

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ SAAD DAHLEB – BLIDA 1

FACULTÉ DES SCIENCES
DÉPARTEMENT D'INFORMATIQUE



MÉMOIRE DE MASTER II

Spécialité : Ingénierie des Logiciels

THÈME

LES RÉSEAUX DE NEURONES POUR LA GÉNÉRATION
DE QUESTIONS ÉDUCATIVES EN LANGUE ARABE

Présenté par MM :

IBRIR ADEL

MIHOUBI AKRAM FAWZI

Devant le jury composé de :

Président

Mme Cheriguene.S

Examineur

M Benaissi.S

Promotrice

Mme Ouahrani.L

Soutenu le 04/10/2022

Remerciements

Au terme de cinq dernières années de formation, accomplies par la réalisation du présent mémoire avec beaucoup de sacrifices et abnégation, nous exprimons notre profonde gratitude à notre promotrice Madame L. OUAHRANI pour nous avoir guider, accorder l'entière disponibilité et offert l'opportunité de travailler sous votre direction.

Notre reconnaissance pour les efforts consentis à notre égard pour mener à bien cette mission.

Nos vifs remerciements s'adressent au Président et aux membres examinateurs du jury, dont vous nous faites le grand honneur d'avoir accepté d'évaluer ce travail, qu'ils trouvent également le témoignage de notre profond respect et considération.

C'est un honneur de partager et d'exprimer notre reconnaissance que nous devons toutes les personnes ayant contribuer à la réalisation de ce travail trouvent par le présent témoignage de notre estime et reconnaissance.

Résumé

La Génération Automatique des Questions (AQG) à partir d'une phrase ou d'un paragraphe est une tâche difficile dans le Traitement Automatique de Langage (NLP). Cette tâche a des implications importantes dans le domaine de l'éducation par les possibilités offertes dans la génération automatique des questions à partir d'un contenu éducatif sans recourir à un effort manuel de la part des éducateurs.

Dans ce contexte, plusieurs études ont été menées sur la génération des questions basées sur des approches d'apprentissage en profondeur. À notre connaissance, les travaux réalisés à ce jour ne disposent pas de modèle de génération de questions en langue arabe et qui ne s'appuient que sur des approches traditionnelles reposant sur des règles érigées manuellement, ces approches sont limitées en termes de diversité linguistique exigeant un effort humain considérable.

Récemment, de nombreux modèles de langage pré-entraînés ont fourni des performances élevées dans plusieurs tâches de NLP pour étudier la génération de questions en langue arabe. Dans ce présent travail, nous avons affiné un modèle pré-entraîné Ara-T5 (*Text-to-Text Transformers for Arabic Language*) à l'aide d'un Dataset Question-Réponse arabe.

Les résultats d'évaluation ont montré que l'approche proposée permet de générer des questions excellentes en termes de fluidité et grammaticalité et bonnes en pertinence.

Mots Clés : Génération automatique de questions ; Transformers ; Apprentissage profond ; Modèle pré-entraîné.

Abstract

Automatic Question Generation (AQG) from a sentence or paragraph is a difficult task in Automatic Language Processing (NLP). This task has important implications in the field of education through the possibilities offered in the automatic generation of questions from educational content without resorting to a manual effort on the part of educators.

In this context, several studies have been conducted on the generation of questions based on deep learning approaches. To our knowledge, the works carried out to date do not have models for generating questions in the Arabic language and which only rely on traditional approaches based on manually erected rules, these approaches are limited in terms of linguistic diversity requiring considerable human effort.

Recently, many pre-trained language models have provided high performance in several NLP tasks to study Arabic language question generation. In this present work, we fine-tune a pre-trained Ara-T5 (Text-to-Text Transformers for Arabic Language) model using an Arabic Question-Answer Dataset.

The evaluation results showed the proposed approach makes it possible to generate excellent questions in terms of fluency and grammaticality and good in relevance.

Key words: Automatic questions generation; Transformers; Deep Learning; Pre-trained Model.

المخلص

يعد إنشاء الأسئلة آلياً أو بصفة تلقائية (AQG) من جملة أو فقرة مهمة صعبة في مجال المعالجة الآلية للغة الطبيعية (NLP). هذه المهمة لها آثار إيجابية هائلة في مجال التعليم من خلال الإمكانيات المتوفرة في التوليد التلقائي للأسئلة من المحتوى التعليمي دون اللجوء إلى الجهد اليدوي من جانب المعلمين.

في هذا السياق، تم إجراء العديد من الدراسات حول توليد الأسئلة بناءً على مناهج التعلم العميق. على حد علمنا، لا توجد أعمال قائمة على منهج التعليم العميق التي تم تنفيذها حتى الآن لتوليد الأسئلة باللغة العربية، حيث تعتمد الأساليب التقليدية القائمة حول هذه اللغة على القواعد الموضوعية يدوياً فقط، وهذه الأساليب محدودة من حيث التنوع اللغوي وتتطلب جهداً بشرياً كبيراً.

في الآونة الأخيرة، قدمت العديد من نماذج اللغة المدربة مسبقاً أداءً عاليًا في العديد من مهام المعالجة الآلية اللغوية لتمكننا من دراسة توليد أسئلة اللغة العربية. في هذا العمل الحالي، قمنا بضبط النموذج (Ara-T5) محول نص إلى نص للغة العربية) باستخدام مجموعة بيانات الأسئلة - الأجوبة عربية.

اقترحنا حل لتوليد أسئلة آلياً للغة العربية في مجال التعليم، حيث أظهرت نتائج التقييم أن النهج المقترح يجعل من الممكن طرح أسئلة ممتازة من حيث الطلاقة والنحو وجيدة فيما يتعلق بالملائمة.

الكلمات المفتاحية: توليد الأسئلة آلياً؛ محولات؛ تعلم العميق؛ نموذج مدرب مسبقاً.

TABLE DES MATIÈRES

INTRODUCTION GÉNÉRALE	11
CHAPITRE I: CONCEPTS FONDAMENTAUX DE L'APPRENTISSAGE PROFOND ET DE MODÈLES DE SÉQUENCES.....	15
Introduction.....	15
I.1 Réseaux de neurones récurrents (<i>RNN</i>)	15
I.2 Réseaux à Mémoire Longue et Courte Durée (<i>LSTM</i>)	17
I.3 Le Réseau de Neurones Récurrents à Portes (<i>GRU : Gated Recurrent Unit</i>)	19
I.4 Encodeur - décodeur.....	21
I.5 Mécanisme d'attention	22
I.6 Transformer	23
I.7 Modèles Pré-entraînés	24
I.7.1 BERT	25
I.7.2 Text To Text Transform Learning (T5)	27
I.7.3 Modèle AraT5.....	28
Conclusion	29
CHAPITRE II : ETAT DE L'ART GÉNÉRATION DE QUESTIONS	30
Introduction	30
II.1 Traitement Automatique de Langage (NLP)	30
II.2 Plongement de Mots (<i>Word Embedding</i>)	31
II.3 Génération de Questions (QG)	32
II.3.1 Domain d'application	33
II.4 Les Approches de génération de questions	35
II.4.1 Les Approches basées sur les règles	35
II.4.2 Les approches basées sur les réseaux de neurones	39
II.5 Datasets les plus utilisés dans la génération de questions	40
II.6 Métriques d'évaluation utilisées dans la génération de questions	41
II.7 Travaux connexes	42
II.8 Synthèse des travaux	46

Conclusion	48
CHAPITRE III : GÉNÉRATEUR DE QUESTIONS EN LANGUE ARABE	49
Introduction.....	49
III.1 Définition de la tâche de génération de questions.....	49
III.2 Approche proposé	49
III.3 Méthodologie	51
III.4 Dataset	52
III.4.1 Description du dataset	52
III.4.2 Prétraitement du dataset	53
III.5 L'entraînement du modèle ANQG	56
Conclusion	58
CHAPITRE IV : REALISATION & EXPEREMENTATION	59
Introduction	59
IV.1 Environnement	59
IV.1.1 Google colaboratory	59
IV.1.2 Pytorch Lightning	59
IV.1.3 Tranformers	60
IV.2 Résultats et interprétations.....	60
IV.2.1 Evaluation automatique.....	60
IV.2.2 Evaluation humaine qualitative.....	62
IV.3 Analyse des questions générés par ANQG	62
IV.3.1 Les questions générée de l'ensemble de test	63
IV.3.2 Les questions générée à partir des exemples extrait du web	65
IV.4 Interface du développement.....	66
Conclusion	68
CONCLUSION GENERALE ET PERSPECTIVES	69
REFERENCES BIBLIOGRAPHIQUES.....	71

LISTE DES ACRONYMES

NLP	Natural Language Processing
ANN	Artificial Neural Networks
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated recurrent Unit
Seq2seq	Sequence-to-Sequence
QG	Question Generation
QA	Question Answering
NQG	Neural Question Generation
AQG	Automatic Question Generation
MRC	Machine Reading Comprehension
SQuAD	Stanford Question Answering Dataset
POS	Part of Speech
NER	Named Entity Recognition
BERT	Bidirectional Encoder Representations from Transformers
T5	Text-to-Text Transfer Transformer
ANQG	Arabic Neural Question Generation
CQA	Conversational Question Generation

LISTE DES FIGURES

Figure I.1 : Architecture d'un réseau de neurones récurrent.....	15
Figure I.2 : Les différents types de réseaux RNN.....	16
Figure I.3 : Représentation de la porte d'oubli.....	17
Figure I.4 : Architecture de la porte d'entrée.....	18
Figure I.5 : Mise à jour d'état de cellule dans la porte d'entrée.....	18
Figure I.6 : Architecture de la porte de sortie.....	19
Figure I.7 : Architecture d'une cellule GRU	20
Figure I.8 : Architecture d'Encodeur-Décodeur	21
Figure I.9 : Représentation graphique des zones d'attention.....	22
Figure I.10 : Architecture du Transformer.....	24
Figure I.11 : Chronologie des versions populaires des modèles de langage pré-entraînés.	25
Figure I.12 : Architecture du modèle BERT	26
Figure I.13 : Représentation des entrées du modèle BERT.....	27
Figure I.14 : Concept multi-tâches du modèle T5	27
Figure II.1 : Représentation des mots dans un plan 2D	32
Figure II.2 : Représentation du modèle NQG de Zhou et <i>al.</i>	43
Figure II.3 : Présentation du modèle Gated Self-Attention avec pointeur Maxout.....	44
Figure II.4 : Architecture du modèle BERT-HLSQG.....	46
Figure III.1 : Illustration schématique de la conception de notre système.....	51
Figure III.2 : Deux exemples d'un tuple contexte-question-réponse de ARCD...	53
Figure III.3 : La tokenization d'une phrase à l'aide du Ara-t5 tokenizer.....	55
Figure III.4 : Préparation la phrase d'entrée pour le modèle ANQG	56
Figure IV.1 : évaluation automatique d'un seul échantillon de l'ensemble du test.	61
Figure IV.2 : échantillon des questions générées par le modèle ANQG	64
Figure IV.3 : Interface graphique de modèle ANQG.....	67

LISTE DES TABLEAUX

Tableau I.1: Présentation Comparative des modèles pré-entraînés	25
Tableau I.2: Les datasets utilisées pour le prétraitement de Ara-T5	28
Tableau II.1: les modèles de questions proposés pour la tâche QG.....	36
Tableau II.2 : Les étiquettes de rôles sémantiques.....	38
Tableau II.3 : le découpage usuel du dataset SQuAD.....	45
Tableau II.4 : Résultat de comparaison des modèles sur SQuAD dans <i>Du et al</i>	46
Tableau II.5 : Résultat de comparaison des modèles sur SQuAD dans <i>Zhou et al</i>	47
Tableau III.1 : Détails du dataset utilisé.....	52
Tableau III.2: Partitionnement du dataset.....	54
Tableau III.3: Configuration des hyperparamètres dans notre modèle.....	57
Tableau IV.1 : Résultat de comparaison entre notre modèle ANQG et les meilleurs modèles existants.....	60-61
Tableau IV.2 : Résultat d'évaluation humain de notre modèle.....	62
Tableau IV.3 : échantillon des questions générées par le modèle ANQG extraites du web	65

INTRODUCTION GENERALE

Soulever des questions est une action quotidienne, utile dans le cadre professionnel et occasionnel, il permet d'un côté de satisfaire la nature curieuse de l'être humain en l'aidant à découvrir les limites de son potentiel puis les transcender ou tout simplement répond à ses besoins courants, et de l'autre côté questionner peut jouer un rôle primordial pour juger les connaissances de quelqu'un comme le cas des épreuves scolaires ou les interviews.

Sachant que, poser des questions semble être une composante élémentaire de notre communication en tant qu'êtres humains, cela peut devenir un sujet très complexe à explorer, surtout si l'intention est de transférer cet art de poser des questions aux machines. Pour poser une question, il est important de produire une séquence de langage naturel correcte qui soit sous la forme interrogative, ce qui en soi est une tâche qui n'est pas triviale à réaliser automatiquement.

La difficulté de la tâche de la génération automatique des questions augmente lorsque nous nous attendons à des questions substantielles liées à un sujet donné, généralement écrites dans un texte non structuré. Quel que soit le système qui génère ces questions, il doit extraire les informations pertinentes de ce texte et les utiliser dans la question.

En parallèle de ces défis on trouve les principales motivations de la génération automatique des questions comme elle peut être considérée comme une tâche qui affecte de nombreux aspects de la vie des gens, et peut également aider à fournir des informations robustes pour d'autres tâches dans les domaines de recherche sur le NLP telles que le développement des systèmes de réponse aux questions (*Question Answering*) et les systèmes de compréhension en lecture automatique. Un système QG peut améliorer les entrées et fournir un dataset complet pour de tels systèmes, par conséquent, stimuler la recherche dans ces tâches.

De plus, des questions sont également générées à d'autres fins, telles que la validation des bases de connaissances, évidemment dans le domaine éducationnel un système QG peut bénéficier aux enseignants en facilitant leur travail car, la génération manuelle des questions est une tâche fastidieuse et chronophage, surtout lorsqu'un grand nombre de questions sont

nécessaires pour évaluer les connaissances de l'apprenant. Ainsi, que l'évaluation des réponses peut prendre énormément de temps, surtout avec l'essor des plateformes *e-learning* qui devient un impératif. En outre, le développement d'agents conversationnels les *Chatbots* médicaux qui servent d'assistants de santé intelligents et peuvent déterminer la maladie d'une personne en posant des questions sur les symptômes du patient et qui peuvent être améliorés en utilisant un système QG (le développement d'agents conversationnels).

Les premières tentatives de génération automatique des questions dépendent d'ensemble de règles créées par l'humain, ces règles se concentrent principalement sur la structure syntaxique du texte et sont limitées dans leur application. Les approches récentes d'apprentissage en profondeur (*Deep Learning*) représentent l'état de l'art dans la tâche de génération.

Dans cette étude nous explorons la génération automatique des questions en langue arabe vue que cette précieuse langue est malheureusement pauvre en termes de données ce que nous a motivé à exploiter la performance des différents modèles d'intelligence artificielle utiliser dans le domaine du NLP et nous essayons d'adapter le meilleur modèle Adéquat pour générer des questions éducationnelles en arabe.

Problématique

La nécessité d'évaluation des connaissances dans les milieux éducatifs a mené la communauté scientifique à s'intéresser aux problématiques de la génération automatique de questions.

De nombreuses plateformes *e-learning* telles que les cours en ligne ouverts et massifs (MOOC) et l'apprentissage adaptatif nécessitent de plus en plus un nombre de questions éducationnelles, bien que les experts humains soient actuellement irremplaçables pendant la phase de vérification et il est souvent difficile d'évaluer traditionnellement les acquis de connaissances.

Un système automatique peut toujours soutenir l'éducation.

En effet, le principe général de génération automatique de questions est de produire une question naturelle à partir d'une phrase ou d'un paragraphe. En parallèle, vu que cette étude se focalise sur la langue arabe, plusieurs défis entrent en jeu vu la nature et la spécificité de langue et l'absence d'outils de traitement qui lui sont dédiés.

Objectifs

Le travail se déroule dans le contexte de la langue arabe où plusieurs défis doivent être considérés liés particulièrement au manque de ressources. Notre objectif principal consiste, tout d'abord, à automatiser le processus de génération de questions éducatives en version arabe en se basant sur les techniques d'apprentissage profond.

Vu la nature du problème, nous explorons les modèles neuronaux « séquence à séquence » pour construire un système de génération de questions. Le but du système recherché est de générer des questions pour des tests éducatifs. Pour ce faire, la génération automatique doit être effectuée par la réalisation des étapes suivantes :

- a. Explorer les approches de génération de questions.
- b. Explorer les modèles d'apprentissage profond « sequence to sequence » pour en adapter à la génération de questions (Transformer, Encodeur-decodeur, Lstm, Bi-Lstm, ...)
- c. Acquisition & collecte du dataset arabe pour l'entraînement du modèle proposé.
- d. Concevoir et implémenter le modèle de génération de questions.
- e. Évaluer la qualité du modèle de génération des questions (Evaluation quantitative par des métriques automatiques & une évaluation humaine par une expertise humaine).

Le présent mémoire est structuré en quatre chapitres dont le premier chapitre est consacré à présenter l'état de l'art des modèles d'apprentissage en profondeur (*Deep Learning*) employés dans le NLP. Le second chapitre est consacré à une description détaillée de la tâche relative à la génération de questions.

Le troisième chapitre est dédié à la conception du générateur de questions basées sur l'architecture de Transformer par l'utilisation du modèle pré-entraîné.

Quant au dernier chapitre, nous avons procédé à l'évaluation humaine et automatique du système élaboré.

CHAPITRE I :

CONCEPTS FONDAMENTAUX DE L'APPRENTISSAGE PROFOND ET DE MODÈLES DE SÉQUENCES

Introduction

Ces dernières années, l'apprentissage en profondeur a révolutionné le domaine de traitement automatique de langage naturel (NLP) en améliorant les performances de différentes tâches basées sur NLP telles que : l'analyse des sentiments, la traduction automatique, les réponses aux questions.

Dans ce chapitre nous présentons les modèles du Deep Learning utilisé dans le traitement automatique de langage naturel, et nous élaborons l'architecture et le fonctionnement de chacun ainsi que leurs avantages et inconvénients.

I.1 Réseaux de neurones récurrents (RNN : *Recurrent Neural Network*)

Le RNN est un type de réseaux de neurones adapté aux traitements des données séquentielles, Il est doté d'un état caché pour utiliser les sorties précédentes comme entrée, Cet état lui permet de traiter des séquences d'entrée de n'importe quelle longueur, ce qui le rend parfaitement adapté aux nombreuses tâches de traitement automatique de langage. La figure I.1 illustre l'architecture d'un réseau de neurones récurrent.

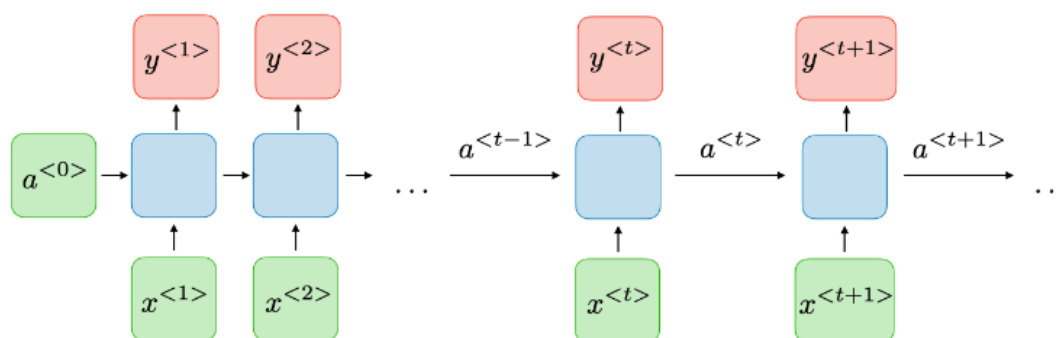


Figure I.1 : Architecture d'un réseau de neurones récurrent [1]

A chaque instant t , l'activation $a^{<t>}$ et la sortie $y^{<t>}$ sont exprimées comme suit :

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad \text{et} \quad y^{<t>} = g_2(W_{ya}a^{<t>} + b_y) \quad (\text{I.1})$$

Où, x_t et a_t sont respectivement le vecteur d'entrée du réseau et le vecteur d'activation à l'instant t , W et B sont respectivement les poids et les biais à apprendre durant l'entraînement du réseau et g_1, g_2 sont des fonctions d'activation.

Nous constatons clairement l'aspect récurrent dans ces calculs (chaque calcul à l'instant t dépend de calculs précédents $t-1$), contrairement au réseau de neurones classique où la sortie dépend uniquement des valeurs d'entrées.

Il existe quatre types de réseau de neurones récurrent à savoir :

- **Un à Un** : Il s'agit d'un type de RNN le plus élémentaire connu sous le nom "*Vanilla Neural Network*" pour la simplicité de son algorithme. Ce type est utilisé pour les problèmes généraux d'apprentissage automatique, qui ont une seule entrée et une seule sortie.
- **Un à Plusieurs** : Ce type de réseau prend une seule entrée et génère plusieurs sorties, il traite généralement des problèmes comme sous-titrage d'une image.
- **Plusieurs à Un** : Cette variante prend une séquence d'entrées et génère une sortie unique, il est utilisé lorsque plusieurs entrées sont nécessaires pour donner une seule sortie. L'analyse de sentiment est un exemple courant dans ce type.
- **Plusieurs à Plusieurs** : Utilisé pour générer une séquence de données de sortie à partir d'une séquence d'unités d'entrée. Il est appliqué dans la résolution des problèmes complexes comme la traduction automatique.

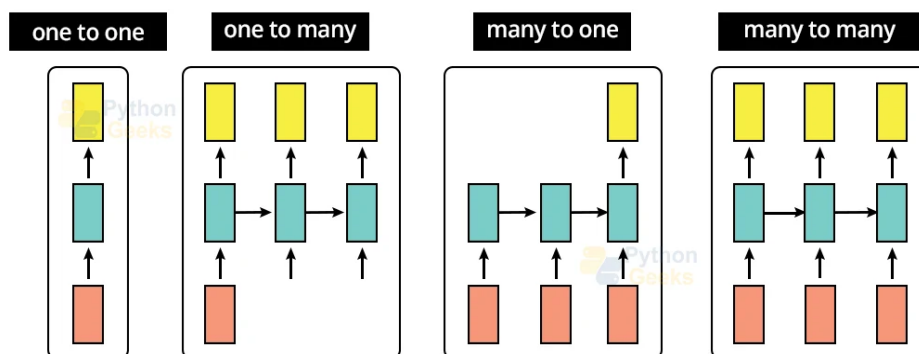


Figure I.2: Les différents types de RNN [2]

Malgré que le RNN est construit spécialement pour gérer les données séquentielles, néanmoins il souffre de problèmes de mémoire court terme et assez difficile de le former sur les séquences longues.

I.2 Réseau à Mémoire Longue et Courte Durée (LSTM : *Long Short-Term Memory*)

Le réseau à mémoire longue et courte durée (LSTM) a été introduit en 1997 par Hochreiter et Schmidhuber [3], Il s'agit d'une version améliorée de réseau de neurones récurrents(RNN) capable d'apprendre les dépendances à long terme.

Une unité LSTM de base est composée d'une cellule qui mémorise les valeurs sur des intervalles arbitraires, et trois portes (porte d'oubli, porte d'entrée, porte de sortie), régulent le flux d'informations entrant et sortant de la cellule. Le fonctionnement global d'un LSTM peut se résumer en 3 étapes :

I.2.1 La porte d'oubli (*Forget Gate*)

La première étape consiste à décider quelles informations doivent être éliminées ou conservées. L'état précédent h_{t-1} et l'entrée actuelle x_t sont transmises via la fonction sigmoïde. Le Forget Gate est calculé comme suit :

$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f) \quad (I.2)$$

Où

W_f correspond au poids des neurones et b_f au biais.

Le résultat f_t est un vecteur dont les valeurs sont comprises entre 0 et 1, plus près de 0 signifie d'oublier et plus près de 1 signifie de garder l'information. L'architecture de cette porte est illustrée dans la figure suivante :

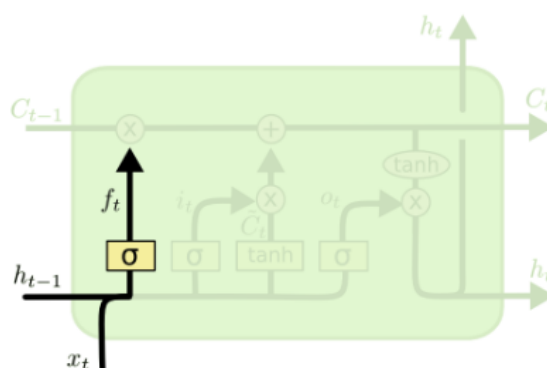


Figure I.3 : Représentation de la porte d'oubli [4]

I.2.2 La porte d'entrée (Input Gate)

Le but de cette étape est d'ajouter des informations utiles à l'état de la cellule. Tout d'abord, les informations sont filtrées à l'aide d'une fonction sigmoïde de la même manière que l'étape précédente à l'aide des entrées h_{t-1} et x_t :

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (I.3)$$

Ensuite les mêmes entrées sont passées à la fonction Tanh (Tangente Hyperbolique) qui génère un vecteur de valeurs \bar{C}_t comprises entre -1 et 1 Tanh. La sortie \bar{C}_t est calculée comme suit :

$$\bar{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (I.4)$$

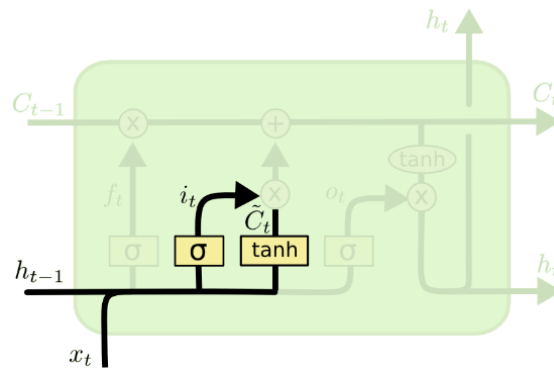


Figure I.4 : Architecture de la porte d'entrée [4]

Pour mettre à jour l'ancien état de la cellule C_{t-1} dans la nouvelle cellule C_t . Nous multiplions l'état de la cellule précédente par le vecteur d'oubli f_t puis nous ajoutons la multiplication de la porte d'entrée i_t et vecteur de valeurs \bar{C}_t à savoir :

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t \quad (I.5)$$

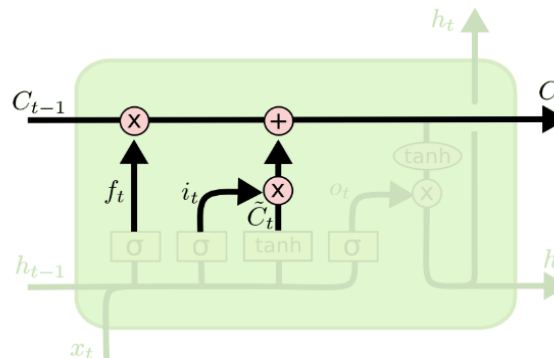


Figure I.5 : Mise à jour d'état de cellule dans la porte d'entrée [4]

I.2.3 La porte de sortie (*Output Gate*)

La porte de sortie définit le prochain état caché qui contient des informations sur les entrées précédentes.

Premièrement, l'état caché précédent h_{t-1} et l'entrée actuelle x_t doivent passer dans une fonction sigmoïde afin de sélectionner les alurs qui seront autorisés à passer :

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (I.6)$$

Ensuite, nous passons l'état de cellule nouvellement modifié C_t à la fonction \tanh , et nous multiplions la sortie \tanh avec la sortie sigmoïde pour décider quelles informations l'état caché doit transporter. La sortie h_t est l'état caché s'écrit :

$$h_t = O_t * \tanh(C_t) \quad (I.7)$$

Le nouvel état de la cellule C_t et le nouvel état caché h_t sont reportés au pas de temps suivant.

L'architecture de cette porte est illustrée dans la figure I.6 :

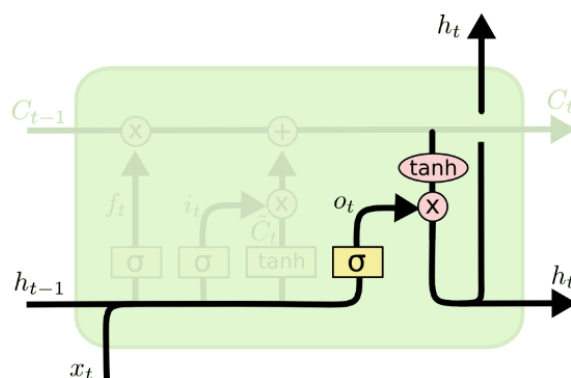


Figure I.6 : Architecture de la porte de sortie [4]

I.3 Le Réseau de Neurones Récurrents à Portes (*GRU : Gated Recurrent Unit*)

Le réseau de neurones récurrent à portes (GRU) est un autre variant de RNN, il a été introduit en 2014 par Cho et *al* [5] pour résoudre le problème de disparition du gradient rencontré par les réseaux récurrents classiques, mais aussi pour proposer une architecture avec moins de paramètres à entraîner par rapport à un LSTM avec des performances similaires dans la plupart des tâches, l'architecture du modèle GRU est illustré dans la figure suivante :

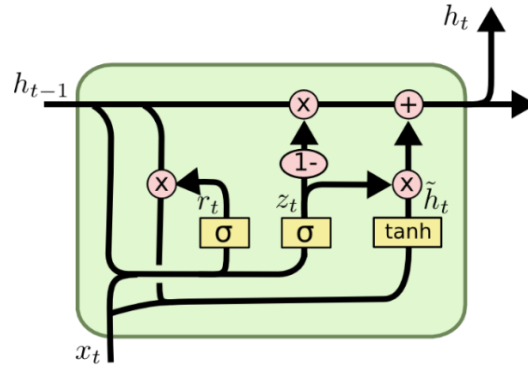


Figure I.7 : Architecture d'une cellule GRU[4]

Les différents composants d'une cellule GRU sont décrits ci-après :

I.3.1 Porte de mise à jour

La porte de mise à jour détermine la quantité d'information qui doit être transmise au futur. Elle est similaire de la porte de sortie dans l'unité de LSTM, l'équation de la porte est illustrée ci-dessous :

$$z_t = \sigma(W_z * x_t + U_z * h_{t-1}) \quad (I.8)$$

Où W_z et U_z sont des matrices de poids pour la porte de mise à jour.

I.3.2 Porte de réinitialisation

La porte de réinitialisation détermine la quantité d'information qui doit être oubliée, elle est calculée par la formule suivante :

$$r_t = \sigma(W_r * x_t + U_r * h_{t-1}) \quad (I.9)$$

Où

W_t et U_t représentent des matrices de poids comme dans z_t .

I.3.3 Contenu actuel de la mémoire

Dans cette étape, le nouveau contenu de la mémoire \hat{h}_t utilise la porte de la réinitialisation pour stocker les informations pertinentes du passé, il est calculé comme suit :

$$\hat{h}_t = \tanh(W \cdot x_t + r_t \odot U \cdot h_{t-1}) \quad (I.10)$$

Où

$r_t \odot h_{t-1}$ étant le produit Hadamard entre les deux vecteurs r_t et h_{t-1} .

I.3.4 Mémoire finale au pas du temps courant

Afin de transmettre les informations aux autres unités de réseau, nous avons besoin de la porte de mise à jour pour déterminer quelles informations collectées du contenu actuel de la mémoire \hat{h}_t et de temps précédents h_{t-1} .

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \hat{h}_t \quad (\text{I.11})$$

I.4 Encodeur - décodeur

C'est un type de réseaux de neurones introduit en 2014 par Sutskever et *al.* [6], il est constitué de deux composants encodeurs et décodeurs, ces derniers peuvent être un réseau de neurones récurrent RNN ou l'un de ses variantes tels qu'un LSTM ou un GRU afin d'améliorer les performances du système. Le rôle de l'encodeur est de prendre la séquence de texte d'entrée et essaye de convertir toutes les informations nécessaires de la séquence dans un vecteur de contexte (*Encoder vector*), après avoir parcouru toute la séquence, l'encodeur transmet le vecteur du contexte au décodeur pour l'aider à faire des prédictions précises. Ce dernier lit le vecteur de contexte et produit la séquence de sortie.

L'un des principaux avantages de ce modèle est qu'il permet de prendre en entrée une séquence de mots de longueur variable et de générer en sortie une autre séquence de mots de longueur variable, Cela ouvre la porte à des applications très intéressantes comme la réponse aux questions, et la traduction du texte.

La figure suivante montre l'architecture de l'Encodeur-Décodeur.

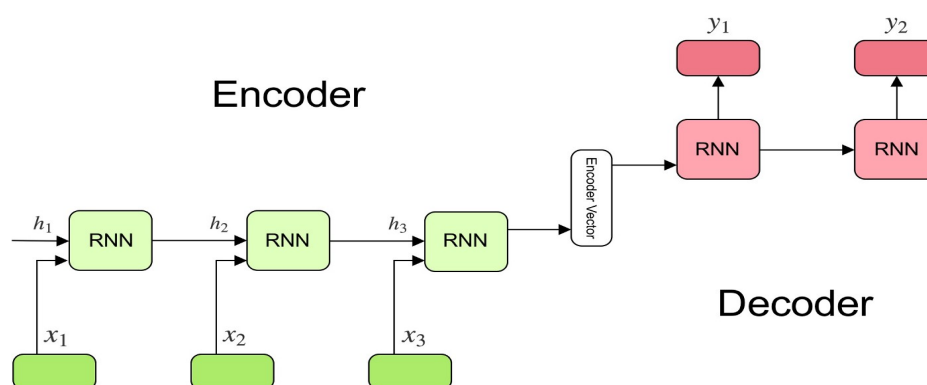


Figure I.8 : Architecture de Encodeur -Décodeur [7]

I.5 Mécanisme d'attention

L'inconvénient du modèle précédant Encodeur-décodeur est l'incapacité du système à se souvenir de séquences plus longues. Souvent, il oublie les premières parties de la séquence une fois qu'il a traité l'intégralité de la séquence. De plus, les performances se dégradent rapidement au fur et à mesure que la longueur de la séquence d'entrée augmente. Ce problème réside au niveau de l'encodeur qui prend toutes les informations de la séquence d'entrée dans un vecteur de longueur fixe, et le décodeur ne prend éventuellement pas en compte les premiers éléments de la séquence d'entrée, le mécanisme de l'attention a été introduit pour résoudre ce problème. L'attention est portée sur l'idée de libérer l'architecture Encodeur-Decodeur de la représentation à longueur fixe et améliorer les performances des séquences plus longues, Ceci est réalisé en conservant les sorties Encoder de chaque étape de la séquence entrée, et en entraînant le modèle à apprendre à prêter une attention sélective à ces entrées et à relier aux éléments de la séquence de sortie. En d'autres termes, chaque élément de la séquence de sortie dépend d'éléments sélectifs de la séquence d'entrée.

La figure I.9, présente la répartition de l'attention pour une traduction français-anglais. L'axe des x et l'axe des y correspondent respectivement aux mots de la phrase source (Anglais) et de la traduction générée (Français), les carrés clairs sont ceux pour lesquels l'attention est importante.

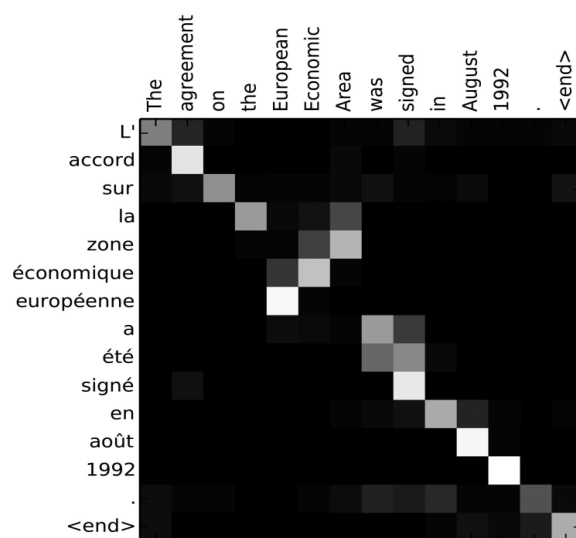


Figure I.9: Représentation graphique des zones d'attention [8].

Pour conclure, le mécanisme d'attention permet au modèle de se concentrer et d'accorder plus d'attention aux parties pertinentes de la séquence d'entrée et d'avoir meilleur résultat dans des applications où il est nécessaire de modéliser les dépendances à long terme.

I.6 Transformer

Le Transformer est un modèle de type séquence à séquence (Seq2seq), il a été introduit en 2017 par *Vaswani et al* [9] dans l'article : *Attention is All You Need*, contrairement au modèle précédent ou le traitement de l'encodeur et le décodeur était effectué avec un réseau récurrent. Le Transformer est basé uniquement sur mécanisme d'attention pour assurer l'interdépendance des mots. Il se compose d'encodeurs et décodeurs, chacun est constitué de N blocs identiques empilés les uns sur les autres.

L'encodeur est composé de deux composantes : Une couche auto-attention '*self-attention*' qui calcule la relation entre les différents mots de la séquence d'entrée, et un réseau de neurones à propagation avant (*Feed-Forward Neural Network*) appliqué à chaque vecteur d'attention pour le préparer à l'encodeur suivante. L'encodeur utilise une connexion résiduelle autour de ses deux couches et une couche de normalisation pour améliorer les performances et le temps de formation du modèle.

Pour le décodeur, il a la même architecture que l'encodeur avec une couche supplémentaire appelée '*Encoder-Decoder Attention*' placée entre la couche auto-attention et le réseau de neurones à propagation avant (*Feed-Forward Neural Network*).

L'entrée du premier encodeur est constituée de la somme de plongement de mots (Word-embedding) de la séquence d'entrée et le vecteur du codage positionnel. Ces vecteurs sont transmis à la couche auto-attention puis dans les réseaux de neurones à propagation avant, la sortie de chaque bloc d'encodeur est utilisée comme entrée dans l'encodeur suivant jusqu'à le dernier encodeur qui est utilisé par chaque couche "*Encoder-Decoder Attention*" de décodeur.

Le fonctionnement du décodeur est similaire à celui de l'encodeur, mais avec l'ajout de l'attention sur la représentation transmise par le dernier encodeur lui permet de se concentrer sur les informations pertinentes de la séquence d'entrée. La sortie du dernier décodeur est suivie par une couche linéaire qui convertit le vecteur de la sortie en un autre vecteur avec la taille du vocabulaire d'un modèle, ce vecteur transmis à la couche Softmax qui transforme les valeurs en probabilité, le mot avec la probabilité la plus élevée est ajouté à la séquence de sortie. L'architecture du transformer est illustrée dans la figure suivante

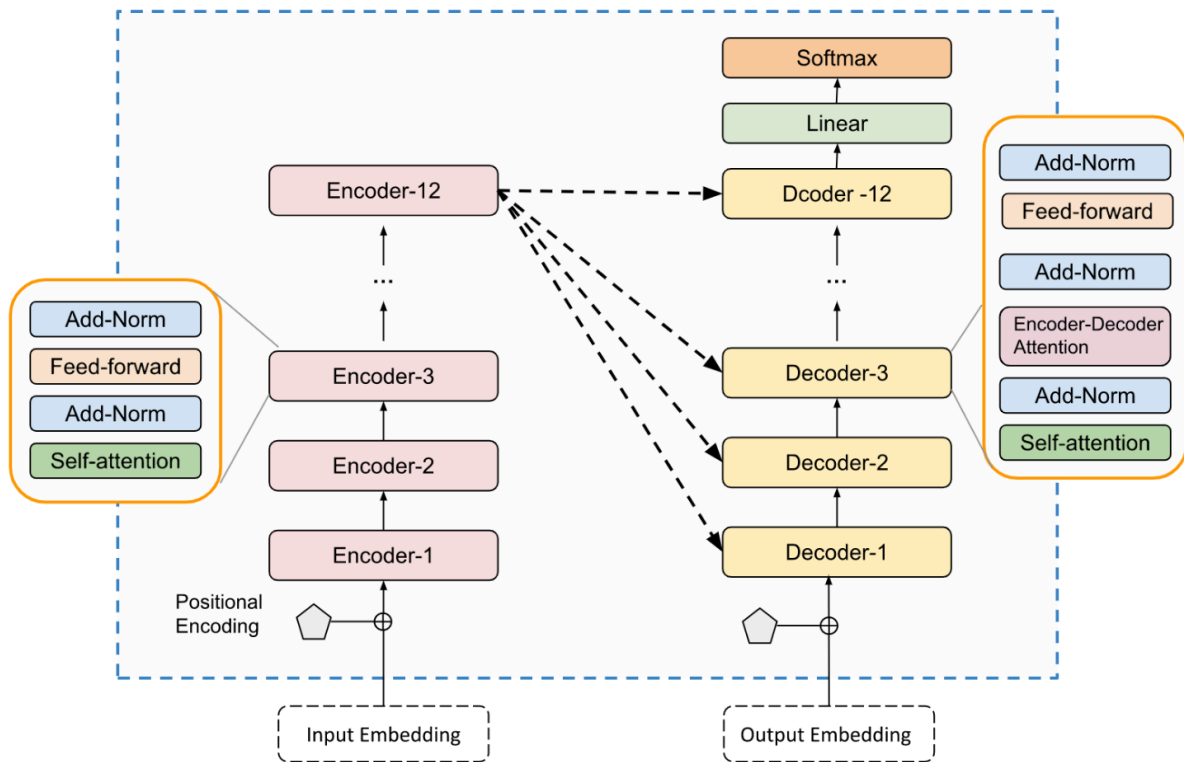


Figure I.10 : Architecture du Transformer

I.7 Modèles Pré-entraînés

L'émergence des modèles pré-entraînés a fait entrer le Traitement du langage naturel (NLP) dans une nouvelle ère, non seulement ils permettent aux amateurs d'effectuer des tâches simples de NLP, mais en plus aident également les experts à obtenir de meilleurs résultats sans créer un modèle à partir de zéro.

Les modèles pré-entraînés sont des modèles d'apprentissage en profondeur basés sur l'architecture de Transformer [9], Ils sont pré-entraînés sur une quantité massive de données non étiquetées afin d'apprendre les représentations universelles d'un langage, ensuite les modèles pourront être affiné sur des datasets étiquetés spécifique à la tache ciblé, ce qui conduit à des meilleures performances par rapport à un entraînement sur les données étiquettes seules. Il existe de nombreux modèles de langage pré-entraîné, ils diffèrent selon l'architecture (Encoder , Décoder ,les deux composant encodeur-décodeur), le dataset de prétraitement, l'objectif de l'entraînement, le nombre d'hyper paramètres, les plus utilisés sont BERT [12] , T5 [14] et GPT [58], le tableau suivant résume les différentes architectures des modèles pré-entraînés avec leurs tâches correspondantes.

Tableau I.1: Présentation Comparative des modèles pré-entraînés

Modèles	Architecture	Tâches
BERT, ALBERT, DistilBERT, ELECTRA, RoBERTa	Encodeur	Reconnaissance d'Entité Nommées (NER) Classification des phrases.
GPT, GPT-2	Décodeur	Génération de texte.
T5, BART	Encodeur-Décodeur	Toutes les tâches de traitement automatique de langage (NLP)

La figure ci-dessous, donne une présentation à la chronologie des versions populaires des modèles pré-entraînés :

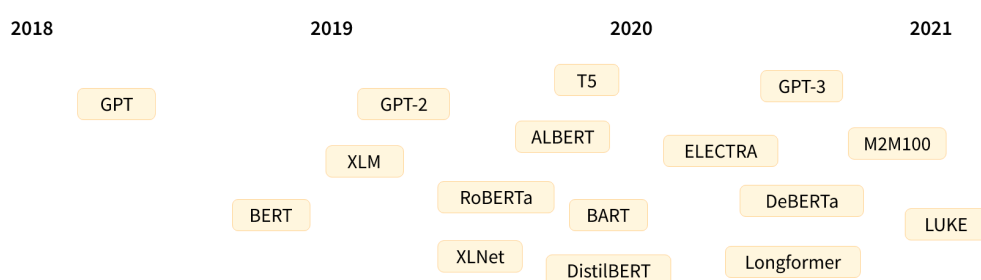


Figure I.11 : Chronologie des versions populaires des modèles pré-entraînés [11]

Grâce au pré-entraînement, le modèle permet d’avoir une meilleure initialisation, ce qui entraîne généralement à une meilleure performance et convergence plus rapide sur la tâche ciblée [10].

I.7.1 BERT (*Bidirectional Encoder Representation from Transformers*)

BERT (*Bidirectional Encoder Representation from Transformers*) est un modèle de langage pré-entraîné introduit en 2018 par devlin et al [12]. L’architecture du modèle BERT est une pile d’encodeurs de l’architecture de transformer, ce modèle de langage est bidirectionnel ce qui signifie qu’il apprend les informations de gauche à droite et de droite à gauche, cette

particularité lui permet d'avoir une meilleure compréhension de texte. Il existe deux versions du modèle BERT : une version de base qui contient 12 couches d'encodeurs, 12 têtes d'attention (*Multi Head Attention Mask*), 110 millions de paramètres et la version Large contient 24 couches d'encodeurs, 16 têtes d'attention et 340 millions de paramètres. L'architecture du modèle BERT est illustrée dans la figure suivante :

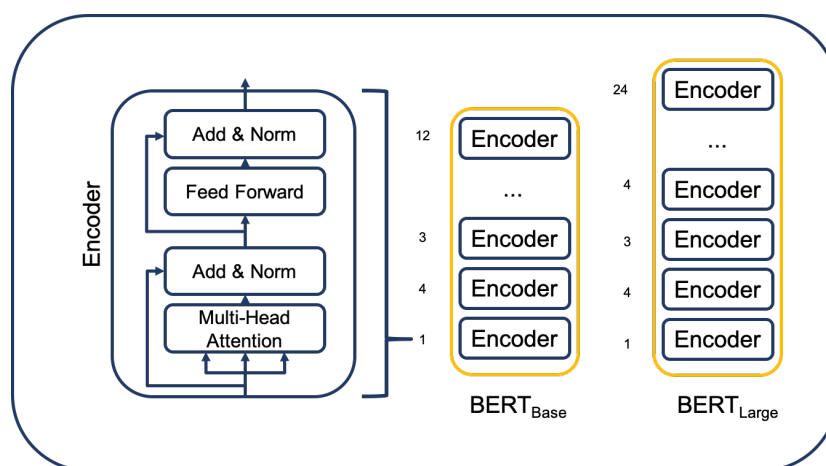


Figure I.12 : Architecture du modèle BERT [13]

Le modèle BERT est pré-entraîné sur deux tâches de NLP non supervisées sur un grand corpus: La première tâche est appelé la modélisation du langage masqué qui permet au modèle d'apprendre une représentation bidirectionnelle de la phrase, pour ce faire, le modèle masque au hasard 15% des mots, après il est entraîné pour prédire le mot manquant. La seconde tâche est la prédiction de la phrase suivante qui permet au modèle d'apprendre les relations entre les phrases, le modèle est entraîné avec deux phrases en entrée A et B, 50% des entrées la phrase B est la phrase suivante de A, tandis que dans 50% restants une phrase aléatoire de corpus est choisie comme deuxième phrase.

Pour aider le modèle à distinguer les phrases en entraînement, l'entrée de l'encodeur est traitée en trois étapes avant qu'elle soit transmis au modèle (voir la figure I.15) :

La première étape consiste à insérer un jeton [CLS] au début de phrase et un jeton [SEP] est inséré à la fin de chaque phrase, alors que la seconde étape ajoute des segments A pour la première phrase et B pour la deuxième, cela permet à l'encodeur de distinguer l'ordre des phrases, la dernière est d'indiquer la position de chaque jeton dans la séquence.

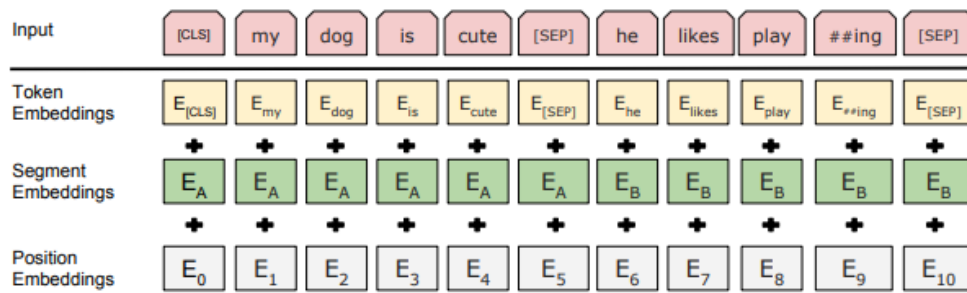


Figure I.13: Représentation des entrées du modèle BERT [12]

I.7.2 Text-To-Text Transform Learning (T5):

Le T5 est un modèle Encodeur-Décodeur basé sur le Transformer [9], il a été introduit en 2019 par Raffel et al dans l'article 'Exploring the limits of transfer Learning with a Unified Text to text transformer' [14]. Les auteurs ont proposé un nouveau corpus C4 'Colossal clean Crawled Corpus' une version nettoyée de Common Crawl pour pré-entraîné le modèle T5. Dans l'article, les auteurs ont étudié les différentes architectures et les méthodes introduites pour l'apprentissage par transfert dans NLP au cours des dernières années. Après les études, ils ont constaté que les modèles encodeur-décodeur surpassent généralement les autres modèles de langage (décodeur seulement).

La particularité du modèle T5 est qu'il convertit tous les problèmes NLP en format identique texte à texte, l'entrée et la sortie du modèle sont toujours des textes, cela permet au modèle d'effectuer n'importe quelles tâche NLP avec le même modèle, la même fonction du coût et les hyper paramètres en même temps.

Pour spécifier quelle tâche doit être effectuée par le modèle T5, il suffit d'ajouter un préfixe à l'entrée du modèle. La figure suivante illustre le concept multitâche du modèle T5

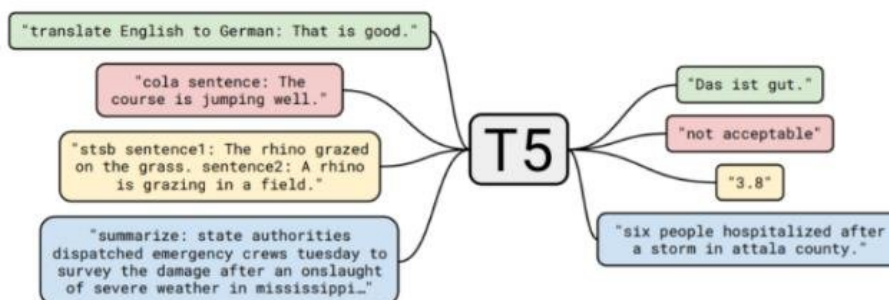


Figure I.14 : Concept multi-tâches du modèle T5 [14]

I.7.3 Modèle Ara-T5

Le Modèle Ara-T5 (*Text-To-Text Transformers for Arabic Language Generation*) est la version arabe du modèle T5, il a été introduit en 2021 par Nagoudi et al [52], L'architecture du modèle Ara-T5 repose sur l'architecture de Transformer [9], il est constitué d'une pile de 12 encodeurs et 12 décodeurs.

Nagoudi et al ont montré que le modèle Ara-T5 fournit des meilleures performances que le modèle mT5 (*Multilingual Pre-trained Text-to-Text Transformer*) dans la plupart des tâches de Traitement Automatique de Langage NLP y compris la génération de questions (QG).

I.7.3.1 Pré-entraînement de Ara-T5

Le Ara-T5 est pré entraîné sur une tâche globale consiste à prédire les mots manquants, ce pré-entraînement a été fait à l'aide de deux grands datasets riche en vocabulaire, le premier appelé *MSA Data* contient 70GO du texte MSA (Arabe Standard Moderne) 7.1B jetons (tokens) provenant de différentes sources :AraNews , El-Khair , Gigaword , OSCAR , OSIAN , Wikipédia Arabic et Hindawi Books , le deuxième est appelé Twitter Data comprend 178GO et 21.9B jetons (tokens) provenant des tweets en arabe choisis aléatoirement .

Tableau I.2 : Les datasets utilisées pour le prétraitement de Ara-T5

Source	Size	Jetons
AraNews	8.6GO	847.8M
Books	650MO	72.5M
El-khair	16GO	1.6B
Gigawords	10GO	1.1B
OSIAN	2.8GO	292.6M
OSCAR-MSA	31GO	3.4B
OSCAR-Egyptian	32MO	3.8M
WIKI	1.4GO	156.5M
MSA-Total	70GO	7.1B
Twitter (1.5B)	178GO	21.9B
Total	248GO	29.0B

Conclusion

Dans ce chapitre, nous avons abordé les modèles d'apprentissage en profondeur dédiés au traitement des données séquentielles, ces modèles sont devenus indispensables pour résoudre les tâches de traitement automatique de langage naturel (NLP). Dans le chapitre suivant nous intéresserons à la tâche de génération automatique de questions (AQG), son domaine d'application, et les différentes approches de génération de questions, ainsi que les travaux connexes à cette action.

CHAPITRE II :

ETAT DE L'ART ET TRAVAUX CONNEXES CONCERNANT LA GÉNÉRATION DE QUESTIONS

Introduction

La génération automatique de questions (AQG) à partir d'un passage de texte est cruciale dans de nombreux domaines, elle aide l'enseignant à économiser du temps et d'efforts pour créer du contenu pédagogique, réduire le coût du développement des outils éducatifs intelligents ainsi augmenter la performance des assistantes vocales. Malgré qu'elle a un impact sur vaste domaine, la création de systèmes de génération de questions est une tâche loin d'être facile. Dans ce chapitre, nous allons introduire le Traitement Automatique de Langage (NLP) et le concept du plongement de mots, nous présenterons ensuite la tâche de la génération automatique de questions (AQG) et ses différentes approches ainsi que son domaine d'application. Puis nous étudierons les travaux qui traitent la génération de questions basées sur l'apprentissage en profondeur et nous terminerons par une synthèse des travaux connexes.

II.1 Traitement automatique des langues (NLP)

Dernièrement, le NLP est exploité pour une large variété de cas d'usage tels que la traduction automatique, l'analyse des sentiments, le résumé du texte, la détection de Spam, l'agent virtuel (chatbot). Le Traitement automatique des langues (NLP) ou *Natural Language Processing* (NLP) en anglais est une branche de l'intelligence artificielle qui vise à donner aux ordinateurs la capacité de comprendre le texte et les mots parlés de la même manière que les êtres humains. Le NLP peut être divisé en deux grandes parties : la première est la compréhension du Langage Naturel ou *Natural Language Understanding (NLU)* en anglais, son rôle est de comprendre la structure et la signification du langage humain. La deuxième partie est la Génération du Langage Humain ou *Natural Language Generation (NLG) en anglais* son but est de créer et générer du texte à la manière d'être humain. Le NLP comprend plusieurs méthodes pour interpréter le langage humain, nous citons ci-après les méthodes couramment utilisées :

- **Méthodes basées sur des règles** : les plus anciennes de NLP, elles sont utilisées pour résoudre des problèmes assez simples, comme extraire de données structurées à partir de données non structurées par exemple : classer les mails indésirables dans les spams.

Néanmoins, ces méthodes simples peuvent être rapidement dépassées par la complexité du langage naturel.

- **Méthodes basées sur l'apprentissage automatique** : les approches de l'apprentissage automatique sont utilisées pour résoudre des problèmes plus difficiles, plus axées dans la compréhension du langage que la méthode basée sur les règles. Les algorithmes d'apprentissage automatique apprennent à effectuer des tâches grâce à des données d'entraînement, généralement ils utilisent des méthodes statistiques telles que la Classification Naïve Bayésienne, Régression Logistique.

- **Méthodes basées sur l'apprentissage en profondeur** : Ces méthodes utilisent des réseaux de neurones qui permettent d'extraire automatiquement des caractéristiques (*Features*). Grâce à leur puissance, les algorithmes du Deep Learning parviennent à effectuer des tâches complexes de NLP comme la traduction.

II.2 Plongements de Mots (*Word embedding*)

Les plongements de mots ou *Words Embeddings* en anglais sont des représentations vectorielles de longueur fixe pour les mots [15], Il s'agit de représenter chaque mot d'un vocabulaire étudié par un vecteur de nombres réels, décrits dans un modèle vectoriel (*Vector Space Model*), ces représentations vectorielles sont censées conserver les similitudes sémantiques et syntaxiques entre les mots.

Le word embedding repose sur la théorie linguistique introduite par *Zellig Harris* et connue sous le nom hypothèse distributionnelle (*Distributional Hypothesis*) [16], qui pourrait être résumée comme suit : “ Des mots qui ont un contexte similaire auront des significations similaires ”.

Il existe plusieurs méthodes pour obtenir ces représentations, les premières remontent aux années 1960 basées sur des méthodes de réduction de dimensionnalité, dernièrement des nouvelles techniques basées sur des modèles probabilistes et des réseaux de neurones ont permis d'obtenir de meilleures performances comme *Word2Vec* qui a été développé par une équipe de recherche de Google sous la direction de *Tomas Mikolov* [17].

Pour conclure, les plongements de mots ont permis d'améliorer les performances des méthodes de traitement du langage NLP.

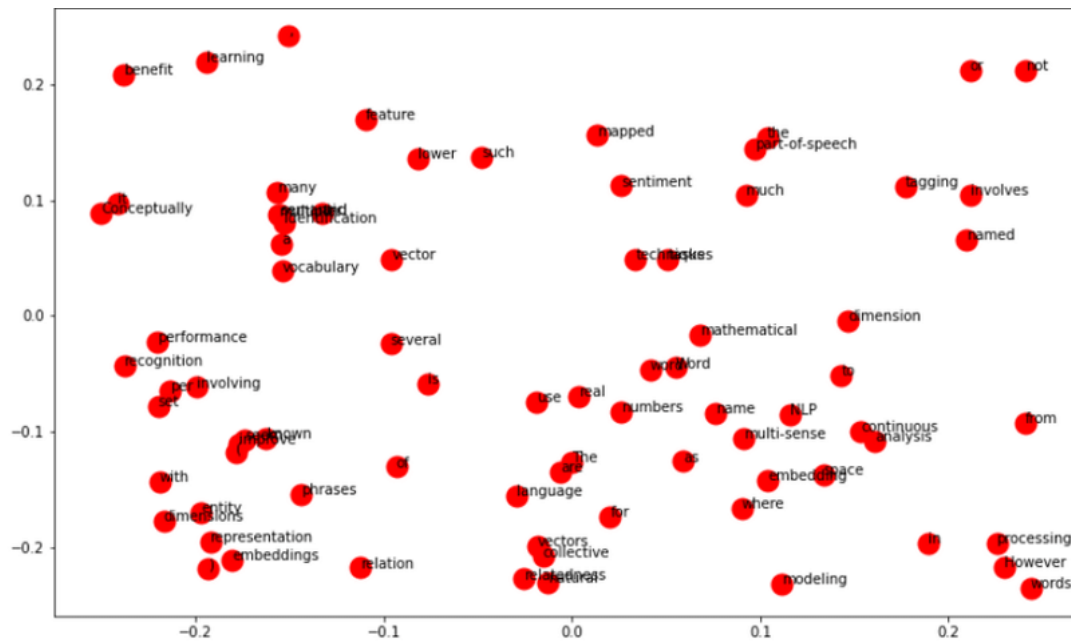


Figure II.1 : Représentation des mots dans un plan 2D [18]

II.3 La génération de questions

Selon la définition introduite en 2008 par *Rus et al* [59], et reprise par *Piwek et Boyer en 2012* [19]: “La génération de questions est la tâche qui consiste à générer automatiquement des questions à partir de différentes sources de données, dont notamment du texte, des bases de données, ou des bases de connaissances. La génération de questions est considérée comme une tâche linguistique basée sur les 4 étapes suivantes : (1) quand poser la question, (2) sur quoi porte la question, c’est-à-dire la sélection du contenu, (3) déterminer le type de question, et (4) construire la question ”. Cette définition indique que le type d'entrée pour la génération de questions peut varier : il peut être une phrase, un paragraphe ou une carte sémantique.

En ce qui concerne les types de questions qui peuvent être posées, elles peuvent être réparties dans les catégories suivantes :

- **Les questions à choix multiples (QCM)** : Une question à choix multiples est un type de question où l'on demande au répondant de choisir un ou plusieurs éléments parmi une liste limitée de choix.
- **Les textes à trous (*Gap-fill*)** : Ce type de question est constitué d’une phrase affirmative où un ou plusieurs mots sont supprimés, pour répondre à la question, les apprenants doivent déterminer les mots manquants dans un texte. (Exemple : “Ce matin,

je suis allé à la boulangerie pour _____ du pain. Sur le chemin du retour, la pluie s'est _____ à tomber”).

- **Les questions Oui / Non** : Une question oui ou non consiste en un énoncé qui requiert une réponse oui ou non (ou Vrai /Faux), elles sont conçues pour tester rapidement et efficacement les connaissances de l'apprenant sur une idée ou un concept particulier. Par exemple : “Le Machine Learning s’applique seulement sur des données structurées.” Vrai ou faux ?
- **Les questions factoides (factuelles)** : Les questions factoides sont des questions qui demandent des informations factuelles, où on peut répondre en une phrase maximum et dans la plupart des cas, ils commencent par : ‘Qui, ‘Où’, ‘Quand’. Par exemple : Qui a inventé l'Intelligence Artificielle ?

Kurdi et *al.* [20] ont conclu que les questions “textes à trous” et “factoides” sont les types de questions les plus générés, l'une des raisons est que les questions “textes à trous” ne demandent pas d'effort à les générer puisque la phrase est construite en supprimant un ou plusieurs mots, de plus, les questions "factoides" sont moins problématiques car il n'est pas nécessaire de créer l'ensemble des mauvais choix avec la bonne réponse comme dans les questions de choix multiples.

II.3.1 Domaine d’application

De nos jours, la génération des questions est une tâche répandue dans le domaine du traitement automatique de langage NLP, elle peut être utilisée dans de nombreux scénarios tels que les systèmes de tutorats automatique, l'amélioration des performances des modèles QA et la possibilité pour les chatbots de mener une conversation.

Dans cette section, nous décrivons plusieurs domaines d'application majeurs où la génération de questions (QG) a été adoptée.

a-L’éducation

La génération automatique de questions joue un rôle important dans le domaine de l'éducation, elle peut créer des quiz, des tests en ligne à partir de contenu éducatif sans effort manuel de la part des éducateurs.

En outre, la génération de questions est souvent utilisée comme un élément indispensable dans les systèmes de tuteurs intelligents pour évaluer les connaissances des apprenants et stimuler

l'auto-apprentissage.

Wang et al.[21] ont proposé un modèle de génération de questions basé sur un réseau de neurones récurrent spécialement conçu pour la génération des quiz à partir de contenu éducatif. *Heilman et Smith.*[22] ont développé une approche pour générer des questions afin d'évaluer l'acquisition par les élèves de connaissances à partir de documents de lecture.

b- Réponse aux questions (*Question Answering*)

Les modèles de réponse aux questions permettent de répondre automatiquement à des questions posées par les humains, ils sont entraînés sur des datasets contenant des paires de questions-réponses.

La construction de ces datasets est très coûteuse et prend beaucoup de temps. La génération de questions peut être utilisée comme stratégie d'augmentation des données afin de réduire le travail humain nécessaire à la création des paires question-réponse et améliorer l'entraînement des modèles réponses aux questions QA.

Duan et al. [23] ont proposé un modèle de génération de questions pour améliorer les systèmes de réponses aux questions, *Fan et al.* [24] a exploité la génération de questions (QG) pour générer un grand nombre de paires questions-réponses.

c- Les chatbots

Un chatbot est un logiciel utilisé pour imiter la conversation humaine par le biais des chats textuels et des commandes vocales [25]. Aujourd'hui, ils sont de plus en plus présents dans de nombreux domaines de notre vie quotidienne.

Dans la santé, les chatbots peuvent être utilisés pour déterminer la maladie d'une personne en posant des questions sur des symptômes, de côté des entreprises, ces dernières peuvent exploiter pour fournir un service client 24H/7J. Les chatbots pourraient améliorer leur crédibilité en posant des questions correctes liées aux sujets de conversation précédents.

Récemment, pour améliorer l'interactivité et la persistance des dialogues, une tâche de génération de questions conversationnelle (CQG) est proposée pour mener une conversation de type question-réponse, elle vise à générer une question à partir d'un texte d'entrée et de l'historique de la conversation, Pan et al. [26] ont proposé une nouvelle approche nommée *Reinforced Dynamic Reasoning Network* qui s'appuie sur l'encodeur décodeur pour générer des questions conversationnelles.

II.4 Les approches de génération de questions

Yao et Zhang [28] a divisé la génération de questions en trois catégories principales : Approches basées sur des modèles, basées sur la syntaxe et basées sur la sémantique. Ajoutons à cela, l'émergence des techniques d'apprentissage automatique notamment celles qui utilisent les réseaux de neurones, les articles les plus récents étudient “ Neural Question Generation NQG” une approche basée sur les réseaux de neurones.

Dans cette section, nous présenterons les approches de génération de questions qui pourraient être classées en deux catégories : Les approches basées sur les règles qui créent des questions grâce à des modèles construits manuellement, et les approches basées sur les réseaux de neurones qui génèrent des questions à partir d'un grand dataset.

II.4.1 Les approches basées sur les règles

Les approches basées sur les règles s'appuient généralement sur des règles de transformation conçues manuellement pour convertir des phrases déclaratives en interrogatives, dans la plupart des cas, les approches basées sur les règles ont été abordées dans le passé comme *Mitkov et Ha (2003)* [29].

Les travaux basés sur des règles existantes peuvent être classés en trois approches : basées sur les modèles, basées sur la syntaxe et basées sur la sémantique.

II.4.1.1 Les approches basées sur les modèles (Template-based approach)

Cette approche consiste à analyser la structure d'une phrase avec un analyseur de structure, puis à créer des questions à l'aide des modèles prédéfinis. Ces modèles sont essentiellement de taille fixe avec des espaces réservés qui sont ensuite remplis avec le terme ou l'ensemble de termes extraits du texte d'entrée.

Les approches basées sur des modèles conviennent principalement aux applications ayant un objectif particulier, qui relève parfois d'un domaine fermé. L'un de ces avantages est la possibilité de générer des questions bien formulées.

De nombreux chercheurs se sont concentrés sur la construction manuelle des modèles de questions, puis sur leur application à la génération de questions, *Ureel et al.* [30] ont décrit le système informatique appelé *Ruminator* qui prend le contexte résumé comme entrée et génère un grand nombre de questions faciles à l'aide des règles et des modèles simples : *Zheng et al* [31] ont proposé une méthode basée sur des modèles pour construire des questions à partir de

texte ou paragraphe chinois, les questions générées sont classées par un modèle neuronal pour obtenir les meilleures questions.

Voici un exemple illustratif sur les modèles de questions [32] proposés pour la tâche QG. (voir le tableau II.1)

Tableau II.1: les modèles de questions proposés pour la tâche QG[32]

Type	Question
Définition	Qu'est-ce que <X>?
	Qu'avez-vous à l'esprit lorsque vous pensez à <X>?
	Qu'est-ce que <X> vous rappelle?
Caractéristique/propriété	Quelles sont les propriétés de <X>?
	Quels sont les problèmes (opposés) de <X>?
	Quelles sont les fonctionnalités de <X>?
Exemple	Qu'est-ce qu'un exemple de <X> ?
Vérification	Y a-t-il un problème avec les arguments concernant <X> ?
Jugement	Qu'aimez-vous lorsque vous pensez ou entendez parler de <X> ?
Interprétation	Comment peut-on utiliser <X> aujourd'hui ?
Attente	Comment sera <X> dans le futur, basé sur la façon dont il est maintenant ?
Quantification	De combien de sous-thèmes vos partenaires ont-ils parlé ?
	Sur quels sous-thèmes vos partenaires se concentrent-ils ?

Comparaison de concepts	Quelle est la différence ou les relations entre ces sous-thèmes?
-------------------------	--

Par exemple, les modèles de questions de la classe définition peuvent être remplis par la phrase nominale « Énergie d'Activation », Cela amène aux questions suivantes :

- Qu'est-ce que l'Énergie d'Activation ?
- Qu'avez-vous à l'esprit lorsque vous pensez à l'Énergie d'Activation ?
- Qu'est-ce que l'Énergie d'Activation vous rappelle ?

II.4.1.2 Les approches basées sur la syntaxe

Les approches basées sur la syntaxe déterminent d'abord la structure syntaxique d'un texte donné, puis appliquent des règles de transformation syntaxique et des placements de mots d'interrogation ("Qui", "Quand", "Où"...) pour obtenir les questions [33].

Les approches basées sur la syntaxe sont plus prometteuses pour les phrases courtes, mais ne garantissent pas toujours la grammaticalité [28].

Wyse et Piwek [34] , Heilman et Smith [35] utilisent une approche très similaire en manipulant des arbres syntaxiques de l'analyseur syntaxique de Stanford (*Klein et Manning* [36]) pour la construction des questions.

Ali et al. [37] ont proposé un système qui automatise la génération de questions à partir d'une phrase à l'aide d'un analyseur syntaxique, L'étiquetage morpho-syntaxique (*Part-of-Speech POS*) et Reconnaissance d'entités nommées (*Named-Entity recognition NER*).

II.4.1.3 Les approches basées sur la sémantique

Les approches basées sur la sémantique effectuent une analyse sémantique des textes pour générer des questions. De nombreux travaux ont utilisé l'étiquetage sémantique des rôles (*Semantic Role Labeling*), les étiquettes de rôles sémantiques incluent le sujet, le verbe, le prédicat, l'objet, le temps, etc. (voir tableau II.2).

L'étiqueteur de rôles sémantiques identifie les arguments obligatoires comme A0 (sujet), A1(objet), et les arguments optionnels comme AM-TMP, AM-LOC et AM_MNR qui sont utiles pour déterminer le pronom interrogatif (Quand ? Où ?) de la question générée . Par exemple, l'étiquette AM-MNR qui indique la manière, invite le pronom interrogatif "Comment".

À titre d'illustration, considérons la phrase suivante [38] :

[_{A0} Peter] called [_{AM-TMP} on Monday].

La question qui pourrait être générée : *When did Peter call ?*

Le tableau suivant donne un aperçu des étiquettes de rôles sémantiques

Tableau II.2 : Les étiquettes de rôles sémantiques [38]

Label	Rôle
A0	Proto-agent (souvent sujet grammatical))
A1	Proto-patient (souvent objet grammatical)
A2	Instrument, attribut, bénéfique, montant, etc.
A3	Point de départ ou état
A4	Point final ou état
AM-LOC	Emplacement
AM-DIR	Direction
AM-TMP	Temps
AM-CAU	Cause
AM-PNC	Objectif
AM-MNR	Manière
AM-EXT	Le degré
AM-DIS	Discours
AM-ADV	Adverbial
AM-MOD	Verbe modal
AM-NEG	Négation

Schwartz et al. (2004) [39] ont introduit un générateur de questions de contenu basé sur un système appelé “ NLPwin ” utilisé par Microsoft. Il utilise la forme logique pour représenter les relations sémantiques des arguments dans une phrase et génère des questions de type WH-questions.

Yao et Zhang [28] ont proposé une nouvelle approche de génération de questions basée sur la sémantique en transformant la représentation de la sémantique de récursivité minimale (*Minimal Recursion Semantics*) des phrases déclaratives en celle des phrases interrogatives.

II.4.2 Les approches basées sur les réseaux de neurones

Les approches de génération de questions basées sur les réseaux de neurones ont montré de grands avantages par rapport aux approches traditionnelles. À la fois en termes de fluidité et de diversité des questions. Grâce à ces approches, la génération automatique de questions devient entièrement axée sur les données et nécessite peu d'efforts humains par rapport à celles qui utilisent des règles de transformations.

La plupart des approches neuronales considèrent la tâche de génération de questions comme un problème de séquence à séquence (Seq2Seq), et conçoivent différents types d'encodeurs et de décodeurs pour améliorer la qualité des questions générées.

Les modèles séquence à séquence convertissent le texte d'entrée et éventuellement la réponse ciblée en représentation intermédiaire via un encodeur, puis utilisent un décodeur pour générer des questions à partir de représentation intermédiaire, il existe deux différents types du modèle séquence à séquence (Seq2seq) pour la génération de questions.

II.4.2.1 Modèle Seq2Seq basée sur RNN

Avec le succès de la traduction automatique neuronale, les chercheurs ont commencé à appliquer l'architecture encodeur décodeur RNN avec le mécanisme d'attention sur la génération de questions.

Du et al.[27] ont introduit le premier modèle de génération de questions neuronales NQG en utilisant l'architecture encodeur décodeur basée sur RNN plus précisément LSTM bi-directionnel avec le mécanisme d'attention, le rôle de mécanisme d'attention est d'aider le décodeur à prêter attention aux parties les plus pertinentes du texte d'entrée lors la génération de questions.

Zhou et al [40] ont proposé un modèle de génération de questions en utilisant l'encodeur-décodeur basé sur GRU-bidirectionnel avec mécanisme d'attention, de plus, ils ont ajouté le mécanisme de copie [41] pour traiter le problème de mots rares et inconnus. Le modèle prend la position de réponse dans la phrase d'entrée et les caractéristiques lexicales ainsi que le passage comme entrée de l'encodeur.

II.4.2.2 Modèle Seq2Seq basée sur les Transformers

Récemment, l'architecture Transformer s'est avérée très efficace sur diverses tâches NLP, surmontant les problèmes causés par RNN tel que le coût de calcul et les problèmes de dépendance à long terme.

Mishra et al [42] ont fourni une étude comparative de plusieurs méthodes de génération automatique de questions AQG, les résultats d'évaluation automatique et humaine montrent que les méthodes basées sur les Transformers surpassent les autres méthodes AQG existantes sur les longues réponses.

Chan et Fan., [43] ont proposé un modèle QG au niveau de phrase et de paragraphe en utilisant un modèle pré-entraîné BERT qui utilise uniquement la partie d'encodeur de transformer pour la génération de questions.

II.5 Datasets les plus utilisés dans la génération de questions

Dans cette partie, nous passons en revue les datasets utilisés dans le domaine de génération de questions, nous remarquons que dans les travaux antérieurs, les chercheurs utilisent généralement les datasets de la tâche question à réponse (QA) et de compréhension en lecture automatique (MRC) pour la tâche de génération de questions vu le manque des datasets QG.

II.5.1 Stanford Question Answering Dataset (SQuAD)

SQuAD [44] est un grand dataset de compréhension en lecture automatique (MRC), qui contient des paires de questions-réponses construites par des humains à partir des articles de Wikipédia. Dont les réponses peuvent être n'importe quelle séquence de jetons dans le texte donné.

Il existe deux versions : SQuAD 1.1 (2016) contient plus de 100,000 paires de question - réponse sur plus de 500 articles de Wikipédia, et SQuAD 2.0(2018) combine les données de SQuAD 1.1 avec plus de 50,000 questions sans réponse.

II.5.2 MicroSoft MACHine Reading COMprehension (MS MARCO)

MS MARCO [45] est un dataset à grande échelle axée sur la compréhension en lecture automatique MRC, la réponse aux questions et le classement des passages, il comprend 1,010,916 questions (avec ou sans réponse) échantillonnées à partir des requêtes de recherche Bing.

II.5.3 News QA

News QA [46] est un dataset de compréhension en lecture automatique qui contient 119,633 paires de question-réponse générées par un humain sur un ensemble de plus de 10,000 articles de presse de CNN choisis aléatoirement.

II.5.4 Arabic-SquAD

La version arabe traduite à partir du dataset SQuAD [44], les auteurs ont traduit les 231 premiers articles du dataset de l'entraînement à l'aide de traduction automatique neuronal (NMT) de Google Translate. Cela a résulté 48 344 questions sur 10 364 paragraphes.

II.5.5 Arabic Reading Comprehension Dataset (ARCD)

ARCD a été créé en 2019 par Mozannar et al [47], ce dataset contient 1395 questions posées par les humains sur des articles de Wikipédia, ARCD a été utilisé dans plusieurs systèmes de questions-réponses et a montré de bonnes performances.

II.6 Métriques d'évaluation utilisées dans la génération de questions

Les métriques d'évaluation automatique sont largement utilisées dans les tâches de traitement automatique de langage NLP, elles permettent de fournir des scores pour analyser la performance du modèle et de le comparer avec d'autres modèles.

Vu qu'il existe peu de métriques spécialement conçues pour les systèmes de génération de questions, la majorité des chercheurs adoptant les métriques de similarité tels que: BLEU [48], METEOR [49] et ROUGE [50] comme métrique d'évaluation des modèles de génération de questions, ces métriques étaient initialement destinées pour évaluer les performances des systèmes de traduction automatique.

II.6.1 BLEU

BLEU (*BiLingual Evaluation Understudy*) [48] est l'une des métriques les plus utilisées pour les tâches séquence à séquence, elle compare les n-grammes entre les séquences générées et les séquences référence, les scores comparés par paires sont moyennés pour produire le score final évaluant la qualité globale du texte généré. BLEU-N est le score BLEU qui utilise jusqu'à n-grammes pour compter les co-occurrences.

II.6.2 METEOR

METEOR (*Metric for Evaluation of Translation with Explicit ORdering*) [49] calculé avec la moyenne harmonique de la Précision et du Rappel unigramme, où le rappel pèse plus que la précision, chaque correspondance d'unigramme n'est pas seulement basée sur la correspondance exacte, mais aussi sur la racine, les paraphrases et les synonymes. La métrique a été conçue pour résoudre certains des problèmes rencontrés dans le BLEU, et produit également une bonne corrélation avec le jugement humain.

II.6.3 ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [50] est un ensemble de métriques destinées à évaluer la synthèse automatique et la traduction automatique.

Il existe cinq variantes de la métrique ROUGE : ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S, ROUGE-SU. La variante la plus utilisée dans la tâche de génération de questions est Rouge-L, les articles récents utilisent cette dernière pour évaluer la tâche de QG qui traite les séquences de mots en recherchant la plus longue sous-séquence commune (*longest common subsequence LCS*) entre la séquence hypothèse et la référence ou LCS est une séquence qui apparaît dans le même ordre relatif mais pas nécessairement contiguë.

Le principal avantage de rouge-L par rapport à rouge-1 ou rouge-2 est qu'il ne dépend pas de correspondances consécutives de n-grammes et qu'il a donc tendance à capturer la structure de la phrase beaucoup plus précisément.

II.7 Travaux connexes

Il existe plusieurs travaux dans le domaine de génération de questions, nous intéressons dans cette section aux travaux basés sur les approches du *Deep Learning*.

II.7.1 Learning to Ask :Neural Question Generation for Reading Comprehension [27]

Du et al [27] ont été les premiers à proposer un modèle de génération de questions neuronal (*Neural Question Generation NQG*) en utilisant un modèle d'encodeur-décodeur avec mécanisme d'attention, qui reçoit en entrée une phrase à l'aide de l'encodeur et génère une question sur la phrase grâce au décodeur.

Comme il est mentionné il est avant, le rôle de mécanisme d'attention est d'aider le décodeur à prêter attention aux parties les plus pertinentes de la phrase d'entrée, Dans cet article deux variantes des modèles ont été étudiées : le première modèle encode uniquement les informations au niveau de la phrase , l'autre modèle encode à la fois au niveau de la phrase et du paragraphe.

Les modèles sont entraînés sur le dataset question-réponse SQuAD [44] traité et divisé en trois sous-ensembles : ensemble d'apprentissage / ensemble de validation / ensemble de test contenant respectivement 70484 / 10570 / 11877 paires .

Les résultats de l'évaluation automatique montrent que le modèle surpasse le meilleur modèle basé sur des règles selon les métriques BLEU [48], METEOR [49], ROUGE-L [50]. Et dans les évaluations humaines, les questions générées par le modèle sont jugées plus naturelles (grammaticalité, fluidité).

II.7.2 Neural Question Generation from Text

Zhou et al.[40] ont proposé d'appliquer le modèle Encodeur-Décodeur pour générer des questions significatives et diverses à partir d'un passage de texte contenant la réponse correspondante. L'encodeur est composé d'un GRU bidirectionnel, permettant à l'entrée d'être lue dans les deux sens, et le décodeur est composé d'une couche GRU avec un mécanisme d'attention.

L'entrée de l'encodeur a été enrichie de fonctionnalités telles que l'étiquetage morpho-syntaxique (Part-of-Speech (POS)) et Reconnaissance d'entités nommées (*Named Entity Recognition* NER).

Il utilise également, des informations sur la position de la réponse dans la séquence d'entrée, identifiée par l'étiquette BIO : B est le début de la réponse, I est pour les mots qui appartiennent à la réponse et enfin O pour tous autres mots qui n'en fait pas partie, les auteurs ont utilisé le dataset question-réponse SQuAD [44] découpé en trois sous-ensembles: ensemble d'entraînement, validation et de test, contiennent respectivement 86635/8965/8964 triplets phrase-réponse -question.

L'évaluation a été faite à l'aide d'évaluation humaine et la métrique BLEU-4 [48] où ils ont obtenu 13.29. La figure suivante présente un aperçu sur le modèle de génération de questions de Zhou et al, [40]

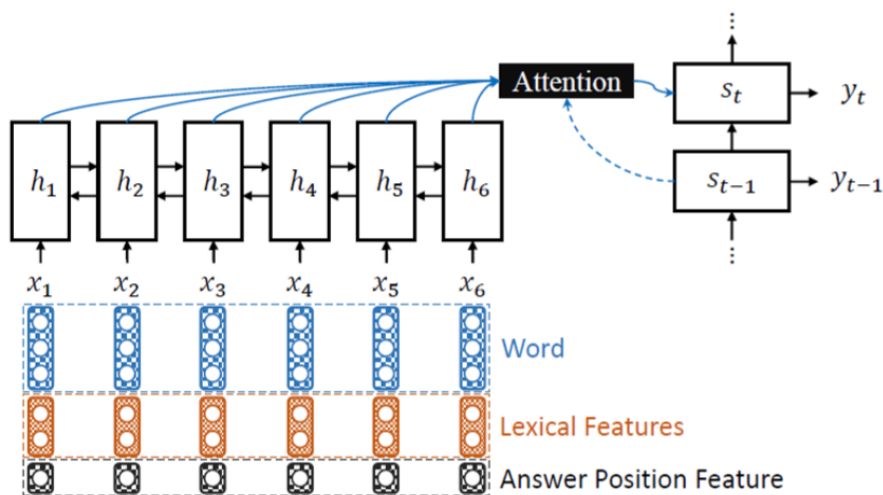


Figure II.2 : Représentation du modèle NQG de Zhou et al [40]

II.7.3 Paragraph-level Neural Question Generation with Maxout Pointer and Gated Self-attention Networks [51]

Un paragraphe contient un contexte beaucoup plus riche qu'une phrase, selon Du et al. [27] 20% des questions dans SQuAD [44] nécessitent des informations au niveau du paragraphe. Et que l'utilisation du paragraphe entier peut améliorer les performances du modèle de génération automatique des questions (AQG).

Cependant, plus le texte d'entrée s'allonge, plus il est difficile pour le modèle de capturer les informations pertinentes et d'éviter les informations non pertinentes. Pour surmonter les défis de la génération de questions au niveau du paragraphe. Zhao et al., [51] ont proposé un nouveau modèle de générations de question applicable à la fois aux entrées au niveau de la phrase et du paragraphe, il s'agit d'un modèle d'attention séquence à séquence avec un mécanisme de pointeur *Maxout* pour éviter les problèmes de répétitions des mots dans la séquence de sortie (Figure II.3).

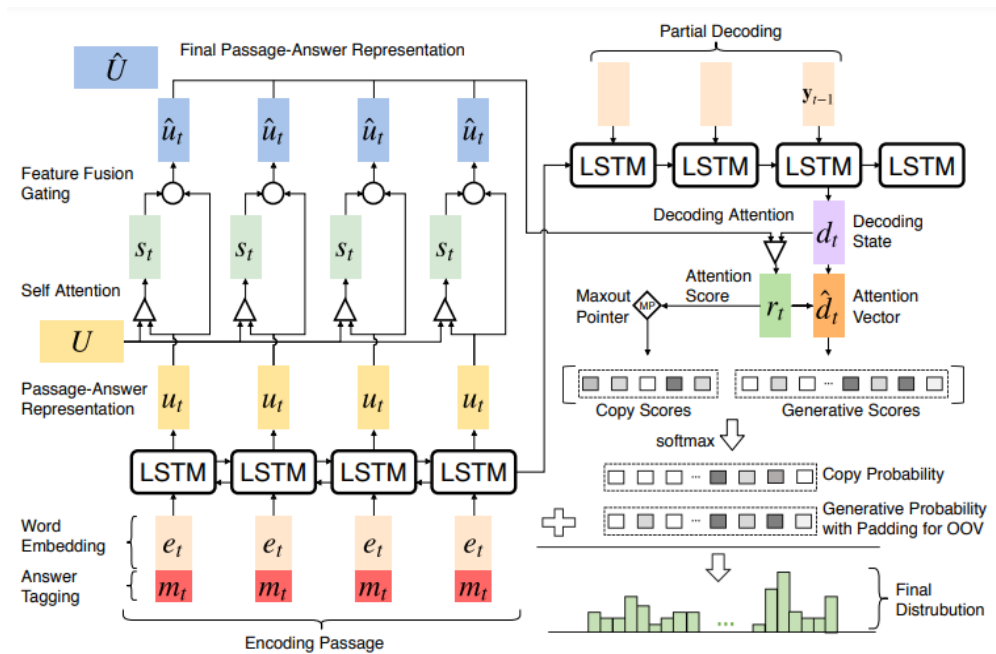


Figure II.3 Présentation du modèle Gated Self-Attention avec pointeur Maxout [51]

Le modèle a été entraîné sur le dataset SQuAD [44] et MS MARCO [45], pour pouvoir comparer directement avec les travaux Du et al. [27] et Zhou et al. [40] ils ont adopté deux types de découpage du dataset SQuAD : Split 1 similaire à Du et al. [27] , et Split 2 équivalent à Zhou et al. [40] voir le tableau II.3)

Tableau II.3 : Découpage usuel du dataset SQuAD

	Train	Test	Dev
SQuAD 73K (Du et al. [27])	73240	11877	10570
SQuAD 81K (Zhou et al. [51])	81577	8964	8964

L'évaluation du modèle a été effectuée grâce aux métriques BLEU [48], METEOR [49] , ROUGE-L [50]. Les résultats d'évaluation ont montré que le modèle surpasse les approches QG de référence [27], [40] au niveau de phrase et au niveau de paragraphe (voir Tableau II.4 et II.5).

II.7.4 A Recurrent BERT-based Model for Question Generation [43]

Ying-Hong Chan et Yao-Chung Fan ont proposé un modèle de génération de questions au niveau de phrase et de paragraphe en utilisant un modèle pré-entraîné BERT [13] (*Bidirectional Encoder Representations from Transformers*).

Dans cet article, les auteurs ont introduit trois architectures neuronal : le premier est un modèle BERT simple appelé BERT-QG qui offre des performances médiocres et dévoile les inconvénients d'implantation directe du modèle BERT sur la tâche de génération de questions. En conséquence le deuxième modèle proposé appelé BERT-SQG (*BERT-Sequential Question Generation*) est un modèle séquentiel qui prend des informations à partir des résultats décodés précédemment, le modèle BERT-SQG atteint un bon score faisant passer le résultat de point de 16.8 à 21.04 (BLEU 4).

Le dernier modèle est appelé BERT-HLSQG (*Highlight Sequential Question Generation*), son fonctionnement est de marquer la réponse de la séquence d'entrée avec des jetons (tokens) [HL] pour éviter l'ambiguïté dans la sélection des réponses pour la génération de questions, cette conception améliore encore le Score de 21.04 à 22,17 (BLEU4).

L'architecture du modèle BERT-HLSQG est illustrée dans la figure II.4

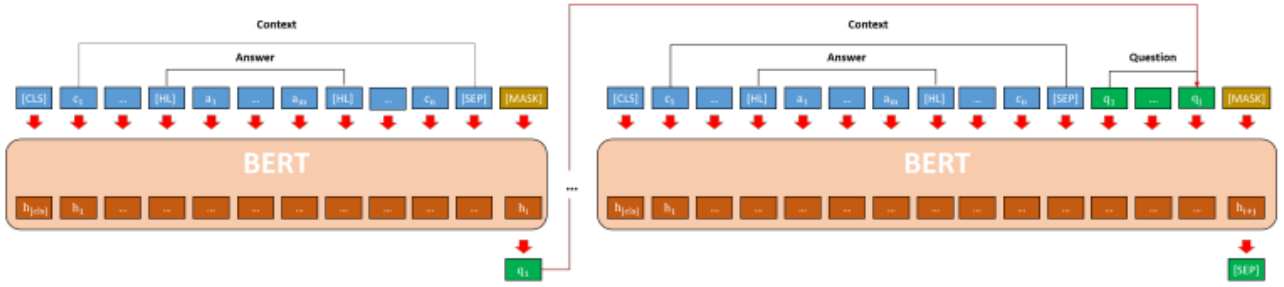


Figure II.4 : Architecture du modèle BERT-HLSQG [43]

Les modèles proposés sont entraînés sur le dataset SQuAD avec deux partitionnement SQuAD 73K et SQuAD 81K (voir le tableau II.3) pour pouvoir comparer avec les travaux précédents. L'évaluation du modèle a été effectuée grâce aux métriques BLEU [48], METEOR [49], ROUGE-L [50]. Les résultats d'évaluation montrent que le modèle BERT-SQG surpasse les modèles comparés [27] [40] et le modèle BERT-HLSQG atteint le meilleur score au niveau de phrase et paragraphe dans les deux partitions SQUAD 73K et SQUAD 81K sur les métriques usuelles de génération de questions. (Voir le tableau II.4 et II.5).

II.8 Synthèse des travaux

Dans cette partie, nous résumons les résultats d'évaluation automatique obtenus dans les travaux connexes cités ci-dessus.

Nous illustrons les performances des modèles dans le tableau II.4, II.5 selon les métriques BLEU 1-4, METEOR, ROUGE-L sur le dataset SQuAD suite au découpage de données dans Du et al [27] et Zhou et al [40]. Les indices (s) et (p) indiquent respectivement que le modèle QG prend le texte de contexte d'entrée au niveau de phrase et au niveau du paragraphe.

Tableau II.4 : Résultat de comparaison des modèles selon découpage de SQuAD dans Du et al [27]

Modèle	BLEU 1	BLEU 2	BLEU 3	BLEU 4	METEOR	ROUGE-L
[27] _s	43.09	25.96	17.50	12.28	16.62	39.75
[51] _s	43.47	28.23	20.40	15.32	19.29	43.91
[51] _p	45.07	29.58	21.60	16.38	20.25	44.48
[43] _s	48.29	33.12	24.78	19.14	22.89	47.07

$[43]_P$	49.73	34.60	26.13	20.33	23.88	48.23
----------	--------------	--------------	--------------	--------------	--------------	--------------

Tableau II.5 : Résultat de la comparaison des modèles selon découpage de SQuAD dans *Zhou et al* [27]

Modèle	BLEU 1	BLEU 2	BLEU 3	BLEU 4	METEOR	ROUGE-L
$[40]_P$	-	-	-	13.29	-	-
$[51]_S$	44.51	29.07	21.06	15.82	19.67	44.24
$[51]_P$	45.69	30.25	22.16	16.85	20.62	44.99
$[43]_S$	50.71	35.44	26.95	21.20	24.02	48.68
$[43]_P$	51.54	36.45	27.96	22.17	24.80	49.68

Conclusion

Dans ce chapitre, nous avons présenté l'état de l'art de la génération automatique de questions et leur avantage dans le traitement automatique de langage ensuite, nous nous sommes passés en revue sur les approches existantes de la génération de questions.

Les récents travaux ont montré que les approches basées sur l'apprentissage en profondeur dépassent les approches traditionnelles, malheureusement nous n'avons pas trouvé des travaux sur la génération de questions neuronal (NQG) en langue arabe.

Dans le chapitre suivant, nous présenterons notre approche basée sur l'apprentissage en profondeur plus précisément sur le modèle pré-entraîné pour la génération de questions en langue arabe.

CHAPITRE III :

GÉNÉRATEUR DE QUESTIONS EN LANGUE ARABE

Introduction :

Dans le chapitre précédent, nous avons présenté l'état de l'art sur la génération de question, nous détaillons dans ce chapitre basée sur l'apprentissage en profondeur pour la génération de questions en langue arabe, à notre connaissance il s'agit parmi les premiers travaux de génération de questions neuronal en langue arabe ce qui fait l'objet de notre travail.

Ce chapitre s'organise de la façon suivante : nous présentons d'abord la tâche de génération automatique de questions (AQG), puis nous exposons l'approche proposée pour construire le modèle de génération de question nommé ANQG, nous expliquerons ensuite la méthodologie que nous avons suivie pour construire modèle ANQG, après nous présentons les datasets utilisés pour l'entraînement du modèle et nous finirons par les détails d'entraînement du modèle.

III.1 Définition de la tâche de génération de questions

Pour décrire formellement la tâche de génération de questions QG, étant donné un Contexte $X = (x_1, \dots, x_n)$ et une réponse ciblée A (une étendue de texte dans le passage) en entrée, l'objectifs de la tâche QG vise à générer une question $Q = (q_1, \dots, q_n)$ posé sur une réponse ciblée A dans le passage X , autrement dit cette tâche est défini comme trouver la meilleure question prédit \bar{Q} qui maximise la vraisemblance conditionnelle compte tenu du passage X et de la réponse A :

$$\bar{Q} = \arg \max P(Q|X, A) \quad (\text{III.1})$$

III.2 Approche Proposée

La langue arabe est une langue morphologiquement riche avec relativement peu de ressources et une syntaxe moins explorée par rapport à l'anglais, en raison de ces limites la tâche de génération de questions en langue arabe s'est avérée difficile à aborder avec peu de données et les moyens informatiques limités.

Récemment, les modèles pré-entraînés sont conçus pour résoudre le problème de manque de données étiquetées, étant donné qu'ils sont entraînés sur une quantité massive de données non-étiquetées pour apprendre la représentation de langages, les modèles pré-entraînés pourront

être affinées sur une tâche spécifique de Traitement Automatique de Langage TAL et avoir souvent de meilleurs résultats qu'un modèle conçu à partir de zéro avec moins de données étiquetées.

En raison de leur capacité de transférer les connaissances apprises dans le prétraitement vers une tâche spécifique, et avoir obtenu les meilleurs résultats dans des nombreuses tâches, ce paradigme d'affiner un modèle pré-entraîné devient une pratique indispensable dans le domaine TAL, par conséquent, L'utilisation des modèles pré-entraînés dans la tâche de génération de questions en langue arabe peut être très prometteuse.

Dans ce travail, nous avons fait un affinement partiel du modèle Ara-T5 la version arabe du modèle T5 ce modèle peut naturellement utilisée pour la tâche de génération de questions (QG) sans avoir à modifier l'architecture du modèle contrairement aux autres modèles comme BERT.

III.2.1 Affinement du le modèle Ara-T5

L'affinement (*Fine-Tuning*) du modèle Ara-t5 consiste à ré-entraîné le modèle et mettre à jour les poids à l'aide des données spécifiques à la tâche de Traitement Automatique de Langage NLP.

Afin d'affiner un modèle il est important de définir les paramètres qui contrôlent le processus d'entraînement du modèle appelé souvent les hyperparamètres, qui se décline selon la description ci-après :

- ❖ **Taux d'apprentissage (*Learning Rate*)** : Le taux d'apprentissage est un hyperparamètre qui contrôle les poids de réseau de neurones par rapport au gradient de perte. Il définit la rapidité avec laquelle le réseau de neurones met à jour les concepts qu'il a appris.
- ❖ **Fonction d'optimisation** : L'optimisation est la technique pour mettre à jour les différents paramètres qui peuvent réduire la fonction de perte avec beaucoup moins d'efforts.
- ❖ **Taille de lot (*Batch-size*)** : La taille de lot est un hyperparamètre qui contrôle le nombre d'échantillons à traiter avant la mise à jour des paramètres internes du modèle.
- ❖ **Nombre d'epoch** : Le nombre d'epoch est un hyperparamètre qui définit le nombre de fois que l'algorithme d'apprentissage fonctionnera sur le dataset d'entraînement.

III.3 Méthodologie

Dans cette partie, nous présentons le processus suivi dans la conception de notre modèle de génération de questions en langue arabe (*Arabic Neural Question Generation (ANQG)*) composé de 3 étapes :

- Le chargement du dataset
- L'entraînement du modèle sur le dataset d'apprentissage (*Training-set*), Et la validation des résultats sur l'ensemble de validation (*Validation set*),
- Évaluer les performances du modèle sur l'ensemble de test (*Test-set*).

Chaque étape est revue plus en profondeur dans les sections qui suivent. La figure III.1 suivante illustre les processus dans la conception du modèle QG

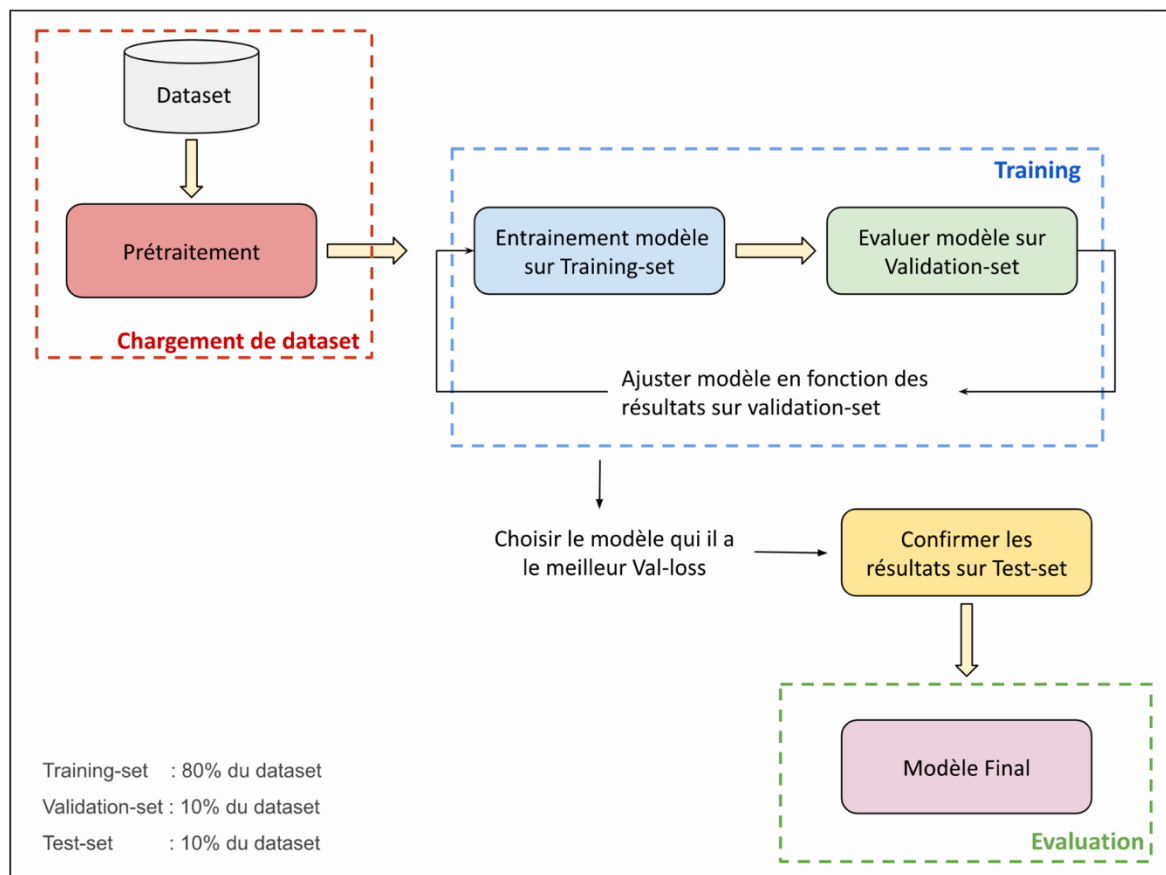


Figure III.1: Illustration schématique de la conception du système

III.4 Dataset

Étant donné qu'il n'existe pas de dataset dédiés pour la tâche de génération de questions (QG), les travaux antérieurs utilisent les datasets de la tâche opposée questions-réponses (*Question-answering*). Ces datasets sont conçus spécialement pour la tâche de QA, autrement dit les réponses de chaque question est une partie du contexte.

III.4.1 Description du dataset

Nous menons nos expériences sur quatre datasets en langue arabe accessibles au public : Tydiqa [53], MLQA [54], Arabic-SQuAD [47], ARCD [47], le format de ces quatre datasets correspond au célèbre dataset SQuAD 1.0 [44].

Chaque exemple du dataset comporte un contexte, et plus de 4 questions et réponses pour chaque contexte, les datasets sont structurés comme suit :

1. Contexte : Le paragraphe à partir duquel la question est posée
2. QAS : La liste comprend les éléments suivants :
 - (a) Id : Un identifiant unique pour la question
 - (b) Question : Une question.
 - (c) Answers : une liste qui comprend les éléments suivants :
 - (i) Answer : la réponse à la question
 - (ii) Answer - start : contenant l'index de départ de la réponse dans le contexte.

Après avoir extrait les paragraphes avec chaque question et réponse qui lui est associée, nous obtenons un tuple de forme < Contexte, Réponse, Question>. Le tableau suivant résume le nombre de tuples dans chaque dataset.

Tableau III.1 : Détails du dataset utilisé

Dataset	Train-set (tuple)	Test-set (tuple)	Total (tuple)
Arabic-SQuAD [47]	38.885	9459	48.344
ARCD [47]	695	700	1395
TyDiQA-GoldP (arabic) [52]	14724	921	15.645
MLQA (arabic) [53]	4120	1300	5800
Fusionner tous les datasets			71.184

Nous présentons deux exemples du tuple contexte-question-réponse dans le dataset ARCD d'article : جمال خاشقجي.

<p>Titre d'article: جمال خاشقجي</p> <p>Contexte: " صحفي (2018 أكتوبر 2 -، المدينة المنورة 1958 أكتوبر 13) جمال أحمد حمزة خاشقجي "، وإعلامي سعودي، رأس عدة مناصب لعدد من الصحف في السعودية، وتقلد منصب مستشار، كما أنه مدير عام قناة "العرب الإخبارية سابقًا".</p> <p>Question 1 : من هو جمال أحمد حمزة خاشقجي؟ La réponse de la question 1 : صحفي وإعلامي</p> <p>Question 2: في أي صحيفة قام بكتابة عمود منذ عام 2017؟ La reponse de la question 2 : واشنطن بوست</p>
--

Figure III.2 : Deux exemples d'un tuple contexte-question-réponse de ARCD.

III.4.2 Prétraitement du dataset

Le prétraitement des données est une première étape essentielle dans la construction d'un modèle du Deep Learning, cette étape peut faire référence à la suppression des données qui peuvent affecter négativement aux performances du modèle ou au nettoyage et la préparation de données avant qu'elles soient utilisées.

Afin de garantir la qualité de données utilisées dans notre modèle, nous avons suivi les étapes suivantes :

4.2.1 Nettoyage du dataset

Cette étape consiste à convertir les données textuelles en format cohérent afin que le modèle puisse comprendre la langue. Cette étape comporte les opérations suivantes :

1. Élimination de l'accentuation "التشكيل"

Avant	Après
الدَّوْلَةُ العُثْمَانِيَّةُ	الدولة العثمانية
العَرَبِيَّةُ	العربية

2. Élimination des parties étendues dans les mots

Avant	Après
محمد	محمد
العربية	العربية

3. Élimination les mots non-arabes

Avant	Après
الحوسبة المتنقلة (Mobile Computing) هي التفاعل بين الإنسان والحاسوب	الحوسبة المتنقلة هي التفاعل بين الإنسان والحاسوب
المطبخ الكوري (韩国料理؛ 한국 요리)، أو (韩食)؛ hansik (한식)	المطبخ الكوري

4. Elimination des questions non pertinentes

Après avoir inspecté le dataset, nous avons constaté que 20 455 questions n'étaient pas valides. Une question invalide est une question qui ne commence pas par le pronom interrogatif (هل؟، أي؟، ما؟، كم؟، من؟، متى؟، كيف؟، أين؟) étant donné qu'un mot interrogatif jouent un rôle important dans la caractérisation d'une question, nous avons supprimé les exemples qui contiennent ces questions.

Les questions non valides
تسمية بعض المكونات من الأجهزة
توجد الجزيرة في أعلى أي من الأرخبيلات؟
قبل إنشاء كلية الهندسة تم اجراء دراسات مماثلة في اي كلية نوتردام؟

4.2.2 Partitionnement de données (Data Split)

Après le nettoyage du dataset, le dataset final contient près de 31k tuple de forme (Contexte, Réponse, Question), nous avons consacré 80% du dataset pour l'entraînement, 10% pour l'ensemble de la validation, 10% pour l'ensemble du test.

Tableau III.2: Partitionnement du dataset

Total	Entraînement (80%)	Validation (10%)	Test (10%)
31472	25177	3147	3148

4.2.2 Tokenization

La tokenization est une étape très importante dans le traitement automatique de langage (NLP), elle permet de préparer les entrées pour le modèle. Cette étape consiste à diviser du texte en ensemble de fragments significatifs, ou chaque fragment est appelé jeton (*Tokens* en anglais), les jetons peuvent être des phrases, des mots ou des caractères. Ensuite chaque jeton est transformé en un identifiant unique correspondants (Input Ids) pour qu'il soit utilisé comme entrée pour le modèle.

Étant donné que notre ensemble de données n'est pas très différent en termes de vocabulaire de celui que le modèle (Ara-T5) a été entraîné, nous utilisons le tokenizer du modèle Ara-T5, ce dernier contient un vocabulaire de 110K jeton.

Concernant les jetons qui n'apparaissent pas dans le vocabulaire, le modèle utilise l'algorithme de WordPiece qui consiste à diviser un jeton en plusieurs sous-jetons qui peuvent être reconnus par le modèle, dans le cas contraire le modèle remplace le jeton inconnu par un jeton [UNK].

Il est à noter que le modèle Ara-t5 reçoit des phrases d'entrée d'une longueur fixe et qui ne doit pas dépasser longueur maximale 512 jetons, pour les phrases plus courtes que la longueur maximale, le modèle ajoute des jetons vide [PAD] aux phrases pour compenser la longueur. Le modèle ajoute à la fin de chaque phrase d'entrée le jeton `</s>`. Voici un exemple de tokenization d'une phrase :

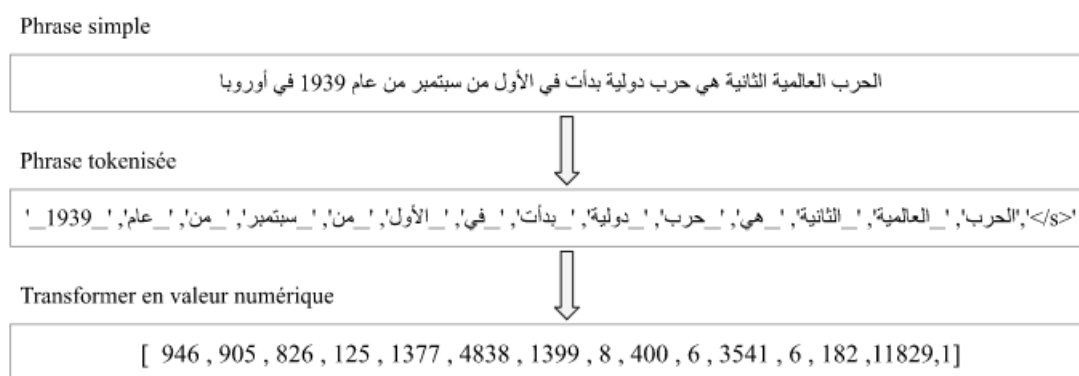


Figure III.3 : La tokenization d'une phrase à l'aide du Ara-t5 tokenizer

4.2.3 Préparation des entrées pour modèle Ara-T5

Dans ce travail, nous nous intéressons à la génération de questions sensibles à la réponse *Answer-Aware Question Generation* en anglais, autrement dit le modèle nécessite le contexte et la réponse pour générer une question. Pour utiliser le modèle Ara-T5 en génération de questions sensible à la réponse, l'entrée doit être comme suit :

Pour un paragraphe de contexte donné $C = [c_1, \dots, c_{|C|}]$ et une réponse La séquence d'entrée X est alignée comme : $X = (\text{contexte: } C, \text{ réponse : } A)$.

La figure suivante illustre les étapes suivies pour préparer les phrases d'entrée au modèle ANQG

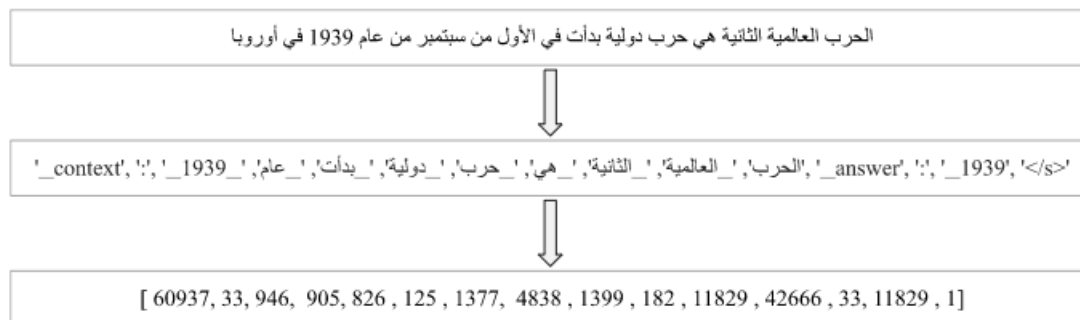


Figure III.4 : Préparation la phrase d'entrée pour le modèle ANQG

III.5 L'entraînement du Modèle ANQG

L'entraînement du modèle ANQG est précédé par le téléchargement du modèle pré-entraîné AraT5-Base à partir de la bibliothèque Transformers de *HuggingFace* par deux lignes de code à savoir :

```
from transformers import AutoModel
model = AutoModel.from_pretrained("UBC-NLP/AraT5-base")
```

Ensuite, on procède à l'affinement du modèle AraT5-Base sur dataset de questions-réponses (cf. Section III.4), la longueur de séquence d'entrée est limitée à 512 jetons, et 128 jetons pour la séquence de sortie (question générée). Pour choisir le taux d'apprentissage (*Learning-Rate*) optimal, nous avons entamer plusieurs entraînements avec différentes valeurs afin d'obtenir meilleur résultat correspondant au taux d'apprentissage optimum égale à $3e-4$. Toutes les expériences sont réalisées avec une taille de lots (*Batch-size*) égal à 8, en raison de la limite des ressources nous n'avons pas pu essayer d'autres valeurs supérieures.

Lors de l'entraînement du modèle, Nous utilisons la technique d'optimisation l'arrêt précoce pour réduire le sur apprentissage (*Overfitting*). Ce dernier, permet d'arrêter l'entraînement lorsque la valeur perte de validation commence à augmenter. L'implémentation de l'arrêt précoce est définie par les deux lignes suivantes :


```
from pytorch_lightning.callbacks.early_stopping import Early Stopping
trainer = trainer (callbacks = [ EarlyStopping (monitor = "val_loss", mode = "min ")])
```

Tableau III.3: Configuration des hyperparamètres pour l'entraînement du modèle ANQG.

Batch size	Taux d'apprentissage	Nombre d'epoch	Fonction d'optimisation
8	3e-4	5	Adamw

Conclusion

Dans ce chapitre, nous avons présenté notre approche basée sur l'architecture de Transformer plus précisément les modèles pré-entraînés ,ensuite nous avons expliqué la méthodologie suivie dans la conception de notre modèle, ainsi que les détails d'entraînement du modèle de génération de questions .

Dans le chapitre suivant, nous présentons l'environnement de développement et les ressources informatiques utilisées dans ce travail et les résultats obtenus.

CHAPITRE IV :

EXPÉRIMENTATION DU MODÈLE DE GÉNÉRATION DE QUESTIONS NEURONALES EN LANGUE ARABE (ANQG)

Introduction :

Le choix du bon matériel aura un impact significatif sur les performances du modèle.

Dans ce chapitre, nous présentons dans la première section l'environnement de développement et les ressources informatiques déployées dans ce travail, puis nous évaluons les performances du modèle ANQG et nous interprétons les résultats obtenus.

IV.1 Environnement de développement

IV.1.1 Google Colaboratory

En raison du volume de travail fourni et l'entraînement des modèles qui nécessite beaucoup de ressources matérielles en termes de RAM, GPU, nous avons recouru à l'utilisation de la version Pro de *Google Colaboratory* pendant toute cette phase de développement et de concrétisation de notre modèle.

Google Colaboratory est un environnement proposé par Google Research souvent raccourci en "Collab". Il permet aux développeurs d'implémenter des codes en Python par le biais d'un navigateur seulement. Cet environnement nous a donné la possibilité d'utiliser un accélérateur graphique. GPU : Tesla P100 de 16 GB, une RAM de 35.5GB de RAM.

IV.1.2 Pytorch Lightning

Pytorch Lightning (PL) [55] est un *Framework* créé par *W. Falcon*, Le PL structure le code Pytorch afin qu'il puisse résumer les détails de d'entraînement du modèle, PL permet à nos implémentations d'être plus lisible et exécutable sur différentes ressources GRU, TPU, CPU sans changer le code.

Pour développer un modèle neuronal dans le Pytorch Lightning, nous devons implémenter la classe Lightning Module qu'est une sous-classe de NN. Module de Pytorch, la classe Pytorch Lightning organise le code en 6 section : Initialisation, l'étape de l'entraînement (*Training step*), l'étape de validation (*Validation step*), l'étape de teste (*Test step*), l'étape de prédiction (*Predict step*), configuration de l'optimiseur (*Configure optimiser*).

De plus, nous avons utilisé deux autres classes de Pytorch Lightning : le *Model checkpoint* qui permet d'enregistrer meilleur modèle selon une métrique donnée, et la classe *Early Stopping* qui surveille la métrique et arrête l'entraînement lorsqu'elle n'améliore pas.

IV.1.3 Transformers

Transformers est une bibliothèque open source publiée par *HuggingFace* [56], elle fournit des API et des outils qui permettent de télécharger et d'utiliser les modèles pré-entraînés de Traitement Automatique de Langage (NLP).

La bibliothèque Transformers prend en charge l'interopérabilité des Frameworks entre PyTorch, TensorFlow et JAX, ce qui permet d'offrir une flexibilité dans l'utilisation de plusieurs *Framework* dans l'entraînement d'un modèle.

IV.2 Résultats et interprétations

Les méthodes d'évaluation des modèles de génération de questions sont généralement divisées en deux catégories, les métriques d'évaluation humaine et évaluation automatique. Dans cette section, nous examinons les résultats d'évaluations automatiques et humaines de notre modèle comparé avec les meilleurs modèles de génération de questions pour la langue anglaise en l'absence de travaux relatifs à la génération de questions neuronal en langue arabe.

IV.2.1 Evaluation Automatique

La plupart des travaux antérieurs sur la génération de questions utilisent les métriques d'évaluation automatique pour mesurer les performances des modèles, quoique ces métriques pourraient ne pas être adaptées pour évaluer la tâche de génération de questions, puisque la majorité des métriques comparent uniquement la similarité entre les questions références (source) et les questions générées, il se peut que la question générée soit complètement différente par rapport à la question référence, mais soit validée par l'humain. Nous rapportons les résultats obtenus par les métriques BLEU, ROUGE-L, METEOR dans le tableau IV.1

Tableau IV.1 : Résultat de comparaison entre notre modèle ANQG et les meilleurs modèles existants en anglais

Modèle	BLEU 1	BLEU 2	BLEU 3	BLEU 4	METEOR	ROUGE-L
[27]s	43.09	25.96	17.50	12.28	16.62	39.75
[51]s	43.47	28.23	20.40	15.32	19.29	43.91
[51]p	45.07	29.58	21.60	16.38	20.25	44.48
[43]s	48.29	33.12	24.78	19.14	22.89	47.07
[43]p	49.73	34.60	26.13	20.33	23.88	48.23
ANQG	37.62	27.80	20.89	15.87	33.19	43.37

Nous constatons que notre modèle ANQG obtient le meilleur score par la métrique METEOR, tandis que, sur les métriques BLEU et ROUGE-L, les modèles comparés ont des scores un peu élevés par rapport à notre modèle. L'une des raisons est que ces métriques recherchent la correspondance exacte des mots et il ne prend pas en compte la similarité sémantique comme la métrique METEOR, et vu la richesse de langue arabe face à l'anglais, BLEU et ROUGE peuvent ne pas donner des scores assez élevés. (Voir La figure IV.1)

Contexte	تم تفكيك الإمبراطورية النمساوية المجرية في عام 1918 بعد نهاية الحرب العالمية الأولى. وكان اباطرتها : الإمبراطور فرانس جوزيف الأول هابسبورغ لورين في الفترة من 1867 إلى (1916 والإمبراطورة إليزابيث) من 1867 إلى (1898، تبعتها الإمبراطور تشارلز الأول إمبراطور النمسا) من 1916 إلى (1918).
Réponse	1918
Question référence	متى انفصلت النمسا و المجر؟
Question générée par le modèle ANQG	متى تفككت الإمبراطورية النمساوية المجرية؟

Figure IV.1 : évaluation automatique d'un seul échantillon de l'ensemble du test.

Le score obtenu pour cet exemple :

BLEU 1	BLEU 2	BLEU 3	BLEU 4	ROUGE-L	METEOR
20	0	0	0	40.29	20

Nous constatons que les résultats d'évaluation sur la métrique BLEU sont souvent nul, malgré que la question générée par le modèle est loin d'être mauvaise, au contraire elle est correcte et valide.

Pour conclure, les métriques de l'évaluation automatique sont basées sur un seul critère de similarité, ces derniers ne peuvent pas quantifier les questions correctes, puisque les questions correctes ont tendance d'être significative et syntaxiquement, sémantiquement correctes. En conséquence, les travaux récents comptent sur l'évaluation manuelle pour évaluer les performances du modèle de génération de questions.

IV.2.2 Evaluation humaine qualitative

Afin de mieux évaluer les performances de notre modèle, nous avons effectué une évaluation humaine sur un échantillon de cent (100) questions générées du jeu de test, un expert en langue arabe a jugé les questions générées en tenant compte deux aspects :

-La correction syntaxique : Indique la grammaticalité et la fluidité. La grammaticalité sert à déterminer si la question respecte les règles grammaticales d'une langue. Par contre, la fluidité consiste à évaluer la simplicité et le naturel des termes dans les questions générées.

-La pertinence : Sert à déterminer si la question générée est significative et est reliée au passage en entrée ou non.

Pour chacun des deux aspects, un score est attribué, compris entre 1 et 5, (où 5 est le meilleur). Nous présenterons dans le tableau IV.2 la moyenne des scores obtenus pour notre modèle.

Tableau IV.2 : Résultat d'évaluation humain de notre modèle

Aspects	Correction Syntaxique	Pertinence
ANQG	4.5/5	3.5/ 5

Nous constatons que les annotations des experts arabes sont assez bonnes, la lisibilité des questions générées présente un taux de 90% avec une pertinence moyenne de 70%.

En conclusion, ces annotations manuelles des experts confirment la capacité du modèle de générer des questions correctes et pertinentes.

IV.3 Analyse des questions générée par ANQG

Nous présentons ci-après des exemples des questions générées par notre modèle ANQG à partir de contexte et réponse de l'ensemble de test et un autre ensemble de données extrait du web.

IV.3.1 Les questions générées d'ensemble de test

Nous présentons ci-après des exemples des questions générées par notre modèle à partir d'un contexte et une réponse, ces exemples ne sont qu'un petit échantillon des questions générées pour montrer la capacité du modèle proposé ANQG.

Contexte

تم تفكيك الإمبراطورية النمساوية المجرية في عام 1918 بعد نهاية الحرب العالمية الأولى. وكان اباطرتها: الإمبراطور فرانس جوزيف الأول هابسبورغ لورين) في الفترة من 1867 إلى (1916 والإمبراطورة إليزابيث) من 1867 إلى (1898، تبعها الإمبراطور تشارلز الأول إمبراطور النمسا) من 1916 إلى. (1918)

Réponse

1918

Question référence

متى انفصلت النمسا والمجر؟

Question générée

متى تفككت الإمبراطورية النمساوية المجرية؟

Contexte

اسكتلندا [1] دولة في شمال غرب أوروبا، تعتبر جزء من الدول الأربع المكونة المملكة المتحدة. تحتل الثلث الشمالي من جزيرة بريطانيا العظمى وتحدها جنوبا إنجلترا ويحدها شرقا بحر الشمال وغربا المحيط الأطلسي. عاصمتها أدينبرة، وأهم مدنها وأكبرها مدينة غلاسكو. كانت اسكتلندا مملكة مستقلة حتى 1 مايو 1707 حين تم إقرار قانون الوحدة لعام 1707 والذي اتحدت بموجبه مملكتي إنجلترا واسكتلندا في ما يعرف اليوم بمملكة بريطانيا العظمى.

Réponse

أدينبرة

Question référence

ما هي عاصمة اسكتلندا؟

Question générée

ما هي عاصمة اسكتلندا؟

Contexte

في عصر النهضة في أوروبا، من حوالي 1400 فصاعداً، كان هناك إحياء للتعليم الكلاسيكي مصحوباً بتطور حركة الإنسانية التي عادت إلى عصر النهضة والتي ركزت بشكل أكبر على دور الفرد في المجتمع مما كان عليه الحال خلال فترة القرون الوسطى. وأرجعت المباني لمعماري معينين برونيليشي، البرتي، مايكل أنجلو، بالاديو وبدأ عبادة الفرد. لم يكن هناك حتى الآن خط فاصل بين الفنان والمهندس المعماري والمهندس، أو أي من المهن ذات الصلة، وكانت التسمية في الغالب واحدة من

Réponse

أوروبا

Question référence

ما هي القارة التي حدثت فيها هذه النهضة؟

Question générée

أين كان عصر النهضة؟

Contexte

ترجع تسمية الإمارات العربية المتحدة إلى الإمارات السبع التي شكلت اتحاداً فيما بينها، وهي: إمارة أبوظبي وإمارة دبي وإمارة عجمان وإمارة الشارقة وإمارة رأس الخيمة وإمارة أم القيوين وإمارة الفجيرة.

Réponse

سبع

Question référence

كم عدد الإمارات في دولة الإمارات العربية المتحدة؟

Question générée

كم عدد الإمارات العربية المتحدة؟

Contexte

مات المستشار الألماني أدولف هتلر في 30 أبريل 1945 منتحراً عن طريق تناول مادة السيانيد السامة وإطلاق النار على نفسه وهي الرواية العامة المقبولة لطريقة موت الزعيم النازي. ولكن هذه الطريقة المزدوجة في الانتحار والظروف الأخرى التي أحاطت بالحادث شجعت البعض على إطلاق الشائعات بأن هتلر لم ينتحر وأنه عاش حتى نهاية الحرب العالمية الثانية مع الاختلاف حول ما حدث لجثته. وقد أكدت الوثائق السوفيتية المفرج عنها من جهاز بي وجهاز الأمن الفيدرالي الروسي

Réponse

إطلاق النار على نفسه

Question référence

كيف توفي أدولف هتلر؟

Question générée

كيف توفي أدولف هتلر؟

Figure IV.2 : échantillon des questions générées par le modèle ANQG

IV.3.2 Les questions générées à partir des exemples extraits du web :

Nous présentons ci-après des exemples des questions générées par notre modèle ANQG à partir des exemples différents de l'ensemble de test.

Tableau IV.3: échantillon des questions générées par ANQG extraites du web

L'entrée du modèle	La sortie du modèle
<p>Contexte :</p> <p>شارك منتخب الجزائر لكرة القدم أربع مرات في نهائيات كأس العالم لكرة القدم دورات 1982 و1986 و2010 و2014. لكنه لم يتجاوز دور المجموعات إلا في نهائيات كأس العالم</p> <p>Réponse :</p> <p>أربع مرات</p>	<p>Question générée :</p> <p>كم مرة شاركت الجزائر في كأس العالم لكرة القدم؟</p>
<p>Contexte :</p> <p>يعتبر يوم 16 أبريل من كل عام، يوم العلم بالجزائر وهو يوم وفاة العالم الجزائري عبد الحميد ابن باديس؛ ولد الامام عبد الحميد بن باديس سنة 1889 بقسنطينة بالجزائر من أسرة معروفة بالعلم والجاه تعلم القرآن وحفظه في سن 13 سنة من عمره على يد الشيخ حمدان الونيسي.</p> <p>Réponse :</p> <p>16 أبريل</p>	<p>Question générée :</p> <p>متى يوم العلم في الجزائر؟</p>
<p>Contexte:</p> <p>تُفرز العنبيات البنكرياسية الإنزيمات الهضمية في حين تفرز جزر لانغرهانس الهرمونات. يطرح البنكرياس الإنزيمات الهضمية في الاثناعشري، في حين تُطرح الهرمونات في المجرى الدموي مباشرة.</p> <p>Réponse :</p> <p>المجرى الدموي</p>	<p>Question générée :</p> <p>أين يتم طرح الهرمونات؟</p>
<p>Contexte :</p> <p>التعلم العميق هو مجال بحث جديد يتناول إيجاد نظريات وخوارزميات تتيح للآلة أن تتعلم بنفسها عن طريق محاكاة الخلايا العصبية في جسم الإنسان. وأحد فروع العلوم التي تتناول علوم الذكاء الاصطناعي</p> <p>Réponse :</p> <p>هو مجال بحث جديد يتناول إيجاد نظريات وخوارزميات تتيح للآلة أن تتعلم بنفسها</p>	<p>Question générée :</p> <p>ما هو التعلم العميق؟</p>

<p>Contexte :</p> <p>جافا سكريبت هي لغة برمجة عالية المستوى تستخدم أساساً في متصفحات الويب لإنشاء صفحات أكثر تفاعلية. يتم تطويرها حالياً من طرف لجنة TC39 المدارة بواسطة منظمة EACMA للمعايير</p> <p>Réponse :</p> <p>لغة برمجة عالية المستوى تستخدم أساساً في متصفحات الويب</p>	<p>Question générée :</p> <p>ما هي جافا سكريبت؟</p>
<p>Contexte :</p> <p>اندلعت ثورة التحرير الجزائرية في 1 نوفمبر 1954 ضد الاستعمار الفرنسي الذي احتلّ البلاد منذ سنة 1830 (احتلال دام 132 سنة)، ودامت ثورة التحرير الجزائرية طيلة سبعة سنوات ونصف من الكفاح المسلح والعمل السياسي</p> <p>Réponse :</p> <p>سبعة سنوات ونصف الويب</p>	<p>Question générée :</p> <p>كم استمرت ثورة التحرير الجزائرية؟</p>
<p>Contexte :</p> <p>فيروس كورونا هي عائلة من الفيروسات التي يمكنها أن تسبب أمراضاً مثل الزكام والالتهاب التنفسي الحاد الوخيم (السارس) ومتلازمة الشرق الأوسط التنفسية (ميرس). في عام 2019، اكتُشِف نوع جديد من فيروسات كورونا تسبب في تفشي مرض كان منشأه في الصين.</p> <p>Réponse :</p> <p>عام 2019</p>	<p>Question générée :</p> <p>متى تم إكتشاف فيروس كورونا؟</p>

IV.4 Interface du développement

Pour rendre notre travail apercevable, nous avons développé une interface graphique à l'aide d'une bibliothèque Python open source appelée de Gradio [57], pour fournir un environnement générateur de questions, cet outil contient deux rubriques pour les entrées du modèle (Contexte, Réponse), et une rubrique pour la question générée par notre modèle ANQG.

Dans ce qui suit, nous illustrons une capture d'écran de l'interface graphique.

Arabic Question Generation (ANQG)

Get the Question from given Context and a Answer

Context	Question
<input type="text"/>	<input type="text"/>
Answer	Flag
<input type="text"/>	
Clear	Submit

Figure IV.3: Interface graphique du modèle ANQG

Conclusion :

Dans ce chapitre nous avons évalué les performances de notre modèle ANQG par les métriques d'évaluations manuelle et automatique, Les résultats expérimentaux évalués sur des métriques automatiques démontrent que le modèle proposé à savoir ANQG apporte les scores performants, l'évaluation humaine a confirmé les résultats obtenus et elle a jugé que le modèle peut générer des questions de qualité prometteuse. Cette évaluation nous permet de valider notre approche basée sur les modèles pré-entraînés avec moins de ressources informatiques et datasets.

CONCLUSION GÉNÉRALE

Ce travail consiste à élaborer un modèle de génération de questions neuronal à partir d'un paragraphe en langue arabe, dont l'idée majeure visée par cette étude est de contribuer à la réduction des coûts des plateformes ou un tutorat intelligent ayant un impact positif dans le développement du domaine de l'enseignement numérique.

Souvent, les travaux réalisés dans la génération de questions en langue arabe ont abordé des approches traditionnelles qui s'appuient généralement sur des règles conçues manuellement. Ce qui entraîne un manque important de diversité dans les questions générées.

Par ailleurs, de travaux récents présentés pour la génération de questions d'autres langues présentent des approches intéressantes basées sur le Deep Learning qui sont entièrement axées sur les données et qui nécessitent peu d'efforts avec de meilleurs résultats comparativement aux approches traditionnelles.

Ces travaux nous ont motivés à présenter une approche de réseaux de neurones entièrement basée sur les données pour la génération automatique de questions en langue arabe. Pour ce faire, nous avons choisi d'utiliser des modèles pré-entraînés qui conduisent généralement à des meilleures performances.

Le travail a été précédé par une présentation de l'état de l'art des modèles d'apprentissage en profondeur utilisé dans le traitement automatique de langage NLP, nous avons vu leur fonctionnement et leur architecture.

Ensuite nous avons effectué une synthèse sur les approches appliquées aux modèles de génération de questions. A l'issue de l'étude de la tâche QG, il a été conclu que les travaux les plus prometteurs ont intégré l'architecture de Transformer.

Nous avons proposé par la suite notre approche qui consiste à utiliser un modèle pré-entraîné basé sur l'architecture de transformer pour générer les questions à partir d'un paragraphe en langue arabe

Nous avons également présenté les évaluations qui ont été mises en place pour valider notre approche, les résultats d'évaluation automatique sont incohérents pourraient être liée à la manière dont les métriques ont été calculées la similarité entre la question référence et la

question générée par notre modèle, en ce qui concerne l'évaluation humaine ont montré que les questions générées par notre modèle sont fluides et grammaticalement correctes.

Malgré les résultats prometteurs, quelques limites ont été soulevées à savoir :

- Notre modèle ANQG a été entraîné sur un dataset conçu pour la tâche de réponses aux questions (QA) ceci a un impact significatif sur les performances du modèles.
- Vu la limitation de ressources, nous n'avons pas pu utiliser un batch size supérieur à 8, cet hyper-paramètre influence les résultats du modèle.

Afin d'améliorer les performances du modèle ANQG nous proposons un ensemble de pistes permettant son développement à savoir :

- Affiner le modèle ANQG sur des données plus pédagogiques pour pouvoir générer des questions de type d'examen.
- Ajouter la tâche d'extraction des réponses aux modèle ANQG pour générer des questions à partir d'un contexte seulement (*End-to-End Question Generation*)

REFERENCES BIBLIOGRAPHIQUES

- [1] « CS 230 - Recurrent Neural Networks Cheatsheet ». <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks> (consulté le 3 juillet 2022).
- [2] P. Geeks, « Recurrent Neural Network - RNN in Machine Learning », *Python Geeks*, 1 avril 2022. <https://pythongeeks.org/recurrent-neural-network/> (consulté le 3 juillet 2022).
- [3] S. Hochreiter et J. Schmidhuber, « Long Short-Term Memory », *Neural Comput.*, vol. 9, n° 8, p. 1735-1780, nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [4] « Understanding LSTM Networks -- colah's blog ». <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (consulté le 3 juillet 2022).
- [5] K. Cho, B. van Merriënboer, D. Bahdanau, et Y. Bengio, « On the Properties of Neural Machine Translation: Encoder-Decoder Approaches ». arXiv, 7 octobre 2014. doi: 10.48550/arXiv.1409.1259
- [6] I. Sutskever, O. Vinyals, et Q. V. Le, « Sequence to Sequence Learning with Neural Networks », sept. 2014, doi: 10.48550/arXiv.1409.3215.
- [7] « Understanding Encoder-Decoder Sequence to Sequence Model | by Simeon Kostadinov | Towards Data Science ». <https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346> (consulté le 3 juillet 2022).
- [8] D. Bahdanau, K. Cho, et Y. Bengio, « Neural Machine Translation by Jointly Learning to Align and Translate ». arXiv, 19 mai 2016. Consulté le: 3 juillet 2022. [En ligne]. Disponible sur: <http://arxiv.org/abs/1409.0473>
- [9] A. Vaswani *et al.*, « Attention Is All You Need ». arXiv, 5 décembre 2017. doi: 10.48550/arXiv.1706.03762.
- [10] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, et X. Huang, « Pre-trained Models for Natural Language Processing: A Survey », *Sci. China Technol. Sci.*, vol. 63, n° 10, p. 1872-1897, oct. 2020, doi: 10.1007/s11431-020-1647-3.
- [11] « How do Transformers work? - Hugging Face Course ». <https://huggingface.co/course/chapter1/4> (consulté le 3 juillet 2022).
- [12] J. Devlin, M.-W. Chang, K. Lee, et K. Toutanova, « BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding ». arXiv, 24 mai 2019. Consulté le: 3 juillet 2022.
- [13] « Fine-tune BERT Model for Sentiment Analysis in Google Colab », *Analytics Vidhya*, 28 décembre 2021. <https://www.analyticsvidhya.com/blog/2021/12/fine-tune-bert-model-for->

- sentiment-analysis-in-google-colab/ (consulté le 3 juillet 2022).
- [14] C. Raffel *et al.*, « Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer ». arXiv, 28 juillet 2020. Consulté le: 4 juillet 2022.
- [15] F. Almeida et G. Xexeo, « Word Embeddings: A Survey », p. 10.
- [16] Z. S. Harris, « Distributional Structure », *WORD*, vol. 10, n° 2-3, p. 146-162, août 1954, doi: 10.1080/00437956.1954.11659520.
- [17] T. Mikolov, K. Chen, G. Corrado, et J. Dean, « Efficient Estimation of Word Representations in Vector Space ». arXiv, 6 septembre 2013. Consulté le: 26 juillet 2022.
- [18] S. Chatterjee, K. Chakrabarti, A. Garain, F. Schwenker, et R. Sarkar, « JUMRv1: A Sentiment Analysis Dataset for Movie Recommendation », *Appl. Sci.*, vol. 11, p. 9381, oct. 2021, doi: 10.3390/app11209381.
- [19] P. Piwek et K. E. Boyer, « Varieties of Question Generation: Introduction to this Special Issue: Introduction to this Special Issue », *Dialogue Discourse*, vol. 3, n° 2, Art. n° 2, mars 2012, doi: 10.5087/dad.2012.201.
- [20] G. Kurdi, J. Leo, B. Parsia, U. Sattler, et S. Al-Emari, « A Systematic Review of Automatic Question Generation for Educational Purposes », *Int. J. Artif. Intell. Educ.*, vol. 30, n° 1, p. 121-204, mars 2020, doi: 10.1007/s40593-019-00186-y.
- [21] Z. Wang, A. S. Lan, W. Nie, A. E. Waters, P. J. Grimaldi, et R. G. Baraniuk, « QG-net: a data-driven question generation model for educational content », in *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, London United Kingdom, juin 2018, p. 1-10. doi: 10.1145/3231644.3231654.
- [22] M. Heilman et N. A. Smith, « Good Question! Statistical Ranking for Question Generation », in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, California, juin 2010, p. 609-617. Consulté le: 28 juillet 2022. [En ligne]. Disponible sur: <https://aclanthology.org/N10-1086>
- [23] N. Duan, D. Tang, P. Chen, et M. Zhou, « Question Generation for Question Answering », in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, sept. 2017, p. 866-874. doi: 10.18653/v1/D17-1090.
- [24] Y. Fang, S. Wang, Z. Gan, S. Sun, J. Liu, et C. Zhu, « Accelerating Real-Time Question Answering via Question Generation », sept. 2020, doi: 10.48550/arXiv.2009.05167.
- [25] A. S. Sreelakshmi, S. B. Abhinaya, A. Nair, et S. Jaya Nirmala, « A Question Answering and Quiz Generation Chatbot for Education », in *2019 Grace Hopper Celebration India*

- (GHCI), Bangalore, India, nov. 2019, p. 1-6. doi: 10.1109/GHCI47972.2019.9071832.
- [26] B. Pan, H. Li, Z. Yao, D. Cai, et H. Sun, « Reinforced Dynamic Reasoning for Conversational Question Generation », in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, juill. 2019, p. 2114-2124. doi: 10.18653/v1/P19-1203.
- [27] X. Du, J. Shao, et C. Cardie, « Learning to Ask: Neural Question Generation for Reading Comprehension », juill. 2017, p. 1342-1352. doi: 10.18653/v1/P17-1123.
- [28] X. Yao et Y. Zhang, « Question Generation with Minimal Recursion Semantics », Saarland University & University of Groningen, 2010.
- [29] R. Mitkov et L. A. Ha, « Computer-Aided Generation of Multiple-Choice Tests », in *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, 2003, p. 17-22. Consulté le: 30 juillet 2022.
- [30] I. Ureel, F. Leo C., R. Kenneth D., B. Chris, et Larry, « Question Generation for Learning by Reading »:, Defense Technical Information Center, Fort Belvoir, VA, janv. 2005. doi: 10.21236/ADA466018.
- [31] H.-T. Zheng, J. Han, J. Chen, et A. Sangaiah, « A novel framework for Automatic Chinese Question Generation based on multi-feature neural network model », *Comput. Sci. Inf. Syst.*, vol. 15, n° 3, p. 487-499, 2018, doi: 10.2298/CSIS171121018Z.
- [32] N.-T. Le et N. Pinkwart, « Evaluation of a question generation approach using semantic web for supporting argumentation », *Res. Pract. Technol. Enhanc. Learn.*, vol. 10, n° 1, Art. n° 1, déc. 2015, doi: 10.1007/s41039-015-0003-3.
- [33] R. Zhang, J. Guo, L. Chen, Y. Fan, et X. Cheng, « A Review on Question Generation from Natural Language Text », *ACM Trans. Inf. Syst.*, vol. 40, n° 1, p. 1-43, janv. 2022, doi: 10.1145/3468889.
- [34] B. Wyse et P. Piwek, « Generating Questions from OpenLearn study units », p. 8, 2009.
- [35] M. Heilman et N. A. Smith, « Question Generation via Overgenerating Transformations and Ranking »:, Defense Technical Information Center, Fort Belvoir, VA, janv. 2009. doi: 10.21236/ADA531042.
- [36] D. Klein et C. D. Manning, « Fast Exact Inference with a Factored Model for Natural Language Parsing », p. 8.
- [37] H. Ali, Y. Chali, et S. A. Hasan, « Automation of Question Generation From Sentences », p. 11.
- [38] M. Flor et B. Riordan, « A Semantic Role-based Approach to Open-Domain Automatic

- Question Generation », in *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, New Orleans, Louisiana, juin 2018, p. 254-263. doi: 10.18653/v1/W18-0530.
- [39] L. Schwartz, T. Aikawa, et M. Pahud, « Dynamic Language Learning Tools », p. 4.
- [40] Q. Zhou, N. Yang, F. Wei, C. Tan, H. Bao, et M. Zhou, « Neural Question Generation from Text: A Preliminary Study ». arXiv, 18 avril 2017. Consulté le: 6 août 2022.
- [41] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, et Y. Bengio, « Pointing the Unknown Words ». arXiv, 21 août 2016. Consulté le: 6 août 2022. [En ligne]. Disponible sur: <http://arxiv.org/abs/1603.08148>
- [42] S. K. Mishra, P. Goel, A. Sharma, A. Jagannatha, D. Jacobs, et H. Daumé III, « Towards Automatic Generation of Questions from Long Answers ». arXiv, 15 avril 2020. doi: 10.48550/arXiv.2004.05109.
- [43] Y.-H. Chan et Y.-C. Fan, « A Recurrent BERT-based Model for Question Generation », nov. 2019, p. 154-162. doi: 10.18653/v1/D19-5821.
- [44] P. Rajpurkar, J. Zhang, K. Lopyrev, et P. Liang, « SQuAD: 100,000+ Questions for Machine Comprehension of Text ». arXiv, 10 octobre 2016. Consulté le: 26 août 2022.
- [45] P. Bajaj *et al.*, « MS MARCO: A Human Generated MACHine Reading COMprehension Dataset ». arXiv, 31 octobre 2018. doi: 10.48550/arXiv.1611.09268.
- [46] A. Trischler *et al.*, « NewsQA: A Machine Comprehension Dataset ». arXiv, 7 février 2017. doi: 10.48550/arXiv.1611.09830.
- [47] H. Mozannar, E. Maamary, K. El Hajal, et H. Hajj, « Neural Arabic Question Answering », in *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, Florence, Italy, août 2019, p. 108-118. doi: 10.18653/v1/W19-4612.
- [48] K. Papineni, S. Roukos, T. Ward, et W.-J. Zhu, « Bleu: a Method for Automatic Evaluation of Machine Translation », in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, juill. 2002, p. 311-318. doi: 10.3115/1073083.1073135.
- [49] M. Denkowski et A. Lavie, « Meteor Universal: Language Specific Translation Evaluation for Any Target Language », in *Proceedings of the Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland, USA, juin 2014, p. 376-380. doi: 10.3115/v1/W14-3348.
- [50] C.-Y. Lin, « ROUGE: A Package for Automatic Evaluation of Summaries », in *Text Summarization Branches Out*, Barcelona, Spain, juill. 2004, p. 74-81. Consulté le: 29 août

2022.

- [51] Y. Zhao, X. Ni, Y. Ding, et Q. Ke, « Paragraph-level Neural Question Generation with Maxout Pointer and Gated Self-attention Networks », 2018, p. 3901-3910. doi: 10.18653/v1/D18-1424.
- [52] E. M. B. Nagoudi, A. Elmadany, et M. Abdul-Mageed, « AraT5: Text-to-Text Transformers for Arabic Language Generation », in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland, mai 2022, p. 628-647. doi: 10.18653/v1/2022.acl-long.47.
- [53] J. H. Clark *et al.*, « TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages », *Trans. Assoc. Comput. Linguist.*, vol. 8, p. 454-470, 2020, doi: 10.1162/tacl_a_00317.
- [54] P. Lewis, B. Oğuz, R. Rinott, S. Riedel, et H. Schwenk, « MLQA: Evaluating Cross-lingual Extractive Question Answering ». arXiv, 3 mai 2020. doi: 10.48550/arXiv.1910.07475.
- [55] « Lightning-AI/lightning: Build and train PyTorch models and connect them to the ML lifecycle using Lightning App templates, without handling DIY infrastructure, cost management, scaling, and other headaches. » <https://github.com/Lightning-AI/lightning> (consulté le 27 septembre 2022).
- [56] T. Wolf *et al.*, « Transformers: State-of-the-Art Natural Language Processing », in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, oct. 2020, p. 38-45. doi: 10.18653/v1/2020.emnlp-demos.6.
- [57] A. Abid, A. Abdalla, A. Abid, D. Khan, A. Alfozan, et J. Zou, « Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild ». arXiv, 6 juin 2019. doi: 10.48550/arXiv.1906.02569.
- [58] T. B. Brown *et al.*, « Language Models are Few-Shot Learners ». arXiv, 22 juillet 2020. doi: 10.48550/arXiv.2005.14165.
- [59] V. Rus, Z. Cai, et A. Graesser, « Question Generation: Example of A Multi-year Evaluation Campaign », p. 3.