

Research and Methodology in Data Science

Open Domain Question Answering

Idles MAMOU
Amine DJEGHRI

January 29, 2021

1 Introduction

2 État de l'art

- 2.1 Modèles Knowledge Based
- 2.2 Modèles IR Based

3 Notre Contribution

4 Expérimentations

5 Perspectives

Qu'est ce que les systèmes Questions-Réponses ?

- Des systèmes permettant de répondre automatiquement à des questions en langage naturel.
- Peuvent s'appliquer à des données **structurées** (Bases de Connaissances) ou **non structurées** (Texte Brut).
- Travaillent sur un domaine **ouvert** (ex: Web/Wikipedia) ou **fermé** (ex: Appli BI).



Figure: Résultat d'une question sur Google

- 60's/70's : Premiers systèmes QA à base de règles syntaxiques. ex: BASEBALL (1961), Lunar (1977).
- Vers 00's: Émergence des conférences TREC-QA et QA@CLEF et premiers modèles IR Based.
- 2007: Développement de Freebase (devenu WikiData) et DBpedia et essor du "Knowledge Based QA".
- Depuis 2013: Publication de larges Dataset en conséquence du développement du Deep Learning.

État de l'art

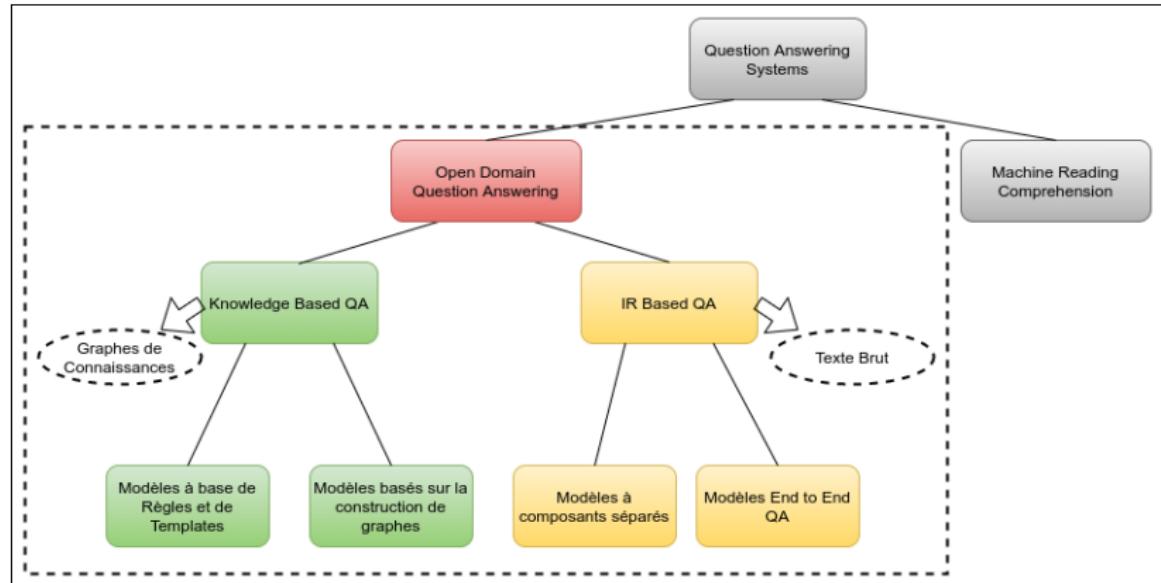


Figure: Catégories des modèles de Question Answering

Modèles Knowledge Based

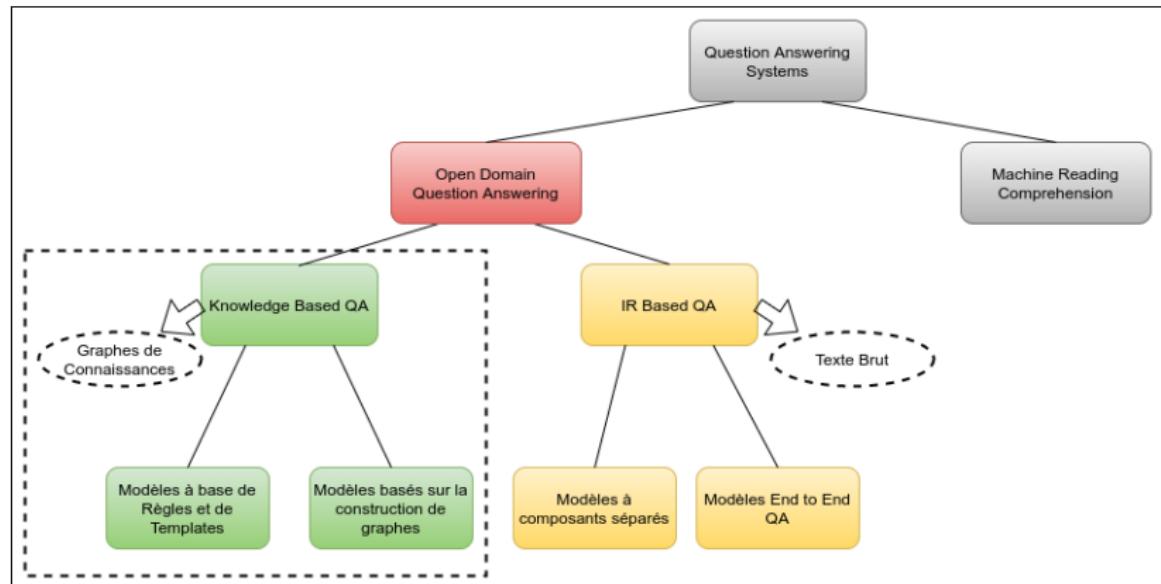


Figure: Catégories des modèles de Question Answering

- Données sauvegardées sous forme **structurée** dans des Bases de Connaissance.
=> FreeBase (2008), DBpedia (2015), YAGO...
- Objectif :
 - Passer du langage naturel en un langage de requête (ex SPARQL) exécutable sur la base de connaissances.

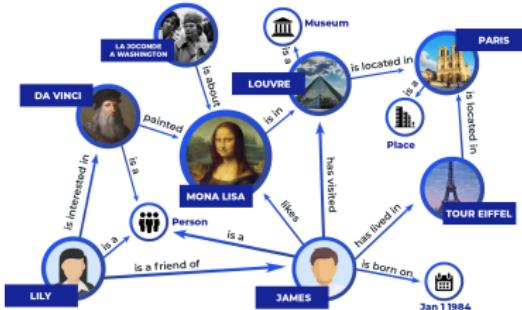


Figure: Exemple de Graphe de Connaissances

Deux Catégories de modèles à base de connaissances:

- Modèles à base de templates:
 - Reposent sur des règles, des lexiques et des templates prédéfinis à la main.
=> Berant et al (2013) [3], Bast et al (2015)[2]
- Modèles basés sur la construction de graphes:
 - Construction d'un sous graphe de connaissances à l'aide de réseaux de neurones pour réduire l'espace de recherche.
=> Yih et al (2015)[11], Bao et al (2016)[5], Bhutani et al (2019)[12], Lan et al (2020)[6]

Modèles KB à base de Templates

Le modèle **AQQU** (*Bast et al 2015*)[2]

- 3 templates prédéfinis.
- Identification des entités candidates (Projection sur les templates).
- Construction du sous graphe centré sur les entités identifiées.
- Extraction de features et passage au Learning to Rank.

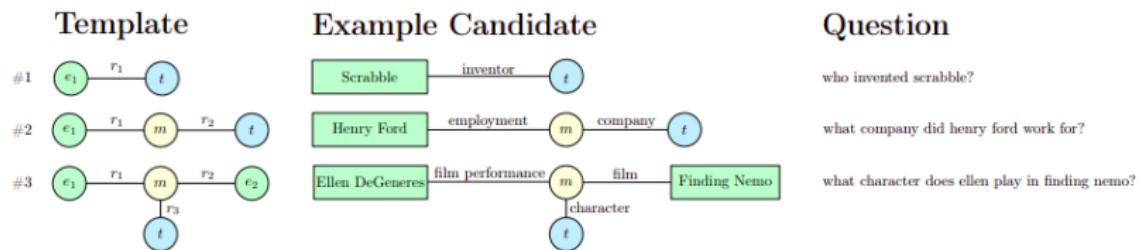


Figure: Les types de Templates utilisés dans AQQU

Le modèle **AQQU** (*Bast et al 2015*)[2]

- Avantages:
 - ➊ Beaucoup de questions (simples) calquent très bien sur les templates.
- Inconvénients:
 - ➋ Nécessite l'intervention humaine pour construire les templates.
 - ➋ Inefficace pour répondre à des questions plus complexes.

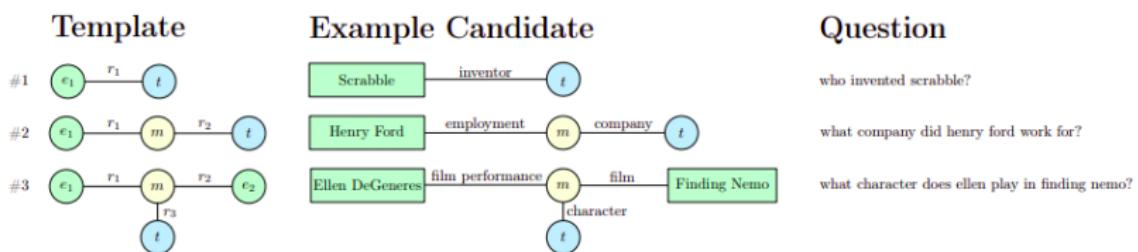
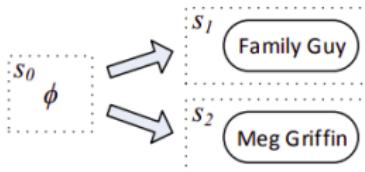


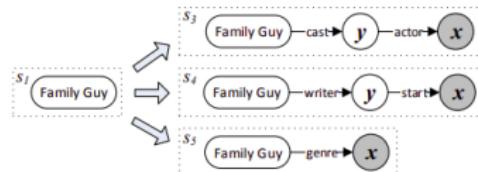
Figure: Les types de Templates utilisés dans AQQU

Modèles KB Basés sur la Construction de Graphes

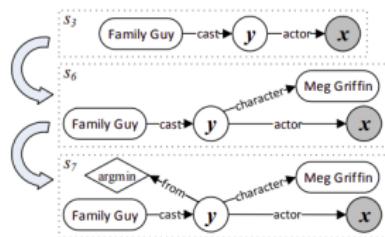
STAGG: Staged Query Graph Generation (Yih et al 2015)[11]



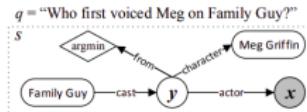
(I) Extraction des entités principales (POS Tagging)



(II) Identification du meilleur chemin à l'aide de CNN



(III) Ajout des contraintes (autres entités ou agrégation)
au chemin retenu



- (1) EntityLinkingScore("FamilyGuy", "Family Guy") = 0.9
- (2) PatChain("who first voiced meg on <>", cast-actor) = 0.7
- (3) QuesEP(q , "family guy cast-actor") = 0.6
- (4) ClueWeb("who first voiced meg on <>", cast-actor) = 0.2
- (5) ConstraintEntityWord("Meg Griffin", q) = 0.5
- (6) ConstraintEntityInQ("Meg Griffin", q) = 1
- (7) AggregationKeyword(argmin, q) = 1
- (8) NumNodes(s) = 5
- (9) NumAns(s) = 1

(IV) Extraction de caractéristiques pour le Learning to Rank

STAGG: Staged Query Graph Generation (*Yih et al* 2015)[11]

- Avantages:

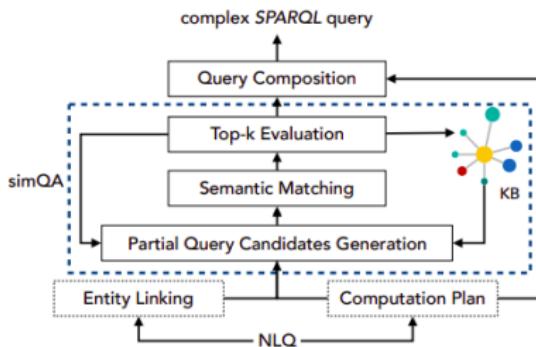
- ① Réduction de l'intervention humaine.
- ② Réduction de l'espace de recherche.

- Inconvénients:

- ① Applicable que sur des chemins de longueur maximale 2.
- ② Pas adapté aux questions complexes.

TextRay (Bhutani et al 2019)[12]

- **Hypothèse:** Un graphe de requête complexe peut être construit par la génération de multitudes de graphes simples.



- Avantages:
 - ➊ Plus robuste face aux questions complexes.
- Inconvénients:
 - ➊ Dépend fortement des performances du STAGG.
 - ➋ Processus Itératif lourd.

(Lan et al 2020)[6]

- **Idée:** Re-visiter le STAGG pour permettre la construction de chemins de longueur supérieure à 2.
- Introduction des contraintes **avant** la génération complète du chemin.
- Beam Search pour la recherche du meilleur chemin.

=> Diminution de l'espace de recherche.

=> Plus de flexibilité face aux questions complexes.

Performances des modèles Knowledge Based sur les benchmarks de l'état de l'art:

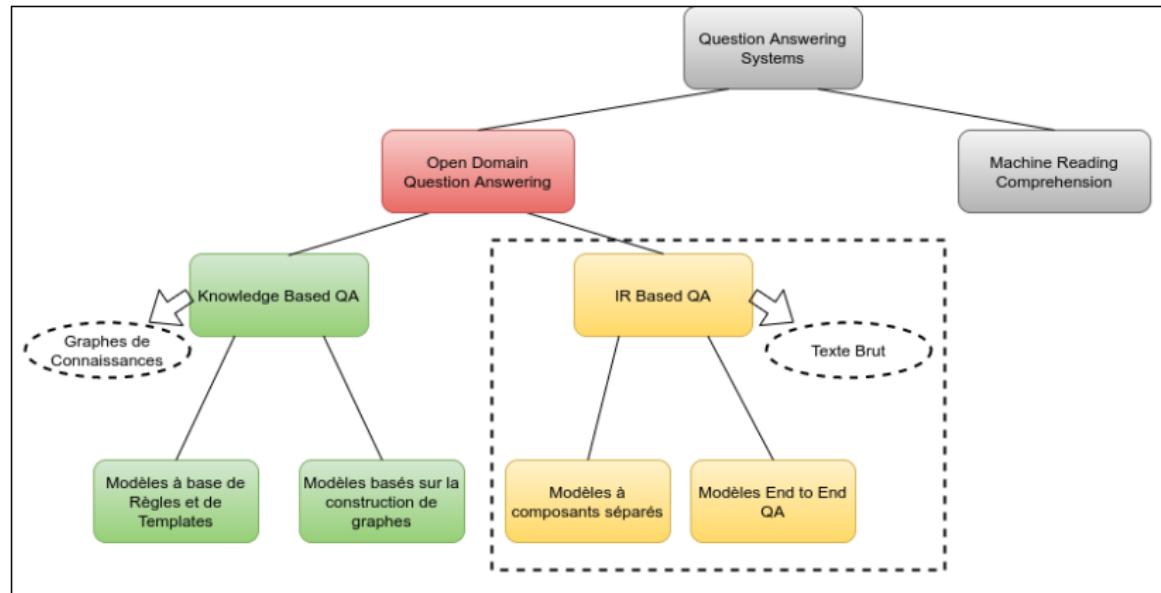
	ComplexWeb-Questions	WebQuestionsSP	ComplexQuestions
STAGG (Yih et al, 2015)	-	69.0	37.0
MultiCG (Bao et al, 2016)	-	-	42.3
Luo et al (2018)	-	-	42.8
Chen et al (2019)	29.8	68.5	35.3
TextRay (Buhtani et al, 2019)	33.9	60.3	-
Lan et al (2020)	40.4	74.0	43.3

Figure: **F1 score (%)** des différents modèles de l'état de l'art du Knowledge Based QA

=> Dépendance vis-à-vis des ontologies des bases de connaissances.

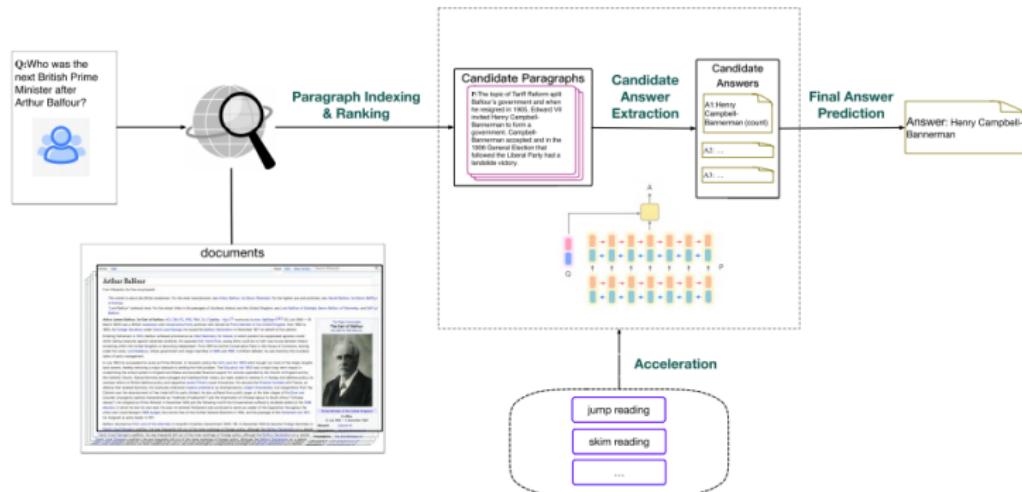
=> Nécessité du maintien à jour de la base.

Modèles IR Based



- Données sauvegardées sous forme **non structurée** dans des bases de documents .
=> Wikipedia, Web...

- Données sauvegardées sous forme **non structurée** dans des bases de documents .
=> Wikipedia, Web...
- ① Recherche et filtrage des passages candidats.
 - ② Extraction des réponses depuis les passages obtenus en 1.



Deux Catégories de modèles IR Based:

- Modèles à composants séparés
 - Séparation des deux modules.
 - Modèle de RI pour le ranking.
 - Modèle de MRC pour l'extraction de réponses.
- => Chen et al (2017)[1]
- Modèles End to End QA:
 - Entraînement simultané des deux modules pour la tache de Question Answering.
- => Wang et al (2018)[9], Lin et al (2018)[10], Hu et al (2019)[7]

Modèles IR Based à Composants Séparés

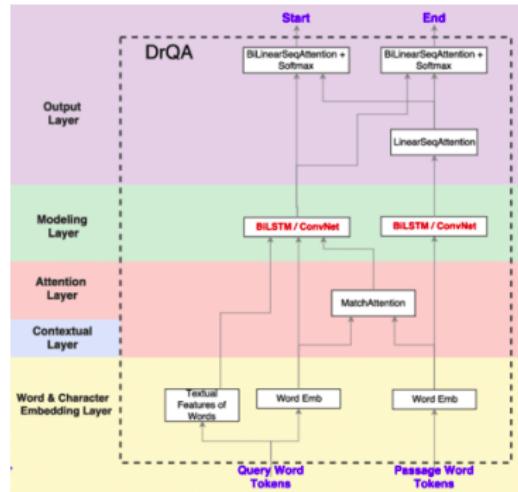
DrQA(Chen et al 2017)[1]

① Document Retriever:

- TF-IDF sur des bigrammes de mots.

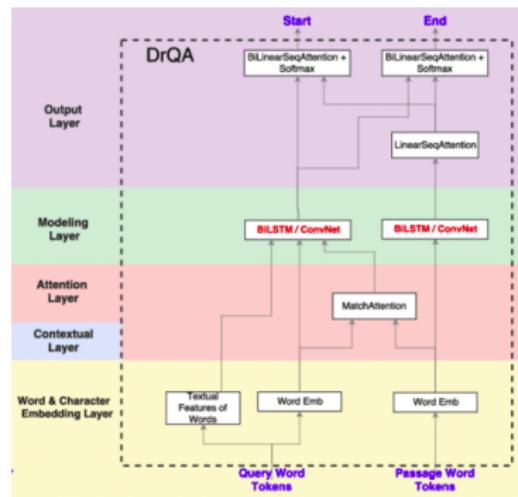
② Document Reader: Inspiré des modèles de MRC.

- Projection GloVe sur la question et les documents.
- LSTM + co-attention pour la représentation contextuelle
- Deux modules linéaires pour prédire le début et la fin de la réponse.



DrQA (Chen et al 2017)[1]

- Avantages:
 - ① Profite du compromis Performance/Rapidité offert par le modèle TF-IDF.
- Inconvénients:
 - ① Seul le Reader est entraîné pour la tache de Question Answering.
 - ② Propagation d'erreurs depuis le premier module de RI.

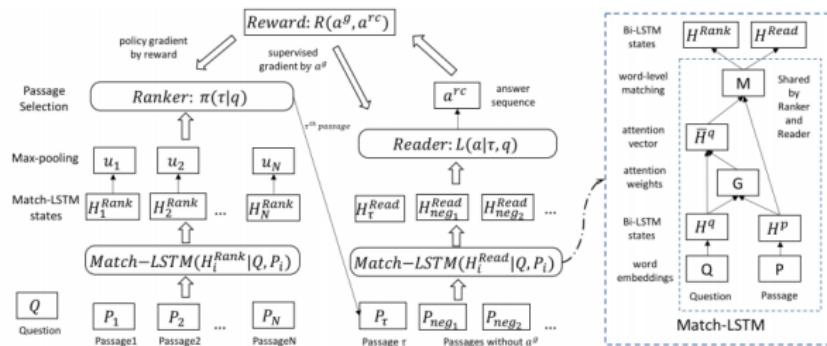


Modèles IR Based End to End

R3: Reinforced-Ranker-Reader(Wang et al 2018)[9]

Idée: Lier les deux modules Ranker-Reader par de l'apprentissage par renforcement.

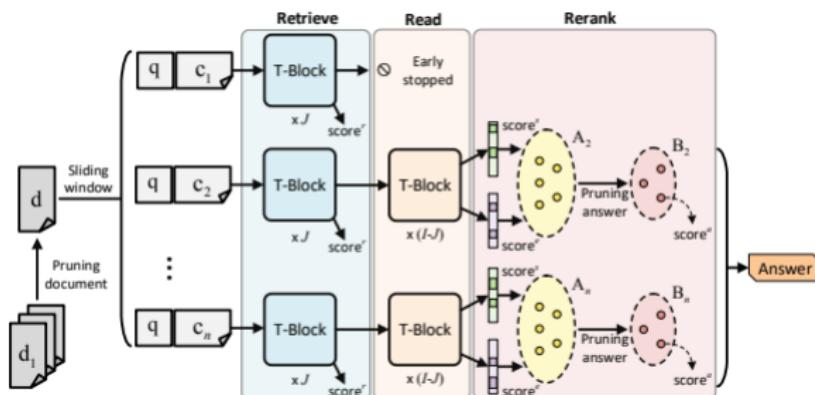
- Pré-filtrage par BM-25.
- Représentation contextuelle des passages via Match-LSTM et Ranking.
- Attribution de récompense au Ranker en fonction de la qualité des passages choisis. (REINFORCE Williams 1992).



RE3QA(*Hu et al 2019*)[7]

Idée: Partage du même encodage par les deux blocs.

- Pré-filtrage par TF-IDF.
- Passage par les blocs en surface de BERT + MLP pour le ranking (deuxième filtrage).
- Passage par les blocs BERT les plus profonds pour l'extraction de la réponse.



R3 - R3QA

- Avantages:
 - ① Les deux modules sont entraînés pour la tache de Question Answering.
- Inconvénients:
 - ① Pas applicable sur de larges corpus de données. ==> Pré-filtrage obligatoire.

Notre Contribution

Objectif : s'affranchir de la dépendance vis à vis des modèles de correspondances lexicales (BM25,TF-IDF,..).

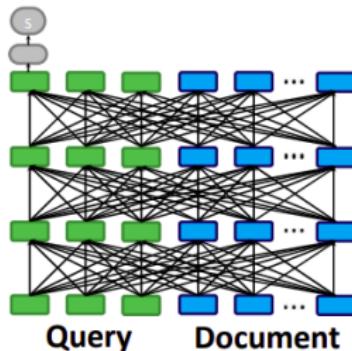
Idée:

Exploiter deux travaux :

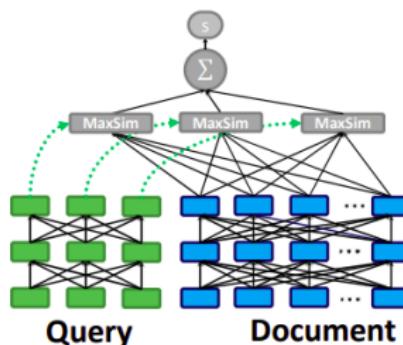
- Le **mécanisme d'interaction tardive** développé par *Khattab et Zaharia* en 2020 pour le modèle de RI "ColBERT"[8].
- Le framework de **calcul de similarité à grande échelle** Facebook AI Similarity Search "FAISS"[4].

Late Interaction (CoBERT 2020[8]):

- Retarder l'interaction entre la question et les passages.
- Encodage BERT des deux parties indépendamment.
=> Encodage des passages en amont de l'étape d'inférence.
=> Gain considérable en Coût.



| (c) All-to-all Interaction
(*e.g., BERT*)



| (d) Late Interaction
(*i.e., the proposed CoBERT*)

FAISS : Facebook AI Similarity Search (Johnson et al 2017)[4]

- Calcul de similarité entre millions (voire milliards) de vecteurs à moindre coût.
- Approximation des KNN par Compression, Partitionnement et Indexation vectorielle.
- Exploitation efficace des GPU.

Module de Ranking

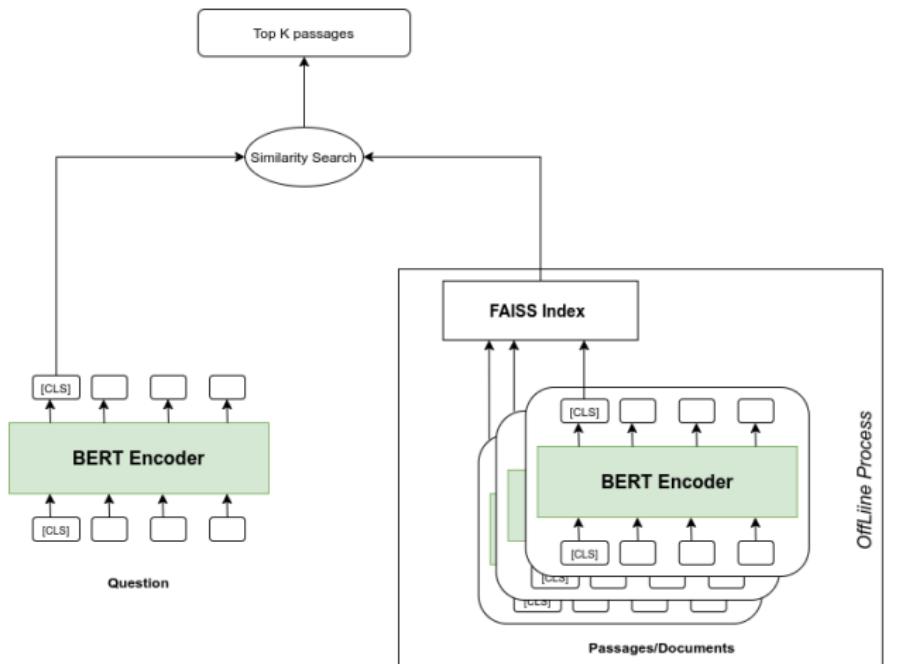


Figure: Architecture du module de Ranking

Apprentissage

- Apprentissage PairWise.
- Pour chaque question, le passage contenant la réponse, un (ou des) passages ne contenant pas la réponse.
==> Utilisation des dataset de la Compréhension de Textes.

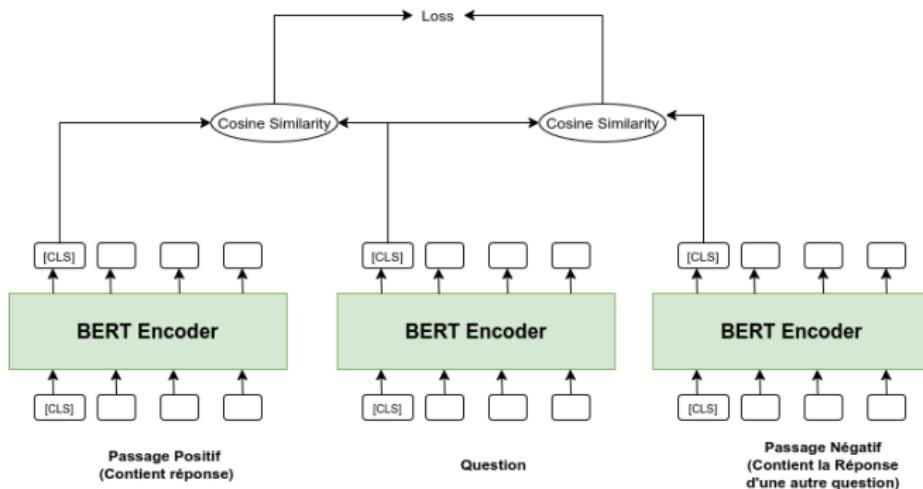


Figure: Training Time

Fonction de coût:

$$\text{Loss}(q_i, p_i^+, p_i^-) = -\log \frac{e^{\text{sim}(q_i, p_i^+)}}{e^{\text{sim}(q_i, p_i^+)} + e^{\text{sim}(q_i, p_i^-)}}$$

Où q_i représente la ième question, p_i^+ et p_i^- représentent respectivement les passages positifs et négatifs.

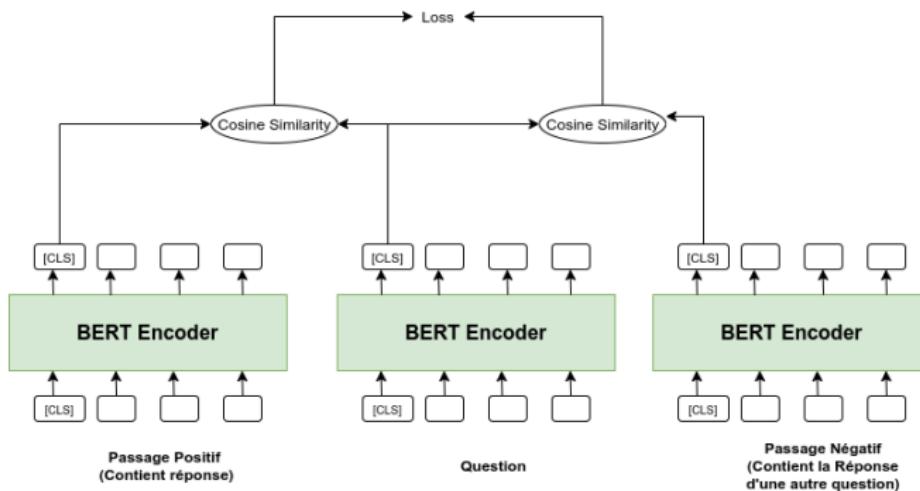


Figure: Training Time

Prédire l'extrait le plus vraisemblable dans les k passages retenus.

- Utilisation de la projection BERT de chaque token.
- Apprentissage de deux modules linéaires pour le calcul du début et de la fin de l'extrait le plus vraisemblable tel que:

$$P_{debut}(m) = \text{softmax}(mW_{debut})$$
$$P_{fin}(m) = \text{softmax}(mW_{fin})$$

où m est la représentation BERT du mot et W_{debut} et W_{fin} les paramètres des deux modules linéaires.

==> Les modèles de MRC dépassent de loin les performances humaines dans cette tache.

Expérimentations

DATA:

- **SQuAD-open:** (*Chen et al, 2017*)
=> Basé sur le dataset de MRC SQuAD (2016).
=> 100k paires questions-réponses de Wikipedia.
- **SearchQA:** (*Dunn et al, 2017*)
=> 140k paires questions-réponses depuis Google Search Engine
- **TriviaQA:** (*Joshi et al, 2017*)
=> 110k paires de questions-réponses obtenues sur des sites d'anecdotes.
- **Quasar-T:** (*Dhingra et al, 2017*)
=> 43k paires de diverses sources du Web.

dataset	query form	answer form	question source	context source	granularity
SQuAD-open [37]	full question	span	crowdsourced	Wikipedia	document level
SearchQA [15]	key word/ phrase	span	Jeopardy!	Google search results	paragraph level
TriviaQA [14]	full question	span	Trivia websites	Wikipedia & Bing search results	document level
Quasar-T [30]	full question	span	Free Database	CluWeb09	paragraph level

Figure: Les benchmarks de IR Based Question Answering

Métriques:

- Exact Match (Accuracy).
- Moyenne de F1 score.

- => Implémentation : Python3, PyTorch.
- => Construction Training Set à partir de MS-MARCO et SQuAD.
- => Librairie "*transformers*" pour BERT pré-entraîné. Dim=256.
- => Librairie "*FAISS*" pour le calcul de similarité à grande échelle.
- => Learning rate: parmi $\{10^{-3}, 10^{-4}, 10^{-5}\}$
- => Batch Size: 30

Résultats

	SQuAD-open		SearchQA		TriviaQA		QuasarT	
	EM	F1	EM	F1	EM	F1	EM	F1
DrQA (2017)	28.4	-	51.4	58.2	48.0	52.1	36.9	45.5
<i>R</i> ³ (2018)	29.1	37.5	49.0	55.3	47.3	53.7	35.3	41.7
DS-QA (2018)	28.7	36.6	58.5	64.5	48.7	56.3	37.3	43.6
ORQA (2019)	20.2	-	-	-	45.1	-	-	-
<i>RE</i> ³ QA (2019)	41.9	50.2	-	-	65.5	71.2	-	-
Notre Modèle	54.3	56.1	62.4	66.3	65.1	70.3	54.2	46.4
Notre Modèle (avec Transfert)	55.2*	57.3*	63.5	67.1	65.2	70.6	55.6*	47.2*

Figure: Comparaison des résultats

	Temps de Réponse Question (sec)	Temps de Construction Index (heures)
TF-IDF+Reader	0.09	0.5
Notre Modèle	0.88	10

Figure: Comparaison des temps de réponse et de construction d'index par rapport au TF-IDF

	Ranking		Reading Comprehension		
			Encoding	Attention	Prediction
DrQA (2017)		TF-IDF	Glove	co-attention	Bi-LSTM
R ³ (2018)	BM25+MatchLSTM		Glove	co-attention	Match-LSTM
DS-QA (2018)	TF-IDF+LSTM+MLP		Glove	self-attention	Bi-LSTM
ORQA (2019)	ICT+BERT	BERT		self-attention	BERT
RE ³ QA (2019)	TF-IDF	BERT		self-attention	BERT
Notre Modèle	BERT	BERT	self-attention		BERT

Figure: Comparaison de la structure des modèles

- Lier les deux modules pour obtenir un modèle de Question-Answering de bout en bout.
- Allier les techniques de recherche d'information et les bases de connaissances pour les questions complexes.

- [1] Jason Weston Antoine Bordes Danqi Chen Adam Fisch. "Reading Wikipedia to Answer Open-Domain Questions". In: *Proc. Annu. Meeting Assoc. Comput. Linguistics (ACL)*, vol. 1. Vancouver, BC, Canada (2017), 1870–1879.
- [2] E.Haussmann H.Bast. "More Accurate Question Answering on Freebase". In: *CIKM'15, October 19–23 Melbourne, Australia* (2015).
- [3] R.Frostig J.Berant A.Chou and P.Liang. "Semantic Parsing on Freebase from Question-Answer Pairs". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (2013), 1533–1544.
- [4] Maffhijs Douze Jeff Johnson and Herve J ´ egou. "Billion-scale similarity search with GPUs". In: *arXiv preprint arXiv:1702.08734* (2017).
- [5] Zhao Yan] Ming Zhou Tiejun Zhao Junwei Bao Nan Duan. "Constraint-Based Question Answering with Knowledge Graph". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (2016), 2503–2514.
- [6] Jiang J Lan Y. "Query Graph Generation for Answering Multi-hop Complex Questions from Knowledge Bases[". In: */Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. (2020), pp. 969–974.
- [7] Z. Huang M. Hu Y. Peng and D. Li. "Retrieve, read, rerank: Towards End-to-End multi-document reading comprehension". In: *in Proc. 57th Annu. Meeting Assoc. for Comput. Linguistics, Florence, Italy* (2019), 2285–2295.
- [8] M.Zaharia O.Khattab. "CoBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT". In: *T. In Proceedings of Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China* (2020).
- [9] X. Guo Z. Wang T. Klinger W. Zhang S. Chang G. Tesauro B. Zhou S. Wang M. Yu and J. Jiang. "R3: Reinforced ranker-reader for opendomain question answering". In: *in Proc. 32nd AAAI Conf. Artif. Intell., New Orleans, LA, USA* (2018), 5981–5988.

- [10] Z. Liu Y. Lin H. Ji and M. Sun. "Denoising distantly supervised opendomain question answering". In: *in Proc. 56th Annu. Meeting Assoc. Comput. Linguistics* (2018), 1736–1745.
- [11] Wen tau Yih Ming-Wei Chang Xiaodong He Jianfeng Gao. "Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* (2015), 1321–1331.
- [12] Nikita Bhutani Xinyi Zheng. "Learning to Answer Complex Questions over Knowledge Bases with Query Composition". In: *CIKM '19, November 3–7, 2019, Beijing, China* (2019).