Moteur de recherche documentaire en langage naturel

Ying Zhang*, Matthieu Petit Guillaume*, Aurelien Krauth*

* Leviatan, 725 Boulevard Robert Barrier, 73100 Aix-les-Bains, France
y.zhang@leviatan.fr, matthieu@leviatan.fr, aurelien@leviatan.fr

1 Résumé étendu

Nous présentons un résumé étendu de l'article (Zhang et al., 2022) présenté à la journée scientifique LIFT-TAL 2022. À l'heure d'internet, il est de plus en plus facile et accessible de rechercher de l'information sur de nombreux types de sujets. Les archives documentaires et notamment celles générées par la presse spécialisée, jouent un rôle important chez les professionnels qui ont opéré leur transformation vers le numérique. Mais que deviennent ces archives documentaires et notamment les anciens numéros de magazines spécialisés ? Ceux-ci regorgent d'informations riches et précieuses dont la numérisation représente une solution efficace de stockage et un moyen rapide de recherche d'informations précises et pertinentes mis en oeuvre au travers d'une plateforme web. Dans le cadre de ce projet, nous avons stocké 1.1 To de magazines français initiaux aux formats pdf ou jpg.

Nous proposons un nouveau moteur de recherche documentaire en langage naturel, permettant d'accéder facilement à des informations précises dans des archives documentaires de masse. Le projet a été déployé dans un environnement de production avec un partenaire industriel et a été séparé en 4 composants principaux :

1. Prétraitement des magazines : Il s'agit d'un ensemble de prétraitements afin de transformer les magazines en version numérique. Nous avons principalement recours à un traitement OCR (Optical Character Recognition), au regroupement des textes par analyse de leurs informations géométriques, un ensemble de fonctions de nettoyage, une analyse linguistique et une transformation de paragraphe en word-embeddings (Yang et al., 2020).
2. Stockage des données : Les magazines originaux sont stockés dans un bucket AWS S3, les données numériques (sortie de l'étape 1) sont stockées dans un index Elasticsearch.
3. Filtrages des paragraphes à analyser pour une question posée : Nous avons 500 000 paragraphes stockés dans Elasticsearch. Étant donné qu'une question est posée par l'utilisateur, nous avons utilisé plusieurs stratégies de filtrages afin de récupérer uniquement les premiers 1000 paragraphes les plus pertinents pour des raisons de temps d'analyse.
4. Inférence de requête : Nous avons déployé une API de modèle MRC (Machine Reading Comprehension) afin de réaliser l'inférence de requête. Ce modèle a ensuite été ajusté, sur une base de modèle de langue CamemBERT (Martin et al., 2020) et de plusieurs jeux de données disponibles (Keraron et al., 2020; D'Hoffschmidt et al., 2020).