



HAL
open science

Documentary Research in Natural Language (D.R.N.L.): Plateforme d'accès numérique aux archives documentaires en langage naturel

Ying Zhang, Matthieu Petit Guillaume, Aurélien Krauth

► **To cite this version:**

Ying Zhang, Matthieu Petit Guillaume, Aurélien Krauth. Documentary Research in Natural Language (D.R.N.L.): Plateforme d'accès numérique aux archives documentaires en langage naturel. Journées Jointes des Groupements de Recherche Linguistique Informatique, Formelle et de Terrain (LIFT) et Traitement Automatique des Langues (TAL), Nov 2022, Marseille, France. pp.74-83. hal-03846823

HAL Id: hal-03846823

<https://hal.science/hal-03846823>

Submitted on 14 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Documentary Research in Natural Language (D.R.N.L.) : Plateforme d'accès numérique aux archives documentaires en langage naturel

Ying ZHANG¹ Matthieu PETIT GUILLAUME¹ Aurélien KRAUTH¹

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

RÉSUMÉ

Nos travaux de recherche sont motivés par un besoin industriel consistant initialement à la gestion, au stockage et à l'accès à d'anciens journaux et magazines archivés. Notre partenaire industriel possède plusieurs téraoctets de magazines et de journaux. Ces documents sont rédigés dans différentes langues (français, russe, portugais, etc.), répartis dans plusieurs dossiers représentant chacun un type de magazine précis. Ils sont stockés en format PDF et JPG. Dans le cadre de cet article, nous avons centré notre recherche sur le traitement des documents français. Nous proposons une plateforme D.R.N.L. (Documentary Research in Natural Language) permettant le traitement, le stockage et l'accès à des archives documentaires avec quatre composants principaux : 1. Prétraitement des magazines, 2. Stockage des données, 3. Filtrages des documents à analyser pour une question posée et 4. Inférence de requête.

ABSTRACT

Documentary Research in Natural Language (D.R.N.L.): Platform for digital access to documentary archives in natural language

Our research was motivated by an industrial need. It initially involves managing, storing and accessing old archived newspapers and magazines. Our partners own terabytes of magazines and newspapers. These documents are written in different languages (French, Russian, Portuguese, etc.) divided into several folders, each representing a specific type of magazine. They are stored in PDF and JPG format. In the context of this article, we have focused our research on the processing of French documents. We offer a D.R.N.L. (Documentary Research in Natural Language) allowing the processing, storage and access of documentary archives in four main components: 1. pre-processing of magazines, 2. data storage, 3. filtering of documents to be analyzed for a question asked and 4 query inferences.

MOTS-CLÉS : Compréhension automatique de texte, Système de questions-réponses, Analyse automatique des archives documentaires, Moteur de recherche de données dédiées

KEYWORDS: Machine reading comprehension, Question answering system, Automatic analysis of documentary archives, Dedicated data search engine

1 Introduction du projet D.R.N.L.

A l'heure d'internet, il est de plus en plus facile et accessible de rechercher de l'information sur de nombreux de sujets.

Les archives documentaires et notamment celles générées par la presse spécialisée, jouent un rôle important chez les professionnels, même si celles-ci ont opéré leur transformation vers le numérique. Ainsi entre 1990 et 2004, la diffusion annuelle au format papier a reculée de près de 38% (Tessier, 2007).

Mais que deviennent ces archives documentaires et notamment les anciens numéros de magazines et de journaux 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 œuvre au travers d'une interface homme machine (IHM).

Notre partenaire a stocké plusieurs téraoctets de magazines et de journaux. Ces documents sont rédigés dans différentes langues (français, russe, portugais, etc.). Dans la première phase de ce projet, nous avons centré notre recherche sur le traitement des documents français. Cela implique 1.1 téraoctets de magazines et journaux français.

Dans cet article, nous proposons une nouvelle plateforme D.R.N.L. (Documentary Research in Natural Language) permettant le traitement, le stockage et l'accès à des archives documentaires. Notre plateforme a été séparée en deux parties : 1. Développement de l'IHM et 2. Recherche en TAL (Traitement Automatique des Langues). Dans cet article, nous ne présentons pas la réalisation de l'IHM ni la gestion des utilisateurs.

L'objectif de ces travaux de recherche est de stocker des données numériques standardisées et normalisées, de rechercher des données à l'aide de mots-clés, d'inférer des requêtes et de proposer des réponses.

La plateforme est implémentée sous quatre composants principaux :

1. Prétraitement des magazines,
2. Stockage des données,
3. Filtrage des documents à analyser pour une question posée, et
4. Inférence de requête.

Cet article est organisé de la façon suivante. Nous présentons les difficultés rencontrées et les solutions retenues pour les composants principaux ci-dessus dans les sections 2 à 5. Ensuite nous présentons les expérimentations et les déploiements. Enfin, nous concluons et donnons quelques perspectives.

2 Prétraitement des magazines

Les archives documentaires sont stockées en PDF ou en image. Ce composant permet d'unifier les archives documentaires et de nous envoyer les sorties en deux catégories.

La première catégorie contient les sorties des informations des paragraphes internes, tels que le texte brut, les parties du discours, les lemmes, la langue, la position du paragraphe dans la page, l'extraction des mots-clés et la transformation du plongement lexical (word embeddings) etc.

La deuxième catégorie contient les sorties des informations des paragraphes externes, par exemple, le contexte du paragraphe (les informations du paragraphe précédent et du paragraphe suivant), le nom du document, le numéro de la page du paragraphe, le nombre de pages total, et l'année de parution etc.

2.1 Problématiques

Les documents sont hétérogènes. Les mises en page sont incohérentes, même pour différentes pages du même document. Dans un même document, nous observons des pages à deux colonnes, des pages à trois colonnes, et des pages divisées en partie haute, centrale et basse etc. Le nombre et la largeur des colonnes ne sont pas identiques avec de nombreux petits blocs d'annonces. Les en-têtes et les pieds des pages amènent également beaucoup d'informations redondantes. Il existe des magazines bilingues (en français-anglais ou en français-allemand). Ces éléments augmentent la difficulté de la normalisation.

2.2 Solutions

Notre solution est principalement divisée en quatre étapes. La première étape consiste en une analyse d'OCR (Smith, 2007), la détection de position de chaque texte et une génération des PDF éditables. Ensuite nous avons réalisé une manipulation et analyse d'objets géométriques basés sur des coordonnées cartésiennes. Nous fusionnons deux textes dans un paragraphe, si la distance entre ces deux textes consécutifs est inférieure à 5 unités de longueur (Gillies, 2013). La sortie de cette étape est un ensemble de paragraphes.

La deuxième étape est le nettoyage. Nous nettoyons toutes les en-têtes et les pieds de page en se basant sur la position et la longueur du texte. Le projet actuel ne permet pas de nettoyer le contenu de l'annonce. Nous discuterons de ce point plus loin dans la section perspectives.

Dans la troisième étape, nous utilisons SpaCy (DataCamp, 2020) afin d'effectuer une analyse linguistique sur chaque paragraphe, comprenant principalement la détection de la langue, la

tokenisation et la lemmatisation. Enfin nous utilisons Yake (Campos et al., 2020) afin de faire une extraction de mots-clés.

Dans la dernière étape, nous transformons les textes en plongement lexical (word embeddings) en utilisant un modèle de langage pré-entraîné. Bien que dans la première phase du projet, nous ne traitons que des documents français, pour une mise à l'échelle plus facile vers des plateformes multilingues, nous choisissons le modèle multilingue de l'équipe Google (Chidambaram et al., 2019; Yang et al., 2020) pour mettre en œuvre cette étape.

3 Stockage des données

Le stockage des données est séparé en deux parties. 1. Les documents originaux et les PDF éditables sont stockés dans un bucket AWS S3 (AWS, 2006), 2. Les données numériques sont stockées dans un index Elasticsearch (Elasticsearch, 2018). Les PDF éditables sont utilisés pour l'affichage sur l'IHM. Les données numériques sont utilisées pour le calcul de la réponse.

À ce stade de cet article, nous avons stocké 1.1 téraoctets d'archives documentaires en français dans le S3 et plus de 500 000 paragraphes dans l'index d'Elasticsearch.

4 Filtrages des paragraphes à analyser pour une question posée

Le système D.R.N.L. permet de traiter deux types de requête : 1. Recherche à l'aide de mot(s)-clé(s) et 2. Inférence d'une question posée.

Pour traiter le premier type de requête, nous utilisons le résultat de l'extraction de mots-clés stockés dans les données numériques. Si ce résultat est vide pour un mot-clé indiqué par l'utilisateur, nous utilisons le résultat de lemmatisation stocké dans les données numériques (voir la section 2.2). Nous ne détaillons pas cette implémentation. Dans cette section et la suivante, nous nous concentrons sur le cœur de notre recherche, à savoir comment recommander des réponses à une question donnée dans le contexte de données massives et de contraintes par les besoins industriels.

4.1 Problématiques

Dans la phase initiale du projet, nous avons réalisé un prototype avec moins de 500 paragraphes stockés dans l'index d'Elasticsearch. Dans ce prototype, il n'y a pas d'étape de filtrage, et nous obtenons un bon fonctionnement.

Une fois que la quantité de données dans Elasticsearch augmente, des problèmes de rapidité et de passage à l'échelle sont apparus. Étant donné que notre processus d'inférence utilise un modèle MRC

(Machine Reading Comprehension) affiné et basé sur CamemBERT (Martin et al., 2020), son fonctionnement consomme beaucoup de ressources de calcul. Nous le présenterons dans la section 5. Avec notre test et calcul, une analyse de 500 000 paragraphes sans filtrage, prend environ 45 minutes afin de recevoir les résultats. Ce test est basé sur un déploiement sur machine GPU NVIDIA Tesla V100.

D'autre part, la précision des réponses a également chuté de manière significative. Nous voulons souligner notre observation sur le traitement des noms propres. Par exemple, pour la question « *Qui est le président de Dior ?* », en supposant que nous ayons les deux textes suivants : 1. « *Le président de Chanel est Bruno Pavlovsky.* » et 2. « *La nomination inattendue de Pietro Beccari à la tête de Dior récemment.* », le modèle MRC préfère recommander « *Bruno Pavlovsky* » comme réponse à la question. La raison est que la parenté sémantique de ces deux mots (*Dior* et *Chanel*) est trop élevée, autrement dit, les word embeddings de ces deux mots sont très similaires. Au contraire, la relation sémantique entre « *La nomination inattendue de Pietro Beccari à la tête de Dior récemment.* » et « *Qui est le président de Dior ?* » est plus éloignée que la relation sémantique entre « *Le président de Chanel est Bruno Pavlovsky.* » et « *Qui est le président de Dior ?* ». Lorsque le nom propre dans la question fait référence à une marque ou à un objet moins connu, comme un modèle précis d'une marque de téléphone mobile, ou un nouveau parfum, le retour du système sera encore moins pertinent. Nous observons que les word embeddings de ces noms propres sont convertis en un vecteur fixe par le modèle.

En raison des contraintes industrielles, nous ne pouvons pas construire une base de connaissance afin de traiter ce problème. Face à une grande quantité de données, il est extrêmement difficile pour notre prototype de faire des inférences correctes sur des questions avec un nom propre.

Pour les raisons ci-dessus, nous avons ajouté le filtrage comme étape essentielle. L'objectif de cette étape : étant donnée une question posée par l'utilisateur, nous avons utilisé plusieurs stratégies de filtrages afin de récupérer les premiers 1000 paragraphes les plus pertinents.

4.2 Solutions

Lors d'une recherche Elasticsearch, un score est calculé pour chaque document dans le résultat. Ce score représente la pertinence du document afin de pouvoir en classer les résultats.

Une question est généralement relativement courte, mais un paragraphe est régulièrement long. En raison d'une grande différence de longueur entre la question et le paragraphe, Elasticsearch donne des scores plus élevés aux paragraphes courts. Par exemple, pour une question « *Quand la société Dior a-t-elle été créée ?* », nous avons deux textes, le premier texte est un titre d'un article « *Société Dior et sa création* », le deuxième texte est un paragraphe décrivant l'historique de croissance de la société Dior. Le score du premier texte est plus élevé que le deuxième texte. C'est parce que Elasticsearch calcul son score selon la fréquence du terme, la fréquence inverse du terme et la longueur du champ (Denton, 2017).

Cette étape de filtrage prend donc en compte trois axes dans le calcul :

1. La proposition du moteur de recherche d'Elasticsearch et la longueur du texte. Si nous avons assez de paragraphes candidats, nous ignorons les paragraphes courts (le nombre de tokens est inférieur à 20).
2. La cohérence du nom propre. S'il existe un nom propre dans la question, celui-ci doit également exister dans le paragraphe courant.
3. Le filtrage de la similarité cosinus basé sur le word embeddings entre les différents paragraphes candidats. Si deux paragraphes candidats ont une similarité très élevée, nous n'en prenons qu'un pour l'analyse. C'est principalement parce qu'un même contenu ou un même sujet est rapporté à plusieurs reprises par différents éditeurs ou journaux.

Enfin, nous prenons les premiers 1000 paragraphes les plus pertinents comme les paragraphes candidats, analysés par le modèle MRC.

5 Inférence de requête

Du point de vue du domaine de la connaissance, un système de questions-réponses peut être divisé en deux types : « domaine fermé » et « domaine ouvert ». Si les archives documentaires appartiennent toutes au même domaine, alors l'approche d'une ontologie dédiée et une base de connaissance pourront grandement améliorer la précision des réponses (Lopez et al., 2007; Otegi et al., 2015; Siciliani, 2018; Franco et al., 2020).

Dans le cadre du projet D.R.N.L., une grande quantité de données brutes hétérogènes couvre de nombreux domaines tels que la politique, les affaires, le divertissement, et la mode etc. Avec de nombreuses contraintes industrielles, nous devons développer une solution rapide et efficace. Enfin, nous nous concentrons sur l'étude du modèle MRC (Machine Reading Comprehension).

5.1 État de l'art

Le MRC est un sujet très important dans le domaine TALN (Traitement Automatique du Langage Naturel). Des recherches récentes sur l'utilisation des modèles de langue contextualisés pré-entraînés avec des architectures à base de Transformer (Vaswani et al., 2017) ont obtenu un grand succès dans de nombreuses tâches de TALN.

CamemBERT (Martin et al., 2020) est considéré comme l'un des meilleurs modèles français. Nous utilisons ce modèle comme le modèle de langue contextualisé pré-entraîné pour commencer notre expérimentation.

5.2 Modèle MRC ajusté

L'équipe Google AI Langue a expliqué la méthode d'ajustement d'un système MRC basé sur le modèle de langue BERT (Devlin et al., 2019) en utilisant les jeux de données SQuAD v1.1 (Rajpurkar et al., 2016) et SQuAD v2.0 (Rajpurkar et al., 2018). Nous n'aborderons pas le détail de cette méthode dans cet article.

Nous avons bien respecté les consignes précisées dans (Devlin et al., 2019) et avons gardé les valeurs par défaut des hyper-paramètres proposées par Transformer (Huggingface, 2020) pour ajuster notre modèle MRC en utilisant trois jeux de données publiques et un jeu de données privées de notre partenaire. Les trois jeux de données en source ouverte proviennent du projet Piaf (Bras, 2019), du projet FQuAD (D'Hoffschmidt et al., 2020) et du projet French-SQuAD (Kabbadj, 2018). Le F1-score atteint une valeur moyenne de 80.6 sur cet ensemble de jeux de données.

6 Déploiements et expérimentations

Le système D.R.N.L. est un système qui permet d'accéder au service en temps réel, par conséquent, la performance de ce système est très importante. En termes de déploiement et d'optimisation de code, nous avons fait plusieurs tentatives. Dans ce système, la partie la plus gourmande en ressources est l'analyse du modèle MRC des 1000 paragraphes les plus pertinents (voir la section 4.2) pour une question donnée.

Nous transformons les analyses en forme d'itération *question1 : [text1, text2, text3...text1000]* vers les analyses en forme *[question1 : text1, question1 : text2, question1 : text3...question1 : text1000]*, en utilisant le traitement « batching » du pipeline de Transformer (Huggingface, 2021). Pour garantir un temps de traitement raisonnable, cette partie est déployée sur une machine GPU NVIDIA Tesla V100.

Nous avons mis en place d'autres solutions d'optimisation du programme. Par exemple, nous pré-analysons et stockons les word embeddings et les résultats d'analyse linguistique dans l'index d'Elasticsearch. Ces optimisations ont permis une importante amélioration du temps de traitement, qui est passé de dizaines de secondes, voire 1 minute, à environ 4-5s.

Nous avons testé ce système avec 4050 questions pré-annotées. 3702 questions présentent des réponses et 348 questions liées aux bons documents mais n'ayant pas de réponses précises. Nous proposons au maximum 10 réponses pour chaque question selon l'ordre du score de la fiabilité.

Dans les 3702 questions avec réponses, nous avons 3077 bonnes réponses trouvées. Parmi ces 3077 bonnes réponses, 2906 réponses ont reçu un score de fiabilité élevé du MRC (>0.2), 171 réponses ont reçu un score de fiabilité faible (<0.2). Le système a proposé 355 mauvaises réponses mais trouvées

dans le bon document. Enfin, le système a proposé 271 mauvaises réponses qui ne sont pas présentes dans les bons documents.

Dans les 348 questions sans réponses, nous avons retrouvé 160 bons documents (79 questions ont reçu une réponse avec un score de fiabilité élevé et 81 questions ont reçu une réponse avec un score de fiabilité faible). 188 questions n'ont pas pu être rattachées au bon document.

7 Conclusions et perspectives

Dans cet article, nous présentons une nouvelle plate-forme D.R.N.L. basée sur les technologies les plus récentes en TALN, permettant de stocker et d'accéder de façon plus efficace et plus durable aux archives documentaires.

Les perspectives de cette recherche sont multiples et concernent aussi bien le court terme que le long terme. En ce qui concerne le court terme, il s'agit d'un passage à échelle en multilingue. Pour le long terme, nous pouvons distinguer deux perspectives.

Notre IHM D.R.N.L. permet non seulement de présenter les dix premiers résultats d'une question, mais également de gérer un espace membre dédié afin que les utilisateurs puissent partager et contribuer aux contenus les plus pertinents associés aux sujets donnés. Nous avons prévu de construire un système de recommandation en utilisant les contributions des utilisateurs.

La deuxième perspective a été mentionnée aux sections 2.1 et 2.2. Il s'agit d'un modèle de classification de textes afin d'identifier les annonces. Aujourd'hui les annonces nous amènent beaucoup de bruits dans le stockage. Afin d'entraîner ce modèle, nous voulons utiliser la méthode présentée dans (Sun et al., 2019). Nous commencerons à annoter les données dans un proche avenir.



FIGURE 1 : Interface d'inférence pour la question « Qui est le directeur marketing de Darty ? »

Références

- AWS. (2006). Amazon Simple Storage Service. Available at: <https://aws.amazon.com/fr/s3/>
- BRAS, M. (2019). Piaf. Available at: <https://www.etalab.gouv.fr/ia-decouvrez-et-participez-au-projet-piaf-pour-des-ia-francophones>
- CAMPOS, R., MANGARAVITE, V., PASQUALI, A., JORGE, A., NUNES, C., & JATOWT, A. (2020). YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509, 257–289. DOI : <https://doi.org/10.1016/j.ins.2019.09.013>
- CHIDAMBARAM, M., YANG, Y., CER, D., YUAN, S., SUNG, Y. H., STROPE, B., & KURZWEIL, R. (2019). Learning cross-lingual sentence representations via a multi-task dual-encoder model. *ACL 2019 - 4th Workshop on Representation Learning for NLP, Repl4NLP 2019 - Proceedings of the Workshop*, 250–259. DOI : <https://doi.org/10.18653/v1/w19-4330>
- D’HOFFSCHMIDT, M., VIDAL, M., BELBLIDIA, W., & BRENDLÉ, T. (2020). FQuAD: French question answering dataset. DOI : <https://doi.org/10.18653/v1/2020.findings-emnlp.107>
- DATA CAMP. (2020). spaCy. *Python Cheat Sheet*. Available at: <https://www.datacamp.com/cheat-sheet/spacy-cheat-sheet-advanced-nlp-in-python>
- DENTON, A. (2017). *Making your search not suck with Elasticsearch — Part 6: Totally irrelevant*.
- DEVLIN, J., CHANG, M. W., LEE, K., & TOUTANOVA, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*. DOI : <https://doi.org/10.18653/v1/N19-1423>
- ELASTICSEARCH. (2018). Elasticsearch. *International Journal of Modern Trends in Engineering & Research*, 5(5), 23–28.
- FRANCO, W., VIKTOR, C., OLIVEIRA, A., MAIA, G., BRAYNER, A., VIDAL, V. M. P., CARVALHO, F., & PEQUENO, V. M. (2020). Ontology-based question answering systems over knowledge bases: A survey. *ICEIS 2020 - Proceedings of the 22nd International Conference on Enterprise Information Systems*. DOI : <https://doi.org/10.5220/0009392205320539>
- GILLIES, S. (2013). *The shapely user manual, Version 1.3*. December 31, 2013.
- HUGGINGFACE. (2020). Fine-tuning BERT on SQuAD1.0. <https://huggingface.co/transformers/v2.8.0/examples.html#squad>
- HUGGINGFACE. (2021). Pipeline batching. https://huggingface.co/docs/transformers/main_classes/pipelines#pipeline-batching
- KABBADI, A. (2018). Something new in French Text Mining and Information Extraction (Universal Chatbot): Largest Q&A French training dataset (110 000+).
- LOPEZ, V., UREN, V., MOTTA, E., & PASIN, M. (2007). AquaLog: An ontology-driven question answering system for organizational semantic intranets. *Web Semantics*. DOI : <https://doi.org/10.1016/j.websem.2007.03.003>
- MARTIN, L., MULLER, B., SUAREZ, P. J. O., DUPONT, Y., ROMARY, L., DE LA CLERGERIE, É. V., SEDDAH, D., & SAGOT, B. (2020). CamemBERT: A tasty French language model. DOI : <https://doi.org/10.18653/v1/2020.acl-main.645>
- OTEGI, A., ARREGI, X., ANSA, O., & AGIRRE, E. (2015). Using knowledge-based relatedness for information retrieval. *Knowledge and Information Systems*. DOI : <https://doi.org/10.1007/s10115-014-0785-4>

- RAJPURKAR, P., JIA, R., & LIANG, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*. DOI : <https://doi.org/10.18653/v1/p18-2124>
- RAJPURKAR, P., ZHANG, J., LOPYREV, K., & LIANG, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*. DOI : <https://doi.org/10.18653/v1/d16-1264>
- SICILIANI, L. (2018). Question answering over knowledge bases. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. DOI : https://doi.org/10.1007/978-3-319-98192-5_47
- SMITH, R. (2007). Tesseract OCR Engine. *Lecture. Google Code. Google Inc.*
- SUN, C., QIU, X., XU, Y., & HUANG, X. (2019). How to Fine-Tune BERT for Text Classification? *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. DOI : https://doi.org/10.1007/978-3-030-32381-3_16
- TESSIER, M. (2007). La presse au défi du numérique. In *RAPPORT AU MINISTRE DE LA CULTURE ET DE LA COMMUNICATION*.
- VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł., & POLOSUKHIN, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*. DOI : <https://doi.org/10.48550/arXiv.1706.03762>
- YANG, Y., CER, D., AHMAD, A., GUO, M., LAW, J., CONSTANT, N., ABREGO, G. H., YUAN, S., TAR, C., SUNG, Y., STROPE, B., & KURZWEIL, R. (2020). *Multilingual Universal Sentence Encoder for Semantic Retrieval*. 87–94. DOI : <https://doi.org/10.18653/v1/2020.acl-demos.12>