Approche de génération de réponse à base de transformers

Imen Akermi, Johannes Heinecke, Frédéric Herledan

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Approche de génération de réponse à base de transformers

Imen Akermi Johannes Heinecke Frédéric Herledan
Orange / DATA-AI, Lannion, France
{imen.elakermi,johannes.heinecke,frederic.herledan}@orange.com

RÉSUMÉ

Cet article présente une approche non-supervisée basée sur les modèles Transformer pour la génération du langage naturel dans le cadre des systèmes de question-réponse. Cette approche permettrait de remédier à la problématique de génération de réponse trop courte ou trop longue sans avoir recours à des données annotées. Cette approche montre des résultats prometteurs pour l'anglais et le français.

ABSTRACT

Transformer based approach for answer generation

This paper presents an unsupervised approach for Natural Language Generation within the framework of question-answering task. This approach aims at solving the too-short or too-long answer generation problem without requiring annotated data. The proposed approach shows promising results for both English and French.

MOTS-CLÉS : systèmes de question-réponse, génération du langage naturel, analyse de dépendance.

KEYWORDS: question answering systems, natural language generation, dependency analysis.

1 Introduction

La popularité récente des assistants intelligents a accru l'intérêt pour les systèmes de question-réponse (SQR) qui sont devenus un élément central des échanges « Humain-Machine » puisqu'ils permettent aux utilisateurs d'avoir directement des réponses à leurs questions en langage naturel en utilisant leur propre terminologie sans avoir à parcourir une longue liste de documents pour trouver les réponses appropriées.

La plupart des travaux de recherche existants focalisent sur la complexité majeure de ces systèmes résidant dans le traitement et l'interprétation de la question qui exprime le besoin en information de l'utilisateur, sans pour autant donner de l'importance à la représentation de la réponse donnée en sortie. Généralement, la réponse est soit représentée par un ensemble de termes court répondant exactement à la question (cas des SQR qui extraient les réponses à partir de données structurées), soit par un passage d'un document qui indique la réponse exacte et qui peut intégrer d'autres informations inutiles ne relevant pas du contexte de la question posée.

La figure 1 présente un exemple de réponses données par deux systèmes différents.

Compte-tenu de la particularité des SQR qui extraient les réponses à partir de données structurées, les utilisateurs ne reçoivent, généralement, qu'une réponse courte et limitée à leurs questions comme c'est illustré par l'exemple de la figure 1. De plus, dans le cas où la question est posée oralement, ce type de représentation de réponse peut ne pas convenir aux attentes de l'utilisateur.

Volume 2 : Traitement Automatique des Langues Naturelles, pages 2–11.
HAL : hal-02784751.

FIGURE 1 – Réponses affichées par deux assistants intelligents, GOOGLE ASSISTANT et ALEXA pour la question « *Qui est le directeur de thèse d'Albert Einstein ?* »

En effet, le type de réponse donné par le premier système peut être perçu comme trop bref ne rappelant pas le contexte de la question et pour le deuxième type de réponse, on a tendance à renvoyer un passage qui contient la réponse mais qui inclut également des informations qui peuvent être hors du contexte de la question et jugées par l'utilisateur comme inutile.

C'est dans ce contexte que nous présentons dans cet article une approche qui permet de générer une réponse concise en langage naturel pour le français et l'anglais. Cette approche fait partie d'un SQR que nous avons proposé dans Rojas Barahona *et al.* (2019) et que nous présenterons brièvement dans cet article.

2 État de l'art

Malgré l'abondance des travaux dans le domaine des SQR, la problématique de formulation des réponses a reçu très peu d'attention. Une première approche traitant indirectement cette tâche a été proposée dans Brill *et al.* (2001, 2002). En effet, les auteurs avaient pour but de diversifier les motifs possibles de réponses en permutant les termes de la question en vue de maximiser le nombre de documents extraits qui peuvent contenir la réponse. Une autre approche de représentation de réponse basée sur des règles de reformulation a été également proposée dans Agichtein & Gravano (2000) et Lawrence & Giles (1998) dans le contexte d'expansion de requêtes pour la recherche de documents et non pour l'extraction de la réponse exacte.

Le peu de travaux qui se sont intéressés à cette tâche dans le cadre des SQR l'ont adressée sous l'angle de la génération de résumé de texte (Ishida *et al.*, 2018; Iida *et al.*, 2019; Rush *et al.*, 2015; Chopra *et al.*, 2016; Nallapati *et al.*, 2016; Miao & Blunsom, 2016; See *et al.*, 2017; Oh *et al.*, 2016; Sharp *et al.*, 2016; Tan *et al.*, 2016; dos Santos *et al.*, 2016). La majorité de ces travaux n'ont considéré que les questions de causalité de type « *pourquoi* » où les réponses sont des paragraphes. Pour rendre ces réponses plus concises, ils procèdent à un compactage des paragraphes extraits.

D'autres approches (Kruengkrai *et al.*, 2017; Girju, 2003; Verberne *et al.*, 2011; Oh *et al.*, 2013) ont exploré cette tâche comme un problème de classification où il s'agit de prédire si un passage de texte pourra constituer une réponse à une question donnée.

Il faut noter que ces approches ont pour seul but de diversifier au maximum les formules de réponses possibles à des questions pour augmenter la probabilité d'extraire la bonne réponse et non pour générer une réponse qui soit conviviale pour l'utilisateur. Il faut également souligner que ces approches ne sont valables que pour les SQR qui génèrent les réponses sous forme d'extrait de texte et ne pourront pas être appliquées aux réponses courtes.

Notre approche de génération de réponse diffère de ces travaux dans le sens qu'elle est non supervisée, peut s'adapter à n'importe quel type de questions factuelles (à l'exception des questions de type *Pourquoi* et *Comment*) et qu'elle s'appuie que sur des données facilement accessibles et non annotées. En effet, nous partons de l'hypothèse intuitive qu'une réponse concise et facilement prononçable par un assistant intelligent n'est en fait qu'une reformulation de la question posée. Cette approche est intégrée à un système (Rojas Barahona *et al.*, 2019) qui permet d'extraire la réponse à une question à partir de données structurées.

Dans ce qui suit, nous détaillerons dans la section 3 l'approche que nous avons proposé pour la génération de réponse en Langage Naturel et nous évoquerons brièvement le SQR développé. Nous présenterons dans la section 4 les expérimentations que nous avons conduit pour évaluer cette approche. Nous conclurons dans la section 5 avec les limites relevées et les perspectives envisagées.

3 Approche de génération de réponse en langage naturel

L'approche de génération de réponse que nous décrivons dans cet article est un composant d'un SQR qui a été développé dans Rojas Barahona *et al.* (2019) et qui consiste à traduire une question en langue naturelle (français ou anglais) dans une représentation formelle pour ensuite transformer cette représentation formelle en une requête Sparql [1]. Grâce à la requête Sparql nous cherchons la réponse dans une base de connaissance RDF, dans notre cas Wikidata [2]. La réponse est toujours une liste d'URI ou de valeurs.

Bien que nous arrivons à trouver la réponse exacte à une question, sa représentation n'est pas conviviale pour l'utilisateur. De ce fait, nous proposons une approche non-supervisée qui intègre l'utilisation des modèles Transformers tels que BERT (Devlin *et al.*, 2019) et GPT (Radford *et al.*, 2018). Le choix d'une approche non supervisée émane du fait qu'il n'existe pas un corpus d'apprentissage associant une question à une réponse compacte, exhaustive et qui permettrait d'appliquer en mode supervisé une architecture neuronale End-to-End apprenant à générer une phrase répondant à une question.

Cette approche part du fait que nous avons déjà extrait la réponse exacte à une question posée. Nous supposons qu'une réponse bien formulée n'est que la reformulation de la question même associée à la réponse exacte. Cette approche est constituée de deux étapes fondamentales. La première étape consiste à effectuer une analyse de dépendance de la question en entrée et nous procédons dans une deuxième étape à la génération de la réponse.

3.1 Analyse en dépendance

Pour l'analyse en dépendance, nous utilisons une version améliorée de UdpipeFuture (Straka, 2018) qui était le système gagnant (3e avec les métriques *Labelled Attachment Score* (LAS) et *Content Word LAS* (CLAS) et 1er en utilisant la métrique *Morphology-Aware LAS* (MLAS)) de la tâche de l'analyse en dépendance (Zeman *et al.*, 2018). UdpipeFuture est un analyseur, qui fait l'étiquetage en partie de discours et la lemmatisation avec un LSTM. L'analyse en dépendance est faite avec un parser à base de graphes, inspiré de Dozat *et al.* (2017).

Notre modification consiste à intégrer les plongements contextuels à UdpipeFuture lors de l'apprentissage. Pour cela, nous nous sommes orientés vers BERT multilingue (Devlin *et al.*, 2019), XLM-R

1. https://www.w3.org/TR/sparql11-overview/
2. https://www.wikidata.org/

(Conneau *et al.*, 2019) (pour l'anglais et français), RoBERTA (Liu *et al.*, 2019) (pour l'anglais) FlauBERT (Le *et al.*, 2020) et CamemBERT (Martin *et al.*, 2019) (pour le français) lors de l'apprentissage des treebanks French-GSD et English-EWT [3], issues du projet Universal Dependencies (UD) (Nivre *et al.*, 2016) [4]. L'ajout des plongements contextuels a significativement augmenté les résultats pour les trois métriques, LAS, CLAS et MLAS (cf. table 1).

français (Fr-GSD)			
embeddings	MLAS	CLAS	LAS
Straka (2018)	77,29	82,49	85,74
FlauBERT	79,53	84,16	87,98
BERT	81,64	86,21	89,68
CamemBERT	82,17	86,45	89,67
XLM-R	**82,62**	**86,94**	**89,82**

anglais (En-EWT)			
embeddings	MLAS	CLAS	LAS
Straka (2018)	74,71	79,14	82,51
BERT	81,16	85,89	88,63
RoBERTA	82,38	86,89	89,40
XLM-R	**82,91**	**87,24**	**89,54**

TABLE 1 – Analyse en dépendance du français et l'anglais (UD v2.2), les meilleurs résultats en gras

Néanmoins, pour l'analyse en dépendances des questions simples de type quiz, les deux treebanks UD (En-EWT et Fr-GSD) ne sont pas adaptés, car leurs corpus d'apprentissage ne contiennent pas ou très peu de questions [5]. Les résultats de l'analyse des questions avec des modèles appris sur les données standard de UD sont montrés en table 2.

français (Fr-GSD)			
embeddings	MLAS	CLAS	LAS
BERT	60,52	73,04	79,27
CamemBERT	**61,32**	**75,26**	**80,49**
FlauBERT	58,09	70,96	78,40
Word2Vec	59,83	74,43	80,14
XLM-R	59,23	73,52	79,27

anglais (En-EWT)			
embeddings	MLAS	CLAS	LAS
BERT	**80,45**	**88,02**	**90,58**
RoBERTa	80,68	89,17	91,49
XLM-R	80,68	89,42	91,88

TABLE 2 – Analyse des questions avec des modèles appris sur UD sans modifications

Afin d'améliorer l'analyse, nous avons enrichi les treebanks d'apprentissage Fr-GSD et En-EWT en annotant 309 questions anglaises des challenges QALD7 (Usbeck *et al.*, 2017) et QALD8 [6] (ainsi 91 questions pour le test) en supprimant les doublons. Pour le français, nous avons traduit des questions issues de QALD7, et formulé des questions nous-mêmes (66 test, 267 apprentissage). Les annotations ont été effectuées par deux linguistes avec le guide d'annotation du project Universal Dependencies [7] et la documentation des treebanks Fr-GSD (pour les questions françaises) et En-EWT (pour l'anglais).

Comme la table 3 le montre, la qualité de l'analyse augmente considérablement, les embeddings CamemBERT (pour le français) et BERT (anglais) ont de nouveau le meilleurs impact.

Nous nous appuyons sur la version UdpipeFuture que nous avons améliorée avec BERT/CamemBERT et qui donne les meilleurs résultats en terme d'analyse en dépendance pour procéder au découpage de la question en fragments textuels (appelés également *chunks*) : $Q = \{c_1, c_2, \ldots, c_n\}$.

3. Comme la campagne d'évaluation Conll2018, nous utilisons la version 2.2 des treebanks UD pour la comparaison
4. https://universaldependencies.org/
5. Il existe également un treebank de questions, French-FQB (Seddah & Candito, 2016), mais la plupart des questions de ce treebank sont plutôt des questions longues, et peu similaires aux questions « typiques » du quiz.
6. https://github.com/ag-sc/QALD
7. https://universaldependencies.org/guidelines.html

français (Fr-GSD)

embeddings	MLAS	CLAS	LAS
BERT	91,20	96,10	97,55
CamemBERT	**92,12**	**97,37**	**98,26**
FlauBERT	90,53	94,74	96,86
Word2Vec	90,88	95,79	97,21
XLM-R	91,23	96,14	97,56

anglais (En-EWT)

embeddings	MLAS	CLAS	LAS
BERT	**84,85**	**91,92**	**94,24**
RoBERTa	83,08	91,67	93,85
XLM-R	83,08	90,66	93,59

TABLE 3 – Analyse des questions avec des corpus d'apprentissage UD de base enrichis de questions

Si on prend l'exemple de la question *Quel est le parti politique du maire de Paris ?*, l'ensemble des fragments textuels serait *Q = {Quel, est, le parti politique du maire de Paris}*.

3.2 Génération de la réponse

Lors de cette phase, nous procédons d'abord à un premier test de l'ensemble Q pour vérifier si le fragment textuel qui contient un marqueur de question (exp : *quel*, *quand*, *qui* etc.) représente le sujet *nsubj* dans la question analysée. Si c'est le cas, nous remplaçons tout simplement ce fragment textuel par la réponse que nous avons identifié précédemment. Reprenons l'exemple de la question *Quel est le parti politique du maire de Paris ?*, le système détecte automatiquement que le fragment textuel contenant le marqueur de question *Quel* représente bien le sujet et sera donc remplacé directement par la réponse exacte *Le Parti Socialiste*. Par suite, la réponse compacte générée sera *Le Parti Socialiste est le parti politique du maire de Paris*.
Dans le cas contraire, nous procédons à la suppression du fragment textuel contenant le marqueur de question que nous avons détecté et nous ajoutons la réponse R à l'ensemble Q : $Q = \{c_1, c_2, \ldots, c_{n-1}, R\}$

À partir de l'ensemble de fragments textuels Q, nous générons par permutation toutes les structures de réponses possibles qui peuvent former la phrase répondant à la question traitée :

$S = \{s_1(R, c_1, c_2, \ldots, c_{n-1}), s_2(c_1, R, c_2, \ldots, c_{n-1}), \ldots, s_m(c_1, c_2, \ldots, c_{n-1}, R)\}$

Ces structures seront évaluées par un Modèle de Langue (LM) basé sur des modèles Transformers qui permettra d'extraire la séquence de fragments textuels la plus probable qui servira de réponse :

$$structure^* = s \in S; p(s) = \underset{s_i \in S}{\operatorname{argmax}}\, p(s_i)$$

Une fois la structure, qui représentera la réponse à la question traitée, est identifiée, l'étape de génération du terme manquant est initiée. En effet, nous supposons qu'il pourrait y avoir un terme qui ne figure pas nécessairement dans la question ou dans la réponse mais qui est par contre nécessaire à la génération d'une bonne structure grammaticale de la réponse.

Par suite, pour prédire ce terme manquant nous faisons appel à BERT en tant que modèle de langue « *masqué* ». Dans le cas ou BERT renvoie une séquence de caractère non alphabétique, nous concluons que la structure optimale, telle qu'elle était prédite par le LM, ne nécessite pas d'être complétée par un terme supplémentaire.

L'exemple suivant illustre les différentes étapes de l'approche proposée :

Question : *Dans quel pays se trouve le Lac de Garde ?*

1. Découpage de la question en fragments textuels (à partir de l'analyse en dépendance obtenu avec UDPipe) → { *Dans quel pays*, *se trouve*, *le Lac De Garde* }
2. Suppression des marqueurs de question (*Dans quel pays*) → { *se trouve*, *le Lac De Garde* }
3. Ajout de la réponse brute → { *se trouve*, *le Lac de Garde*, *Italie* }
4. Génération des structures de réponses possibles
   → — *se trouve, le Lac de Garde, Italie*
   — *Italie, se trouve, le Lac de Garde*
   — ...
5. Évaluation des structures par un modèle de langage → *structure** = {*le Lac de Garde*, *se trouve*, *Italie*}
6. Génération des termes manquants à *structure** (avec BERT pour l'anglais ou CamemBERT pour le français)
   → *le Lac de Garde se trouve [terme manquant] Italie* : « en »

Réponse : *le Lac de Garde se trouve en Italie.*

4 Évaluation

Les corpus de tests existants pour l'évaluation des SQR, sont plus adaptés aux systèmes qui génèrent la réponse exacte à la question et donc une réponse courte ou sont plus axés vers la tâche de *Machine Reading Comprehension* où la réponse consiste en un passage de texte contenant la réponse exacte. Nous avons créé un jeu de données qui consiste à associer des questions extraites du corpus QALD-7 challenge (Usbeck *et al.*, 2017) avec des réponses en langage naturel qui ont été définies manuellement par un linguiste et que nous avons revues individuellement. Ce corpus appelé QueReo consiste en 150 questions avec les réponses exactes extraites par le SQR que nous avons décrit précédemment. On note en moyenne trois réponses possibles en langage naturel pour chaque question. Ce corpus existe en versions française et anglaise.

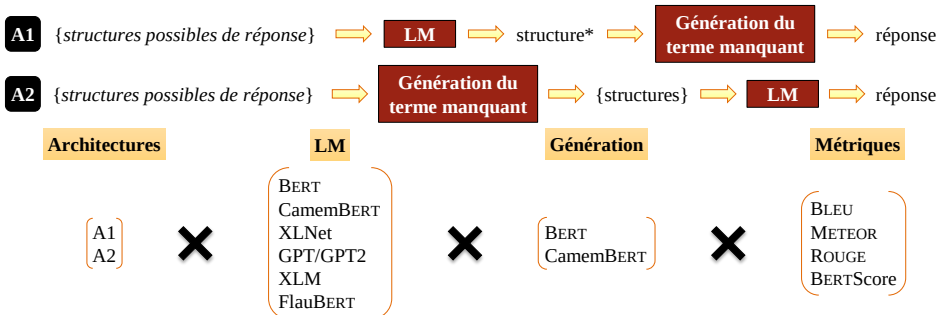

FIGURE 2 – Cadre d'expérimentation

Comme illustré dans la figure 2, deux architectures possibles de l'approche que nous proposons pour la génération de réponse ont été évaluées. La première architecture *A1* consiste à générer toutes les structures possibles de réponses pour les faire évaluer après par un LM qui permettrait la sélection de

la réponse optimale puis de générer le terme manquant à cette structure. L'architecture *A2* commence à générer le terme manquant pour chaque structure dans *Q* qui seront ensuite évaluer par le LM. Pour le moment, nous supposons qu'il y a seulement 1 terme manquant par structure.

Pour évaluer l'approche proposée, nous avons opté pour trois métriques N-gram (BLEU, METEOR et ROUGE) utilisées dans la littérature pour évaluer ce type de tâche et la métrique BERTScore qui exploite les plongements lexicaux pré-entraînés de BERT pour calculer la similarité entre la réponse générée et la réponse de référence. Pour pouvoir comparer les différentes variantes de l'approche, nous nous sommes référés au test de Friedman (Milton, 1939) qui permet de détecter les écarts de performances entre plusieurs modèles évalués par plusieurs métriques en se basant sur les rangs moyens.

La table 4 (corpus français) et la table 5 (corpus anglais) ne présentent que les résultats obtenus pour les cinq meilleurs modèles selon le classement du test de Friedman. Les modèles de langue utilisés sont adaptés selon la langue du corpus mis en test. Vanté par ses mérites en tant que modèle génératif très puissant entraîné sur un très large corpus de données, le modèle GPT a été également testé avec le corpus français pour voir s'il arrive à détecter la meilleure structure à choisir pour une question. Les valeurs mises entre parenthèses représentent le rang d'un modèle selon la métrique utilisée.

rang moyen	arch.	LM	modèle de génération	%BleuS score	%BleuS rang	%MeteorS score	%MeteorS rang	%RougeS score	%RougeS rang	%BertS score	%BertS rang
8,5	A1	BERT	CamemBERT	68,21	14	95,90	2	84,48	11	94,45	7
12	A1	GPT	CamemBERT	68,63	13	96,04	1	84,04	15	93,85	19
13,5	A1	GPT	BERT	70,39	6	94,90	38	85,70	4	94,61	6
13,5	A1	BERT	BERT	69,68	9	95,03	35	85,26	6	94,79	4
13,75	A2	BERT	BERT	75,08	1	94,78	48	85,91	3	94,98	3
16	A2	GPT	CamemBERT	69,37	10	95,73	13	83,59	21	93,82	20
18,5	A1	XLM	BERT	67,02	19	95,04	34	84,89	7	94,12	14
21,75	A2	XLM	BERT	68,67	12	94,44	59	84,67	8	94,42	8
25,25	A2	BERT	CamemBERT	63,82	52	95,82	7	83,13	26	93,98	16
26	A2	XLNET	BERT	70,87	4	94,05	74	84,65	9	93,98	17

TABLE 4 – Classement des modèles selon le test de Friedman - Corpus français

Selon la table 4, c'est CamemBERT en tant que modèle de génération du terme manquant à la réponse, associé à BERT et GPT en tant que modèles de langue, qui se classe en premières places selon la moyenne des rangs des quatre métriques d'évaluation, pour traiter les questions du corpus français. Comme attendu, le modèle GPT, malgré le fait qu'il est principalement entraîné sur un corpus de texte écrit en anglais, arrive quand même à prédire la meilleure structure de réponse pour du texte écrit en français. Ce modèle, dans sa deuxième version améliorée GPT2, arrive également à se hisser aux premiers rangs pour le corpus de questions anglais, associé à BERT comme modèle génératif du terme manquant.

La différence de performance selon l'architecture adoptée varie aussi selon la langue dans laquelle la question est écrite. En effet, nous remarquons que par rapport aux cinq premiers rangs, c'est l'architecture *A1* faisant évaluer les structures par le modèle de langue avant de générer le terme manquant, qui donne les meilleurs résultats pour le corpus français, alors que c'est la deuxième architecture *A2* qui est la plus pertinente pour les questions en anglais.

rang moyen	arch.	LM	modèle de génér.	%BleuS score	%BleuS rang	%MeteorS score	%MeteorS rang	%RougeS score	%RougeS rang	%BertS score	%BertS rang
1,25	A2	GPT2	BERT	77,82	1	94,41	1	93,48	1	97.30	2
2	A1	GPT	BERT	77,56	2	94,02	3	93,22	2	97,51	1
5	A2	XLM	BERT	76,90	5	94,14	2	92,17	7	97,16	6
8	A2	GPT	BERT	75,26	14	93,72	10	92,68	3	97,19	5
9,5	A1	XLM	BERT	76,71	7	93,92	4	91,63	12	96,79	15
12,75	A1	XLNET	BERT	74,96	15	93,77	7	91,3	17	96,8	12
16,5	A1	GPT2	BERT	75,75	10	93,49	21	91,11	24	96,87	11
17	A2	XLNET	BERT	74,19	17	93,57	15	91,2	19	96,78	17
20,5	A2	BERT	BERT	74,56	16	93,75	8	90,51	35	96,67	23
22	A1	BERT	BERT	75,37	12	93,41	25	91,12	22	96,59	29

TABLE 5 – Classement des modèles selon le test de Friedman - Corpus anglais

Il faut noter que dû à la spécificité de l'approche que nous proposons et qui fait usage des termes de la question pour générer une réponse en langage naturel, nous avons obtenu des scores relativement élevés. Nous pensons que la mesure de corrélation avec une évaluation humaine permettrait de déterminer la métrique d'évaluation la plus appropriée.

5 Conclusion

Nous avons présenté dans cet article une approche non-supervisée qui se base sur des modèles Transformer pour la génération de réponse en langage naturel dans le cadre des systèmes de question-réponse. L'évaluation que nous avons effectuée prouve que cette approche est prometteuse. Nous envisageons d'utiliser cette approche pour construire des corpus d'apprentissage de type question-réponse en langage naturel, qui permettraient d'entraîner des approches neuronales de type end-to-end.

Références

AGICHTEIN E. & GRAVANO L. (2000). Snowball : Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, p. 85–94.

BRILL E., DUMAIS S. & BANKO M. (2002). An analysis of the askmsr question-answering system. In *Proceedings of the ACL conference on Empirical methods in natural language processing*, p. 257–264 : Association for Computational Linguistics.

BRILL E., LIN J., BANKO M., DUMAIS S. & NG A. (2001). Data-intensive question answering. In *In Proceedings of the Tenth Text REtrieval Conference (TREC)*, p. 393–400.

CHOPRA S., AULI M. & RUSH A. M. (2016). Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 93–98.

CONNEAU A., KHANDELWAL K., GOYAL N., CHAUDHARY V., WENZEK G., GUZMÁN F., GRACE E., OTT M., ZETTLEMOYER L. & STOYANOV V. (2019). Unsupervised Cross-lingual Representation Learning at Scale. arXiv preprint : 1911.02116.

DEVLIN J., CHANG M.-W. C., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language under-standing. In *NAACL*, p. 4171–4186, Minneapolis : Association for Computational Linguistics.

DOS SANTOS C., TAN M., XIANG B. & ZHOU B. (2016). Attentive pooling networks. arXiv preprint : 1602.03609.

DOZAT T., QI P. & MANNING C. D. (2017). Stanford's Graph-based Neural Dependency Parser at the CoNLL 2017 Shared Task. In *Proceedings of the CoNLL 2017 Shared Task. Multilingual Parsing from Raw Text to Universal Dependencies*, p. 20–30, Vancouver, Canada : Association for Computational Linguistics.

GIRJU R. (2003). Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering-Volume 12*, p. 76–83 : Association for Computational Linguistics.

IIDA R., KRUENGKRAI C., ISHIDA R., TORISAWA K., OH J.-H. & KLOETZER J. (2019). Exploiting background knowledge in compact answer generation for why-questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, p. 142–151.

ISHIDA R., TORISAWA K., OH J.-H., IIDA R., KRUENGKRAI C. & KLOETZER J. (2018). Semi-distantly supervised neural model for generating compact answers to open-domain why questions. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

KRUENGKRAI C., TORISAWA K., HASHIMOTO C., KLOETZER J., OH J.-H. & TANAKA M. (2017). Improving event causality recognition with multiple background knowledge sources using multi-column convolutional neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*.

LAWRENCE S. & GILES C. L. (1998). Context and page analysis for improved web search. *IEEE Internet computing*, **2**(4), 38–46.

LE H., VIAL L., FREJ J., SEGONNE V., COAVOUX M., LECOUTEUX B., ALLAUZEN A., CRABBÉ B., BESACIER L. & SCHWAB D. (2020). FlauBERT : Unsupervised Language Model Pre-training for French. In *LREC 2020*. arXiv preprint : 1912.05372.

LIU Y., OTT M., GOYAL N., DU, JINGFEI ADN JOSHI M., CHEN D., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). RoBERTa : A Robustly Optimized BERT Pretraining Approach. arXiv preprint : 1907.11692.

MARTIN L., MULLER B., SUÁREZ P. J. O., DUPONT Y., ROMARY L., VILLEMONTE DE LA CLERGERIE É., SEDDAH D. & SAGOT B. (2019). CamemBERT : a Tasty French Language Model. arXiv preprint 1911.03894.

MIAO Y. & BLUNSOM P. (2016). Language as a latent variable : Discrete generative models for sentence compression. *EMNLP 2016*.

MILTON F. (1939). A correction : The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association. American Statistical Association*, **34**(205), 109.

NALLAPATI R., ZHOU B., DOS SANTOS C., GÜLÇEHRE Ç., XIANG B. *et al.* (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *CoNLL 2016*, p. 280–290.

NIVRE J., MARNEFFE M.-C. D., GINTER F., GOLDBERG Y., GOLDBERG Y., HAJIČ J., D. M. C., MCDONALD R., PETROV S., PYYSALO S., SILVEIRA N., TSARFATY R. & ZEMAN D. (2016). Universal Dependencies v1 : A Multilingual Treebank Collection. In *the tenth international conference on Language Resources and Evaluation*, p. 23–38, Portorož, Slovenia : European Language Resources Association.

OH J.-H., TORISAWA K., HASHIMOTO C., IIDA R., TANAKA M. & KLOETZER J. (2016). A semi-supervised learning approach to why-question answering. In *Thirtieth AAAI Conference on Artificial Intelligence*.

OH J.-H., TORISAWA K., HASHIMOTO C., SANO M., DE SAEGER S. & OHTAKE K. (2013). Why-question answering using intra-and inter-sentential causal relations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1733–1743.

RADFORD A., NARASIMHAN K., SALIMANS T. & SUTSKEVER I. (2018). Improving language understanding by generative pre-training. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

ROJAS BARAHONA L. M., BELLEC P., BESSET B., DOS SANTOS M., HEINECKE J., ASADULLAH M., LEBLOUCH O., LANCIEN J.-Y., DAMNATI G., MORY E. & HERLÉDAN F. (2019). Spoken Conversational Search for General Knowledge. In *SIGdial Meeting on Discourse and Dialogue*, p. 110–113, Stockholm : Association for Computational Linguistics.

RUSH A. M., CHOPRA S. & WESTON J. (2015). A neural attention model for abstractive sentence summarization. arXic preprint : 1509.00685.

SEDDAH D. & CANDITO M. (2016). Hard Time Parsing Questions : Building a QuestionBank for French. In *the tenth international conference on Language Resources and Evaluation*, Portorož, Slovenia : European Language Resources Association.

SEE A., LIU P. J. & MANNING C. D. (2017). Get to the point : Summarization with pointer-generator networks. arXiv preprint : 1704.04368.

SHARP R., SURDEANU M., JANSEN P., CLARK P. & HAMMOND M. (2016). Creating causal embeddings for question answering with minimal supervision. *EMNLP 2016*.

STRAKA M. (2018). UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In *Proceedings of the CoNLL 2018 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies*, p. 197–207, Brussels : Association for Computational Linguistics.

TAN M., DOS SANTOS C., XIANG B. & ZHOU B. (2016). Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 464–473.

USBECK R., NGOMO A.-C. N., HAARMANN B., KRITHARA A., RÖDER M. & NAPOLITANO G. (2017). 7th Open Challenge on Question Answering over Linked Data (QALD-7). In M. DRAGONI, M. SOLANKI & E. BLOMQVIST, Éds., *Semantic Web Challenges*, p. 59–69, Cham : Springer International Publishing.

VERBERNE S., VAN HALTEREN H., THEIJSSEN D., RAAIJMAKERS S. & BOVES L. (2011). Learning to rank for why-question answering. *Information Retrieval*, **14**(2), 107–132.

ZEMAN D., HAJIČ J., POPEL M., POTTHAST M., STRAKA M., GINTER F., NIVRE J. & PETROV S. (2018). CoNLL 2018 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies. In D. ZEMAN & J. HAJIČ, Éds., *Proceedings of the CoNLL 2018 Shared Task : Multilingual Parsing from Raw Text to Universal Dependencies*, p. 1–21, Brussels : Association for Computational Linguistics.