Transformer based Natural Language Generation for Question-Answering

Imen Akermi   Johannes Heinecke   Frédéric Herledan
Orange / DATA AI
22307 Lannion cedex, France
imen.elakermi@orange.com
johannes.heinecke@orange.com
frederic.herledan@orange.com

Abstract

This paper explores Natural Language Generation within the context of Question-Answering task. The several works addressing this task only focused on generating a short answer or a long text span that contains the answer, while reasoning over a Web page or processing structured data. Such answers' length are usually not appropriate as the answer tend to be perceived as too brief or too long to be read out loud by an intelligent assistant. In this work, we aim at generating a concise answer for a given question using an unsupervised approach that does not require annotated data. Tested over English and French datasets, the proposed approach shows very promising results.

1 Introduction

Question-Answering systems (QAS) aim at analyzing and processing user questions in order to provide relevant answers (Hirschman and Gaizauskas, 2001). The recent popularity of intelligent assistants has increased the interest in QAS which have become a key component of "Human-Machine" exchanges since they allow users to have instant answers to their questions in natural language using their own terminology without having to go through a long list of documents to find the appropriate answers.

Most of the existing research work focuses on the major complexity of these systems residing in the processing and interpretation of the question that expresses the user's need for information, without considering the representation of the answer itself. Usually, the answer is either represented by a short set of terms answering exactly the question (case of QAS which extract answers from structured data), or by a text span extracted from a document which, besides the exact answer, can integrate other unnecessary information that are not relevant to the context of the question asked. The following presents two answers for *Who is the thesis supervisor of Albert Einstein?* possibly generated by two systems :

S1 | *Alfred Kleiner*

S2 | *Albert Einstein is a German-born theoretical physicist who developed the theory of relativity, one of the two pillars of modern physics.*

Given the specificity of QAS which extract answers from structured data, users generally receive only a short and limited answer to their questions as illustrated by the example above. This type of answer representation might not meet the user expectations. Indeed, the type of answer given by the first system can be perceived as too brief not recalling the context of the question. The second system returns a passage which contains information that are out of the question's scope and might be deemed by the user as irrelevant.

It is within this framework that we propose in this article an approach which allows to generate a concise answer in natural language (e.g. *The thesis superviser of Albert Einstein was Alfred Kleiner*) that shows very promising results tested over French and English questions. This approach is a component of a QAS that we proposed in Rojas Barahona et al. (2019) and that we will briefly present in this article.

In what follows, we detail in section 3 the approach we propose for answer generation in Natural Language and we briefly discuss the QAS developed. We present in section 4 the experiments that we have conducted to evaluate this approach.

2 Related Work

The huge amount of information available nowadays makes the task of retrieving relevant informa-

Proceedings of The 13th International Conference on Natural Language Generation, pages 349–359,
Dublin, Ireland, 15-18 December, 2020. ©2020 Association for Computational Linguistics

tion complex and time consuming. This complexity has prompted the development of QAS which help spare the user the search and the information filtering tasks, as it is often the case with search engines, and directly return the exact answer to a question asked in natural language.

The QAS cover mainly three tasks: question analysis, information retrieval and answer extraction (Lopez et al., 2011). These tasks have been tackled in different ways, considering the knowledge bases used, the types of questions addressed (Iida et al., 2019; Zayaraz et al., 2015; Dwivedi and Singh, 2013; Lopez et al., 2011) and the way in which the answer is presented. In this article, we particularly focus on the answer generation process.

We generally notice two forms of representation addressed in literature. The answer can take the form of a paragraph selected from a set of text passages retrieved from the web (Asai et al., 2018; Du and Cardie, 2018; Wang and Jiang, 2016; Wang et al., 2017; Oh et al., 2016), as it can also be the exact answer to the question extracted from a knowledge base (Wu et al., 2003; Bhaskar et al., 2013; Le et al., 2016).

Despite the abundance of work in the field of QAS, the answers generation issue has received little attention. A first approach indirectly addressing this task has been proposed in Brill et al. (2001, 2002). Indeed, the authors aimed at diversifying the possible answer patterns by permuting the question's words in order to maximise the number of retrieved documents that may contain the answer to the given question. Another answer representation approach based on rephrasing rules has also been proposed in Agichtein and Gravano (2000); Lawrence and Giles (1998) within the context of query expansion task for document retrieval and not purposely for the question-answering task.

The few works that have considered this task within the QAS framework have approached it from a text summary generation perspective (Ishida et al., 2018; Iida et al., 2019; Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016; Miao and Blunsom, 2016; See et al., 2017; Oh et al., 2016; Sharp et al., 2016; Tan et al., 2016; dos Santos et al., 2016). These works consist in generating a summary of a single or various text spans that contain the answer to a question. Most of these works have only considered causality questions like the ones starting with "*why*" and whose answers are paragraphs. To make these answers more concise, the extracted paragraphs are summed up.

Other approaches (Kruengkrai et al., 2017; Girju, 2003; Verberne et al., 2011; Oh et al., 2013) have explored this task as a classification problem that consists in predicting whether a text passage can be considered as an answer to a given question.

It should be noted that these approaches only intend to diversify as much as possible the answer representation patterns to a given question in order to increase the probability of extracting the correct answer from the Web and do not focus on the answer's representation itself. It should also be noted that these approaches are only applicable for QAS which extract answers as a text snippet and cannot be applied to short answers usually extracted from knowledge bases. The work presented in Pal et al. (2019) tried to tackle this issue by proposing a supervised approach that was trained on a small dataset whose questions/answers pairs were extracted from machine comprehension datasets and augmented manually which make generalization and capturing variation very limited.

Our answer generation approach differs from these works as it is unsupervised, can be adapted to any type of factual question (except for *why*) and is based only on easily accessible and unannotated data. Indeed, we build upon the intuitive hypothesis that a concise answer and easily pronounced by an intelligent assistant can in fact consist of a reformulation of the question asked. This approach is a part of a QAS that we have developed in Rojas Barahona et al. (2019) that extracts the answer to a question from structured data.

In what follows, we detail in section 3 the approach we propose for answer generation in Natural Language and we briefly discuss the QAS developed. We present in section 4 the experiments that we have conducted to evaluate this approach. and we conclude in section 5 with the limitations noted and the perspectives considered.

3 NLG Approach for Answer Generation

The answer generation approach proposed is a component of a system which was developed in Rojas Barahona et al. (2019) and which consists in a spoken conversational question-answering system which analyses and translates a question in natural language (French or English) in a formal representation that is transformed into a Sparql query[1].

[1] https://www.w3.org/TR/sparql11-overview/

The Sparql query helps extracting the answer to the given question from an RDF knowledge base, in our case Wikidata[2]. The extracted answer takes the form of a list of URIs or values.

Although the QAS that we have developed (Rojas Barahona et al., 2019) is able to find the correct answer to a question, we have noticed that its short representation is not user-friendly. Therefore, we propose an unsupervised approach which integrates the use of Transformer models such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2018). The choice of an unsupervised approach arises from the fact that there is no available training dataset associating a question with an exhaustive and concise answer at the same time. such dataset could have helped use an End-to-End learning neural architecture that can generate an elaborated answer to a question.

This approach builds upon the fact that we have already extracted the short answer to a given question and assumes that a user-friendly answer can consist in rephrasing the question words along with the short answer. This approach is composed of two fundamental phases: The dependency analysis of the input question and the answer generation using Transformer models.

3.1 Dependency parsing

For the dependency analysis, we use an extended version of UDPipeFuture (Straka, 2018) which showed its state of the art performance by becoming first in terms of the Morphology-aware Labeled Attachment Score (MLAS)[3] metric at the CoNLL Shared Task of dependency parsing in 2018 (Zeman et al., 2018). UDPipeFuture is a POS tagger and graph parser based dependency parser using a BiLSTM, inspired by Dozat et al. (2017).

Our modification consisted in adding several contextual word embeddings (with respect to the language). In order to find the best configuration we experimented with models like multilingual BERT (Devlin et al., 2019), XLM-R (Conneau et al., 2019) (for both, English and French), RoBERTA (Liu et al., 2019) (for English), FlauBERT (Le et al., 2020) and CamemBERT (Martin et al., 2019) (for French) during the training of the treebanks French-GSD and English-EWT[4], of the Universal Dependencies project (UD) (Nivre et al., 2016)[5]. Adding contextual word embedding increases significantly the results for all metrics, LAS, CLAS and MLAS (cf. table 1). This is the case for all languages (of the CoNLL shared task), where language specific contextual embeddings or multingual ones (as BERT or XLM-R) improved parsing (Heinecke, 2020)

French (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**

English (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: Dependency Analysis for English and French (UD v2.2) using different contextual word embeddings, best results in bold

In order to parse simple, quiz-like questions, the training corpora of the two UD treebanks are not appropriate (enough), since both treebanks do not contain many questions, if at all[6].

An explanation for bad performance on questions of parser models trained on standard UD is the fact, that in both languages, the syntax of questions differs from the syntax of declarative sentences: apart from *wh* question words, in English the *to do* periphrasis is nearly always used in questions. In French, subject and direct objects can be inversed and the *est-ce que* construction appears frequently. Both, the English *to do* periphrasis and the French *est-ce que* construction are absent in declarative sentences. Table 2 shows the (much lower) results when parsing questions using models trained only on the standard UD treebanks.

In order to get a better analysis, we decided to

[2] https://www.wikidata.org/

[3] MLAS is a metric which takes into account POS tags and morphological features. It is inspired by the Content-Word Labeled Attachment Score (CLAS, Nivre and Fang (2017) which differentiates between content word and function words. Both are derived from the standard Labeled Attachment Score (LAS) metric.

[4] As for the Shared Task CoNLL 2018, we use version 2.2 to be able to compare with the official results

[5] https://universaldependencies.org/

[6] At least for French a question treebank exists within the UD project (French-FQB, Seddah and Candito (2016)). However its questions are rather long and literary, not like thoses used in quizzes.

French (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

English (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: Dependency Analysis of questions using models trained on the standard UD treebanks

annotate additional sentences (quiz-like questions) and add this data to the basic treebanks.

For English we annotated 309 questions (plus 91 questions for validation) from the QALD7 (Usbeck et al., 2017) and QALD8 corpora[7]. For French we translated the QALD7 questions into French and formulated others ourselves (276 train, 66 validation). For the annotations we followed the general UD guidelines[8] as well as the treebank specific guidelines of En-EWT and Fr-GSD.

As table 3 shows, the quality of the dependency analysis improves considerably. The contextual word embeddings CamemBERT (for French) and BERT (English) have the biggest impact.

French (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

English (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: Dependency analysis of questions using models trained on enriched UD treebanks

We rely on the UdpipeFuture version which we have improved with BERT (for English)/CamemBERT (for French) and which gives the best results in terms of dependency analysis, in order to proceed with the partitioning of the question into textual fragments (also called *chunks*): $Q = \{c_1, c_2, \ldots, c_n\}$.

If we take the example of the question *What is the political party of the mayor of Paris?*, the set of textual fragments would be $Q = \{$*What, is, the political party of the mayor of Paris* $\}$.

3.2 Answer generation

During this phase, we first carry out a first test of the set Q to check whether the text fragment which contains a question marker (exp: *what*, *when*, *who* etc.) represents the subject *nsubj* in the analysed question. If so, we simply replace that text fragment with the answer we identified earlier. Let us take the previous example *What is the political party of the mayor of Paris?*, the system automatically detects that the text fragment containing the question marker *What* represents the subject and will therefore be replaced directly by the exact answer *The Socialist Party*. Therefore, the concise answer generated will be *The Socialist Party is the political party of the mayor of Paris*.

Otherwise, we remove the text fragment containing the question marker that we detected and we add the short answer R to Q: $Q = \{c_1, c_2, \ldots, c_{n-1}, R\}$

Using the text fragments set Q, we proceed with a permutation based generation of all possible answer structures that can form the sentence answering the question asked:

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

These structures will be evaluated by a Language Model (LM) based on Transformer models which will extract the most probable sequence of text fragments that can account for the answer to be sent to the user:

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

Once the best structure is identified, we initiate the generation process of possible missing words. Indeed, we suppose that there could be some terms which do not necessarily appear in the question or in the short answer but which are, on the other hand, necessary to the generation of a correct grammatical structure of the final answer.

This process requires that we set two parameters, the number of possible missing words and their

[7] https://github.com/ag-sc/QALD

[8] https://universaldependencies.org/guidelines.html

positions within the selected structure. In this paper, we experiment the assumption that one word could be missing and that it is located before the short answer within the identified structure, as it could be the case for a missing article (*the, a*, etc.) or a preposition (*in, at*, etc.) for example.

Therefore, to predict this missing word, we use BERT as the generation model (GM) for its ability to capture bidirectionally the context of a given word within a sentence. In case when BERT returns a non-alphabetic character sequence, we assume that the optimal structure, as predicted by the LM, does not need to be completed by an additional word. The following example illustrates the different steps of the proposed approach:

Question: *When did princess Diana die?*

1. Question parsing and answer extraction using the system proposed in Rojas Barahona et al. (2019):
   short_answer = {*August 31, 1997*}

2. Chunking the question into text fragments using the UDPipe based dependency analysis:
   Q={*When*, *did die*, *princess Diana*}

3. Removing question marker fragment (*when*) and updating the verb tense and form using a rule-based approach that we have defined:
   Q={*died*, *princess Diana*}

4. Adding the short answer:
   Q={*died*; *princess Diana*; *August 31, 1997*}

5. Generating the set of possible answer structures *S*:
   S={*died princess Diana August 31, 1997*;
   August 31, 1997 died princess Diana;
   princess Diana died August 31, 1997;
   ...}

6. Evaluating the different answer structures using a LM: $p(structure*) = \text{argmax}_{s_i \in S}\, p(s_i)$:
   $structure^*$ = *princess Diana died August 31, 1997*

7. Generating possible missing word for *structure** with BERT:
   Princess Diana died [missing_word] August 31, 1997
   (missing_word = *on*)

Answer: *Princess Diana died on August 31, 1997.*

4 Experiments and Evaluation

The existing QAS test sets are more tailored to systems which generate the exact short answer to a question or more focused on the *Machine Reading Comprehension* task where the answer consists of a text passage from a document containing the short answer. Therefore, we have created a dataset which maps questions extracted from the QALD-7 challenge dataset (Usbeck et al., 2017) with natural language answers which were defined by a linguist and which we individually reviewed. This dataset called QUEREO consists of 150 questions with the short answers extracted by the QAS that we described above. We denote an average of three possible gold sanswers in natural language for each question. French and English versions were created for this dataset.

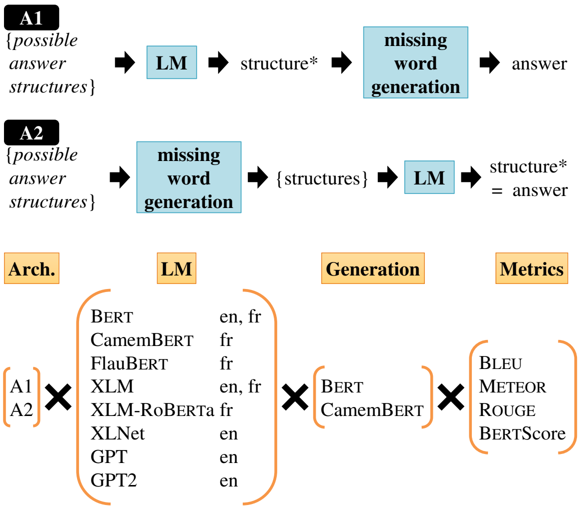


Figure 1: Experiment framework

As illustrated in figure 1, two possible architectures of the approach proposed for answer generation have been evaluated. The first architecture *A1* consists in generating all possible answer structures in order to have them evaluated afterwards by a LM which will identify the optimal answer structure to which we generate possible missing words. Architecture *A2* starts with generating missing words for each structure in *S* which will then be evaluated by the LM. In this paper, we assume that there is only one missing word per structure.

To evaluate the proposed approach, we have referred to standard metrics defined for NLG tasks such as Automatic Translation and Summarization, as they allow to assess to what extent a generated sentence is similar to the gold sentence. We consider three N-gram metrics (BLEU, METEOR and ROUGE) and the BERT score metric which exploits the pre-trained embeddings of BERT to calculate the similarity between the answer generated and the gold answer. To be able to compare the different configurations of the approach, we refer to Friedman's test (Milton, 1939) which allows to detect the performance variation of different configurations of a model evaluated by several metrics based on the average ranks.

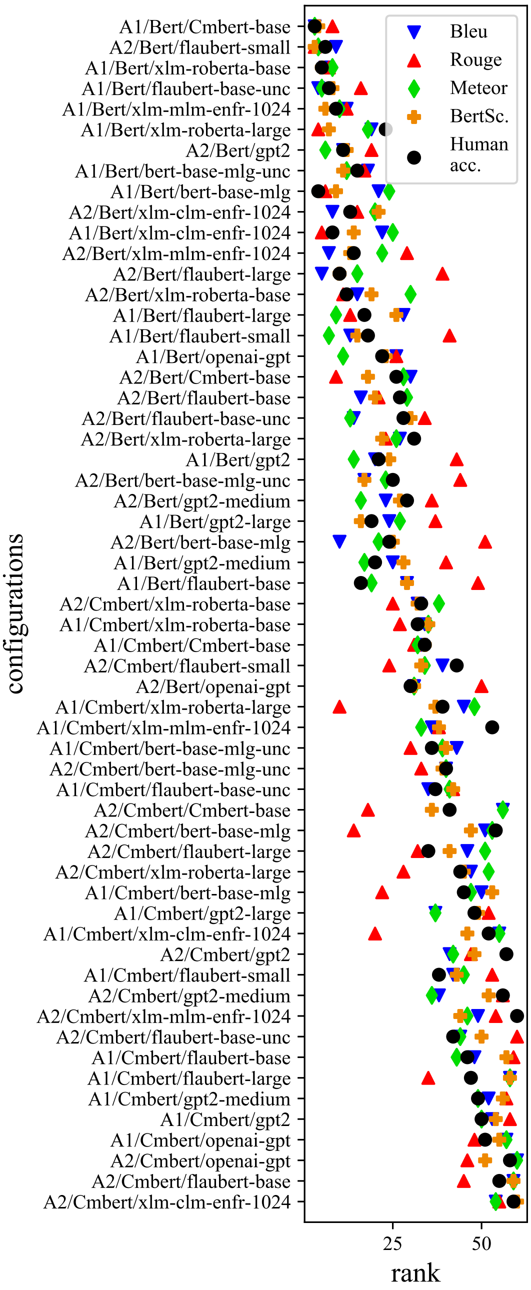
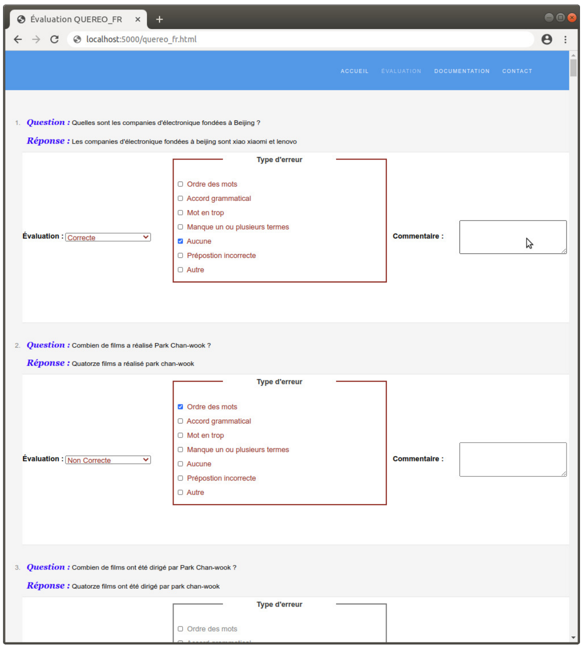


Figure 2: Correlation assessment between human evaluation and the Bleu, Meteor, Rouge and Bert scores - French Q/A ("CmBert" stands for CamemBERT)



1. ***Question :*** Quelles sont les companies d'électronique fondées à Beijing ?

Réponse : Les companies d'électronique fondées à beijing sont xiao xiaomi et lenovo

Évaluation : Correcte

Type d'erreur

- ☐ Ordre des mots
- ☐ Accord grammatical
- ☐ Mot en trop
- ☐ Manque un ou plusieurs termes
- ☑ Aucune
- ☐ Prépostion incorrecte
- ☐ Autre

Commentaire :

2. ***Question :*** Combien de films a réalisé Park Chan-wook ?

Réponse : Quatorze films a réalisé park chan-wook

Évaluation : Non Correcte

Type d'erreur

- ☑ Ordre des mots
- ☐ Accord grammatical
- ☐ Mot en trop
- ☐ Manque un ou plusieurs termes
- ☐ Aucune
- ☐ Prépostion incorrecte
- ☐ Autre

Commentaire :

3. ***Question :*** Combien de films ont été dirigé par Park Chan-wook ?

Réponse : Quatorze films ont été dirigé par park chan-wook

Type d'erreur

- ☐ Ordre des mots

Figure 3: Screenshot of the evaluation tool

We also conducted a human evaluation study for the French and the English versions of the dataset, in which we asked 20 native speakers participants to evaluate the relevance of a generated answer (*correct* or *not correct*) regarding a given question while indicating the type of errors depicted (*grammar*, *wrong preposition*, *word order*, *extra word(s)*, etc). Figure 3 presents the evaluation framework that we have implemented and provided to the participants. The results of each participant are saved in a json-file (figure 4). The inter-agreement rate between participants reached 70% which indicates a substantial agreement. Through the human evaluation study, we wanted to explore to what extent the standard metrics are reliable to assess NLG approaches within the context of question-answering systems.

Table 4 (French dataset) represents the obtained results for the first three best models according to the human evaluation ranking and the Friedman test ranking. We indicate between brackets each model's rank according to the metric used.

We note that the highest human accuracy score for French of about 85% was scored with the first architecture coupled with BERT as the generation model (GM) and CamemBERT as the language model (LM). We also notice that the architecture *A1*, which considers the LM assessment of the structure before generating missing words, performs better. Surprisingly, as a generative model, the multi-

Hum.	Frm.	Arch.	GM	LM	H. Acc	BLEU		METEOR		ROUGE		BERTS	
rank	rank				score	score	rank	score	rank	score	rank	score	rank
1	**1**	**A1**	**BT**	**CmBt**	**84.85**	**86.28**	**[1]**	**96.76**	**[1]**	**93.69**	**[6]**	**97.89**	**[2]**
2	2	A2	BT	FBT-s-c	84.09	85.87	[7]	96.75	[2]	94.22	[1]	97.96	[1]
2	3	A1	BT	XRob	84.09	85.93	[4]	96.63	[6]	93.79	[5]	97.88	[3]
2	9	A1	BT	BT-ml-c	84.09	85.01	[19]	96.52	[22]	93.81	[4]	97.79	[7]
5	4	A1	BT	FBT-b-uc	83.33	86.17	[2]	96.72	[3]	93.56	[14]	97.81	[6]
5	5	A1	BT	mlm-1024	83.33	85.39	[10]	96.60	[8]	93.61	[10]	97.83	[4]
5	6	A2	BT	GPT2	83.33	85.46	[9]	96.67	[4]	93.48	[17]	97.76	[10]
5	10	A2	BT	clm-1024	83.33	85.89	[6]	96.55	[18]	93.57	[13]	97.71	[19]
5	11	A1	BT	clm-1024	83.33	84.99	[20]	96.52	[23]	93.87	[3]	97.76	[12]
5	12	A2	BT	FBT-l-c	83.33	86.15	[3]	96.57	[13]	93.14	[37]	97.79	[8]
5	13	A2	BT	mlm-1024	83.33	85.90	[5]	96.54	[20]	93.30	[27]	97.76	[11]
5	14	A2	BT	XRob	83.33	85.32	[13]	96.46	[28]	93.63	[9]	97.71	[17]

Table 4: Model ranking for French dataset according to the human evaluation study (best in bold) and the Friedman test (best in yellow). "BT" in Column GM stands for BERT-base-multilingual-cased. In column LM we use "CmBT" for CamemBERT-base, "BT-ml-c" for BERT-base-multilingual-cased, "XRob" for XLM-RoBERTa-base, "FBT-s-c" for FlauBERT-small-cased, "FBT-b-uc" for FlauBERT-base-uncased and "clm-1024" for XLM-clm-enfr-1024

lingual BERT model predicts missing words better than CamemBERT for French sentences. These findings are also confirmed by the Friedman test where we can clearly see that the first ranked configuration maps the best configuration selected according to the human accuracy, with a very slight difference for the other four configurations. Let us see if that means that the four metrics are correlated with the human accuracy. According to table 6 which presents the Pearson correlation (Benesty et al., 2009) of the human accuracy with the four metrics and to figure 2 which illustrates the ranking given by each evaluation metric along with the human judgement for each configuration (i.e. *configuration = GM × architecture × LM*) tested, we can clearly see that the human evaluation results are positively and strongly correlated with the BLEU, the METEOR and the BERT scores. These metrics are practically matching the human ranking and thus are obviously able to identify which configuration gives better results. The rouge metric, used for French question/answer evaluation, is moderately correlated with the human evaluation which means that we should not only rely on this metric when assessing such task. On the other hand, when the ROUGE metric is considered with the other metrics, it helps to get closer to the human judgement.

Table 5 presents the results for the English dataset and shows that the best accuracy scored is about 72% with *A1*, BERT as the generative model and the Generative Pretrained Transformer (GPT) as the language model. According to the first three configurations, architecture *A2* prevails and the GPT transformer takes over the other language models. These results are also confirmed by the Friedman test with a very slight difference on the ranking and also upheld with the correlation scores between the human assessment and each of the four metrics as shown by figure 5 and table 6.

These findings mean that we actually can rely on the use of these standard metrics to evaluate the answer generation task for question-answering.

We also tried to analyse the errors indicated by the participants. As we can note from figure 6, the most common error reported for both English and French datasets is the word order which sheds the light on a problem related to the language model assessment phase. The second most reported error addresses the generation process, whether to indicate that there are one or more missing words within the answer (French) or the presence of some odd words (English).

When trying to get an insight on the answers generated by the current intelligent systems such as Google assistant and Alexa, we noted that these systems are very accurate when extracting the correct answer to a question and can sometimes generate user-friendly answers that help recall the question context, specially with Alexa. However, we noticed that most of the answers generated by these systems are more verbose than necessary, we also found out that when addressing yes/no questions, these systems generally settle for just a *yes* or *no* without elaborating, or, on the other hand, present a text span extracted from a Web page and let the user guess the answer. Let us take for example the following question *Was US president Jackson involved in a war?*

Hum.	Frm.	Arch.	GM	LM	H. Acc	BLEU		METEOR		ROUGE		BERTS	
rank	rank				score	score	rank	score	rank	score	rank	score	rank
1	**1**	**A2**	**BT-ml**	**GPT**	**72.36**	**78.25**	**[2]**	**94.63**	**[1]**	**92.83**	**[2]**	**97.21**	**[3]**
1	2	A1	BT-ml	GPT	72.36	78.25	[1]	94.51	[2]	92.53	[10]	97.23	[2]
3	2	A1	BT	GPT	71.55	77.12	[6]	94.45	[3]	92.80	[5]	97.32	[1]
3	4	A2	BT	GPT2	71.55	76.98	[7]	94.40	[7]	92.86	[1]	97.17	[5]
3	5	A2	BT-ml	GPT2	71.55	77.53	[4]	94.39	[8]	92.82	[4]	97.14	[6]
3	6	A2	BT-ml	GPT2-l	71.55	77.85	[3]	94.41	[5]	92.65	[7]	97.07	[10]
3	7	A2	BT-ml	GPT2-m	71.55	77.42	[5]	94.40	[6]	92.58	[9]	97.10	[9]
3	7	A2	BT	GPT2-m	71.55	75.96	[13]	94.41	[4]	92.60	[8]	97.18	[4]
3	10	A2	BT	GPT	71.55	76.28	[11]	94.30	[9]	92.76	[6]	97.14	[7]
10	7	A2	BT	GPT2-l	70.73	76.74	[8]	94.26	[10]	92.83	[3]	97.14	[8]
10	30	A1	BT-ml	BT-b-uc	70.73	74.85	[30]	93.94	[31]	90.86	[26]	96.61	[28]

Table 5: Model ranking for English dataset according to the human evaluation study (best in bold) and the Friedman ranking (best in yellow). In Column GM we use "BT-ml" for BERT-base-multilingual-cased and "BT" for BERT-large-cased. In column LM "GPT" stands for for OpenAI-GPT, "GPT2-l" for GPT2-large, "GPT2-m" for GPT2-medium, "GPT2" for GPT2, "BT-b-uc" for BERT-base-uncased, "mlm-2048" for XLM-mlm-en-2048 and "BT-l-c" for BERT-large-cased.

```
[
  {
    "ID":
       "quereo_5.4",
    "QUESTION":
       "Quelles sont les companies
        d'électronique fondées
        à Beijing ?",
    "SHORT_ANSWER":
        [ "Xiaomi",  "Lenovo" ],
    "GENERATED_ANSWER":
       "Les companies d'électronique
        fondées à  beijing sont xiao
        xiaomi et lenovo",
    "MISSING_WORD":
       "Xiao",
    "EVALUATION":
       "correcte",
    "ERROR":
        [ "aucun" ],
    "COMMENT": ""
  },
  {
    "ID":
       "quereo_8.8",
    "QUESTION":
       "Combien de films
        a réalisé Park Chan-wook ?",
    "SHORT_ANSWER":
        [ "quatorze" ],
    "GENERATED_ANSWER":
       "Quatorze films a réalisé
        park chan-wook",
    "MISSING_WORD":
       ".",
    "EVALUATION":
       "incorrecte",
    "ERROR":
      [ "ordre",  "accord" ],
    "COMMENT": ""
  },
  ...
]
```

Figure 4: Extract of a human evaluation result

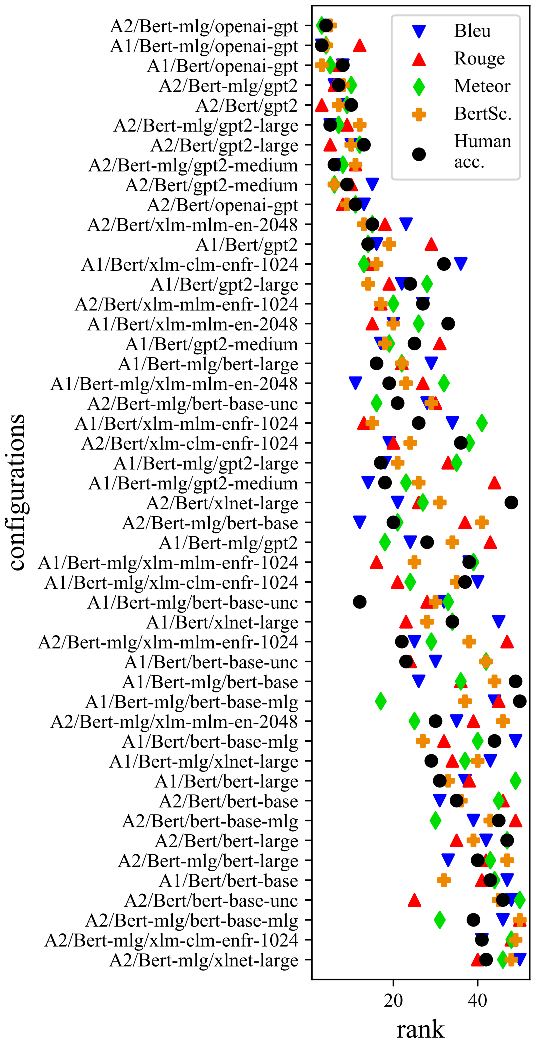


Figure 5: Correlation assessment between human evaluation and the Bleu, Meteor, rouge and Bert scores - English Q/A

Metrics	Pearson Correlation	
	QUEREO-fr	QUEREO-en
BLEU	98%	85%
METEOR	99%	80%
ROUGE	46%	83%
BERT-score	97%	88%

Table 6: Pearson Correlation of the four metrics with the human evaluation/judgement

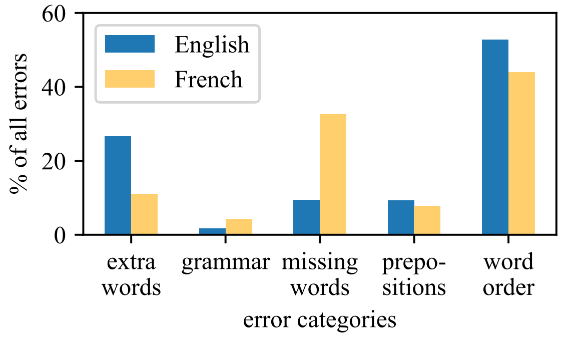


Figure 6: Distribution of generation errors

> *Andrew Jackson, who served as a major general in the War of 1812, commanded U.S. forces in a five-month campaign against the Creek Indians, allies of the British.*

> *Here's something I found on the Web. According to constitutioncenter.org: After the War of 1812, Jackson led military forces against the Indians and was involved in treaties that led to the relocation of Indians.*

The user has to focus on the returned text fragment in order to guess that the answer to his question is actually *yes*. This issue was particularly noted when addressing French questions. If we also take the example *How many grandchildren did Jacques Cousteau have ?* the two systems answer as follows:

> *Fabien Cousteau, Alexandra Cousteau, Philippe Cousteau Jr., Céline Cousteau.*

> *Jacques Cousteau's grandchildren were Philippe Cousteau Jr., Alexandra ousteau, Céline Cousteau, and Fabien Cousteau*

However, the user is not asking about the names of Cousteau's grand-children and has to guess by himself that the answer for this question is *four*. A more accurate answer should indicate the exact answer to the question and then elaborate *Jacques Cousteau had four grand-children.* But these systems perform better in case when the terms employed in the question are not necessarily relevant to the answer. If we take the example of the question *who is the wife of Lance Bass*, the approach that we propose will generate *The wife of Lance Bass is Michael Turchin.* As we can note the answer generated was not adapted to the actual answer, while the other systems are able to detect such nuance:

> *Lance Bass is married to Michael Turchin. They have been married since 2014.*

This issue has still to be addressed.

5 Conclusion and perspectives

We have put forward, in this paper, an approach for Natural Language Generation within the framework of the question-answering task that considers dependency analysis and probability distribution of words sequences. This approach takes part of a question/answering system in order to help generate a user-friendly answer rather than a short one. The results obtained through a human evaluation and standard metrics tested over French and English questions are very promising and shows a good correlation with human judgement. However, we intend to put more emphasis on the Language Model choice as reported by the human study and consider the generation of more than one missing word within the answer.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94.

Akari Asai, Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2018. Multilingual extractive reading comprehension by runtime machine translation. https://arxiv.org/abs/1809.03275.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. *Pearson Correlation Coefficient*, pages 1–4. Springer Berlin Heidelberg, Berlin, Heidelberg.

Pinaki Bhaskar, Somnath Banerjee, Partha Pakray, Samadrita Banerjee, Sivaji Bandyopadhyay, and Alexander Gelbukh. 2013. A hybrid question answering system for Multiple Choice Question

(MCQ). In *Question Answering for Machine Reading Evaluation (QA4MRE) at CLEF 2013*.

Eric Brill, Susan Dumais, and Michele Banko. 2002. An analysis of the AskMSR question-answering system. In *EMNLP 2002*, pages 257–264. ACL.

Eric Brill, Jimmy Lin, Michele Banko, Susan Dumais, and Andrew Ng. 2001. Data-intensive question answering. In *TREC 2001*, pages 393–400.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *NAACL 2016*, pages 93–98.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grace, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised Cross-lingual Representation Learning at Scale. https://arxiv.org/abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, Minneapolis. ACL.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's Graph-based Neural Dependency Parser at the CoNLL 2017 Shared Task. In *CoNLL 2017 Shared Task. Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. ACL.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from Wikipedia. In *ACL 2018*, pages 1907–1917, Melbourne, Australia. ACL.

Sanjay K. Dwivedi and Vaishali Singh. 2013. Research and reviews in question answering system. *Procedia Technology*, 10:417–424.

Roxana Girju. 2003. Automatic detection of causal relations for question answering. In *ACL 2003*, pages 76–83. ACL.

Johannes Heinecke. 2020. Hybrid enhanced Universal Dependencies parsing. In *International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 174–180, Online. ACL.

Lynette Hirschman and Robert Gaizauskas. 2001. Natural language question answering: the view from here. *natural language engineering*, 7(4):275–300.

Ryu Iida, Canasai Kruengkrai, Ryo Ishida, Kentaro Torisawa, Jong-Hoon Oh, and Julien Kloetzer. 2019. Exploiting background knowledge in compact answer generation for why-questions. In *AAI Conference on Artificial Intelligence*, volume 33, pages 142–151.

Ryo Ishida, Kentaro Torisawa, Jong-Hoon Oh, Ryu Iida, Canasai Kruengkrai, and Julien Kloetzer. 2018. Semi-distantly supervised neural model for generating compact answers to open-domain why questions. In *32nd AAAI Conference on Artificial Intelligence*.

Canasai Kruengkrai, Kentaro Torisawa, Chikara Hashimoto, Julien Kloetzer, Jong-Hoon Oh, and Masahiro Tanaka. 2017. Improving event causality recognition with multiple background knowledge sources using multi-column convolutional neural networks. In *31st AAAI Conference on Artificial Intelligence*.

Steve Lawrence and C. Lee Giles. 1998. Context and page analysis for improved web search. *IEEE Internet computing*, 2(4):38–46.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. FlauBERT: Unsupervised Language Model Pre-training for French. In *LREC 2020*.

Juan Le, Chunxia Zhang, and Zhendong Niu. 2016. Answer extraction based on merging score strategy of hot terms. *Chinese Journal of Electronics*, 25(4):614–620.

Yinhan Liu, Myle Ott, Naman Goyal, Mandar Du, Jingfei adn Joshi, Danqi Chen, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. https://arxiv.org/abs/1907.11692.

Vanessa Lopez, Victoria Uren, Marta Sabou, and Enrico Motta. 2011. Is question answering fit for the semantic web?: a survey. *Semantic Web*, 2(2):125–155.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2019. CamemBERT: a Tasty French Language Model. https://arxiv.org/abs/1911.03894.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. *EMNLP 2016*.

Friedman Milton. 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*, 34(205):109.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *CoNLL 2016*, pages 280–290.

Joakim Nivre and Chiao-Ting Fang. 2017. Universal Dependency Evaluation. In *NoDaLiDa 2017 Workshop on Universal Dependencies*, pages 86–95, Göteborg.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Yoav Goldberg, Jan Hajič, Manning Christopher D., Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *10th LREC*, pages 23–38, Portorož, Slovenia. ELRA.

Jong-Hoon Oh, Kentaro Torisawa, Chikara Hashimoto, Ryu Iida, Masahiro Tanaka, and Julien Kloetzer. 2016. A semi-supervised learning approach to why-question answering. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Jong-Hoon Oh, Kentaro Torisawa, Chikara Hashimoto, Motoki Sano, Stijn De Saeger, and Kiyonori Ohtake. 2013. Why-question answering using intra-and inter-sentential causal relations. In *ACL 2013*, pages 1733–1743.

Vaishali Pal, Manish Shrivastava, and Irshad Bhat. 2019. Answering naturally: Factoid to full length answer generation. In *2nd Workshop on New Frontiers in Summarization*, pages 1–9, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Lina M. Rojas Barahona, Pascal Bellec, Benoît Besset, Martinho Dos Santos, Johannes Heinecke, Munshi Asadullah, Olivier Leblouch, Jean-Yves Lancien, Géraldine Damnati, Emmanuel Mory, and Frédéric Herlédan. 2019. Spoken Conversational Search for General Knowledge. In *SIGdial Meeting on Discourse and Dialogue*, pages 110–113, Stockholm. ACL.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. https://arxiv.org/abs/1509.00685.

Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. https://arxiv.org/abs/1602.03609.

Djamé Seddah and Marie Candito. 2016. Hard Time Parsing Questions: Building a QuestionBank for French. In *10th LREC*, Portorož, Slovenia. ELRA.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. https://arxiv.org/abs/1704.04368.

Rebecca Sharp, Mihai Surdeanu, Peter Jansen, Peter Clark, and Michael Hammond. 2016. Creating causal embeddings for question answering with minimal supervision. *EMNLP 2016*.

Milan Straka. 2018. UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In *CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels. ACL.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *ACL 2016*, pages 464–473.

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Bastian Haarmann, Anastasia Krithara, Michael Röder, and Giulio Napolitano. 2017. 7th Open Challenge on Question Answering over Linked Data (QALD-7). In *Semantic Web Challenges*, pages 59–69, Cham. Springer International Publishing.

Suzan Verberne, Hans van Halteren, Daphne Theijssen, Stephan Raaijmakers, and Lou Boves. 2011. Learning to rank for why-question answering. *Information Retrieval*, 14(2):107–132.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. https://arxiv.org/abs/1608.07905.

Tong Wang, Xingdi Yuan, and Adam Trischler. 2017. A joint model for question answering and question generation. https://arxiv.org/abs/1706.01450.

Min Wu, Xiaoyu Zheng, Michelle Duan, Ting Liu, Tomek Strzalkowski, and S Albany. 2003. Question answering by pattern matching, web-proofing, semantic form proofing. In *TREC 2003*, pages 500–255.

Godandapani Zayaraz et al. 2015. Concept relation extraction using naïve bayes classifier for ontology-based question answering systems. *Journal of King Saud University-Computer and Information Sciences*, 27(1):13–24.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels. ACL.