IA Générative : Des fondements historiques à l’utilisation des Large Language Models (LLMs) pour l’extraction d’information dans des documents

I. Introduction

L’IA générative peut se définir comme une sous partie de l’intelligence artificielle visant à générer de l’information à partir de la connaissance des données sur lesquelles elle a été entrainée. Cette information peut être de nature différente (texte, image, vidéo, son, etc.). L’information retranscrite peut être existante ou nouvelle et sucite un nombre de cas d’usage extremement varié. Aujourd’hui, l’engoument de l’IA générative touche le monde entier. Mais comment en est on arrivé là ? Quelles ont été les prémisses de cette avancée majeure dans le domaine de l’intelligence artificielle ? C’est ce que nous allons voir en remontant à ces origines.

Historiquement, le terme d’Intelligence Artificielle a été défini pour la première fois par l’informaticien John McCarthy (1927-2011) à Dartmouth en 1956 lors de la première conférence dans le domaine « Dartmouth Summer Research Project on Artificial Intelligence ». Cette discipline qu’est l’IA a hérité des fondements théoriques véritablement établis une décénie plus tôt, entre les années 1940 et 1950. En 1943, McCulloch, cybernéticien et neurophysiologue & Pitts, logicien et psychologue cognitif se sont inspirés des travaux réalisés par Alan Turing en 1936 sur une description mathématique de ce qu’est une machine universelle (On Computable Numbers [1]). Ils ont proposé la première modélisation mathématique d’un neurone biologique et ont montré que des éléments simples connectés dans un réseau neuronal peuvent avoir une immense puissance de calcul (A Logical Calculus of the ideas Imminent in Nervous Activity [2]). Dans cette même période, John von Neumann et d'autres chercheurs réalisèrent des travaux sur l'architecture des ordinateurs, jetant ainsi les bases du développement de l'IA. Puis en 1950, Alan Turing proposa le "test de Turing" pour évaluer l'intelligence des machines. Ces années ont marqué le commencement de l’IA en opposant deux courant très différents, l’IA symbolique (travaux d’Alan Turing) de l’IA dite connexioniste (inspirée de la biologie – travaux de McCulloch & Pitts).

Ces deux mouvements se sont régulièrement affrontés au cours des années qui ont suivi. L’IA connexioniste a vite été limité par la faible capacité de calcul des machines informatiques entre les années 1960 et 1980 laissant l’IA symbolique prendre le dessus sur les travaux de recherche. Les premiers concepts d'IA générative, développés entre les années 1950 et 1980, se basaient sur des algorithmes et des systèmes autonomes, utilisant des règles et des grammaires formelles (IA symbolique).

Evoqué dès 1975 et aboutis en 1986, le concept de rétropropagation au sein des réseaux de neurones couplé à l’amélioration des calculateurs informatiques a remis l’IA connexioniste sur le devant de la scène. Jusqu'en 1990, des systèmes experts et des algorithmes de production automatique de texte ont émergé, jetant les bases de la génération de contenu. Les réseaux neuronaux ont alors commencé à être utilisés pour des tâches génératives simples. En 1997, le logiciel AARON, développé par Harold Cohen, est capable de créer des dessins de manière autonome, marquant une avancée significative dans la génération d'art par des machines.

Les années 2000 marquent une aire de percées pour l'IA générative qui commence à tirer parti des progrès en apprentissage machine et en réseaux neuronaux. Les algorithmes deviennent capables de générer des textes plus cohérents et des images plus réalistes. En témoigne l’avancée majeure obtenues par Geoffrey Hinton qui introduit les "deep belief networks", une forme d'architecture de réseau neuronal profonde qui joue un rôle clé dans l'évolution des modèles génératifs. Les deep belief networks sont des architectures de réseaux de neurones profonds conçues pour l'apprentissage

non supervisé, capables de modéliser des distributions complexes et d'apprendre des représentations hiérarchiques des données. Ils ont été une étape clé dans l'évolution des techniques d'apprentissage profond et conduiront quelques années plus tard (2014) au développement des modèles GANs (Generative Adversarial Networks) par Ian Goodfellow et ses collègues. Cette technique révolutionnaire utilise deux réseaux neuronaux en compétition (un générateur et un discriminateur) pour produire des contenus extrêmement réalistes en se basant sur une estimation de la loi de distribution des données d'entrainement.

En 2015, OpenAI est fondée, avec un focus sur le développement de modèles d'IA avancés, y compris ceux utilisés pour des tâches génératives. C'est en 2017 qu'est probablement apparu l'innovation la plus importante à ce jour en matière d'IA générative avancées. Le modèle de Transformer est introduit par Vaswani et al. dans l'article "Attention is All You Need [3]". Cette architecture basée sur un empilement successif de couches encodeur/decodeur révolutionne le traitement du langage naturel (NLP) et deviendra par la suite la base de nombreux modèles génératifs de texte. Il faudra attendre 3 ans (2018) avant qu'Open AI ne dévoile au grand jour leur premier modèle avancé d'IA générative GPT (Generative Pretrain Transformers). Il s'agit d'un modèle reprenant une partie de l'architecture transformers initiale, le decodeur. S'en suivra le modèle GPT de deuxième génération (GPT-2), publié en 2019, suivi par GPT-3 en 2020, qui impressionne par sa capacité à générer du texte presque indiscernable de celui écrit par des humains.

C'est véritablement en novembre 2022 avec la sortie de ChatGPT que nous entrons dans l'air des larges langages models (LLMs). Aujourd'hui, il existe une multitude de LLMs développés par différents organismes (Open AI, Meta, Anthropic, Mistral, Google, etc.). Bien sûr, au vu de la puissance de ces modèles dans leur capacité à générer de l'information, cela pose forcément des questions sur le plan éthique et sur la régulation de l'IA au vu de son impact sur la société actuelle.

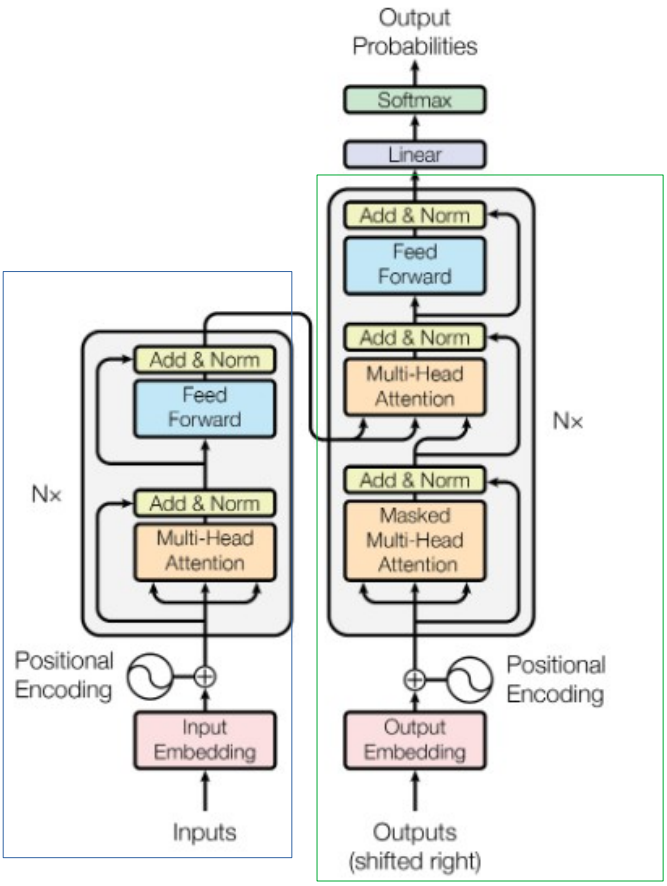
Nous allons maintenant voir comment nous au sein d'Aquila utilisons cette technologie pour répondre aux enjeux de nos clients tout en contenant le niveau de risque que peuvent représenter ces modèles. Nous allons commencer par comprendre ce qu'est un LLM puis, nous présenterons un cas d'étude spécifique développé au sein d'Aquila.

II. Large Language Models (LLMs)

A) Transformers : Concept, Architecture et Fonctionnement

Les LLMs sont des modèles d'IA générative extremement puissants basés sur une architecture de type Transformers. Les Transformers ont vu le jour en 2017 avec le célèbre article scientifique « Attention is all you need [3] ». L'architecture de base d'un Transformer se décompose en deux blocs distincts, le premier étant une succession de couches encodeurs, le second étant une succession de couches décodeurs. Que se soit pour la partie encodeur comme pour la partie décodeur, chaque unité d'encodeur/décodeur se décline en sous couches et il est important de comprendre le fonctionnement de chacune d'entre elle.

Bloc Encodeur



Bloc Décodeur

Figure 1: The Transformer - model architecture.

Input/Output Embedding :

Lorsque nous étudions un cas d'étude spécifique pour un client, les données dont nous disposons peuvent être de nature différentes (textuelle, images, son, etc). L'exemple que nous allons développer dans la section suivante est basé sur des données textuelles. Par conséquent, tous les méchanismes décris par la suite seront basés sur des données textuelles. Les entrées du modèles étant des données textuelles, nous pouvons assimilé un mot à un jeton. Ces jetons doivent être transformées en valeurs numériques pour pouvoir être utilisables par notre modèle. Cette transformation se définit comme la phase d'embedding. Cela consiste simplement en une représentation vectorielle des données d'entrée. Pour cela nous utilisons un modèle spécifique à cette tâche que l'on appelle modèle d'embedding.

Encodeur :

L'objectif de l'encodeur est de générer un modèle de langage en créant des représentations riches et contextuelles des séquences d'entrée, permettant ainsi au modèle de capter les relations et les dépendances entre les jetons de manière efficace. Ils sont bidirectionnels. Cela signifie qu'ils regarderont dans les deux sens lors de l'encodage des données. La sortie est générée de manière non autorégressive. Chaque jeton en sortie est calculé en même temps.

Décodeur :

Le bloc Décodeur au sein du modèle de type Transformers est très proche de celui de l'Encodeur. Au niveau du principe de fonctionnement, la grande différence est le caractère unidirectionnel pour

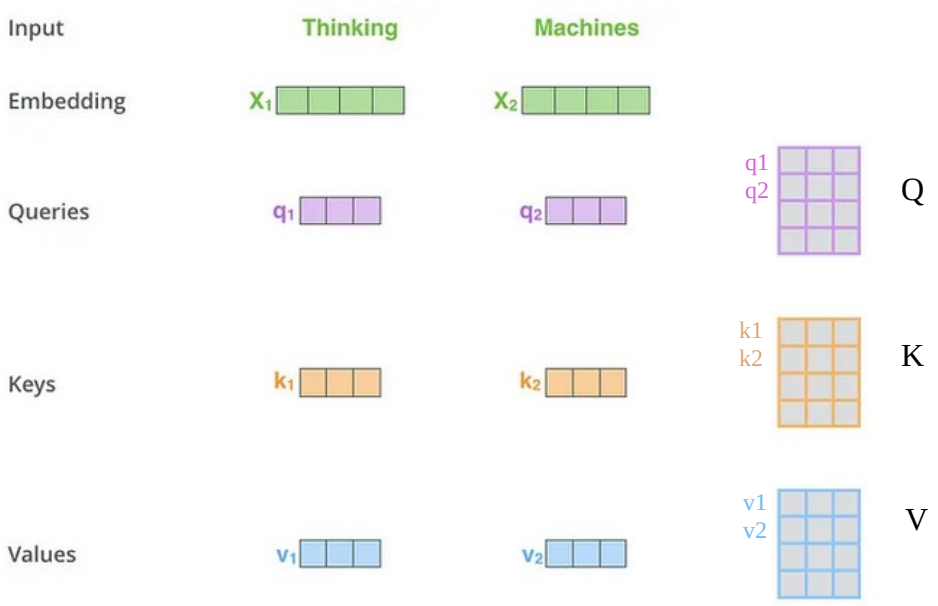
traiter les données. Pour ce qui est de l'architecture, la différence réside dans l'ajout d'une étape supplémentaire dans les sous couches nommée « Masked Multi-Head-Attention ». Cette notion est extrêment importante car c'est elle qui assure la caractéristique autoregressive du modèle, c'est à dire la capacité du modèle à faire de la génération de données. En effet, le terme « Masked » correspond au fait que le Décodeur n'a pas accès aux jetons se situant après le jeton actuellement traité. Les jetons aux positions suivantes sont par conséquent masqués et l'information sur les données est traité en tant que tel par le modèle. Cela garantit que les prédictions pour la position i ne peuvent dépendre que des sorties connues aux positions inférieures à i.

Positional Encoding :

Comme nous pouvons le voir sur la Figure 1 ci-dessus, le bloc encodeur comme décodeur possède une composante dite de « Positional Encoding ». Il s'agit d'un concept permettant au modèle d'injecter de l'informations sur la position relative ou absolue des jetons dans la séquence. C'est un point extrêment important notamment pour la cohérence de ce qui sera générée en sortie du modèle.

Multi-Head-Attention :

Le mechanisme d'attention peut être définit comme une fonction mappant une requête et un ensemble de paires clé-valeur à une sortie, où la requête, les clés, les valeurs et la sortie sont tous des vecteurs. Le résultat est calculé comme une somme pondérée des valeurs, où le poids attribué à chaque valeur est calculé par une fonction de compatibilité de la requête avec la clé correspondante. Chaque jeton est représenté par un vecteur dans le méchanisme d'attention. Il faut donc avoir à l'esprit que nous travaillons à partir de matrice numérique. Voici un exemple pour mieux visualiser la représentation des matrices nécessaires au calcul de l'attention :

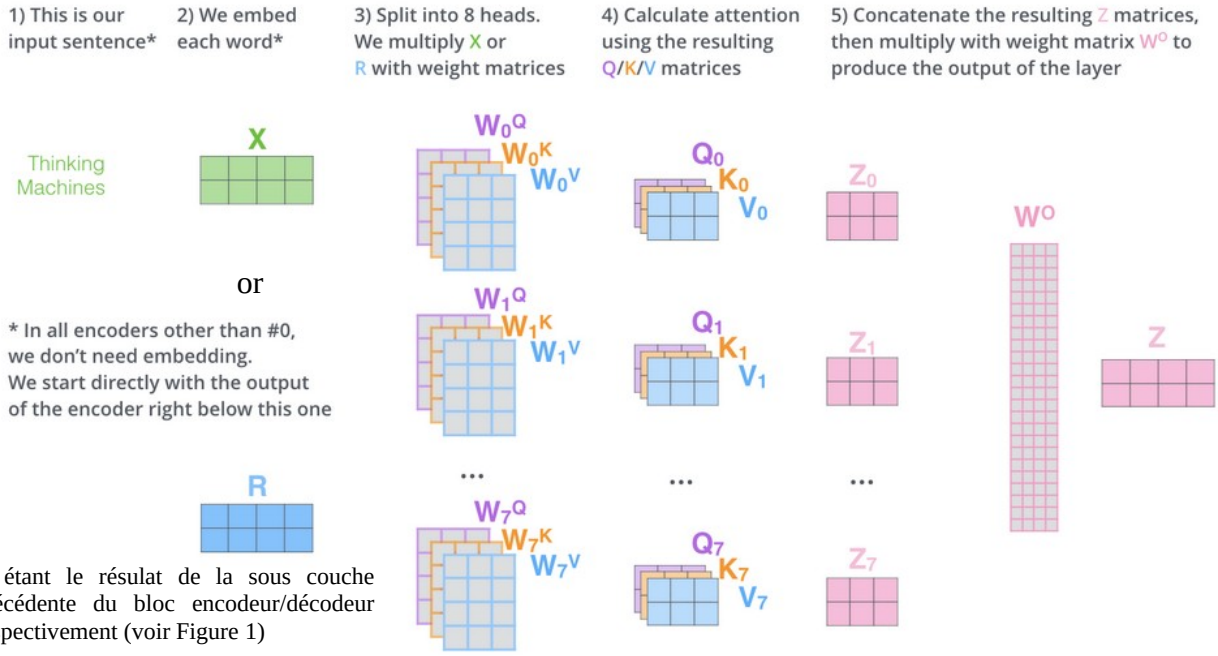


Par conséquent, l'attention se calcule de la manière suivante :

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

d_k est la dimension des matrices Q, K et V. L'application de la fonction softmax permet d'obtenir une pondération normalisée dont la somme pour chaque vecteur étant égale à 1.

Le « Multi-Head-Attention » se compose de plusieurs couches d'attention fonctionnant en parallèle. Cette parallélisation a pour objectif de permettre à différents sous-espaces de représentations de communiquer entre eux et de combiner les informations qu'ils renferment distinctement, là où l'utilisation d'un module unique d'auto-attention calcule une moyenne globale de ces représentations, au détriment de leur précision. Pour obtenir le résultat final de l'attention, il suffit de concaténer les résultats de chaque couche d'attention calculés en parallèle puis, de faire une somme pondérée avec la matrice W0 (matrice de poids).



Add & Norm :

Cette étape est très importante car elle permet tout simplement la convergence de l'algorithme d'optimisation lors des calculs de gradients. Elle empêche l'explosion du gradient (vanishing gradient) lorsque la profondeur du réseau augmente.

Feedforward :

Le réseau feedforward dans les Transformers affine et transforme les représentations intermédiaires des jetons après le mécanisme d'attention. Il ajoute une couche de transformation non linéaire, ce qui permet au modèle de capturer des caractéristiques plus complexes et de mieux représenter les données.

Maintenant que chaque composante de l'architecture des Transformers a été expliquée, nous allons pouvoir aborder les LLMs.

B) Comprendre les LLMs

Le terme de « Large Language Model » se caractérise par plusieurs aspects liés à la taille (de quelques GB à plusieurs centaines de GB), à la complexité et à la capacité de ces modèles de langage. Il faut tenir compte du nombre de paramètres au sein du modèle qui se compte souvent en milliards de paramètres, au volume de données utilisé lors de l'entrainement, à la capacité de traitement d'information complexe, ainsi qu'à leur polyvalence car ils sont capables de traiter une vaste gamme de tâches linguistiques sans nécessiter une spécialisation explicite pour chacune.

Le choix du type de LLM a utilisé dépend fortement du cas d'étude que nous traitons. Chaque architecture est plus ou moins adaptée à la tâche que nous souhaitons accomplir. Nous pouvons distinguer 3 types d'architecture différentes en repartant de la définition de ce qu'est un Transformers :

1. L'architecture Encodeur-Décodeur

   cas d'étude: traduction, question – réponse, génération text to text

   exemples : T5 (Text-To-Text Transfer Transformer), BART (Bidirectional and Auto-Regressive Transformers)

2. L'architecture Encodeur

   cas d'étude: compréhension de texte, classification de texte, similarité de phrase, extraction de feature

   exemples : modèles de type BERT (Bidirectional Encoder Representations from Transformers)

3. L'architecture Décodeur

   cas d'étude: génération, question – réponse, résumé de texte,

   exemples : modèles de type GPT (Generative Pre-trained Transformer)

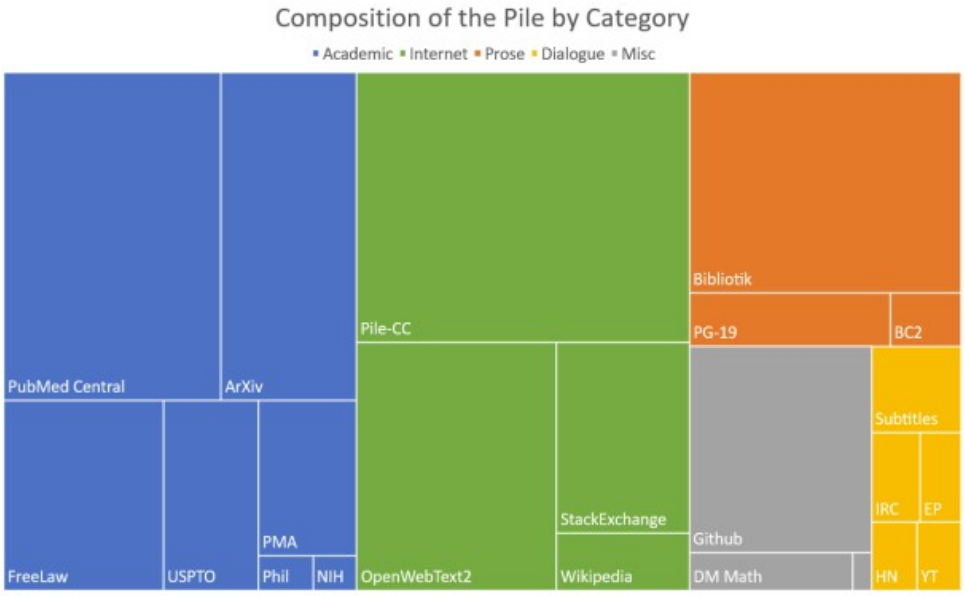
Il faut toutefois prendre cette catégorisation avec du recul car certains modèles peuvent traiter des tâches qui ne sont pas forcément dans les catégories citées ci-dessus en raison notamment de la possibilité de faire un entrainement sur la tâche en question. A titre d'exemple, les LLMs avec une architecture Décodeur peuvent également traiter des cas d'usage sur l'extraction de features lorsque ceux-ci sont dotés d'un entrainement sur la tâche de question-réponse comme c'est le cas pour les modèles de type instruct.

Une autre composante à prendre en compte dans le choix du modèle et l'architecture informatique disponible. C'est probablement le premier facteur déterminant dans le choix du modèle. Ces modèles très gourmands en mémoire (CPU, GPU, TPU, RAM, vRAM) nécessite une infrastructure compatible qui est aujourd'hui la contrainte principale pour les entreprises. Cela représente un investissement et un coût important qui limite leur utilisation. Cependant il faut être conscient que ce n'est pas forcément le modèle le plus volumineux (taille et nombre de paramètres) qui donnera les meilleurs performances pour une tâche précise. Des articles scientifiques ont montré que des modèles de plus petites tailles peuvent atteindre des performances similaires à d'autres modèles plus volumineux. Cela dépend de plusieurs facteurs dont la qualité des données utilisées pour le pré-entrainement du modèle, la stratégie d'entrainement, la stratégie de finetuning pour une tâche spécifique, le choix de l'architecture du modèle, etc...

C) Datasets d'entrainement

Comme son nom l'indique, un LLM est pré-entrainé sur une large quantité de données textuelles. Cette étape est très importante car elle permet au LLM d'apprendre le langage naturel. Aujourd'hui, il existe une multitude de dataset d'entrainement, dans des langues différentes. Voici une liste non exaustive des datasets utilisés pour former des LLMs :

- Common Crawl : En juin 2024, il est composé de 2,7 milliards de pages Web, représentant 382 Tio de contenu non compressé. Les pages capturées proviennent de 52,7 millions d'hôtes ou de 41,4 millions de domaines enregistrés et incluent 945 millions de nouvelles URL.
- The Pile : Ils'agit d'un ensemble de données de modélisation linguistique open source diversifié de 825 Gio , composé de 22 ensembles de données plus petits et de haute qualité combinés ensemble.



Nous n'allons pas énumérer les détails pour chaque dataset. L'intérêt ici est simplement d'avoir conscience que ces modèles ont besoin d'une volumétrie très importante de données pour pouvoir être formés. Voici d'autres datasets utilisés (C4, BigQuery, Wikipedia, ...)

D) Méthode d'évaluation

Evaluer les performances d'un LLM est une étape essentielle mais elle n'est pas forcément la plus simple à réaliser. Certains cas d'étude nécessitent de développer ses propres métriques pour correctement évaluer le modèle. L'ensemble des métriques et méthodes d'évaluations ne sont pas forcément adaptées à tous les modèles et tâches traitées. Il faut donc les sélectionner avec minutie. D'après la littérature, nous pouvons distinguer 4 approches différentes pour mesurer les performances d'un LLM :

- Evaluation quantitative (métriques) : précision, rappel, F1 score, perplexité, exact match, pass@k, BLUE score, CodeBLUE, ROUGE, etc...
- Evaluation de performance : temps nécessaire au modèle pour générer une réponse, nombre de demandes traitées par unité de temps, utilisation de la mémoire, etc...
- Evaluation qualitative : évaluation humaine (cohérence, précision et qualité de la réponse)
- Evaluation des LLMs entre eux via des datasets benchmarks (GLUE, SuperGLUE, SquAD, MMLU, AlpacaEval, HumanEval, etc...

Après avoir exploré les mécanismes fondamentaux des LLMs, notamment leur architecture et leur capacité à comprendre et à générer du texte en s'appuyant sur d'énormes quantités de données, nous allons maintenant aborder un cas d'étude spécifique, qui permettra non seulement d'illustrer les

principes abordés, mais aussi de démontrer comment les LLMs peuvent être appliqués pour résoudre des problèmes réels.

III. Cas d’étude : Extraction d’information au sein de documents

Le traitement manuel des données au sein des entreprises est souvent un travail fastidieux et chronophage. Les employés passent de longues heures à saisir, vérifier et organiser des informations dans des feuilles de calcul ou des bases de données. Ce processus implique une attention constante aux détails pour éviter les erreurs. De plus, les tâches répétitives et routinières, comme l’extraction d’information dans des documents, le classement de documents, la correction des incohérences et la mise à jour des enregistrements, ralentissent le flux de travail et réduisent l'efficacité globale. En l'absence d'automatisation, ce travail manuel peut également retarder les prises de décision, car l'analyse des données dépend de la disponibilité et de la précision des informations, accentuant encore les risques d'erreurs humaines.

C’est pourquoi au sein d’Aquila, nous nous sommes penchés sur l’un de ces cas d’étude que nous trouvions particulièrement pertinent. Nous avons développé une méthodologie permettant d’automatiser l’extraction d’information au sein de documents. S’en suit une phase de retranscription des informations extraites dans un format adapté au cas d’étude traité. Nous avons mis en pratique cette méthodologie pour accompagner différents clients dans leur besoin respectif. Nous utilisons les LLMs et y associons différents concepts dont l’océrisation pour traiter efficacement les documents. Dans la suite, nous allons voir un exemple précis reprenant le besoin d’un client sur lequel nous avons adapté la méthodologie. Nous utiliserons des données fictives pour cause de confidentialité.

<u>Application de la méthodologie sur des document pdf :</u>

A) Océrisation des documents

La première étape consiste à vérifier l’état et la clarté des documents à disposition. S’agit-il de vrais fichiers pdf ou de scans ? Y a-t-il des images ou tableaux comportant du texte dans les documents ? Les caractères au sein des documents sont-ils facilement lisibles ? Les réponses à ces questions sont très importantes car elles impliqueront des étapes supplémentaires de prétraitement des documents avant océrisation. Mais pourquoi parle-t-on d’océrisation ?

L'océrisation, ou reconnaissance optique de caractères (OCR), est une technologie qui convertit le texte d'une image en texte numérique éditable. De manière générale, elle est utilisée pour archiver des documents, automatiser la saisie de données, rendre les documents accessibles aux malvoyants, faciliter la traduction, et permettre la recherche d'informations dans des documents numérisés. Le dernier cas d’utilisation étant notre cas d’étude.

Pour réaliser une bonne océrisation à partir d'un document pdf, il est essentiel de suivre plusieurs étapes pour garantir une extraction de texte précise :

1. Convertir chaque page du PDF en images pour pouvoir appliquer l'OCR sur ces images.
2. Pour améliorer les résultats de l'OCR, il est souvent utile de prétraiter les images. Cela peut inclure la conversion en niveaux de gris, la binarisation (conversion en noir et blanc), le filtrage pour réduire le bruit, etc...
3. Appliquer l'OCR sur chaque image pour extraire le texte de chaque page convertie.

4. Enregistrer le texte extrait dans un fichier texte pour une utilisation ultérieure.
5. Examiner le texte extrait pour corriger les erreurs éventuelles, surtout si l'image ou le texte est complexe. Cela peut inclure des corrections orthographiques, des suppressions d’artefacts textuels ou encore l’organisation du texte (titres, paragraphes), etc ...

L’étape 2 est probablement la plus importante car elle aura un effet sur la qualité de l’extraction du texte au sein des documents, qui elle même, aura un impact lors de l’utilisation du LLM pour extraire les informations répondant à la problématique et au besoin client.

<u>Voici ci-dessous un exemple qui illustre ces étapes :</u>

Nous montrerons le process pour la première page du pdf mais c’est exactement la même chose pour toutes les pages du document.

1) Convertir chaque page du PDF en images :

Première page brute du pdf

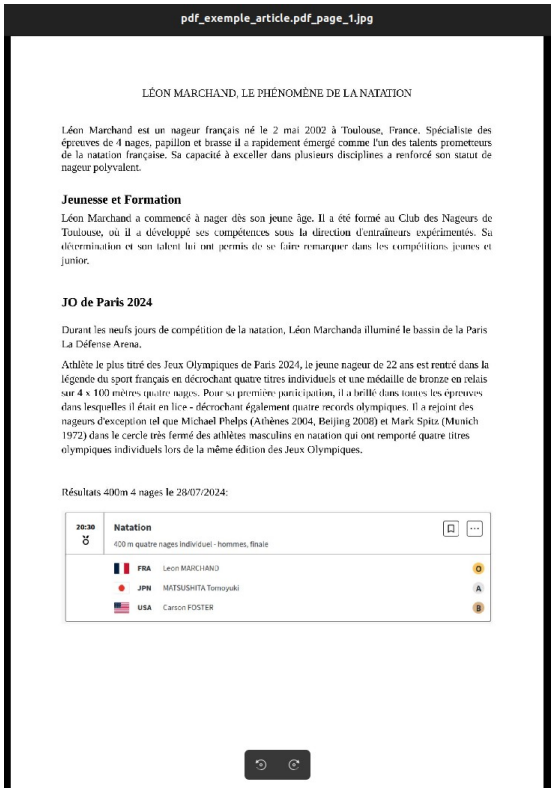


Image de la première page du pdf

2) Prétraitement des images

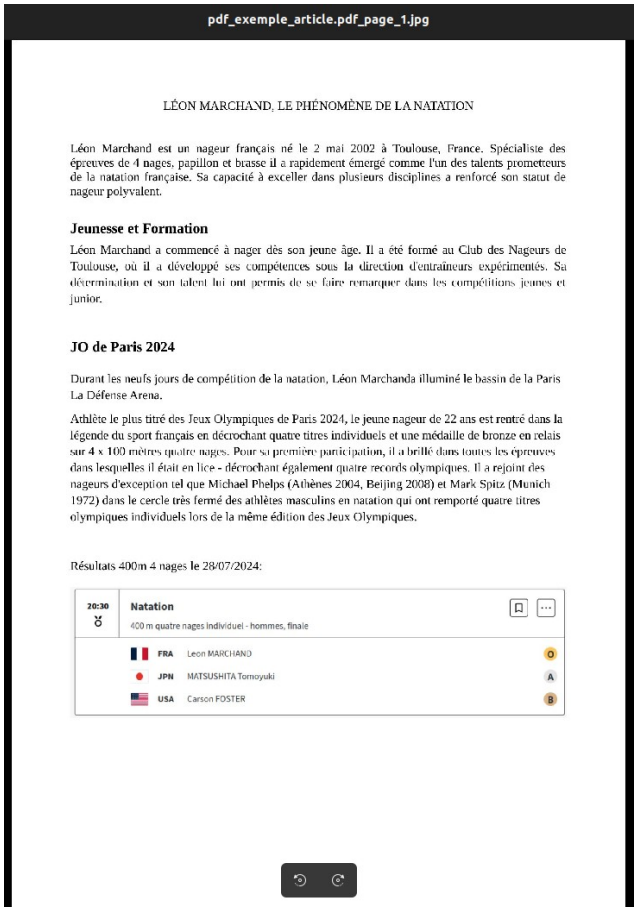


Image de la première page du pdf avant prétraitement

Image de la première page du pdf après prétraitement

Le prétraitement a permis d'améliorer la lisibilité du document bien que le document d'origine était déjà de bonne qualité. Le tableau d'origine n'est plus visible en tant que tableau car les lignes ont disparues. Les couleurs ne sont plus visibles. Cela va permettre d'extraire plus facilement le texte présent initialement dans ce tableau.

3) Appliquer l'OCR sur l'image pour extraire le texte de la page

Voici le résultat de l'océrisation de la première page du pdf après prétraitement :

```
LEON MARCHAND, LE PHENOMENE DE LA NATATION
Léon Marchand est un nageur frangais né le 2 mai 2002 a Toulouse, France. Spécialiste des
épreuves de 4 nages, papillon et brasse il a rapidement émergé comme l'un des talents prometteurs
de la natation frangaise. Sa capacité 4 exceller dans plusieurs disciplines a renforcé son statut de
nageur polyvalent.
Jeunesse et Formation
Léon Marchand a commencé a nager dés son jeune age. Il a été formé au Club des Nageurs de
Toulouse, ott il a développé ses compétences sous la direction d'entraineurs expérimentés. Sa
détermination et son talent lui ont permis de se faire remarquer dans les compétitions jeunes et
junior.
JO de Paris 2024
Durant les neufs jours de compétition de la natation, Léon Marchanda illuminé le bassin de la Paris
La Défense Arena.
Athléte le plus titré des Jeux Olympiques de Paris 2024, le jeune nageur de 22 ans est rentré dans la
légende du sport francais en décrochant quatre titres individuels et une médaille de bronze en relais
sur 4 x 100 métres quatre nages. Pour sa premiére participation, il a brillé dans toutes les épreuves
dans lesquelles il était en lice - décrochant également quatre records olympiques. II a rejoint des
nageurs d'exception tel que Michael Phelps (Athénes 2004, Beijing 2008) et Mark Spitz (Munich
1972) dans le cercle trés fermé des athlétes masculins en natation qui ont remporté quatre titres
olympiques individuels lors de la méme édition des Jeux Olympiques.
Résultats 400m 4 nages le 28/07/2024:
20:30 Natation (a) (-:-]
6 400 m quatre nages individuet - hommes, finale
BE fra leon MARCHAND o
@ JPN MATSUSHITA Tomoyuki A
@ usa CarsonFOSTER B
```

Une fois l'ensemble des pages océrisées, nous concatenons l'ensemble des textes extraits de chaque page pour obtenir le texte global du document original.

Nous sommes maintenant prêt pour extraire les informations du document à l'aide d'un LLM.

B) Utilisation d'un LLM pour extraire les informations du document

Le choix du LLM à utiliser s'est porté sur Mixtral 8x7B-instruct quantisé en 4 bits car il s'agit d'un modèle instruct ce qui signifie qu'il a été entrainé sur la tâche de question/réponse. De plus, d'après les benchmarks d'évaluation, il faisait parti des meilleurs modèles pour cette tâche lorsque nous l'avons utilisé. Nous allons donc pouvoir lui soumettre une question pour extraire les informations que nous souhaitons et il nous retournera une réponse contenant les informations extraites. La quantisation en 4 bits était nécessaire pour pouvoir charger et utiliser le modèle en inférence sur l'architecture informatique à disposition. Il occupe 25GB/32GB de la RAM GPU.

Le document pdf que nous traitons à titre d'exemple correspond à un rapport de 5 pages sur un sportif français ayant participé au JO de Paris 2024. Il détaille entre autre ses résultats obtenus.

Nous avons défini des features à extraire au sein du document telles que :

- Type_Competition : le type de compétition à laquelle l'athléte a participé
- Ville_Competition : la ville dans laquelle a eu lieu la compétition
- Annee_Competition : l'année à laquelle a eu lieu la compétition
- Ville_Naissance : la ville de naissance de l'athléte
- Annee_Naissance : l'année de naissance de l'athléte
- Nom : le nom de l'athléte identifié

- Prenom : le prénom de l’athléte identifié
- Sport : le sport pratiqué par l’athlète
- Epreuve : les épreuves auxquelles l’athlétes a participé avec son classement, son temps/score et la date à laquelle a eu lieu l’épreuve
- Information_Comparaison : information sur l’athlétes avec une comparaison vis à vis de d’autre sportif

Une étape très importante à ce stade est la formulation de la question à transmettre au LLM pour extraire les features. Plus la question sera précise, meilleurs seront nos chances d’extraire les bonnes informations. Pour assurer une certaine qualité de la question nous allons l’associer à un contexte en indiquant des instructions claires pour orienter les réponses. Le contexte représente le texte global extrait des images des pages du pdf. La concaténation des instructions, du contexte et de la question représentera le prompt. Il s’agit tout simplement de l’entrée à transmettre au LLM pour obtenir une réponse.

Voici un schéma récapitulatif du fonctionnement d’un LLM instruct.

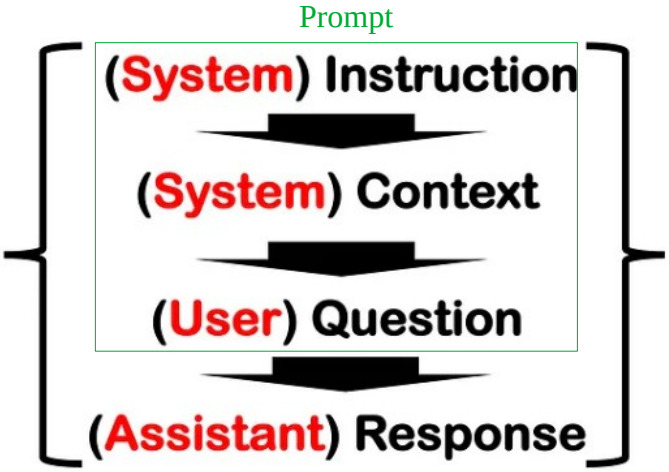


Schéma de fonctionnement du LLM instruct

Voici un exemple de prompt de la feature «Type_Competition» à extraire. Le principe est le même pour les autres features. Seule la question sera différente.

instruction

contexte

question

```
[{'role': 'user',
  'content': 'You must extract several information in the following document: Page 1:\nLEON MARCHAND, LE PHENOMENE
DE LA NATATION\nLéon Marchand est un nageur frangais né le 2 mai 2002 a Toulouse, France. Spécialiste des\népreuve
s de 4 nages, papillon et brasse il a rapidement émergé comme l\'un des talents prometteurs\nde la natation franga
ise. Sa capacité 4 exceller dans plusieurs disciplines a renforcé son statut de\nnageur polyvalent.\nJeunesse et F
ormation\nLéon Marchand a commencé a nager dés son jeune age. Il a été formé au Club des Nageurs de\nToulouse, ott
il a développé ses compétences sous la direction d\'entraineurs expérimentés. Sa\ndétermination et son talent lui
ont permis de se faire remarquer dans les compétitions jeunes et\njunior.\nJO de Paris 2024\nDurant les neufs jour
s de compétition de la natation, Léon Marchanda illuminé le bassin de la Paris\nLa Défense Arena.\nAthléte le plus
titré des Jeux Olympiques de Paris 2024, le jeune nageur de 22 ans est rentré dans la\nlégende du sport francais e
n décrochant quatre titres individuels et une médaille de bronze en relais\nsur 4 x 100 métres quatre nages. Pour
sa premiére participation, il a brillé dans toutes les épreuves\ndans lesquelles il était en lice - décrochant éga
lement quatre records olympiques. II a rejoint des\nnageurs d\'exception tel que Michael Phelps (Athénes 2004, Bei
jing 2008) et Mark Spitz (Munich\n1972) dans le cercle trés fermé des athlétes masculins en natation qui ont rempo
rté quatre titres\nolympiques individuels lors de la méme édition des Jeux Olympiques.\nRésultats 400m 4 nages le
28/07/2024:\n20:30 Natation (a) (-:-]\n6 400 m quatre nages individuet - hommes, finale\nBE fra leon MARCHAND o\n@
JPN MATSUSHITA Tomoyuki A\n@ usa CarsonFOSTER B\n\x0c\nPage 2:\nClassement épreuve\nPlace CNO Nom B\n1 IFFRA MARCH
AND Leon B\nPhase Place Temps\nFinale 1 4:02.95\nSeries 2 4:08.30\n2 @ JPN MATSUSHITA Tomoyuki B\nPhase Place Temp
s\nFinale 2 4:08.62\nSéries 2 4:11.18\n3° HEEUSA FOSTER Carson B\nPhase Place Temps\nFinale 3 4:08.66\nSéries 1 4:
11.07\nRésultats 200m papillon le 31/07/2024:\n20:37 ~~ Natation (a) [:"]\nVw\n0 200 m papillon - hommes, finale\n
BE fra Leon MARCHAND 0\n— :\nmee «(YUN Kristof MILAK A\neh can tlyaKHARUN 8\n\x0c\nPage 3:\nClassement épreuve\nPl
ace CNO Nom B\n1 FRA MARCHAND Leon B\nPhase Place Temps\nFinale 1 1:51.21\nDemi-finales i 1:53.50\nSeries 2 1:55.2
6\n2 SEHUN  MILAK Kristof B\nPhase Place Temps\nFinale 2 1:51.75\nDemi-finales 1 1:52.72\nSeries i 1:53.92\n3 eB C
AN KHARUN Ilya B\nPhase Place Temps\nFinale 3 1:52.80\nDemi-finales 2 1:54.01\nSéries 1 1:54.06\nRésultats 200m br
asse le 31/07/2024:\n22:31 Natation (a) [:-"]\nww\n° 200 m brasse - hommes, finale\nBB Fra Leon MARCHAND °\nAus Za
c STUBBLETY-COOK A\n= y\nwe «(NED Caspar CORBEAU B\n\x0c\nPage 4:\nClassement épreuve\nPlace CNO Nom B\n1 @BFRA MA
RCHAND Leon B\nPhase Place Temps\nFinale 1 2:05.85\nDemi-finales I 2:08.11\nSeries 2 2:09.55.\n2 @MBAUS — STUBBLET
Y-COOK Zac B\nPhase Place Temps\nFinale 2 2:06.79\nDemi-finates 1 2:08.57\nSeries 1 2:09.49\n3 TENED CORBEAU Caspa
r B\nPhase Place Temps\nFinale 3 2:07.90\nDemi-finales 3 2:09.52\nSéries 2 2:09.78\nRésultats 200m 4 nages le 02/0
8/2024:\n20:45 Natation (a) [---]\nww\no 200 m quatre nages individuel - hommes, finale\nBB FRA Leon MARCHAND 0\nP
S Ly\na GBR Duncan SCOTT A\nMB cHN WANG Shun 8\n\x0c\nPage 5:\nClassement épreuve\nPlace CNO Nom B\n1 EBPFRA MARCH
AND Leon B\nPhase Place Temps\nFinale 1 1:54.06\nDemi-finates L 1:56.31\nSéries 2 1:57.86\n2 SHEGBR SCOTT Duncan B
\nPhase Place Temps\nFinate 2 1:55.31\nDemi-finaies 2 1:56.49\nSeries 1 LS7.77\n3° QMCHN WANG Shun B\nPhase Place
Temps\nFinale 3 1:56.00\nDemi-finaies 3 1:56.54\nSéries 3 1:58.09\nRésultats 4x100m 4 nages mixte le 04/08/2024:\n
we\n0 Relais 4x100 m quatre nages - hommes, finale\nMMB un Chine o\n@ usa Etats-Unis A\nfl i FRA France B\nClassem
ent épreuve\nPlaceCNO —Equipe B\n1 QHBCHN République poputaire de Chine B\nPhase Place Temps\nFinale £ 3:27.46\nSé
ries 2 3:31.58\n2 MEBUSA Etats-Unis d\'Amérique B\nPhase Place Temps\nFinale 2 3:28.01\nSéries i 3:31.62\n3° BP FR
A France B\nPhase Place Temps\nFinale 3 3:28.38\nSéries 1 3:31.36\n\x0c\n. These capital information will be liste
d in the next user request'},
 {'role': 'assistant',
  'content': 'I would do the best to find all information that you search. What are these information ?'},
 {'role': 'user',
  'content': 'The information to find are the following: identify and extract all features in the document. - Firs
t, identify and extract the kind of competition. - Second, identify and extract the city where the competion is lo
calized. - Third, identify and extract the year of the competition. - Fourth, identify and extract the city where
the sportive was born. - Fifth, identify and extract the year where the sportive was born. - Sixth, identify and e
xtract the last name of the sportive which participate to the competition. - Seventh, identify and extract the fir
st name of the sportive which participate to the competition. - Heighth, identify and extract the sport of the spo
rtive which participate to the competition. - Nineth, identify and extract all the text that refers to the race wi
th the associated stopwatch, rank place and date. - Tenth, identify and extract the information of comparison whic
h other sportive. Answer me directly with the json good format describe below begining by {: ```json { "Type_Compe
tition":"Kind of competition", "Ville_Competition": "Competition city found", "Annee_Competion": "Competition year
found", "Ville_Naissance": "Birthplace found", "Annee_Naissance": "Birthyear found", "Nom": "Last name found", "Pr
enom": "First name found", "Sport": "Sport found", "Epreuve": "[{{\\"Epreuve\\":\\"event found\\",\\"Classement
\\":\\"Rank place found\\",\\"Chrono\\":\\"stopwatch found\\, \\"Date\\":\\"Date found\\,"}},{{\\"Epreuve\\":\\"ev
ent found\\",\\"Classement\\":\\"Rank place found\\",\\"Chrono\\":\\"stopwatch found\\, \\"Date\\":\\"Date found
\\,"}},...]", "Information_Comparaison": "Comparison information found", } ``` $$$$$$$$$ '}]
```

Et voici ci-dessous le résultat de l’extraction du LLM pour chaque feature définie.

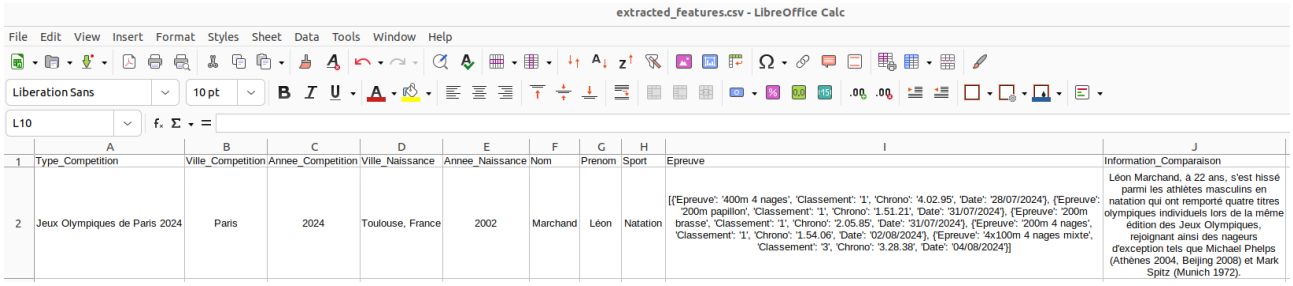
```
Out[11]: {'Type_Competition': 'Jeux Olympiques de Paris 2024',
          'Ville_Competition': 'Paris',
          'Annee_Competition': '2024',
          'Ville_Naissance': 'Toulouse, France',
          'Annee_Naissance': '2002',
          'Nom': 'Marchand',
          'Prenom': 'Léon',
          'Sport': 'Natation',
          'Epreuve': [{'Epreuve': '400m 4 nages',
            'Classement': '1',
            'Chrono': '4.02.95',
            'Date': '28/07/2024'},
           {'Epreuve': '200m papillon',
            'Classement': '1',
            'Chrono': '1.51.21',
            'Date': '31/07/2024'},
           {'Epreuve': '200m brasse',
            'Classement': '1',
            'Chrono': '2.05.85',
            'Date': '31/07/2024'},
           {'Epreuve': '200m 4 nages',
            'Classement': '1',
            'Chrono': '1.54.06',
            'Date': '02/08/2024'},
           {'Epreuve': '4x100m 4 nages mixte',
            'Classement': '3',
            'Chrono': '3.28.38',
            'Date': '04/08/2024'}],
          'Information_Comparaison': "Léon Marchand, à 22 ans, s'est hissé parmi les athlètes masculins en natation qui ont
         remporté quatre titres olympiques individuels lors de la même édition des Jeux Olympiques, rejoignant ainsi des na
         geurs d'exception tels que Michael Phelps (Athènes 2004, Beijing 2008) et Mark Spitz (Munich 1972)."}
```

Résultats de l’extraction du LLM

Avec une simple instruction python, nous obtenons un dataframe que nous pouvons facilement exporter en fichier csv.

Type_Competition	Ville_Competition	Annee_Competition	Ville_Naissance	Annee_Naissance	Nom	Prenom	Sport	Epreuve	Information_Comparaisc
Jeux Olympiques de Paris 2024	Paris	2024	Toulouse, France	2002	Marchand	Léon	Natation	[{'Epreuve': '400m 4 nages', 'Classement': '1'...	Léon Marchand, à 22 an s'est hissé parmi les

Dataframe en langage python



	A	B	C	D	E	F	G	H	I	J
1	Type_Competition	Ville_Competition	Annee_Competition	Ville_Naissance	Annee_Naissance	Nom	Prenom	Sport	Epreuve	Information_Comparaison
2	Jeux Olympiques de Paris 2024	Paris	2024	Toulouse, France	2002	Marchand	Léon	Natation	[{'Epreuve': '400m 4 nages', 'Classement': '1', 'Chrono': '4.02.95', 'Date': '28/07/2024'}, {'Epreuve': '200m papillon', 'Classement': '1', 'Chrono': '1.51.21', 'Date': '31/07/2024'}, {'Epreuve': '200m brasse', 'Classement': '1', 'Chrono': '2.05.85', 'Date': '31/07/2024'}, {'Epreuve': '200m 4 nages', 'Classement': '1', 'Chrono': '1.54.06', 'Date': '02/08/2024'}, {'Epreuve': '4x100m 4 nages mixte', 'Classement': '3', 'Chrono': '3.28.38', 'Date': '04/08/2024'}]	Léon Marchand, à 22 ans, s'est hissé parmi les athlètes masculins en natation qui ont remporté quatre titres olympiques individuels lors de la même édition des Jeux Olympiques, rejoignant ainsi des nageurs d'exception tels que Michael Phelps (Athènes 2004, Beijing 2008) et Mark Spitz (Munich 1972).

Fichier csv

Afin de vérifier si les informations extraites sont bien celles se trouvant dans le document, nous avons développé un process permettant d’identifier et d’encadrer les informations extraites dans le document en y associant le nom de la feature. Nous présentons un exemple avec la première page du document dans laquelle nous retrouvons une partie des features extraites :

LÉON MARCHAND, LE PHÉNOMÈNE DE LA NATATION

Prenom Nom Annee_Naissance Ville_Naissance
Léon Marchand est un nageur français né le 2 mai 2002 à Toulouse, France. Spécialiste des épreuves de 4 nages, papillon et brasse il a rapidement émergé comme l'un des talents prometteurs de la natation française. Sa capacité à exceller dans plusieurs disciplines a renforcé son statut de nageur polyvalent.

Jeunesse et Formation

Léon Marchand a commencé à nager dès son jeune âge. Il a été formé au Club des Nageurs de Toulouse, où il a développé ses compétences sous la direction d'entraîneurs expérimentés. Sa détermination et son talent lui ont permis de se faire remarquer dans les compétitions jeunes et junior.

JO de Paris 2024

Ville_Competition Annee_Competition

Durant les neufs jours de compétition de la natation, Léon Marchanda illuminé le bassin de la Paris La Défense Arena.

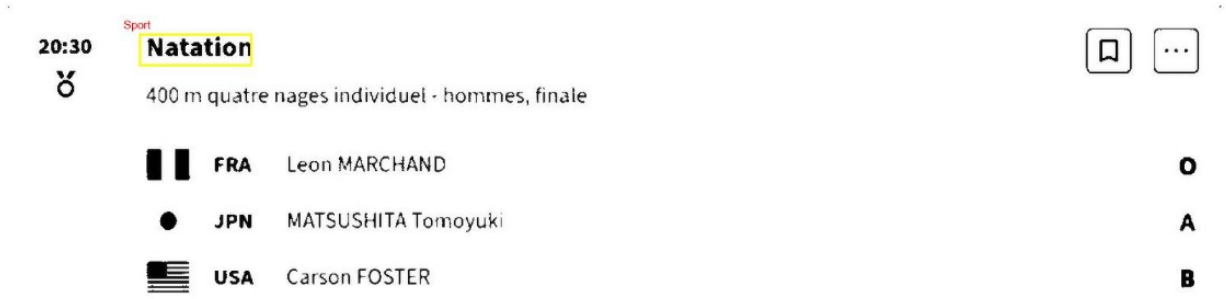
Athlète le plus titré des Jeux Olympiques de Paris 2024, le jeune nageur de 22 ans est rentré dans la légende du sport français en décrochant quatre titres individuels et une médaille de bronze en relais sur 4 x 100 mètres quatre nages. Pour sa première participation, il a brillé dans toutes les épreuves dans lesquelles il était en lice - décrochant également quatre records olympiques. Il a rejoint des nageurs d'exception tel que Michael Phelps (Athènes 2004, Beijing 2008) et Mark Spitz (Munich 1972) dans le cercle très fermé des athlètes masculins en natation qui ont remporté quatre titres olympiques individuels lors de la même édition des Jeux Olympiques.

Type_Competition

Information_Comparaison

Résultats 400m 4 nages le 28/07/2024:

Epreuve



Les LLMs s'avèrent être un outil puissant pour l'extraction d'informations dans des documents. Grâce à leur capacité à comprendre et à traiter des textes complexes, ils peuvent identifier, synthétiser et extraire des données pertinentes avec une grande précision. Que ce soit pour analyser des rapports financiers, des articles scientifiques, des documents juridiques ou encore un rapport des résultats de Léon Marchand au JO de Paris, un LLM facilite l'accès rapide aux informations essentielles, réduisant ainsi le temps et les efforts nécessaires pour les trouver manuellement. De plus, son adaptabilité à différents contextes linguistiques et thématiques en fait un atout indispensable pour les entreprises et les chercheurs cherchant à exploiter efficacement de grandes quantités de données textuelles.

Références:

[1] https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf
[2] https://www.persee.fr/doc/intel_0769-4113_1990_num_9_1_880
[3] https://arxiv.org/pdf/1706.03762
[4]https://www.epitech.eu/2022/06/21/comment-est-nee-l-architecture-informatique-moderne/
[5]https://www.iro.umontreal.ca/~vincentp/ift3395/lectures/backprop_old.pdf
[6]https://arxiv.org/pdf/1406.2661
[7]https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf
[8]https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf
[9]https://arxiv.org/pdf/2005.14165
[10]https://towardsdatascience.com/transformers-141e32e69591
[11]https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/
[12]https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf
[13]https://www.commoncrawl.org/blog/june-2024-crawl-archive-now-available
[14]https://pile.eleuther.ai/