujours mieux que ElasticSearch. Ainsi, on voit que la valeur ajoutée se trouve dans les recherches de plus haut niveau. L'ajout de mots de liaison ne bruite pas la phrase, au contraire elle permet la bonne mis en relation/causalité de la recherche. Ici, la compréhension est plus forte car le contexte des vacances des enfants est directement relié aux vacances scolaires, sans présence de mot clés ou de synonymes. En effet, les enfants sont tous scolarisés et leurs vacances sont donc les vacances scolaires. Même lorsque l'on pose une question précise, le contexte permet de donner en réponse un jeu de données susceptible de contenir la réponse à la question attendue.

2.2 Benchmark quantitatif sur la liste de mots-clés réservée

Afin de mettre en avant certains jeux de données référencés comme importants, des résultats de recherche à des requêtes manuelles ont été mis en place. Cela permet d'avoir une référence entre des requêtes utilisateurs et le résultat attendu. Les résultats sont difficiles à interprétés car ils ne reflètent pas une pertinence globale de l'algorithme mais bien des réponses très précises et orientés. Par exemple, entreprise renvoie directement à la base SIRENE, il est évident que le mot entreprise tout seul apparaît dans de très nombreux jeu de données et que la base Sirene n'est pas forcément le jeu de données le plus proche de la notion.
La comparaison est donc faite sur la 50aine de mots-clés associés à des jeux de données par rapport aux résultats d'une recherche google limitée aux résultats de data.gouv.fr. On compare donc SBERT à Google. Il est inutile de comparer à ElasticSearch car c'est de là que provient le benchmark et les résultats ont été 'connectés' manuellement. Du tableau brut on en retire différents éléments. La comparaison avec google est l'une des seules possibles mais on sait qu'elle dépend de nombreux facteurs inconnus. De plus on remarque que souvent, soit le jeu de donnée est très connu et indexé en premier résultat soit il n'est pas du tout trouvé.

Figure 2: Exemple 'Vacances Scolaires'

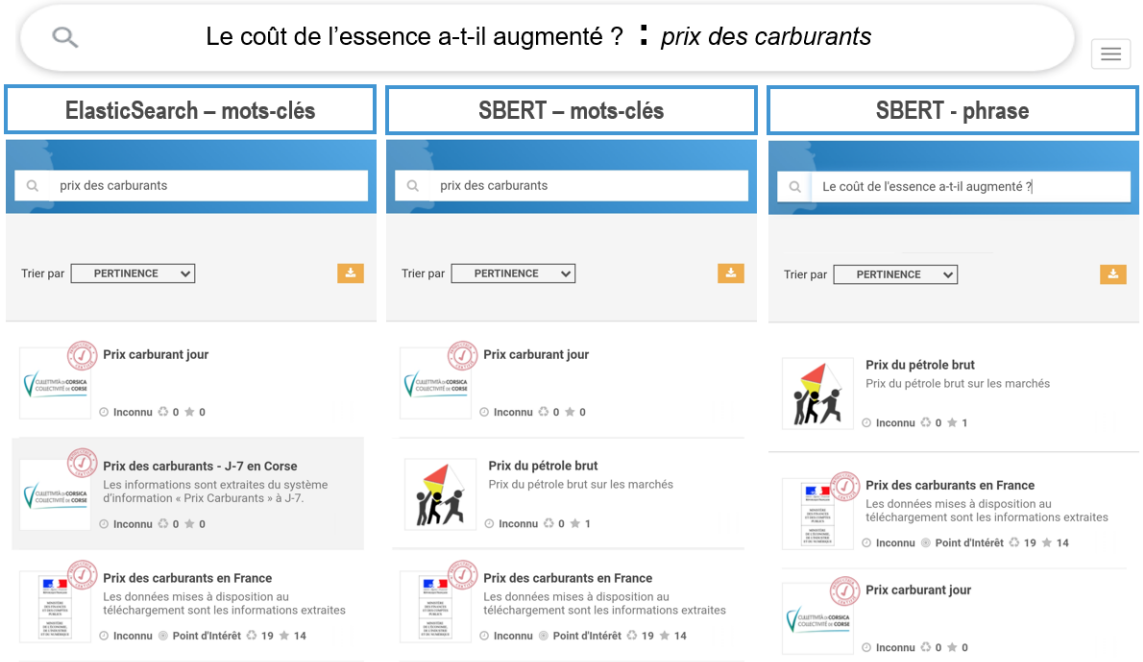

Figure 3: Exemple 'Prix du gasoil'

Table 1: Tableau comparatif des résultats de recherches prédéfinies[1]

Keyword	Expected dataset [2]	Dataset id	Google	SBERT
siren	Base Sirene	5b7ffc618b4c4169d30727e0	1	2
sirene	Base Sirene	5b7ffc618b4c4169d30727e0	1	2
entreprise	Base Sirene	5b7ffc618b4c4169d30727e0	Not Found[3]	Not Found
entreprises	Base Sirene	5b7ffc618b4c4169d30727e0	Not Found	Not Found
siret	Base Sirene	5b7ffc618b4c4169d30727e0	1	9
open damir	Assurance Maladie	54de1e8fc751df388646738b	Not Found	78
opendamir	Assurance Maladie	54de1e8fc751df388646738b	1	Not Found
damir	Assurance Maladie	54de1e8fc751df388646738b	Not Found	Not Found
contours départements	Contours OpenStreetMap	536991b0a3a729239d203d13	Not Found	Not Found
contours départements français	Contours OpenStreetMap	536991b0a3a729239d203d13	Not Found	Not Found
emissions polluantes	/	53ba4c07a3a729219b7bead3	Not Found	32
géofla départements	/	536995f5a3a729239d20487f	16	1
effectifs police municipale	/	5369986ba3a729239d204f55	Not Found	3
marchés public bourgogne	/	5a0effeac751df5832ded865	Not Found	3
Liste gares SNCF	/	59593619a3a7291dd09c8238	Not Found	1
loi de finance 2016	Loi de Finance	5625fba688ee3820e912613c	Not Found	Not Found
lolf 2016	Loi de Finance	5625fba688ee3820e912613c	Not Found	Not Found
formations pas de calais	/	53699058a3a729239d2039a3	Not Found	Not Found
accidents de la circulation	Accidents Circulation	536997eaa3a729239d204dfd	Not Found	4
accidents de la route	Accidents Circulation	536997eaa3a729239d204dfd	Not Found	5
risque de décès un an après accident	/	5630f53b88ee385064531578	Not Found	2
code officiel géographique	COG	58c984b088ee386cdb1261f3	Not Found	8
COG	COG	58c984b088ee386cdb1261f3	Not Found	11
contour commune	Decoupage communal	53699233a3a729239d203e69	Not Found	Not Found
contours communes	Decoupage communal	53699233a3a729239d203e69	Not Found	Not Found
contour communes	Découpage communal	53699233a3a729239d203e69	Not Found	Not Found
code postal	Base officielle codes	545b55e1c751df52de9b6045	Not Found	3
codes postaux	Base officielle codes	545b55e1c751df52de9b6045	Not Found	3
répertoire national des associations	RNA	58e53811c751df03df38f42d	Not Found	6
association	RNA	58e53811c751df03df38f42d	Not Found	Not Found
waldec	RNA	58e53811c751df03df38f42d	1	Not Found
associations	RNA	58e53811c751df03df38f42d	Not Found	Not Found
RNA	RNA	58e53811c751df03df38f42d	Not Found	1
répertoire des associations	RNA	58e53811c751df03df38f42d	1	80
prénoms	Prénoms 1900-2018	5bf42c958b4c4144b0110ce8	Not Found	89
organismes de formation	Org. formations	582c8978c751df788ec0bb7e	Not Found	49
organisme de formation	Org. formations	582c8978c751df788ec0bb7e	Not Found	46
bibliothèques	/	5a4e847cb59508409d014056	Not Found	Not Found
annuaire de l'éducation	/	5889d03fa3a72974cbf0d5b1	Not Found	27
grand débat	/	5c5c3236634f4155110aa4ea	Not Found	7
vie-publique répertoire	/	53699f06a3a729239d20601c	Not Found	Not Found

Contrairement au fonctionnement de SBERT par plus proche voisin qui trouvera forcément toujours le bon jeu de données à partir d'un certain rang dans la recherche. Ainsi, SBERT n'arrive quasiment jamais à renvoyer le bon premier résultat mais retrouve souvent le jdd attendu dans les 10 premier résultats. On remarque aussi le meilleur fonctionnement avec des longues requêtes où l'on a une structure de phrase, un mot explique d'autres ('effectifs de police municipale' ou 'liste des gares SNCF') que des mots-clés isolés ou juxtaposés.

[1] `https://github.com/etalab/datagouv-search-indicator/blob/master/data/datasets.csv`

[2] Position du dataset attendu dans la liste des résultats du moteur de recherche concerné.

[3] Pas dans les 100 premiers résultats.

3 Pistes d'implémentation

L'article suivant résume très bien toute l'approche que l'on décrit.
https://www.elastic.co/blog/text-similarity-search-with-vectors-in-elasticsearch
Les nouvelle version de Elasticsearch >7.3 dispose d'un nouveau champ pour classifier. C'est le type 'dense_vector', c'est là que l'on doit ajouter l'embedding de SBERT pour améliorer les recherches. L'exemple et le script prévu par ElasticSearch dans le lien proposé est basé sur l'encodage du titre des documents avec tensorflow. Utiliser ce module est sûrement plus simple pour la mise à jour de toute l'indexation mais aura des performance moins bonnes qu'en utilisant l'embedding de tout le texte des jdd avec SBERT. Pour ce qui est du calcul des distance, elasticsearch propose déjà une fonction qui calcule la cosine_similarity et qui est compatible.
Voir: https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-script-score-query.html#vector-functions
Malgré tout, je pense que correctement implémenter un embedding pour chaque jdd dans le framework de ElasticSearch actuel peut prendre un peu de temps. Surtout pour s'habituer à ElasticSearch. Mais c'est sûrement le meilleur moyen pour avoir un moteur de recherche plus performant. D'autres liens utiles :

- https://colab.research.google.com/github/dair-ai/covid_19_search_application/blob/master/text_similarity_cord_19.ipynb
- https://medium.com/version-1/vector-based-semantic-search-using-elasticsearch-48d7167b38f

References