

Projet de recherche technologique: état de l'art

Chatbot pour plateforme LMS et Rainbow



Sommaire

Introduction.....	3
1. Présentation du sujet	3
2. Etat de l'art	3
3. Etudes bibliographiques	6
3.1 Natural Language Processing (NLP)	7
3.2 Lemmatisation	7
3.3 Tokenisation	7
3.4 Détermination de la similarité	7
3.4.1 Les méthodes de mesure de la similarité	8
3.5 Stop words.....	8
3.6 Word Embedding.....	9
3.6.1 Indépendant du contexte sans machine learning	9
3.6.2 Indépendant du contexte avec machine learning	10
3.6.3 Dépendant du contexte et basé sur RNN	11
3.6.4 Dépendant du contexte et basé sur un transformateur.....	11
3.7 Question-answering	12
4. Codes et tests	13
4.1 Comparaison méthode de similarité et Word2vec	13
4.2 Question-réponse avec BERT.....	13
4.3 Question-réponse avec cdQA sur un ensemble de fichiers.....	14
Conclusion	16

Introduction

Un chatbot aussi appelé dialogueur ou agent conversationnel est un agent logiciel qui dialogue avec un utilisateur. C'est une machine créée pour converser avec une personne en donnant l'impression de discuter avec une autre personne.

La recherche sur cette interface personne-machine a été influencée par la compétition sur le test de Turing (1950) qui consiste à donner l'illusion qu'un programme pense par un dialogue sensé. Autrement dit, un utilisateur humain formule une demande en langage naturel et le chatbot traite la demande. De nos jours, les chatbots ne se limitent plus à des questions de base, mais intègrent désormais des algorithmes plus évolués permettant une gestion des échanges d'un niveau de complexité plus élevé qu'auparavant. Ils fleurissent sur internet et les entreprises en profitent pour améliorer leur service et traiter les demandes des clients de manières efficaces. Aussi, des chatbots éducatifs se développent de plus en plus pour améliorer l'apprentissage. C'est dans cette optique que se classe notre projet de recherche technologiques (PRT).

1. Présentation du sujet

Rainbow Classroom d'Alcatel-Lucent est une plateforme intuitive pour dispenser des cours en ligne. Avec la crise COVID, les établissements ont compris l'importance de mettre en place des cours virtuels afin d'assurer la continuité pédagogique. Rainbow Classroom est une solution qui intègre des fonctionnalités de communication et de collaboration en temps réel pour les différentes parties prenantes.

Dans le but de parfaire la plateforme, nous travaillons en collaboration avec l'entreprise Alcatel Lucent Entreprise sur le développement d'un chatbot pour plateforme LMS (exemple Moodle) et Rainbow Classroom en utilisant l'intelligence artificiel. Cela permettra :

- Aux professeurs de fournir des détails sur les cours sont pour autant répondre à des questions répétitives
- Aux élèves, d'avoir une réponse adaptée à leur question rapidement

Les chatbots ne remplacent pas l'enseignant mais le secondent et le complètent. Cet outil est un véritable compagnon pour l'élève et permet de différencier les apprentissages pour chacun car la conversation s'adapte aux réponses et aux demandes de celui-ci.

2. Etat de l'art

Cette application est vraiment courante dans le secteur éducatif. Ils existent différents travaux qui ont été réalisés sur le développement de chatbot pour améliorer les conditions d'apprentissage des étudiants en leur permettant de facilement retrouver la réponse à leur question et automatiser une partie du travail des enseignants. Les articles et ouvrages suivants

présentent quelques applications ainsi que les connaissances à avoir pour les mettre en œuvre.

Article 1 : Chatbot de réponse aux questions à l'aide du Deep Learning avec NLP

Cet article de quatre chercheurs indiens en ingénierie et informatique à Institut national de technologie, Tiruchirappalli, Tamil Nadu, Inde et publié en juillet 2021 traite du développement d'un "Question-Answering Chatbot" pour le domaine de l'éducation afin d'aider les étudiants à répondre à leurs questions sur les sujets et les cours et de soulager les enseignants. Pour ce faire, leur travail a été divisé en trois grandes étapes.

La première consiste à créer une base de connaissances qui exploite la spécialité du domaine spécifique. Ceux sont les étudiants qui intègrent les fichiers PDF dans lesquels ils cherchent des informations. Ils subissent plusieurs transformations NLP telles que la tokenisation, la lemmatisation, la singularisation, la suppression de la ponctuation ... afin de réduire la taille du vocabulaire dans la base de connaissances finale. Enfin, le texte est enregistré en tant que base de connaissances.

Dans la deuxième étape, le chatbot utilise les méthodes de récupération d'informations pour vérifier la similitude entre la question et les phrases dans la base de connaissances.

La dernière étape nécessite de reclasser les documents sélectionnés, ceci est possible en utilisant des réseaux de neurones convolutifs (CNN).

La performance du chatbot a été mesurée au travers de deux grandeurs: la précision moyenne (MAP) et le rang réciproque moyen (MRR). La précision moyenne calcule les rangs de toutes les réponses correctes, et le rang réciproque moyen calcule le rang de la première réponse correcte. A la fin des tests, ces grandeurs s'élèvent à 77,42% pour MAP et 84,51% MEP. Ce qui est plutôt correcte comparé aux autres travaux sur le marché.

[Chatbot de réponse aux questions à l'aide du Deep Learning avec NLP | Publication de la conférence IEEE | IEEE Xplore \(insa-strasbourg.fr\)](#)

[Question-Answering-chatbot-for-education/components.py at main · NamrataKadam/Question-Answering-chatbot-for-education \(github.com\)](#)

Article 2 : Un système de réponses aux questions pour les étudiants

Cette étude réalisée par trois chercheurs du département d'information et d'informatique, Chung Yuan Christian University, Taoyuan, Taïwan a été menée en collaboration avec 67 étudiants dans un cours de communication en Soft Power.

Les travaux ont débuté par le traitement de données (soixante-deux fichiers vidéo de légendes et soixante-seize courts articles en cinq chapitres). Ces données sont traitées pour former une

base de données et la particularité de cette étude est l'emploi d'un modèle LDA pour classifier les données selon un sujet précis.

La dernière étape : question answering , a été divisé en trois modules, analyse de phrases, récupération de données et recherche de réponses. En fonction de la question posée, le bot affiche la liste des phrases similaires et en option, le nom de la vidéo ou du document ainsi que le début de la phrase.

[Chatbot : un système de réponse aux questions pour les | étudiants Publication de la conférence IEEE | IEEE Xplore \(insa-strasbourg.fr\)](#)

Resource 1 : Modèle BERT

Dans les systèmes actuels de réponse aux questions, les méthodes de similarité sont divisés en deux catégories : es méthodes d'apprentissage profond et les méthodes conventionnelles. Les méthodes conventionnelles reposent fortement sur des caractéristiques artificielles, ont une faible capacité de généralisation et une précision insuffisante. RNN et CNN ont également une extraction globale de fonctionnalités textuelles. Cet article propose un modèle hybride de correspondance des réponses aux questions basé sur BERT, qui utilise le modèle de pré-formation BERT pour capturer et représenter l'information sémantique de la phrase QAS et la pertinence sémantique entre les deux. Le vecteur caractéristique généré par le modèle BERT est utilisé comme Bi-LSTM _ GCN pour l'entrée du modèle, l'extraction de caractéristiques est effectuée pour obtenir davantage les caractéristiques syntaxiques de la phrase, et enfin le mécanisme d'attention est ajouté pour trouver la réponse cible, et l'efficacité de l'algorithme proposé est vérifiée sur les deux types d'ensembles de données.

[Modèle d'appariement de réponses mixtes aux questions BERT, | Publication de la conférence IEEE | IEEE Xplore \(insa-strasbourg.fr\)](#)

Ressource 2 : Ouvrage sur le NLP

La conception de notre chatbot requiert un grand nombre de connaissances dans le domaine du NLP. Cet ouvrage des éditions PACKT intitulé " Hands-on Natural language processing with python" est un guide pertinent pour la compréhension et la mise en œuvre de différentes étapes du projet. Il présente différentes applications du NLP en lien avec notre projet tels que : la transcription d'une vidéo en texte et les chatbots de question-answering.

[Hands-On Natural Language Processing with Python - ScholarVox Université \(insa-strasbourg.fr\)](#)

3. Etudes bibliographiques

La connaissance bibliographique du sujet est primordial pour mieux comprendre le sujet. D'autant plus qu'il existe plusieurs technologies et méthodes pour la création d'un chatbot. Nous souhaitons que le bot fonctionne ainsi : le professeur crée un ensemble d'intentions qui correspondent aux différentes questions que les étudiants pourraient demander. Si le bot n'a pas trouvé de réponse dans la base de données de cours, il renvoie des recommandations google, YouTube ...

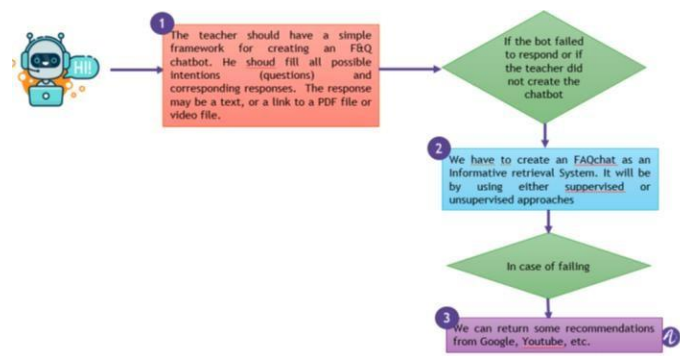


Figure 1 : Etapes de fonctionnement du bot V1

Toutefois, avec une meilleure analyse du sujet, nous avons modifié le fonctionnement pour plus de praticité. En effet, écrire l'ensemble de questions que pourraient avoir les élèves peut facilement devenir pénible pour l'enseignant, le chatbot perdrait alors son utilité. Pour y palier, nous nous sommes orientés vers un apprentissage non supervisé. A partir d'un ensemble de données disponible sur la plateforme LMS ou Rainbow Classroom, le bot doit tous les examiner pour trouver la bonne réponse. Dans un premier temps, nous commençons par un chatbot par cours.

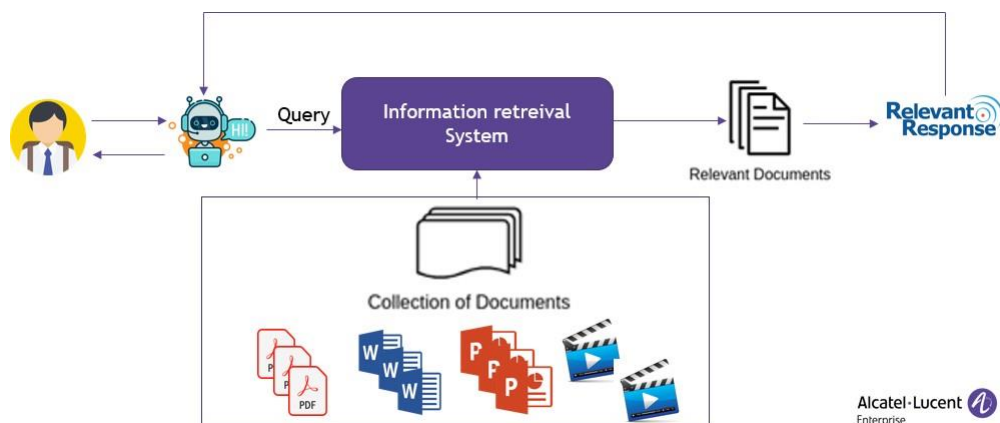


Figure 2 : Etapes de fonctionnement du bot V2

Le contexte ayant été introduit, voici une liste d'éléments nous permettons de le mettre en œuvre.

3.1 Natural Language Processing (NLP)

Traitement automatique du langage naturel est une suite de programmes informatiques développés dans le but de comprendre le langage tel qu'il est écrit ou parlé par les humains. Il résout un certain nombre de tâches comme la traduction automatique, la vérification orthographique ou grammaticale, la classification de texte, la reconnaissance d'entités nommées, leur sélection, la production de résumé, à travers une succession des étapes suivantes :

1. Saisir le texte (ou le son ou la vidéo converti en texte)
2. Segmentation du texte en composantes (segmentation et tokenisation).
3. Nettoyage du texte (filtrage de la « corbeille ») aussi suppression des éléments inutiles.
4. Vectorisation du texte et ingénierie des éléments.
5. Lemmatisation et racinisation – réduction des inflexions pour les mots.
6. Utilisation d'algorithmes et de méthodes d'apprentissage automatique pour les modèles d'entraînement.
7. Interprétation du résultat.

3.2 Lemmatisation

Cette méthode va au-delà de la réduction des mots et considère le vocabulaire complet d'une langue pour appliquer une analyse morphologique aux mots, visant à supprimer uniquement les terminaisons flexionnelles et à renvoyer la forme de base ou de dictionnaire d'un mot, connue sous le nom de lemme, c'est contrairement au stemming.

3.3 Tokenisation

En informatique, l'analyse lexicale ou tokenisation est le processus de conversion d'une séquence de caractères tel qu'un texte en une séquence de jetons lexicaux (chaînes avec une signification assignée et donc identifiée).

3.4 Détermination de la similarité

L'évaluation de la similarité entre les documents textuels est l'un des sujets importants dans plusieurs domaines tels que l'analyse de données textuelles, la recherche d'informations et

l'extraction de connaissances à partir de données textuelles (Text Mining). Dans chacun de ces domaines, les similarités sont utilisées pour différents traitements, premièrement, en analyse de données textuelles, les similarités sont utilisées pour la description et l'exploration de données, ensuite en recherche d'information, l'évaluation des similarités entre documents et requêtes est utilisée pour identifier les documents pertinents par rapport à des besoins d'information exprimés par les utilisateurs, et en Text Mining, les similarités sont utilisées pour produire des représentations synthétiques de vastes collections de documents.

Il existe deux types de similarité : Similarité syntaxique permet de comparer des documents textuels en se basant sur les chaînes de caractères qui les composent en passant par la représentation vectorielle. Par exemple, les chaînes de caractères "voiture" et "voiturier" peuvent être considérées comme très proches, alors que "voiture" et "automobile" pourront être considérées comme très différentes. Et le deuxième type est la similarité sémantique est un concept selon lequel un ensemble de documents ou de termes se voient attribuer une métrique basée sur la ressemblance de leur signification / contenu sémantique.

3.4.1 Les méthodes de mesure de la similarité

Similarité Jaccard (similarité syntaxique) : C'est la forme la plus basique de calcul de similarité entre deux textes car elle ne nécessite pas d'avoir une représentation numérique des textes. L'indice de Jaccard est mesuré en divisant le nombre de mots partagés par les deux textes par le nombre total de mots

Distance euclidienne : c'est la distance la plus courte entre deux vecteurs représentant les documents

Similarité cosinus : la similarité cosinus est fréquemment utilisée en tant que mesure de ressemblance entre deux documents. Il pourra s'agir de comparer les textes issus d'un corpus dans une optique de recherche d'information. C'est-à-dire, un document vectorisé est constitué par les mots de la requête et est comparé par mesure de cosinus de l'angle avec des vecteurs correspondant à tous les documents présents dans le corpus avec aussi et on évalue ainsi lesquels sont les plus proches.

3.5 Stop words

Stop words: ou les mots vides désigne tous les mots n'ayant pas de réelle signification comme les articles, et, mais et comment... En effet, ils sont si courants et reviennent de façon tellement régulière qu'ils ne permettent pas de caractériser, au sens lexical, un texte par rapport à un autre texte. Par exemple, si nous souhaitons rechercher "Qu'est-ce qu'une carte mère ?" sur Google, le moteur de recherche ne recherche que le terme « carte mère ».

3.6 Word Embedding

C'est un ensemble de techniques de machine Learning qui visent à représenter les mots ou les phrases d'un texte par des vecteurs de nombres réels, décrits dans un modèle vectoriel. Ces nouvelles représentations de données textuelles ont permis d'améliorer les performances des méthodes de traitement automatique des langues. C'est une étape essentielle car nos données sont essentiellement des textes mais l'ordinateur ne comprend que des nombres. Il existe plusieurs méthodes de Word Embedding dépendantes ou indépendantes du contexte.

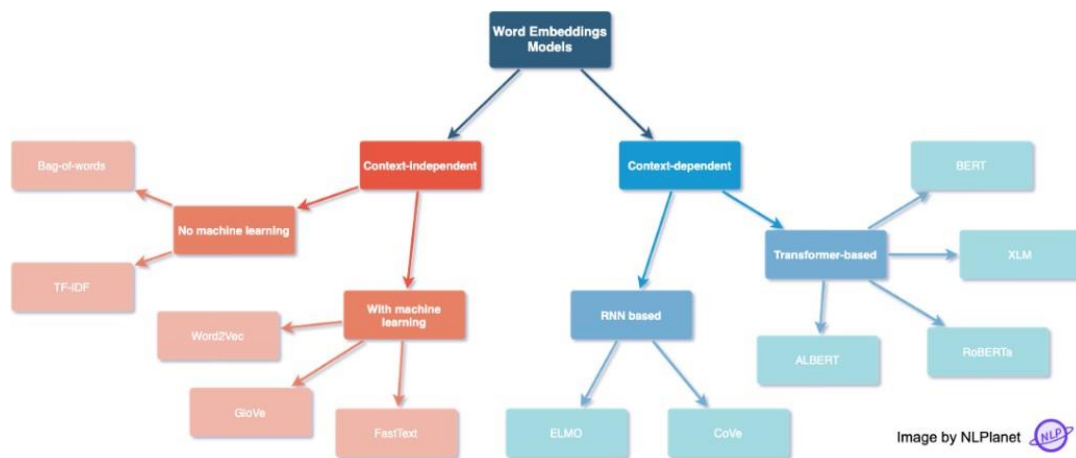


Figure 3 : Quelques méthodes d'embedding

3.6.1 Indépendant du contexte sans machine learning

Bag of Words est la technique qui était majoritairement utilisée avant l'apparition des modèles basés sur les réseaux de neurones. Ce modèle se préoccupe seulement de savoir si des mots connus se trouvent dans le document et non pas où ils se trouvent. Elle élimine toute l'information relative à l'ordre des mots et met l'accent sur l'occurrence des mots dans un document. En effet, le bag-of-words est :

- Très simple à comprendre,
- Peut être utilisé sur un faible échantillon de données alors que les techniques basées sur les réseaux de neurones ont en besoin d'énormément.

- Ne nécessite pas de capacités de calcul importantes (un CPU suffit alors que les techniques de réseaux de neurones nécessitent des GPUs ou des TPUs qui coûtent extrêmement chers).

TF-IDF : il s'agit d'une des techniques de traitement du langage naturel les plus populaires et les plus efficaces, il permet d'estimer l'importance d'un mot par rapport à tous les autres termes dans un texte ou dans une collection de textes. Cette technique est divisée en deux parties : TF, Term Frequency qui s'occupe à calculer le nombre d'occurrences d'un terme en comparaison avec le nombre total de mots dans un texte. IDF:Inverse document Frequency, mesure l'importance d'un terme dans un texte ou une collection de texte. Il est calculé comme un logarithme du nombre de documents divisé par le nombre de documents contenant ce terme. Alors, l'utilisation de la technique TF-IDF revient à évaluer les valeurs TF pour chaque mot, extraire les valeurs IDF pour ces termes pour enfin obtenir les quotients TF-IDF en multipliant les résultats de TF par IDF. Cela forme un dictionnaire de classification pour chaque mot utilisé, suivant leur importance.

3.6.2 *Indépendant du contexte avec machine learning*

Word2Vec repose sur des réseaux de neurones à deux couches et cherche à apprendre les représentations vectorielles des mots composant un texte, de telle sorte que les mots qui partagent des contextes similaires soient représentés par des vecteurs numériques proches. Cet algorithme possède deux architectures neuronales, CBOW et Skip-Gram, parmi lesquelles l'utilisateur peut choisir. CBOW reçoit en entrée le contexte d'un mot, c'est-à-dire les termes qui l'entourent dans une phrase, et essaye de prédire le mot en question. Or Skip-Gram fait exactement le contraire : elle prend en entrée un mot et essaye de prédire son contexte. Dans les deux cas, l'entraînement du réseau se fait en parcourant le texte fourni et en modifiant les poids neuronaux afin de réduire l'erreur de prédiction de l'algorithme. Pour les applications réelles, les Word2Vec sont créés à l'aide de milliards de documents, par exemple pour l'application Google est formé à l'aide de 3 millions mots et phrases.

Glove: Algorithme d'apprentissage non supervisée permettant d'obtenir des représentations vectorielles de mots. Son entraînement est basé sur des statistiques de cooccurrences de mots globales et agrégées calculées sur de vastes corpus.

FastText: Cet algorithme peut accomplir une grande exécution pour la représentation des mots et le regroupement des phrases, uniquement en raison de mots peu communs en utilisant des données au niveau des caractères.

3.6.3 Dépendant du contexte et basé sur RNN

ELMO (Embeddings from Language Model) : algorithme développé en 2018 par AllenNLP, il apprend les représentations de mots contextualisées basées sur un modèle de langage neuronal avec une couche de codage basée sur les caractères et deux couches BiLSTM. Contrairement à GloVe, Bag of Words et Word2Vec les intégrations ELMo sont capables de capturer le contexte du mot utilisé dans la phrase et peuvent générer différentes intégrations pour le même mot utilisé dans un contexte différent dans différentes phrases. Ce modèle a atteint des performances de pointe sur de nombreuses tâches populaires, notamment la réponse aux questions, l'analyse des sentiments et l'extraction d'entités nommées.

CoVe (Contextualized Word Vectors) : utilise un encodeur LSTM profond à partir d'un modèle attentionnel de séquence à séquence formé à la traduction automatique pour contextualiser les vecteurs de mots.

3.6.4 Dépendant du contexte et basé sur un transformateur

BERT (Bidirectional Encoder Representations from Transformers) : modèle de langage développé par Google en 2018. Cette méthode a permis d'améliorer significativement les performances en traitement automatique des langues. Il est basé sur un transformateur entraîné sur un vaste corpus inter-domaines. Applique un modèle de langage masqué pour prédire les mots qui sont masqués de manière aléatoire dans une séquence, suivi d'une tâche de prédiction de phrase suivante pour apprendre les associations entre les phrases.

Contrairement aux précédentes méthodes séquentielles (word2vec et autres) qui lisent les phrases dans un seul sens et qui utilisent un seul vecteur par mot, le fonctionnement bidirectionnel de BERT permet de lire les phrases dans un sens (de gauche à droite) et dans l'autre (de droite à gauche). Cela permet une vraie amélioration pour la compréhension des requêtes longues avec un système de probabilité d'apparition d'un mot dans un contexte. Ainsi un même mot peut à présent avoir plusieurs vecteurs selon son contexte.

XLM (Cross-lingual Language Model) : il s'agit d'un transformateur préentraîné à l'aide de la prédiction du jeton suivant, d'un objectif de modélisation du langage masqué de type BERT et d'un objectif de traduction. XLM utilise une technique de prétraitement connue (BPE) et un mécanisme de formation bilingue avec BERT afin d'apprendre les relations entre les mots dans différentes langues. Le modèle surpasse les autres modèles dans une tâche de classification multilingue (implication de phrases en 15 langues) et améliore considérablement la traduction automatique lorsqu'un modèle pré-entraîné est utilisé pour l'initialisation du modèle de traduction.

RoBERTa (Robustly Optimized BERT Pretraining Approach) : il s'appuie sur BERT et modifie les hyperparamètres clés, supprimant l'objectif de préformation de la phrase suivante et la formation avec des mini-lots et des taux d'apprentissage beaucoup plus importants. Le modèle RoBERTa a été proposé dans RoBERTa: A Robustly Optimized BERT Pretraining Approach par Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. Il est basé sur le modèle BERT de Google publié en 2018.

ALBERT (A Lite BERT for Self-supervised Learning of Language Representations) : il présente des techniques de réduction des paramètres pour réduire la consommation de mémoire et augmenter la vitesse d'entraînement de BERT. L'idée principale d'ALBERT est de réduire le nombre de paramètres (jusqu'à 90% de réduction) en utilisant de nouvelles techniques tout en ne prenant pas un gros coup à la performance. Maintenant, cette version compressée évolue beaucoup mieux que le BERT original, améliorant les performances tout en gardant le modèle petit.

3.7 Question-answering

La réponse aux questions (QA) est une discipline informatique dans les domaines de la recherche d'information et du traitement du langage naturel (NLP), qui s'intéresse à la construction de systèmes qui répondent automatiquement aux questions posées par les humains dans un langage naturel.

3.8 Extraction des données :

Parmi les étapes les plus importantes dans la recherche des informations est l'extraction ou le scraping des données des fichiers étudiés de n'importe quel type, soit de format PDF, PPT ou Word...

Actuellement, les formats les plus utilisés et les plus populaires est le format PDF, les professeurs dépose souvent des documents en anglais et en français en format PDF.

Nous avons suivi une démarche pour tous les formats existe PDF, Word, PPT ; dans un premier temps, nous utiliserons les bibliothèques PPTX2PDF et DOCX2PDF dans le but de transformer les deux types des fichiers en PDF et à l'aide de la bibliothèque PDF2Image nous avons convertir les pages de PDF en images. Enfin, nous suivrons le code de la bibliothèque EasyOcr pour que cette dernière transformer ces images en texte.

4. Codes et tests

Les termes expliqués précédemment n'ont pas été choisis de manière anodine. En effet, dans la création d'un question-answering chatbot, ils reviennent le plus souvent. Dans ce rapport, nous vous proposons trois codes pythons : <https://github.com/Achta00/Chatbot>

- Comparaison de la méthode de similarité avec Word2vec sur un fichier pdf
- Question-réponse avec BERT sur un fichier pdf
- Question-réponse avec cdQA sur un ensemble de fichiers (liens html)

Pour des raisons de fluidité de lecture, nous n'avons pas inséré les codes dans le rapport. Ils sont disponibles via le lien github suivant pour effectuer des tests sur google Colab.

4.1 Comparaison méthode de similarité et Word2vec

Nous avons travaillé sur un code de question réponse avec les deux méthodes la similarité en cosinus et le modèle pré-entraîné de la méthode Word2Vec.

La similarité en cosinus se basé sur la recherche de la similarité entre les embeddings de la question et chaque phrase de documents étudiés. En effet, la phrase qui présente la plus grande similarité avec la question est prédite comme la sortie du modèle pour la question donnée.

Les images suivantes illustrent les résultats des deux méthodes :

```
[ ] print("Question: ", question)
    print("Answer: ", sentences[index])

Question: what are advantages of chatbot
Answer: chatbot short for chatterbot is an artificial intelligence ai feature that can be embedded and used through any major messaging application
```

```
▶ print("Question: ", question)
   print("Answer: ", cleaned_sentences_with_stopwords[index])


Question: what are advantages of chatbot
Answer: some examples of chatbot technology are virtual assistants like amazons alexa and google assistant and messaging apps such as wechat and facebook's messenger
```


Selon la bibliothèque Gensim, le modèle Word2Vec sélectionne une phrase du texte de l'article qui a la plus grande similarité d'intégration Word2Vec avec la question donnée, et la produit comme réponse

4.2 Question-réponse avec BERT

Pour résumé ce code, nous utilisons un modèle de BERT et de tokenizer pré-entraîné à partir de la base de données SQUAD. Ils ne font aucune différence entre minuscule et majuscule.

```
▶ import torch
  from transformers import BertForQuestionAnswering
  model = BertForQuestionAnswering.from_pretrained('bert-large-uncased-whole-word-masking-finetuned-squad')
```

Download: 100%  443/443 [00:00<00:00, 9.77kB/s]

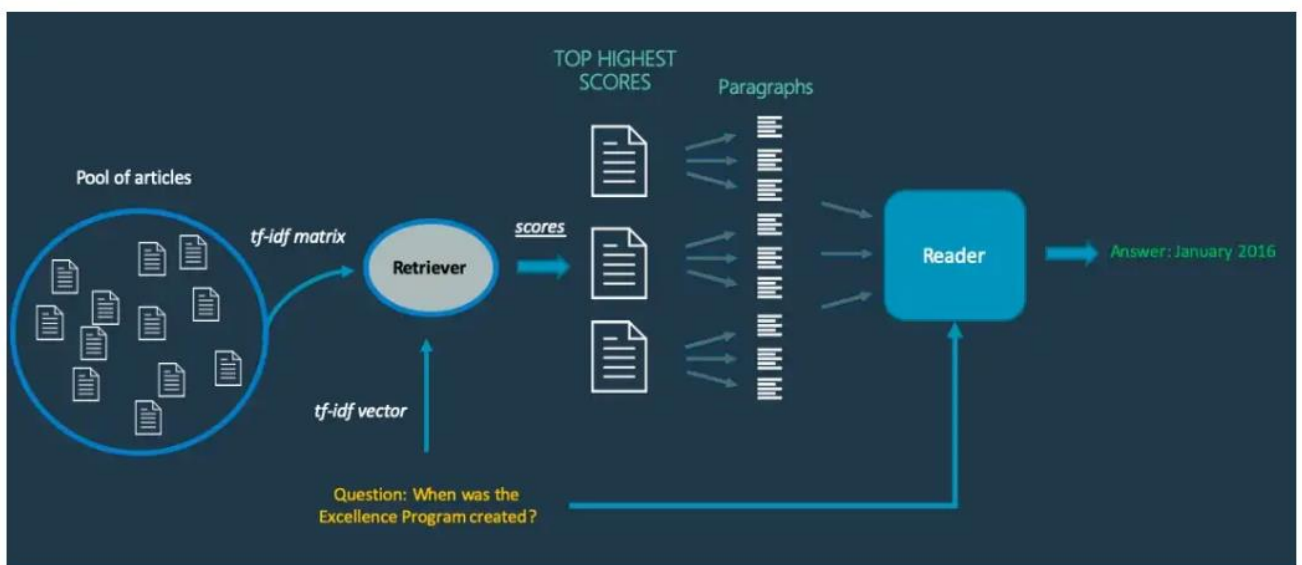
Download: 100%  1.34G/1.34G [00:40<00:00, 37.7MB/s]

```
[ ] from transformers import BertTokenizer
    tokenizer = BertTokenizer.from_pretrained('bert-large-uncased-whole-word-masking-finetuned-squad')
    # Téléchargement d'un tokenizer préentraîné
```

Pour un texte simple, nous créons un tenseur que nous limitons à 512 jetons. Ce qui nous permet de créer un dictionnaire avec les mots du texte. Nous ajoutons les jetons de début de séquence [CLS] et de séparateur [SEP]. Pour cela, nous utilisons la fonction `torch.cat`, qui concatène une liste de tenseurs. A partir d'une lecture du texte par le modèle, nous déterminons la réponse la plus probable. Après analyse avec les méthodes précédentes, le modèle BERT nous permet d'avoir de meilleur résultat. En effet, dans les seuls, il est le seul à avoir trouvé une réponse correcte car celle-ci avait plus de cohérence avec la question. Pour les autres, on obtenait juste une phrase du texte avec l'élément clé de la question.

4.3 Question-réponse avec cdQA sur un ensemble de fichiers

Closed Domain Question Answering est une suite logicielle open source de bout en bout pour la réponse aux questions utilisant des méthodes IR classiques et l'apprentissage par transfert avec le modèle pré-formé BERT.



Mechanism of cdQA pipeline

Le code de la cdQA traite une base de données dans un fichier CSV qui contient la date, le titre, le lien d'article bnp accessible sur internet.

Le package CDQA contient des packages permettant de convertir des fichiers PDF en texte interrogeable dans des blocs de données.

Nous avons appelé le pipeline pour lire le modèle et l'ajuster dans un cadre de données créé.

```
cdqa_pipeline = QAPipeline(reader='./models/bert_qa.joblib')  
cdqa_pipeline.fit_retriever(df=df)
```

Ensuite nous avons commencé à poser des questions et les réponses seront prédites.

```
query = 'Since when does the Excellence Program of BNP Paribas exist?'  
prediction = cdqa_pipeline.predict(query)
```

Conclusion

Dans le cadre de cette première partie du PRT, nous avons compris un certain nombre de notions en lien avec le projet. De plus, à partir des codes testés, nous avons déterminé les méthodes nous permettant d'obtenir les meilleurs résultats. Pour la suite du projet, nous étudierons les points suivants :

- question-réponse avec cdQA pour avoir un temps de latence le plus court possible
- Chatbot bilingue (anglais et français)
- Transcription vidéo en texte

Bibliographie :

[Chatbot/ cdqa.ipynb at main · Achta00/Chatbot · GitHub](#)

<https://github.com/yuvraj123-verma/Smart-Question-Answering-System-on-Document>

<https://github.com/Achta00/Chatbot>

[Chatbot de réponse aux questions à l'aide du Deep Learning avec NLP | Publication de la conférence IEEE | IEEE Xplore \(insa-strasbourg.fr\)](#)

[Question-Answering-chatbot-for-education/components.py at main · NamrataKadam/Question-Answering-chatbot-for-education \(github.com\)](#)

[Hands-On Natural Language Processing with Python - ScholarVox Université \(insa-strasbourg.fr\)](#)

[Modèle d'appariement de réponses mixtes aux questions BERT, | Publication de la conférence IEEE | IEEE Xplore \(insa-strasbourg.fr\)](#)

[Chatbot : un système de réponse aux questions pour les | étudiants Publication de la conférence IEEE | IEEE Xplore \(insa-strasbourg.fr\)](#)