Haystack

haystack by deepset	**End-to-end NLP framework to implement document search**	
	Système d'exploitation :	Multiplateforme
	Développé par :	Deepset
Apache License, Version 2.0 (open source)	Personne de contact :	katy.fokou@smals.be

Fonctionnalités

Haystack est un cadre (*framework)* de développement qui permet d'appliquer des techniques de NLP pour une recherche de documents par mots-clés ou une recherche sémantique en s'appuyant sur les modèles de langage dits *transformer (BERT, RoBERTa, ..)*. La solution finale est construite de façon modulaire comprenant plusieurs nœuds dédiés, chacun, à une tâche NLP. Ces nœuds sont assemblés selon les besoins pour constituer un pipeline. Haystack permet de stocker les textes et les métadonnées dans une variété de bases de données telles que Elasticsearch, FAISS, OpenSearch, etc.

Les principales applications de Haystack sont le QA (*question answering*) et la recherche de documents

Document Search: recherche de documents dans une base de données sur base d'une requête. Le résultat renvoyé par le système est une liste de k documents les plus pertinents par rapport à la requête. Haystack permet d'appliquer aussi bien les techniques type *bag of words* (recherche non sémantique) que les techniques basées sur les modèles de langage neuronaux (recherche sémantique).

Extractive Question Answering (système de questions-réponses) : dans ce cas de figure, l'utilisateur pose une question et le système retrouve les documents les plus pertinents pour ensuite extraire de ces documents les passages qui répondent à la question.

Conclusions & Recommandations

Haystack offre un cadre pratique pour le développement d'une application de recherche sur une collection de documents. Il permet d'intégrer des modules open source divers couvrant les différents étapes du pipeline pour la recherche et le QA: de l'utilisation du *web crawling* pour la collection des textes au déploiement de l'application sous forme d'API. Cet outil est recommandé pour une équipe de data science qui voudrait implémenter une solution open source de recherche sémantique cependant, la qualité des résultats dépendra des modèles de langage utilisés.

Tests et Résultats

Le premier test effectué avec Haystack porte sur la **recherche de documents similaires** à un document donné, ce dernier constituant la requête utilisateur. Pour cela, trois scénarii ont été testés. Dans le premier, les textes et métadonnées sont stockés dans une base de données Elasticsearch et la recherche se fait en utilisant l'algorithme BM25 (correspondance syntactique exacte) implémenté dans Haystack. Dans le deuxième scénario, les textes sont stockés dans la base de données vectorielle FAISS sous forme *d'embeddings* et la recherche se fait avec l'algorithme *approximate nearest neighbors* implémenté dans FAISS (correspondance sémantique). Et enfin, dans le troisième scénario, la recherche se fait en calculant *in-memory* la similarité cosinus entre les textes transformés au préalable en embeddings en utilisant la librairie Sentence Transformer. Dans les deux derniers scenarii, le modèle de langage utilisé est un modèle multilingue de la plateforme HuggingFace. Chaque scénario est déroulé en français et en néerlandais. En observant les résultats obtenus (figure 1), on constate que la recherche sémantique donne de meilleurs résultats en français. Néanmoins, la recherche sémantique est moins performante en néerlandais. La construction modulaire de Haystack permet de mitiger ce problème en intercalant dans notre pipeline un nœud de *routing* qui, selon la langue du texte utilisera l'algorithme BM25 (pour le néerlandais) ou les *embeddings* (pour le français) pour effectuer la recherche.

		Precision	Average Precision
FR	In memory cosine similarity (Embeddings)	0,74	0,70
	Haystack/FAISS (Embeddings)	0,72	0,67
	Haystack/Elasticsearch (BM25)	0,67	0,64
NL	In memory cosine similarity (Embeddings)	0,53	0,45
	Haystack/FAISS (Embeddings)	0,52	0,46
	Haystack/Elasticsearch (BM25)	0,65	0,64

Fig. 1. Résultats recherche de documents similaires.

Le deuxième test de Haystack porte sur **la recherche de documents avec des mots-clés**. Les documents utilisés pour le test sont des documents de la sécurité sociale et la recherche se fait avec des mots-clés tels que « lien familial» ou « faux indépendant ». Pour ce type de recherche, l'utilisation de l'algorithme BM25 disponible dans Haystack donne de meilleurs résultats qu'une recherche basée sur des *embeddings*.

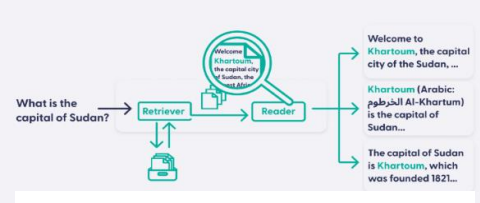

Fig. 2. Exemple d'un système de questions-réponses.
(source : Haystack | Haystack (deepset.ai))

Et enfin le dernier test porte sur les fonctionnalités de **QA**; le schéma ci-dessous (figure 2) illustre le principe de QA tel que présenté par Haystack. Pour tester le système, j'ai utilisé les instructions administratives de l'ONSS comme documents-source et un modèle de langage de QA pour le français (Hugging Face - QA model). Comme on peut le voir dans la figure 3, les résultats sont plutôt mauvais. Cela s'explique entre autre par la qualité du modèle QA qui n'est pas adapté et entrainé sur ce type de documents. Toutefois, la librairie Haystack permet d'affiner les modèles pré-entrainés sur un domaine spécifique.

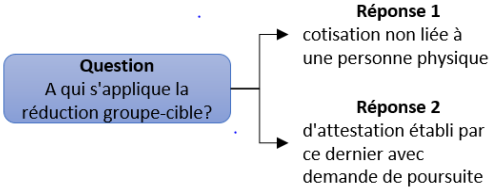

Fig. 3. Test du système de questions-réponses avec des documents de la sécurité sociale.

Conditions d'utilisation & Budget

Haystack est un outil open source sous licence Apache utilisé dans un environnement Python. Pour une installation dans un environnement Windows, il est recommandé d'installer WSL au préalable.