

QUESTION ANSWERING

OUMAIMA BENABOUD
MIAAD 2022-2023

SOMMAIRE

Introduction

Brève histoire du QA en NLP

Applications du QA en NLP

Taxonomie du QA

Approches du QA en NLP

Avancées récentes du QA en NLP

Implémentation

Conclusion



INTRODUCTION

- Le QA est une tâche importante en NLP qui consiste à répondre à une question posée par l'utilisateur en langage naturel en se basant sur un contexte ou une base de connaissances donnés.
- L'objectif du QA est de fournir des réponses précises et pertinentes aux questions posées en langage naturel, telles que celles soumises aux moteurs de recherche ou aux assistants virtuels.

HISTORY OF QA

- BASEBALL 1961:
 - Premier système QA et LUNAR dans les années 60-70.
 - répondait aux questions sur la Ligue Majeure de Baseball
- LUNAR 1971 :
 - développait par NASA
 - répondait aux questions sur l'analyse géologique des roches lunaires.
- Ces systèmes ont eu beaucoup de succès dans leurs domaines respectifs, mais avaient besoin d'une base de connaissances manuellement écrite par des experts.

HISTORY OF QA

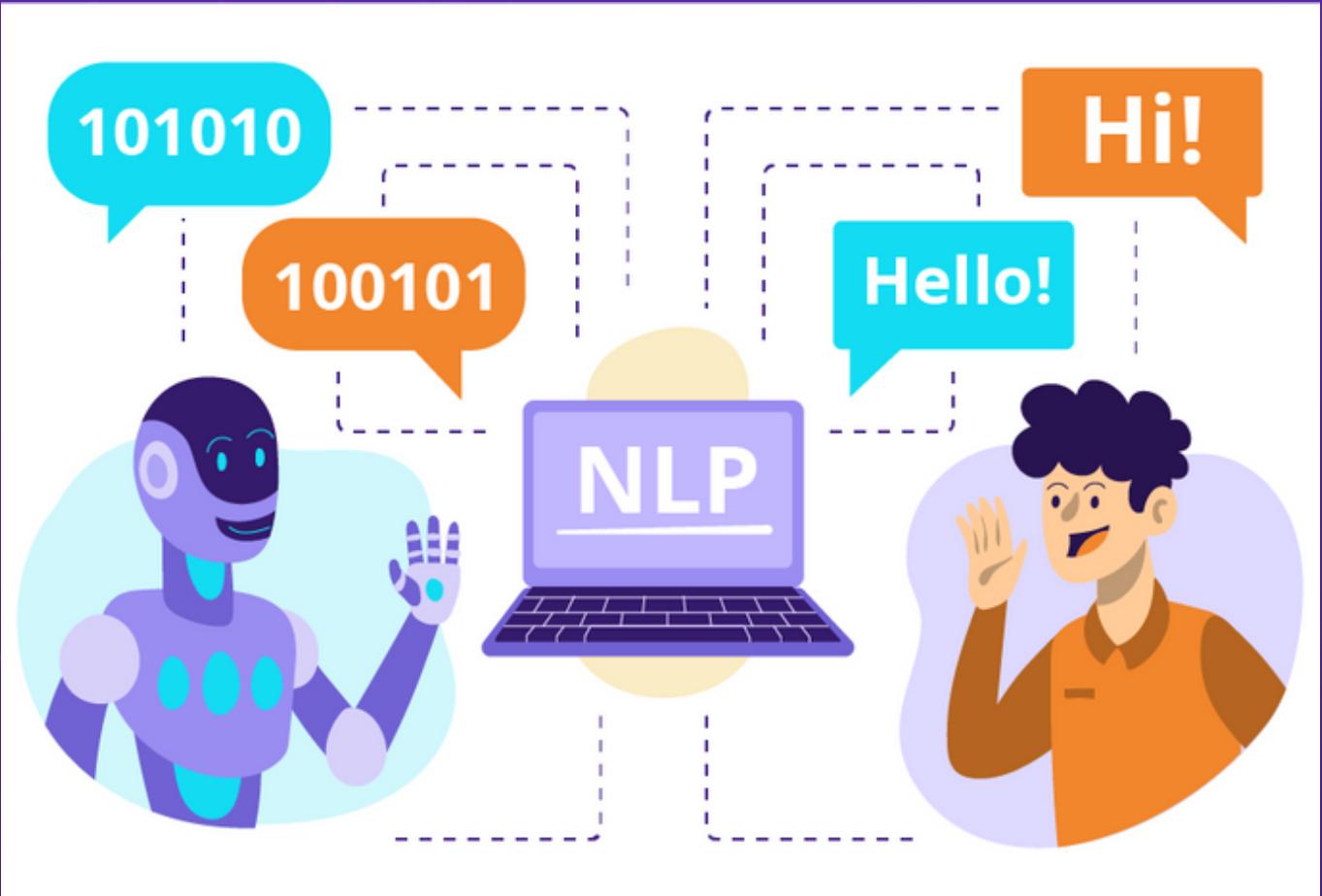
- Des projets ambitieux ont été développés dans les années 70-80 en matière de compréhension de texte et de QA tel que :
- UC (Unix Consultant) :
 - répondait aux questions sur le système d'exploitation Unix
- LILOG :
 - répondait à des questions sur les informations touristiques des villes en Allemagne.
- Des systèmes de QA en langage naturel spécialisés ont également été développés, tels que EAGLi pour les scientifiques de la biologie et la médecine.

HISTORY OF QA

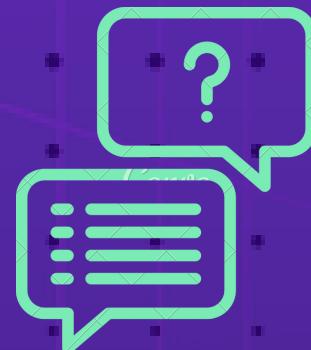
- Cependant, ce n'est que dans les années 90-00 que les systèmes de QA ont commencé à utiliser des techniques de NLP avancées pour comprendre et répondre aux questions en langage naturel.
- Wolfram Alpha 2009 :
 - utilise une base de connaissances encyclopédique combinée à des algorithmes de calcul
 - ne fournit pas de liens vers des pages Web, mais plutôt des réponses directes à des questions portant sur des sujets tels que les mathématiques, la science, la musique, la linguistique, l'histoire, la finance, etc.
- IBM Watson 2011 :
 - développé pour concurrencer des humains au jeu Jeopardy! et a remporté le concours en 2011

APPLICATIONS OF QA

- Le QA est devenu de plus en plus important dans diverses industries, notamment dans les domaines de l'assistance virtuelle, du support client, de la recherche d'information et des moteurs de recherche.



APPLICATIONS OF QA



SUPPORT CLIENT ET CHATBOTS

- répondre aux questions des clients en temps réel, améliorant ainsi l'efficacité du service client et la satisfaction des clients.
- utiliser les chatbots pour fournir un support client automatisé, réduisant ainsi les coûts et améliorant l'efficacité.



ASSISTANTS VIRTUELS

- tels que : Siri d'Apple, Google Assistant et Alexa d'Amazon.
- comprendre et de répondre aux questions posées en langage naturel, offrant ainsi une expérience utilisateur personnalisée et pratique.



MOTEURS DE RECHERCHE

- fournir des réponses précises et pertinentes aux requêtes des utilisateurs.
- afficher directement une réponse extraite de la page Web pertinente, telle que la définition d'un mot ou une information factuelle. Cette fonctionnalité est connue sous le nom de "featured snippets" ou de "position zéro"

QA TAXONOMY

Types et variations

TYPES OF QA

Factoid QA

Ce type de système fournit des réponses à des questions simples qui ont une réponse spécifique, telles que "Quelle est la capitale de la France ?" ou "Qui est l'actuel président des États-Unis ?"

List QA

Ces systèmes fournissent des réponses qui sont des listes d'éléments répondant à certains critères, comme "Quelles sont les plus grandes villes de Californie ?" ou "Qui sont les personnages principaux de Harry Potter ?"

Complex QA

Ces systèmes sont conçus pour répondre à des questions qui nécessitent une inférence ou un raisonnement allant au-delà des simples faits, telles que "Pourquoi les oiseaux migrent-ils ?" ou "Qu'arriverait-il à la Terre si la Lune disparaissait ?"

QA VARIANTS

Les systèmes de QA diffèrent par la manière dont les réponses sont créées :

Extractive QA

- sélectionner une réponse à partir d'un contexte ou d'un texte source donné.
- identifier les extraits de texte pertinents d'un passage qui fournissent la meilleure réponse à la question.
- présenté le texte sélectionné comme la réponse à l'utilisateur

Generative QA

- générer une nouvelle réponse textuelle en fonction du contexte de la question.
- produire une réponse qui n'est pas directement tirée du contexte d'entrée, mais plutôt synthétisée à partir de celui-ci.
- plus flexibles que les modèles extractifs, car ils produisent des réponses plus nuancées ou plus complexes.

Extractive Question Answering

Document

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming.

Question

Is Python dynamically-typed?

Question Answering Model

Answer

Python is dynamically-typed

Generative Question Answering

Document

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming.

Question

Is Python dynamically-typed?

Question Answering Model

Yes, it is

QA VARIANTS

En outre, les systèmes de QA diffèrent par l'origine des réponses :

Closed QA

- Extraction de réponses à partir de données structurées (base de données)
- Conversion des questions en requêtes de base de données.
- Utiliser dans des domaines tels que la médecine, le droit et la finance, où l'on a besoin d'informations précises et spécifiques

Open QA

- Extraction de réponses à partir de vastes collections de données non structurées, telles que l'internet ou de grands corpus.
- Répondre à des questions sur n'importe quel sujet, sans être limités à un domaine ou à une base de connaissances spécifique.

EXEMPLE

- ChatGPT est considéré comme un modèle Open Generative QA, car il est capable de générer des réponses à des questions dans un contexte de domaine ouvert.
- GPT-3 utilise un modèle de génération de langage à grande échelle pour générer du texte en réponse à des questions et des invites.
- Il a la capacité de générer de nouvelles réponses basées sur les informations qu'il a apprises à partir de vastes quantités de données textuelles, ce qui en fait une forme de réponse générative aux questions.



ChatGPT

QA APPROACHES

Rule-based systems

- se sert des règles préétablies pour identifier les réponses aux questions.
- utilisés dans des domaines très spécifiques où les règles sont bien définies.

Knowledge-based systems

- utilisent des bases de connaissances structurées pour répondre aux questions.
- ces bases de connaissances sont créées et maintenues par des experts dans un domaine particulier.

Information retrieval

- rechercher des documents pertinents et extraire les réponses à partir de ces documents.
- utilisés pour répondre à des questions factuelles.

Machine learning-based systems

- ces modèles sont entraînés sur des ensembles de données de questions-réponses pour prédire les réponses à de nouvelles questions.

RECENT ADVANCES IN QA

Pretrained models

- entraînés sur de grands corpus
- adaptés à des tâches de QA spécifiques pour améliorer la performance des systèmes de QA

Transfer learning and domain adaptation

- utiliser un modèle pré-entraîné comme point de départ et les entraînés sur des données de domaines spécifiques pour améliorer leur performance dans ces domaines

Interactive QA

- permettent de poser des questions en utilisant un langage plus informel
- obtenir des réponses précises grâce à la capacité des agents conversationnels à interpréter le contexte et à clarifier les questions.

Multimodal QA

- utilisent à la fois des informations textuelles et visuelles pour répondre aux questions
- combine les avantages de chaque modalité pour améliorer la pertinence des réponses.

DATASETS FOR QA

- SQuAD : questions-réponses basées sur des articles de Wikipédia
- MS MARCO : comprendre des questions complexes et fournir des réponses à partir de pages web
- TREC QA : questions-réponses provenant de diverses sources
- HotpotQA : questions-réponses multi-sauts nécessitant de raisonner sur plusieurs paragraphes
- BioASQ : référence pour l'indexation sémantique biomédicale et la réponse aux questions
- CoQA : questions-réponses conversationnelles nécessitant de maintenir le contexte et comprendre les relations
- MultiRC : ensemble de données pour le raisonnement par phrases multiples
- RACE : ensemble de données pour la compréhension de la lecture automatique et la réponse aux questions en anglais.

IMPLEMENTATION

Phase 1: Data preprocessing

- un exemple de machine learning-based chatbot.
- Un fichier CSV contenant des données d'entraînement a été chargé.
- Les questions contenues dans les données CSV ont été prétraitées.
- Un vecteur Tf-Idf a été ajusté aux questions et un objet OneHotEncoder a été adapté aux tags.

Phase 2: Model creation and training

- Création et entraînement d'un réseau de neurones avec le score Tf-Idf des patterns et la vectorisation des tags.
- Stockage des réponses pour chaque tag dans un fichier JSON.
- Création de la fonction "predict_tag" pour vectoriser la chaîne d'entrée avec Tf-idf et prédire le tag correspondant.

Phase 3: Chatbot deployment

- Crédit de la fonction "chatbot_response" pour prédire le tag avec "predict_tag" et afficher la réponse correspondante dans le dictionnaire des réponses.
- Une fonction "start_chat" a été créée pour lire les entrées de l'utilisateur, prédire le tag avec la fonction "predict_tag" et afficher la réponse avec la fonction "chatbot_response".

THANK YOU!

AVEZ-VOUS DES QUESTIONS ?