A linguistically-motivated evaluation methodology for unraveling model's abilities in reading comprehension tasks

Elie Antoine[1] , **Frédéric Béchet**[1, 4] , **Géraldine Damnati**[2] , **Philippe Langlais**[3]
[1]CNRS, LIS, Aix-Marseille Université, France {first.last}@lis-lab.fr
[2]Orange Innovation, DATA&AI, Lannion, France {first.last}@orange.com
[3]RALI, DIRO, Université de Montréal, Canada felipe@iro.umontreal.ca
[4]International Laboratory on Learning Systems (ILLS - IRL CNRS), Montreal

Abstract

We introduce an evaluation methodology for reading comprehension tasks based on the intuition that certain examples, by the virtue of their linguistic complexity, consistently yield lower scores regardless of model size or architecture. We capitalize on semantic frame annotation for characterizing this complexity, and study seven complexity factors that may account for model's difficulty. We first deploy this methodology on a carefully annotated French reading comprehension benchmark showing that two of those complexity factors are indeed good predictors of models' failure, while others are less so. We further deploy our methodology on a well studied English benchmark by using ChatGPT as a proxy for semantic annotation. Our study reveals that fine-grained linguistically-motivated automatic evaluation of a reading comprehension task is not only possible, but helps understand models' abilities to handle specific linguistic characteristics of input examples. It also shows that current state-of-the-art models fail with some for those characteristics which suggests that adequately handling them requires more than merely increasing model size.

1 Introduction

Generative language models, and very large ones in particular, define the current state-of-the-art in a number of Natural Language Processing tasks. Yet, despite the impressive quantity of scientific studies dedicated to them, the capabilities, limitations, and risks of these models remain largely unknown.

In this work, we argue that black-box evaluations across various tasks, datasets, and languages (Liang et al., 2023; Srivastava et al., 2023) is not enough to portrait current models abilities and instead propose in Section 2 a linguistically fine-grained evaluation methodology that capitalizes on semantic frame annotation (Baker et al., 1998) to characterize examples thanks to a small number of complexity factors we describe in Section 3.

Question Answering (QA) from documents has been extensively studied since the advent of deep neural network-based models, facilitated by large evaluation corpora such as SQuAD (Rajpurkar et al., 2016) and MultiRC (Khashabi et al., 2018), part of the SuperGLUE benchmark (Wang et al., 2019). Transformer-based models consistently top leaderboards[1], outperforming humans. For a nuanced view, see the position paper by (Tedeschi et al., 2023), but we acknowledge this belief as highlighting the challenge of evaluating QA, due to the subjective nature of answer generation and models capturing training data biases (McCoy et al., 2019). Thus, QA offers an interesting playground of our evaluation method that we consider here.

As a proof of concept, we apply our methodology to a publicly available reading comprehension benchmark CALOR (Béchet et al., 2019), which includes French Question-Answer pairs with detailed semantic annotations on the relation linking questions and answers. We demonstrate that certain complexity factors can effectively predict model limitations, regardless of size or architecture. In Section 5, we extended our methodology to the NaturalQA (Kwiatkowski et al., 2019) benchmark, using ChatGPT to compute complexity factors. Our results show that models of various sizes and architectures struggle with certain examples, suggesting that addressing these challenges requires more than just scaling up model size. By presenting a method to automatically select these challenging examples, we provide a means for monitoring further progress in reading comprehension. The data used and collected in this study is available on the following link: https://gitlab.lis-lab.fr/calor-public/complexity-calor.

[1]https://rajpurkar.github.io/SQuAD-explorer,https://super.gluebenchmark.com/leaderboard

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 18376–18392
November 12-16, 2024

2 Method

Our goal is twofold: first, to partition an evaluation corpus into several subsets, each with a distinct (linguistic) level of complexity; and second, to identify linguistically motivated factors that explain the variations in complexity across these subsets. We partition examples based on the analysis of systems' output inspired by the ROVER method (Fiscus, 1997). To ensure independence from any single model when doing so, we propose using a set of models $M = m_1, m_2, \ldots, m_n$ adapted to perform the task and bin examples according to the number of models that agree in their answer with the majority vote. Thus, examples are partitioned into n bins (from total disagreement to full agreement); partition 1 grouping examples where all n systems' outputs differ, while partition n gather examples where all systems agree.

To explain why some subsets are more complex than others, we confront linguistic assumptions formulated as complexity factors to examples in each bin, proceeding as follows:

1. We formulate several assumptions about semantic complexity factors ($F = f_1, f_2, \ldots$) as binary questions applicable to examples in the evaluation corpus. For instance: *Does finding the answer require solving a coreference chain?*

2. For each factor f, we divide the evaluation corpus into two subsets based on whether the examples answer "yes" (E_f=*difficult subset*) or "no" ($\bar{E}_f$=*easy subset*) to the question posed by the factor. When a binary factor requires a threshold to effectively divide the corpus (as in *is the value corresponding to the factor higher ("yes") than the threshold or not ("no")?*) we use quantitative data to set this threshold in order to ensures a balanced division of the corpus.

3. For each factor f and model m, we compute the performance of model m on partitions E_f and $\bar{E}_f$: $S(m, E_f)$ and $S(m, \bar{E}_f)$, and compute $\delta(m, f)$, a score which quantifies performance degradation of model m due to complexity factor f as $\lfloor (S(m, E_f) - S(m, \bar{E}_f)) * 100 \rfloor$.

4. Finally, we calculate a measure of statistical significance for $\delta(m, f)$ with the Mann-Whitney U test with a 5% risk level between the two partitions E_f and $\bar{E}_f$. This test takes into account the value of $\delta(m, f)$ and the characteristics of each set in the partition.

As stated in the introduction, we applied our method to a reading comprehension task, which involves a QA process based on documents. The complexity factors we evaluate in this study were defined through a controlled experiment on the CALOR evaluation corpus, which was manually annotated with semantic frames and enriched with QA based on these frames. This process is described in the next section.

3 Semantic complexity factors

3.1 A semantically-controlled QA corpus

We use the publicly available CALOR corpus (Marzinotto et al., 2018a) which contains documents semantically annotated with the Berkeley Framenet semantic model. This corpus includes French texts from Wikipedia as well as a collection of historical documents covering three main themes: First World War, archaeology, and antiquity. The semantic annotation of this corpus consists of Semantic Frames that describe prototypical situations (e.g., decide, lose, attack, defeat). A trigger word of the Frame, called the Lexical Unit (LU), is identified, followed by the specification of the arguments, known as Frame Elements (FE).

In (Béchet et al., 2019), it was enhanced with semantically controlled question-and-answer examples. This process involved selecting a semantic Frame and a corresponding FE from sentences, then having annotators generate questions whose answers were the selected Frame Elements, with the remaining elements providing context. By varying these selections, a dataset of questions, answers, and their semantic classes was created. Coreference chains were also annotated when needed. This approach produced a corpus of 1785 questions from 54 semantic frames, serving as a valuable resource for validating our methodology under controlled conditions. An example of an annotated sentence from the corpus is shown in Figure 1. Based on these two frame annotations, annotators could have formulated several questions, such as: *"(1) Who lost the majority of their troops on December 10?"* or *"(2) Who started the attack on December 10?"* In both instances, the sentence provides the answer "armies." However, the correct answer, derived from resolving the coreference chain in the paragraph, is "Central Empire coalition."

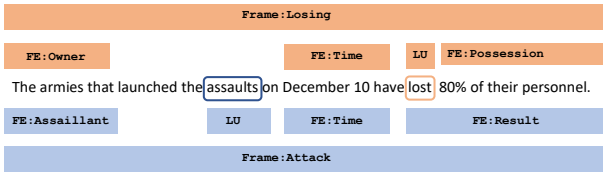


Figure 1: Example of sentence annotated with two semantic frames

3.2 Designing complexity factors

We consider in this study three types of factors based on the semantic frame annotation available in the corpus: factors capturing potential training biases (f_{bias}); factors based on lexical choices and syntactic structures of QA examples (f_{coref}, $f_{trigger}$, $f_{LU\ in\ q}$ f_{dist}) inspired by complexity factors proposed for automatic parsing of frames in (Marzinotto et al., 2018b); finally factors linked intrinsically to the semantic relation defined by a frame ($f_{nb\ FEs}$ and $f_{entropy}$). Here's a concise overview of these factors, with examples for all but f_{bias} and f_{coref} presented in Figure 2.

f_{bias} : **bias in the training/adaptation corpus.** In the experiment section, we use the French QA corpus FQuAD (d'Hoffschmidt et al., 2020) for adapting several models to the QA task. This complexity factor explores the relationship between the frame distribution in this adaptation corpus and the model scores in the evaluation corpus. To explore this factor, we used the tool described in (Marzinotto et al., 2019) to automatically annotate the text data (context) of the FQuAD adaptation corpus with Frames and estimated the frequency of each Frame. We then defined two sets of Frames: $F+$ for the more frequent Frames and $F-$ for the less frequent ones. The set E_f consists of QA examples based on Frames in $F-$ (the rare ones), while $\bar{E}_f$ includes those based on Frames in $F+$ (the common ones).

f_{coref} : **coreference.** The need to resolve a coreference is a potential complexity factor. As mentioned in Section 3.1, coreference chains are annotated for the arguments of the semantic relations linking questions and answers, allowing us to divide the test corpus in two parts: examples with a coreference chain to be resolved to find the answer E_f and the others $\bar{E}_f$. Both examples of question given for figure 1 belongs to E_f as a coreference resolution is needed to find the answer.

$f_{trigger}$: **nature of the semantic relation trigger.** The triggers of a frame in the FrameNet model, called *Lexical Unit - LU*, can be either verbal or nominal. It has been shown (Marzinotto et al., 2018b) that relations triggered by a nominal LU are more difficult to process. We therefore divide the examples in the evaluation corpus according to the nature of the LU: either nominal E_f, or verbal $\bar{E}_f$.

$f_{LU\ in\ q}$: **presence of the frame trigger in the question.** When the same term triggers the semantic relationship in the context and in the question, the example is intuitively simpler to treat. To capture this, we bin examples in subset E_f where the trigger is different between the question and context, and in $\bar{E}_f$ otherwise.

f_{dist} : **syntactic distance between the frame trigger and the answer.** The syntactic distance between the frame trigger and the answer may potentially challenge models as a greater distance may increase ambiguity for finding the answer to the question. We calculate the distance in terms of dependency arcs through a syntactic analysis of the corpus[2] and group together examples with at least two dependency arcs between the trigger and the response in the subset E_f, and group those with only one arc in $\bar{E}_f$.

$f_{nb\ FEs}$: **number of arguments in the frame.** Certain semantic relations exhibit varying numbers of Frame Elements (FEs). The number of FEs within the semantic relation underlying a question-answer pair can influence model efficiency: a higher number of FEs provides a richer contextual basis for accurately identifying the answer, while a smaller number of FEs can make the task more ambiguous. We categorize examples with no more than two annotated FEs into the subset E_f, and those with more than two FEs into $\bar{E}_f$. Our focus is on the manually annotated FEs present in the context of the question, rather than the theoretical number of FEs for the frame in Berkeley FrameNet.

$f_{entropy}$: **measure of entropy in the distribution of LUs for a given frame.** Some frames are consistently triggered by the same terms, while others exhibit much greater diversity, leading to ambiguity in their triggers. This measure of 'surprise' can be quantified through the entropy of the LU distribution in the evaluation corpus. A higher entropy

[2]We used the spaCy toolkit: https://spacy.io

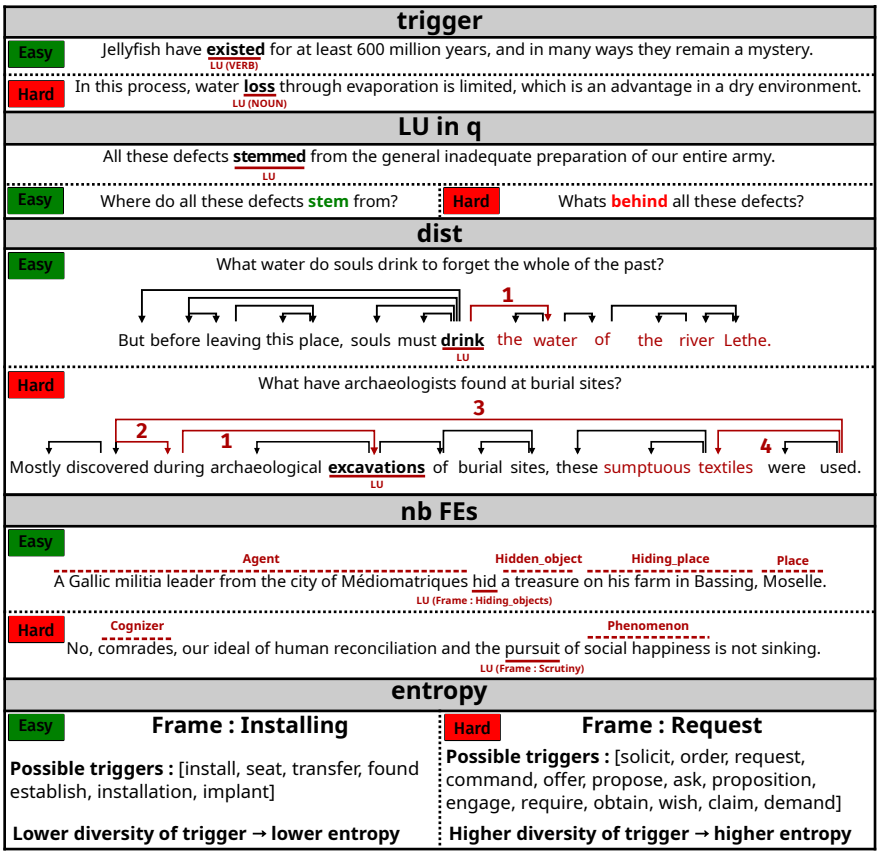


Figure 2: Example of some complexity factors considered

suggests increased ambiguity in frame triggering. We include examples in the subset E_f for frames with an entropy value above a threshold α, and in $\bar{E}_f$ for frames below the same threshold, calculated as the median entropy value across all frames.

4 Controlled experiment

We compare seven pre-trained language models: one is a classification model based on a BERT architecture (Devlin et al., 2019) developed for the French language, CamemBERT (Martin et al., 2020); three models are multilingual generative models based on **T5** (T5-LARGE, FLAN-T5-LARGE (Wei et al., 2021), MT5-LARGE (Xue et al., 2021)), and three models are current Large Language Models (LLMs): LLAMA2 (Touvron et al., 2023), Mixtral 8x7B (Jiang et al., 2024) and ChatGPT-3.5[3].

All these pre-trained models, except ChatGPT3.5 and Mixtral 8x7B, have been adapted to our QA task using the French corpus FQuAD (d'Hoffschmidt et al., 2020). This corpus, constructed similarly to SQuAD (Rajpurkar et al., 2016), contains questions based on French Wikipedia documents.

We used fine-tuning (on FQuAD) for CamemBERT and the T5 models with 2 epochs, and the *Low-Rank Adaptation* method (LoRA) (Hu et al., 2021) on the LLAMA2 model. For GPT-3.5 and Mixtral 8x7B, respectively, a one- and two-shot prompting approach was used, which involved specifying to the model the requirement for an extraction of the original document with one example of input/output in the expected format.

4.1 Evaluation

We evaluate these models on the evaluation corpus with two kinds of metrics: automatic and human metrics. For the automatic metrics we use the *ROUGE-L* score from the ROUGE toolkit[4] (Lin, 2004). This is a similarity score between the extractive reference answer and the systems output. For the human metrics we perform a manual annotation of all the systems' output. Annotators were presented with triplets consisting of a context, a question, and an answer. They were tasked to la-

[3]API from `https://chat.openai.com`

[4]We use the google research implementation available here, with the stemmer and camembert-base tokenizer.

bel each answer as '*correct*', '*partially correct*', or '*incorrect*'. The output from all systems, along with the ground-truth answers, was used to create a total of 14,280 triplets (1,785 triplets per system, including 7 systems and the ground-truth). After removing duplicates in the answers, we obtained a set of 5857 unique triplets, which were then divided into 10 folds and evaluated by 10 human annotators[5]. Any annotator labels that contradicted the ground-truth labels were reviewed to either correct the reference annotations or adjust the annotators' decisions. Two metrics were derived from this manual annotation:

- *Hscore*: This metric assigns a score of 1 to answers labeled as *correct*, 0.5 to those labeled as *partially correct*, and 0 to those labeled as *incorrect*.
- *Hcorrect*: This metric represents the proportion of answers labeled as *correct* by the annotators for a given system.

Model	adapt	#param	Rouge-L	Hscore	% Hcorrect
CamemBERT	FT	335M	0.82	0.85	78.9
T5-L	FT	738M	0.81	0.84	78.0
FLAN-T5-L	FT	783M	0.80	0.85	79.2
MT5-L	FT	1.2B	0.80	0.84	77.5
LLAMA-2	LoRA	7B	0.69	0.78	72.2
Mixtral-8x7b	prompt	47B	0.80	0.87	**82.6**
GPT 3.5	prompt	175B	0.72	**0.88**	82.5
ROVER	-	-	**0.84**	**0.88**	82.3

Table 1: Description of the 7 models used in our experiments with their performance in terms of Rouge-L, Hscore and Hcorrect scores. The last line indicates the performance of systems' combination through the ROVER method.

Overall, the results achieved by the various models are notably lower compared to those showcased on leaderboards of analogous tasks such as SQuAD [6] or MultiRC in SuperGLUE [7]. This discrepancy can be attributed in part to the characteristics of the evaluation corpus and its differences with the adaptation corpus FQuAD as well as the absence of systematic model optimization through hyperparameterization.

The Rouge-L score of the T5-based generation models and the CamemBERT-based classification model are closely aligned, whereas those of the two LLMs, LLAMA-2 and GPT3.5, significantly lag behind. This comes from the fact that the references in the evaluation corpus are extractive (comprising segments of the original text) and that RougeL inherently leans towards models that merely replicate segments without introducing additional words. When considering human evaluation, the results are inverted: generative LLMs that are lightly adapted with prompting, that tend to introduce additional elements for presentation or explanation, are preferred by humans and outperform other models on both *Hscore* and *Hcorrect* metrics.

This analysis underscores the necessity for evaluation metrics beyond string similarity between a single reference and the output of a generative model for abstractive tasks. Notably, unlike GPT-3.5 and Mixtral, the LLAMA-2 model's performance remains low in human evaluations. This discrepancy can be attributed to the ineffective LoRA adaptation, despite being monitored using the Rouge-L score. Although the final Rouge-L score was low, it was comparable to that of GPT-3.5, leading us to initially attribute the low score to the model's abstractive capabilities. However, human evaluation revealed this was not the case. Due to the high cost of human annotation, it was not feasible to use this metric to refine and optimize our adaptation process. Consequently, we exclude the results obtained with LLAMA-2 from now on and use our human metrics instead of Rouge-L.

4.2 Complexity factors

We apply the methodology described in section 2 for partitioning QA examples by complexity and assessing the relevance of the complexity factors describes in section 3.

To sort QA examples by complexity, we utilize the agreement between models, which is assessed using the ROVER score as detailed above. Given that we are working with both extractive and abstractive models, we calculate the *agreement* between the outputs of two models, $M_1(x)$ and $M_2(x)$, for a given input x using the Levenshtein distance, denoted as $dist_L(.)$, between the two strings. The agreement is defined as:

$$agree(M_1, M_2, x) \Leftrightarrow dist_L(M_1(x), M_2(x)) < \alpha$$

In our experiments, we arbitrarily fixed $\alpha = 5$ to allow strings that differ only by the deletion or addition of a specifier to be considered as agreeing.

[5]All these human annotations as well as systems' output and complexity factors annotations are publicly available : https://gitlab.lis-lab.fr/calor-public/complexity-calor

[6]https://rajpurkar.github.io/SQuAD-explorer

[7]https://super.gluebenchmark.com/leaderboard

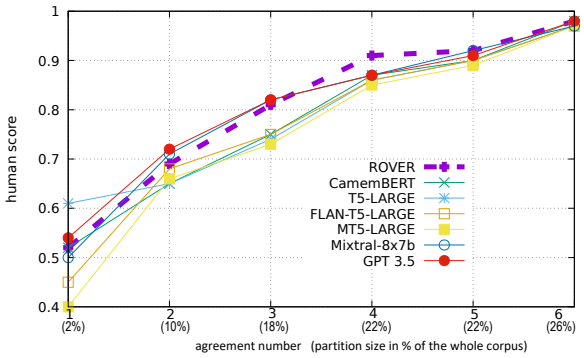


Figure 3: Performance in Hscore according to the agreement number with the ROVER systems' combination method

The ROVER performance is displayed in the last row of Table 1. It performs best for Rouge-L and Hscore metrics and closely approaches the best for Hcorrect. ROVER forms the basis of our proposed method for sorting QA examples by complexity. By using 6 models in the voting process, we categorize examples into 6 partitions ($P1$ to $P6$) based on the level of agreement among systems. $P1$ contains examples where the 6 systems' outputs differ, while $P6$ includes those where all systems agree. In Figure 3, we plot the Hscore of ROVER and all other models across these 6 partitions. The alignment between the number of agreements and complexity measurement is consistent across all models, with ROVER scores closely mirroring Hscore, which increases nearly linearly with agreement count. From this curve, we deduce that our evaluation corpus is relatively easy. Nearly half of the corpus (48%, combining $P5$ and $P6$) has an Hscore over 90% for all models. Of the remaining corpus, 40% ($P3$ and $P4$) are of moderate complexity, where larger models outperform smaller ones. The final 12% are the most difficult examples for all models, regardless of their size.

Is complexity linked to semantic relations? The ROVER partitioning produced reliable clusters but did not clarify why some clusters are more challenging than others. To investigate this, we explore the correlation between semantic relationships linking questions, answers and model performance. Semantic relationships are represented by the frames used to generate the questions (detailed in Section 3.1). We segmented our corpus into 54 sub-corpora based on the frames, allowing us to evaluate each model's performance for each specific frame.

Figure 4 illustrates the distribution of ROVER scores across each frame sub-corpus[8]. This distribution is non-uniform, validating our intuition that model performance varies with underlying semantic relations. This brings us to the second step of our method, which involves validating the complexity factors proposed in Section 3.

	Complexity factor						
models/factors	bias	coref	trigger	LU in q	dist	nb FEs	entropy
size of Ef (%)	42%	6%	37 %	45%	30%	59%	50%
CamemBERT	-1	**-7**	-1	-2	-1	**-4**	-1
T5	0	**-9**	-1	-2	-3	**-4**	**-5**
FLAN	-1	-6	-1	**-3**	**-4**	-3	**-5**
MT5	-1	**-15**	0	**-2**	**-3**	**-4**	**-4**
GPT-3.5	0	-2	0	1	-1	-1	-2
mixtral-8x7b	0	-2	-2	-2	-2	**-4**	-1
ROVER	1	**-7**	0	-2	-1	**-2**	-2

Table 2: Validation results for complexity factors across models, showing δ values in each cell with statistically significant differences in bold. '*Size*' indicates proportions of partitions E_f relative to the total corpus.

Evaluation of complexity factors. Table 2 shows the results for these 7 complexity factors. In each cell, for a model m and a factor f, the value corresponds to the impact of f on m expressed by the difference in terms of Hscore δ presented in Section 2. Values in bold correspond to factors that have validated the Mann-Whitney U test for statistical significance with a 5% risk. This methodology allows us to systematically analyze and quantify the impact of different complexity factors on model performance, providing rigorous statistical validation of observed differences in Hscore between linguistically easier and more complex subgroups. As we can see, the generic factor f_{bias} corresponding to the link between the frequency of a frame in the adaptation corpus and in the evaluation corpus has very little influence on the results.

Factor f_{coref} shows that resolving co-reference chains is a complexity factor for all models but significantly impacts only smaller models like T5 and MT5. While LLMs also experience some performance loss, it is less significant, indicating their better handling of co-references.

The nature of the Frame trigger ($f_{trigger}$) is a complexity factor for all models but differences are not statistically significant. Factor $f_{LU\ in\ q}$ is validated for all models except GPT-3.5, but significant only for FLAN and MT5. Factor f_{dist} mainly affects smaller models, supporting the idea that

[8]Similar distributions were observed across all models, even if there is some variation in the frame ranking. The figures for all models are in Appendix A.8

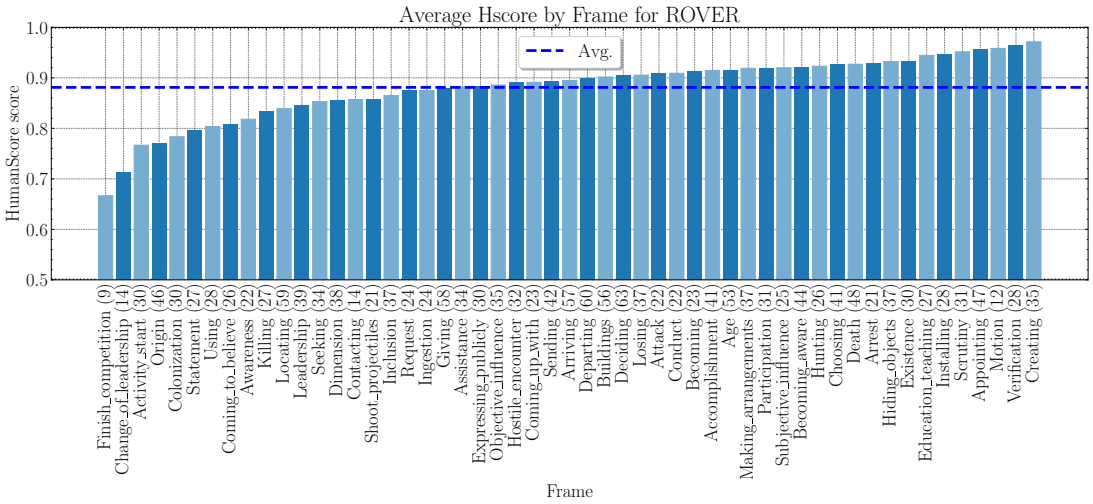


Figure 4: Performance of ROVER according to each frame sorted by Hscore measure. The number of occurrences of each frame in the corpus is given between brackets

LLMs better encode syntactic structures.

Interestingly, the most reliable factors are those intrinsically linked to the semantic relations representing the frames ($f_{nb\ FEs}$ and $f_{entropy}$) rather than their contextual use. Thus, these two factors can be associated with the measure of semantic ambiguity in question/answer relations.

For example, the *Request* frame has over 20 triggers in the Berkeley Framenet lexicon[9]. In our evaluation corpus, it has 33 occurrences with 6 different triggers, resulting in high entropy and Hscore scores from 0.55 to 0.84 depending on the model.

In contrast, the *Installing* frame, defined as "*An Agent places a Component in a Fixed Location so that the Component is attached and interconnected and thereby functional*" has only two triggers (*install* and *installation*). It has 30 occurrences in our corpus with 2 triggers, low entropy, and Hscore scores from 0.79 to 0.90.

Factor $f_{nb\ FEs}$ shows frames with a low number of Frame Elements in their examples (≤ 2). For instance, the *Origin* frame has two 'core' FEs (*Origin* and *Entity*), while the *Giving* and *Contacting* frames have more 'core' and non-core FEs. This aligns with factor $f_{entropy}$, where the *Origin* frame scores below average, while *Giving* is an 'easy' frame.

Selecting semantically complex QA examples. Complexity factors can be used to identify challenging QA examples by considering one or more factors. Our analysis focuses on the most significant factors, $f_{nb\ FEs}$ and $f_{entropy}$. Figure 5 shows Hscore values for subsets of the corpus categorized by examples influenced by neither, one, or both of these factors, plus any additional factors. Most models exhibit the greatest score disparity between subsets with no factors and those with at least one of $f_{nb\ FEs}$ or $f_{entropy}$. The score difference is minimal between subsets with one factor and those with both, except for T5, MT5, and LLaMA-2.

f/P	$P6$	$P5$	$P4$	$P3$	$P2$	$P1$
$P\,(f_{nb\ FEs})$	0.52	0.56	0.62	0.64	0.62	0.80
$P\,(f_{entropy})$	0.51	0.57	0.60	0.59	0.58	0.54

Table 3: Probability of having the $f_{nb\ FEs}$ and $f_{entropy}$ factors according to the agreement partitions of increasing complexity $P6$ to $P1$

The last step of our analysis is to study if our semantic factors can explain the differences in complexity among the different partitions $P1$ to $P6$ obtained through the ROVER method. Table 3 shows the probabilities of the QA examples in each partition P to have factor $f_{nb\ FEs}$ or $f_{entropy}$. As can be observed, probabilities for $f_{nb\ FEs}$ and $f_{entropy}$ increase clearly from $P6$ to $P5$ and to a lesser extent from $P5$ to $P4$, indicating that examples with higher semantic ambiguities are more likely to be occurring in the difficult partitions within $P3$ to $P1$.

5 Experiments with NaturalQA

To evaluate the transferability of our two main complexity factors ($f_{nb\ FES}$ and $f_{entropy}$) to other QA datasets and languages, we used a subset of NaturalQA (Kwiatkowski et al., 2019) and the predic-

[9] https://framenet.icsi.berkeley.edu/frameIndex

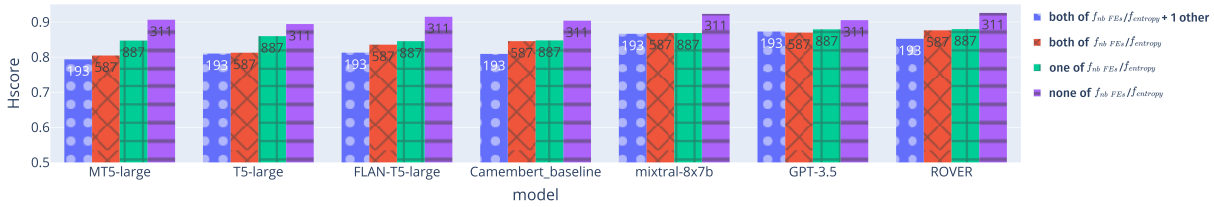


Figure 5: Hscore on 4 partitions of the evaluation corpus according to combinations of complexity factors

tions of 48 models provided by HELM (Liang et al., 2023) on their *natural_qa_openbook_longans*[10] scenario. This subset consists of 1,000 examples from the NaturalQA evaluation distribution, each comprising a question, a "short" answer, and the context, which in this case is the corresponding "long" answer from NaturalQA (typically equivalent to a paragraph). For brevity, we present the results of 8 of the 48 models in Table 5[11], selected to represent the full range of mean F1 scores across all models. Additionally, we display the ROVER score estimated across all (48) models.

Applying $f_{entropy}$ to NaturalQA. For this dataset, lacking an automatic Frame analysis, we used a proxy method: we compiled all potential triggers from Berkeley FrameNet frames and checked their exact presence in the questions. Each question provided a list of triggers and their corresponding frames. Using a custom prompt[12], we employed GPT-3.5 to determine the most appropriate pair for each question.

For example with the question : *How long did the democrats control the house and senate?* we can extract the following list of 11 triggers and their potentials frames : [('Duration_description', 'long'), ('Buildings', 'house'), ('Desiring', 'long'), ('Dimension', 'long'), ('Firefighting', 'control'), ('Controller_object', 'control'), ('Measurable_attributes', 'long'), ('Containing', 'house'), ('Experimentation', 'control'), ('Being_in_control', 'control'), ('Control', 'control')]. The chosen pair in this case being : ('Being_in_control', 'control').

We decided to use a proxy via ChatGPT rather than automatic analysis in a semantic framework for several reasons. First, this approach offers simplicity in implementation and scalability to other languages, requiring only hypothesis extraction via keyword search and API calls. Second, our analysis is focused on questions, not paragraphs of text, unlike the typical training data for most semantic frame models, and we had reservations about the models' performance in this context.

We performed a manual evaluation on 50 sentences, where two annotators assessed ChatGPT's frame predictions as fully correct, partially correct, or erroneous. The results, shown in Table 4, demonstrate overall good performances, with some errors observed.

Evaluation	Full	Partial	Erroneous
Frame Prediction	57	18	25
Frame Elements	66	22	12

Table 4: Manual evaluation (in %) of ChatGPT's frame predictions across 50 random sentences

Out of 1000 examples, 18 had no extractable triggers and were discarded. On the generated frames by ChatGPT, 35 were unknown from our Frame dictionary and were also discarded. We then assessed our $f_{entropy}$ factor by calculating the median entropy across all Berkeley FrameNet frames.

We computed the variation in F1-score between examples that validated $f_{entropy}$ (high entropy) and those that did not. Across all models, the average difference in performance between these subsets was -3.17 (± 1.82) F1 points, indicating that $f_{entropy}$ is also a significant complexity factor for the NaturalQA benchmark. In Table 5, we present the F1 variation for the 8 selected models as well as for ROVER, showing that most models have a significant F1-lost when considering $f_{entropy}$ examples.

Applying $f_{nb\ FEs}$ to NaturalQA. For $f_{nb\ FEs}$, following the prompt method used for $f_{entropy}$, we

[10] https://crfm.stanford.edu/helm/lite/latest/#/groups/natural_qa_openbook_longans

[11] See result for all model in A.9

[12] Appendix A.5

		Factors	
models/factors	F1	nb FEs	entropy
size of E_f (%)		78%	52%
google_text-bison@001	0.81	0	-1
openai_text-davinci-003	0.77	-4	**-5**
ROVER	0.77	-4	-3
meta_llama-3-70b	0.74	-3	**-4**
mistralai_mixtral-8x7b-32kseqlen	0.70	**-6**	**-4**
openai_gpt-3.5-turbo-0613	0.68	-4	**-6**
google_gemma-7b	0.66	-4	-3
AlephAlpha_luminous-extended	0.61	**-7**	**-5**
databricks_dbrx-instruct	0.55	-2	0
ROVER	0.77	-4	-3

Table 5: Validation results for $f_{nb\ FEs}$ and $f_{entropy}$ across models on naturalQA. '*Size*' indicates proportions of partitions E_f relative to the total corpus.

automatically extracted the FEs based on the previously predicted frames using an empirically developed prompt[13]. We extracted Frame Elements for 937 examples where frames were successfully predicted out of 961 attempts (24 were discarded due to output issues).

We then categorized these examples based on our $f_{nb\ FEs}$ factor: those with more than 2 FEs are considered easier, while those with 2 or fewer are considered more challenging. Typically, examples with more than 2 FEs score above average, while those with 2 or fewer score below. However, on average, this difference is smaller compared to $f_{entropy}$. Across all models, the average difference in performance between these subsets is -3.84 (± 2.44) F1 points. This may be due to NaturalQA questions being simpler and containing fewer Frame Elements compared to our original corpus, increasing the proportion of challenging examples from 60% to 78%.

6 Related Work

Our work situates itself within the domain of model evaluation. Our approach contrasts with broad-scale evaluations that span multiple tasks, corpora, and languages (Laskar et al., 2023; Liang et al., 2023; Srivastava et al., 2023; Brown et al., 2020; Wang et al., 2019). It relates to focused studies addressing specific linguistic phenomena such as negations (Truong et al., 2022, 2023; Zhang et al., 2023; Ravichander et al., 2022), ambiguity in inference tasks (Liu et al., 2023), and open information extraction (Lechelle et al., 2019), that utilizes small, meticulously curated datasets to precisely evaluate the capabilities of models for the task. Our study echoes the latter, exploring focused linguistic evaluations.

This study aligns with other efforts evaluating 'closed' LLMs like ChatGPT, accessible only through APIs, on benchmarks such as knowledge-based question answering (KBQA) (Tan et al., 2023). These studies highlight ChatGPT's robust performance across diverse NLP tasks (Kocoń et al., 2023; Laskar et al., 2023), yet also note its potential to lag behind task-specific models.

Overall, this study pushes the idea that we need a more precise evaluation framework and can be related to other studies such as (Ribeiro et al., 2020) that identify *critical failures* in both commercial and state-of-the-art models by proposing a model and task-agnostic testing methodology or (Gehrmann et al., 2023) insisting on the fact that to compare models we need more "*careful annotation process [...] to characterize their output quality and distinguish between them*".

7 Conclusions

This paper presents a methodology for identifying intrinsic complexity factors in NLP tasks. Our results reveal that some examples consistently produce lower scores due to their inherent linguistic complexity. Through an empirical study on a QA task, we identified and validated several factors of semantic complexity, with results directly linked to human evaluations of model predictions. We have also validated these factors on another dataset in another language, confirming their robustness. In addition, we have developed corpora of increasing semantic complexity, suggesting that taking these complexities into account requires more than simply improving the model's parameters.

8 Limitations

The main limitation of our study is to have considered a single task, a limited set of languages (French and English) and corpora (CALOR and NaturalQA). Our focus in this article revolves around the viability of conducting focused, cost-effective studies, requiring less than 100 GPU hours (inclusive of hyperparameter search) and approximately $10 for the GPT-3.5 API. These studies prioritize linguistic analysis to draw conclusions that extend beyond the specific corpus, task, and language. We believe that such complementary studies have a place in academic Natural Language Processing conferences.

[13] Appendix A.5

Acknowledgements

This project was provided with computer and storage resources by GENCI at IDRIS thanks to the grant 2023-AD011012688R2 on the supercomputer Jean Zay's V100/A100 partition. We would like to thank the reviewers for their comment and feedbacks which has helped us improve the first version of this article to its current state.

References

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Frédéric Béchet, Cindy Aloui, Delphine Charlet, Geraldine Damnati, Johannes Heinecke, Alexis Nasr, and Frédéric Herledan. 2019. Calor-quest: generating a training corpus for machine reading comprehension models from shallow semantic annotations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 19–26.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin d'Hoffschmidt, Wacim Belblidia, Quentin Heinrich, Tom Brendlé, and Maxime Vidal. 2020. FQuAD: French question answering dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1193–1208, Online. Association for Computational Linguistics.

Jonathan G Fiscus. 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 347–354. IEEE.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *J. Artif. Int. Res.*, 77.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, et al. 2023. Chatgpt: Jack of all trades, master of none. *Information Fusion*, page 101861.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Huang. 2023. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 431–469, Toronto, Canada. Association for Computational Linguistics.

William Lechelle, Fabrizio Gotti, and Phillippe Langlais. 2019. WiRe57 : A fine-grained benchmark for open information extraction. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 6–15, Florence, Italy. Association for Computational Linguistics.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and et al. Benjamin Newman. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*. Featured Certification, Expert Certification.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah Smith, and Yejin Choi. 2023. We're afraid

language models aren't modeling ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 790–807, Singapore. Association for Computational Linguistics.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Laurent Romary, Éric Villemonte de La Clergerie, Djamé Seddah, Benoît Sagot, et al. 2020. Camembert: a tasty french language model. In *ACL 2020-58th Annual Meeting of the Association for Computational Linguistics*.

Gabriel Marzinotto, Jeremy Auguste, Frederic Bechet, Géraldine Damnati, and Alexis Nasr. 2018a. Semantic Frame Parsing for Information Extraction : the CALOR corpus. In *LREC2018*, Miyazaki, Japan.

Gabriel Marzinotto, Frédéric Béchet, Géraldine Damnati, and Alexis Nasr. 2018b. Sources of Complexity in Semantic Frame Parsing for Information Extraction. In *International FrameNet Workshop 2018*, Miyazaki, Japan.

Gabriel Marzinotto, Géraldine Damnati, Frédéric Béchet, and Benoît Favre. 2019. Robust semantic parsing with adversarial learning for domain generalization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 166–173, Minneapolis, Minnesota. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

Abhilasha Ravichander, Matt Gardner, and Ana Marasovic. 2022. CONDAQA: A contrastive reading comprehension dataset for reasoning about negation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8729–8755, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, and et al. Adrià Garriga-Alonso. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Can chatgpt replace traditional kbqa models? an in-depth analysis of the question answering performance of the gpt llm family. In *International Semantic Web Conference*, pages 348–367. Springer.

Simone Tedeschi, Johan Bos, Thierry Declerck, Jan Hajič, Daniel Hershcovich, Eduard Hovy, Alexander Koller, Simon Krek, Steven Schockaert, Rico Sennrich, Ekaterina Shutova, and Roberto Navigli. 2023. What's the meaning of superhuman performance in today's NLU? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12471–12491, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Thinh Hung Truong, Timothy Baldwin, Karin Verspoor, and Trevor Cohn. 2023. Language models are not naysayers: an analysis of language models on negation benchmarks. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 101–114, Toronto, Canada. Association for Computational Linguistics.

Thinh Hung Truong, Yulia Otmakhova, Timothy Baldwin, Trevor Cohn, Jey Han Lau, and Karin Verspoor. 2022. Not another negation benchmark: The NaN-NLI test suite for sub-clausal negation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 883–894, Online only. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and

Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

Yuhui Zhang, Michihiro Yasunaga, Zhengping Zhou, Jeff Z. HaoChen, James Zou, Percy Liang, and Serena Yeung. 2023. Beyond positive scaling: How negation impacts scaling trends of language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7479–7498, Toronto, Canada. Association for Computational Linguistics.

9 Appendix

A Technical information about the training process and the data

FQuAD dataset download link : https://fquad.illuin.tech/

A.1 Training of CamemBert

CamemBert was finetuned using the default parameters of the HuggingFace trainer for 4 epochs, with model check-pointing keeping the best overall checkpoint.

Training hardware :

```
GPU : 1 x Tesla V100-SXM2-32GB
```

A.2 Training of T5, MT5 and FLAN-T5

The training was performed using a modified version of this training script script from HuggingFace : https://github.com/huggingface/transformers/blob/main/examples/pytorch/question-answering/trainer_seq2seq_qa.py

Training parameters are bellow, all other parameters are the **default** one of the HuggingFace trainer (**transformers** installation from source at commit 686c68f64c9d0181bd54d4d2e2446543c3eca1fa).

```
{
    "max_seq_length": 512,
    "adafactor": true,
    "learning_rate" : 3e-05,
    "num_train_epochs" : 2,
    "evaluation_strategy": "steps",
    "metric_for_best_model": "f1",
    "load_best_model_at_end": true,
    "seed": 260,
    "max_answer_length": 40
}
```

Data format :

```
"question: {question}
contexte : {context}"
```

Training hardware :

```
GPU : 1 x Tesla V100-SXM2-32GB
```

Training time :

- T5 ≈ 2h15mn
- MT5 ≈ 2h30mn
- FLAN-T5 ≈ 2h30mn

In total, a few run of tests (≈ 12) for the prompt, optimizer and learning rate were done with similar running times.

The **inference** time vary a bit between model and is ≈ 30mn.

A.3 Adaptation of llama2-7b

The LoRA adaptation was performed using https://github.com/huggingface/peft library, with the config given bellow.

```
LoraConfig(
    r=32,
    lora_alpha=64,
    target_modules=["q_proj", "v_proj"],
    lora_dropout=0.1,
    bias="none",
    task_type="CAUSAL_LM",
)
```

The modified training argument are given bellow, the rest are default.

```
transformers.TrainingArguments(
    per_device_train_batch_size=1,
    gradient_accumulation_steps=4,
    num_train_epochs=1,
    learning_rate=2e-4,
    fp16=True,
    save_total_limit=3,
    logging_steps=1,
    max_steps=80,
    optim="paged_adamw_32bit",
    lr_scheduler_type="cosine",
    warmup_ratio=0.05,
)
```

Prompt format : The prompt was constructed with the same three examples randomly selected from FQuAD for both training and inference.

```
Below is a paragraph of text, paired with a
question. Extract the sequence of words in
the article that answers the following
question, or answer NULL if there are no
answers.


### Paragraph:
Après le tournage, Hal B. Wallis [...]
### Question:
Qui ne peut pas se libérer pour la scène
envisagée par Wallis ?
### Answer:
"Claude Rains"


### Paragraph:
Riquet étudie de façon approfondie  [...]
### Question:
Quel est l'un des points sur lequel le projet
de Riquet reste imprécis ?
Answer:
"tracé du canal"


### Paragraph:
Dans cet intervalle de 31 jours, [...]
### Question:
Combien sont-ils à être frappés ?
Answer:
"quelques-uns"
```

Training hardware :

```
GPU : 1 x GPU Nvidia A100-80GB
```

Training and inference time :

- training $\approx$ 70sec
- inference $\approx$ 17mn

A.4 rouge-L results and significativity for the complexity factors

A.5 Extraction of Frames and Frame Element on NaturalQA

Prompt for Frame extraction :

```
From a list of (frame, lexical unit) from
FrameNet, predict which is the most likely
for the given question. Only answer with
the correct (frame, lexical unit) pair.
List : {list}
Question : {question}
```

	Complexity factor						
models/factors	bias	coref	trigger	LU in q	dist	nb FEs	entropy
size of Ef (%)	42%	6%	37%	45%	12%	59%	46%
CamemBERT	-1	-4	-1	**-2**	**-7**	**-3**	-1
T5	-1	-9	-2	-1	**-7**	**-5**	**-2**
FLAN	-2	-4	-3	-2	-4	**-5**	**-3**
MT5	0	-13	-1	-1	**-10**	**-4**	-2
llama-2	0	-3	-1	3	-3	**-7**	**-2**
GPT-3.5	0	**4**	-1	0	-4	-4	**-3**
mixtral-8x7b	0	1	-2	-1	-5	**-6**	0

Table 6: Complexity factor validation results with the Rouge-L score. Each box contains the δ value of each factor for each model. Bold indicate statistically significant differences. The *size* line displays the proportions of the E_f partitions relative to the total size of the corpus.

Prompt for Frame Element extraction :

```
From a FrameNet (frame , lu/trigger) pair
and a context extract the corresponding
Frame Elements from the given question.
The LU can't be a FE.  Output a json.
Pair : {pair}
Question : {question}
```

A.6 Complexity factor examples

Number of Arguments in the Frame ($f5$) : Easy (more FEs in context, here > 2) :
Comment est mort Kleitarchos en 341 ?
(*How did Kleitarchos die in 341?*)
Quand les congrès de Zimmerwald et de Kiental ont-ils commencé le processus de renversement de l'ordre établi ?
(*When did the Zimmerwald and Kiental congresses begin the process of overthrowing the established order?*)
Lors de la bataille d'Actium, Caius Sosius a dirigé quelle partie de la flotte ?
(*At the battle of Actium, which part of the fleet did Caius Sosius command?*)
En quelle année Silvestras Žukauskas a-t-il été étudiant à l'école des cadets d'infanterie de Wilna ?
(*In what year was Silvestras Žukauskas a student at the Wilna Infantry Cadet School?*)

Hard (less FEs in context, here 2) :
Qu'est-ce qui est caché ?
(*What's hidden?*)
Quand les Russes attaquent-ils ?
(*When do the Russians attack?*)
Quel est le sujet ?
(*What's the subject?*)
Who shoots the ammunition?
(*Who shoots the ammunition?*)
Qui a découvert de nouvelles techniques de

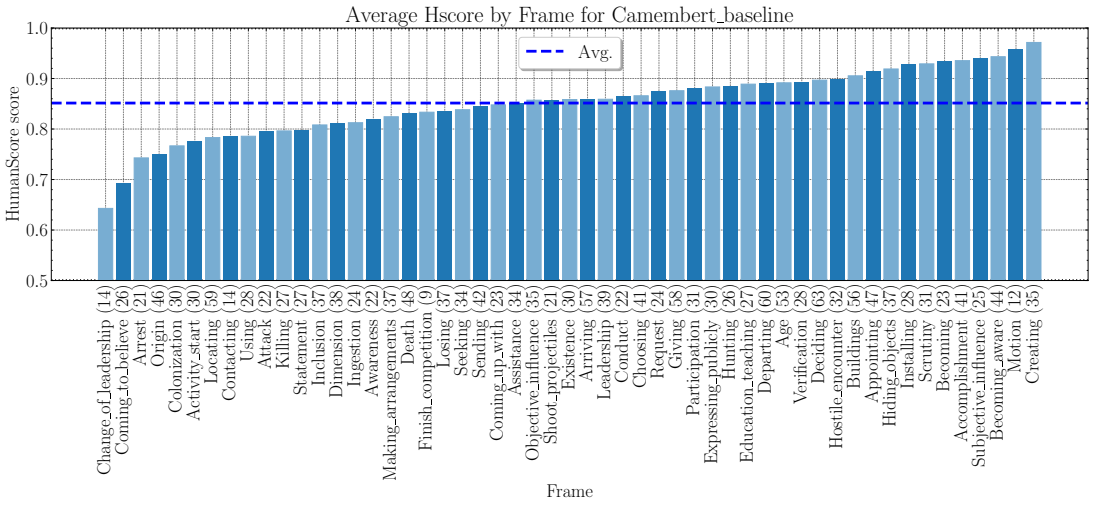
création ?
(*Who's discovered new creative techniques?*)

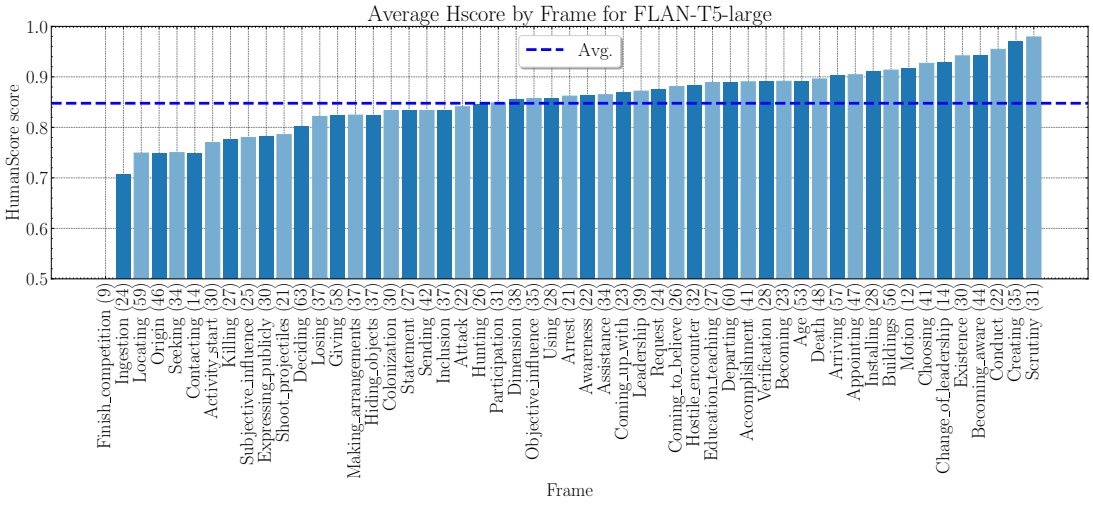
A.7 Annotator compensation

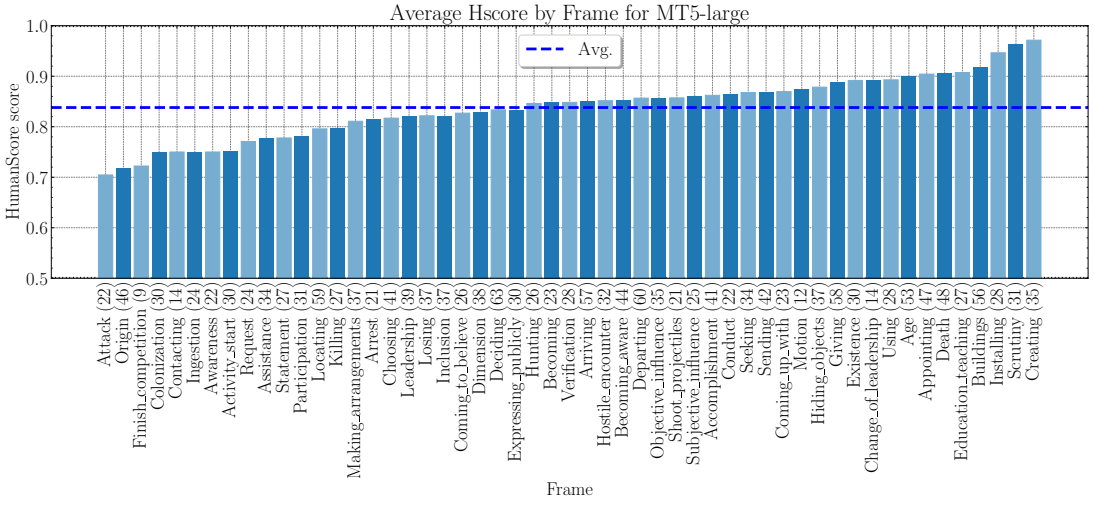
The human annotators are volunteer PhD students from the same laboratory (from different teams to the authors). They were paid 45€ via gift vouchers, as our country's legislation does not allow direct pay-per-task remuneration.

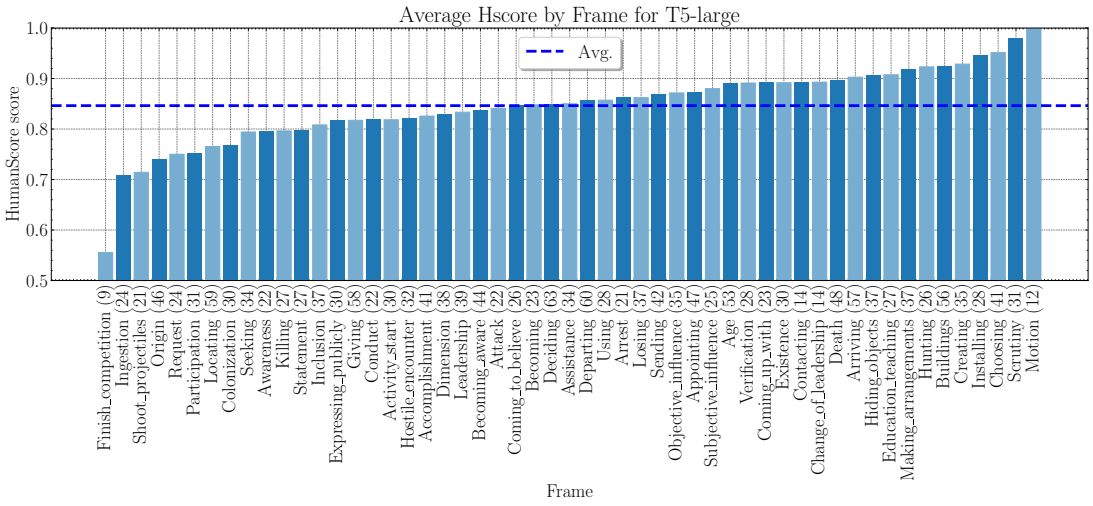
A.8 HumanScore results per frame for all models

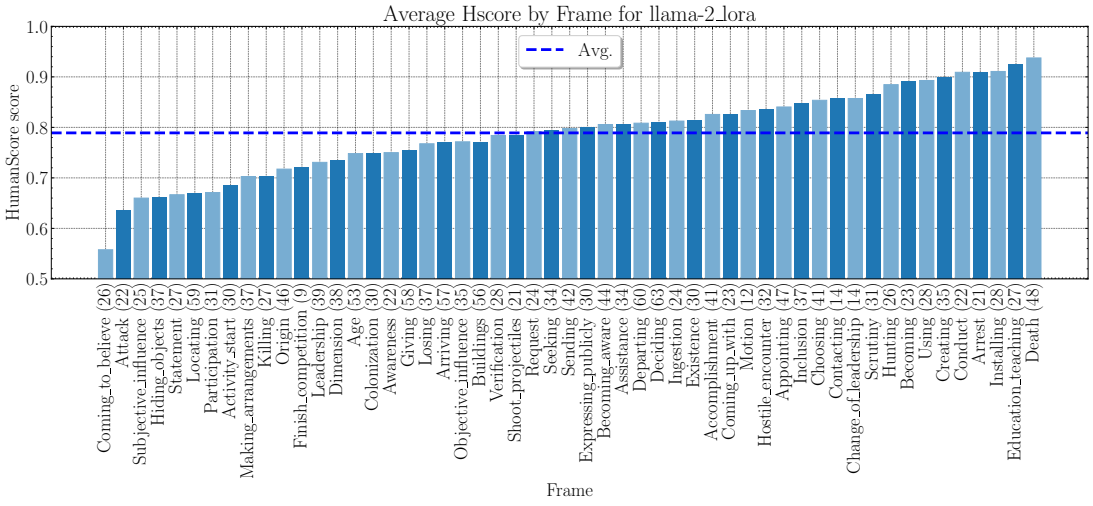
A.9 Result on all model of naturalQA for $f5$ and $f6$

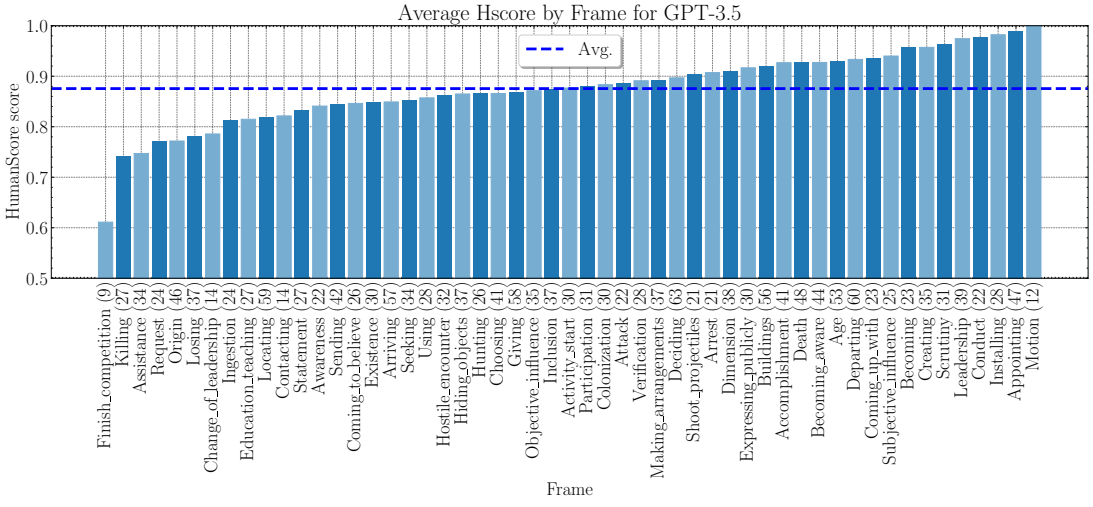

Average Hscore by Frame for Camembert_baseline
HumanScore score
Avg.
Change_of_leadership (14)
Coming_to_believe (26)
Arrest (21)
Origin (46)
Colonization (30)
Activity_start (30)
Locating (59)
Contacting (14)
Using (28)
Attack (22)
Killing (27)
Statement (27)
Inclusion (37)
Dimension (38)
Ingestion (24)
Awareness (22)
Making_arrangements (37)
Death (48)
Finish_competition (9)
Losing (37)
Seeking (34)
Sending (42)
Coming_up_with (23)
Assistance (34)
Objective_influence (35)
Shoot_projectiles (21)
Existence (30)
Arriving (57)
Leadership (39)
Conduct (22)
Choosing (41)
Request (24)
Giving (58)
Participation (31)
Expressing_publicly (30)
Hunting (26)
Education_teaching (27)
Departing (60)
Age (53)
Verification (28)
Deciding (63)
Hostile_encounter (32)
Buildings (56)
Appointing (47)
Hiding_objects (37)
Installing (28)
Scrutiny (31)
Becoming (23)
Accomplishment (41)
Subjective_influence (25)
Becoming_aware (44)
Motion (12)
Creating (35)
Frame

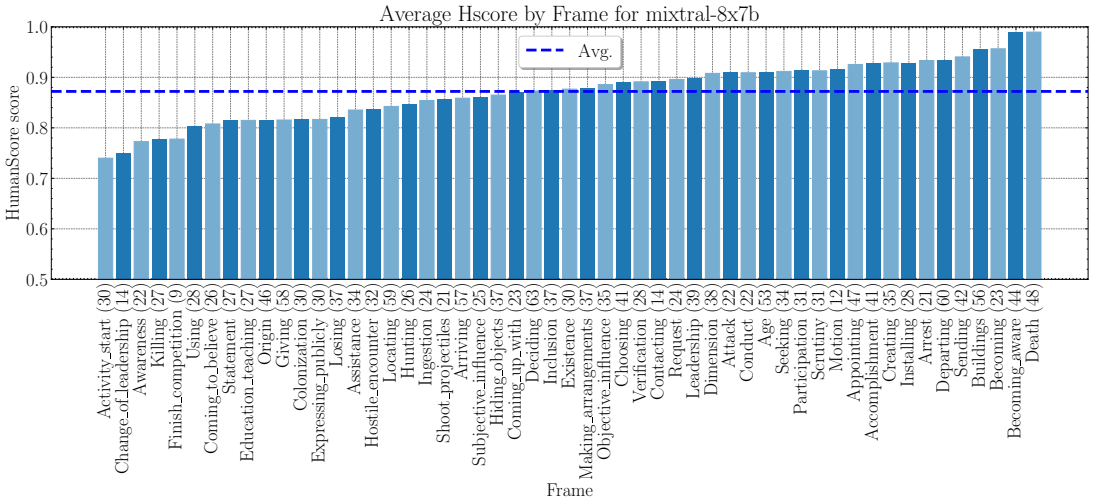

Average Hscore by Frame for FLAN-T5-large
HumanScore score
Avg.
Finish_competition (9)
Ingestion (24)
Locating (59)
Origin (46)
Seeking (34)
Contacting (14)
Activity_start (30)
Killing (27)
Subjective_influence (25)
Expressing_publicly (30)
Shoot_projectiles (21)
Deciding (63)
Losing (37)
Giving (58)
Making_arrangements (37)
Hiding_objects (37)
Colonization (30)
Statement (27)
Sending (42)
Inclusion (37)
Attack (22)
Hunting (26)
Participation (31)
Dimension (38)
Objective_influence (35)
Using (28)
Arrest (21)
Awareness (22)
Assistance (34)
Coming_up_with (23)
Leadership (39)
Request (24)
Coming_to_believe (26)
Hostile_encounter (32)
Education_teaching (27)
Departing (60)
Accomplishment (41)
Verification (28)
Becoming (23)
Age (53)
Death (48)
Arriving (57)
Appointing (47)
Installing (28)
Buildings (56)
Motion (12)
Choosing (41)
Change_of_leadership (14)
Existence (30)
Becoming_aware (44)
Conduct (22)
Creating (35)
Scrutiny (31)
Frame


Average Hscore by Frame for MT5-large
HumanScore score
Avg.
Attack (22)
Origin (46)
Finish_competition (9)
Colonization (30)
Contacting (14)
Ingestion (24)
Awareness (22)
Activity_start (30)
Request (24)
Assistance (34)
Statement (27)
Participation (31)
Locating (59)
Killing (27)
Making_arrangements (37)
Arrest (21)
Choosing (41)
Leadership (39)
Losing (37)
Inclusion (37)
Coming_to_believe (26)
Dimension (38)
Deciding (63)
Expressing_publicly (30)
Hunting (26)
Becoming (23)
Verification (28)
Arriving (57)
Hostile_encounter (32)
Becoming_aware (44)
Departing (60)
Objective_influence (35)
Shoot_projectiles (21)
Subjective_influence (25)
Accomplishment (41)
Conduct (22)
Seeking (34)
Sending (42)
Coming_up_with (23)
Motion (12)
Hiding_objects (37)
Giving (58)
Existence (30)
Change_of_leadership (14)
Using (28)
Age (53)
Appointing (47)
Death (48)
Education_teaching (27)
Buildings (56)
Installing (28)
Scrutiny (31)
Creating (35)
Frame


Average Hscore by Frame for T5-large
Avg.
HumanScore score
1.0
0.9
0.8
0.7
0.6
0.5
Finish_competition (9)
Ingestion (24)
Shoot_projectiles (21)
Origin (46)
Request (24)
Participation (31)
Locating (59)
Colonization (30)
Seeking (34)
Awareness (22)
Killing (27)
Statement (27)
Inclusion (37)
Expressing_publicly (30)
Giving (58)
Conduct (22)
Activity_start (30)
Hostile_encounter (32)
Accomplishment (41)
Dimension (38)
Leadership (39)
Becoming_aware (44)
Attack (22)
Coming_to_believe (26)
Becoming (23)
Deciding (63)
Assistance (34)
Departing (60)
Using (28)
Arrest (21)
Losing (37)
Sending (42)
Objective_influence (35)
Appointing (47)
Subjective_influence (25)
Age (53)
Verification (28)
Coming_up_with (23)
Existence (30)
Contacting (14)
Change_of_leadership (14)
Death (48)
Arriving (57)
Hiding_objects (37)
Education_teaching (27)
Making_arrangements (37)
Hunting (26)
Buildings (56)
Creating (35)
Installing (28)
Choosing (41)
Scrutiny (31)
Motion (12)
Frame


Average Hscore by Frame for llama-2_lora
Avg.
HumanScore score
1.0
0.9
0.8
0.7
0.6
0.5
Coming_to_believe (26)
Attack (22)
Subjective_influence (25)
Hiding_objects (37)
Statement (27)
Locating (59)
Participation (31)
Activity_start (30)
Making_arrangements (37)
Killing (27)
Origin (46)
Finish_competition (9)
Leadership (39)
Dimension (38)
Age (53)
Colonization (30)
Awareness (22)
Giving (58)
Losing (37)
Arriving (57)
Objective_influence (35)
Buildings (56)
Verification (28)
Shoot_projectiles (21)
Request (24)
Seeking (34)
Sending (42)
Expressing_publicly (30)
Becoming_aware (44)
Assistance (34)
Departing (60)
Deciding (63)
Ingestion (24)
Existence (30)
Accomplishment (41)
Coming_up_with (23)
Motion (12)
Hostile_encounter (32)
Appointing (47)
Inclusion (37)
Choosing (41)
Contacting (14)
Change_of_leadership (14)
Scrutiny (31)
Hunting (26)
Becoming (23)
Using (28)
Creating (35)
Conduct (22)
Arrest (21)
Installing (28)
Education_teaching (27)
Death (48)
Frame


Average Hscore by Frame for GPT-3.5
Avg.
HumanScore score
1.0
0.9
0.8
0.7
0.6
0.5
Finish_competition (9)
Killing (27)
Assistance (34)
Request (24)
Origin (46)
Losing (37)
Change_of_leadership (14)
Ingestion (24)
Education_teaching (27)
Locating (59)
Contacting (14)
Statement (27)
Awareness (22)
Sending (42)
Coming_to_believe (26)
Existence (30)
Arriving (57)
Seeking (34)
Using (28)
Hostile_encounter (32)
Hiding_objects (37)
Hunting (26)
Choosing (41)
Giving (58)
Objective_influence (35)
Inclusion (37)
Activity_start (30)
Participation (31)
Colonization (30)
Attack (22)
Verification (28)
Making_arrangements (37)
Deciding (63)
Shoot_projectiles (21)
Arrest (21)
Dimension (38)
Expressing_publicly (30)
Buildings (56)
Accomplishment (41)
Death (48)
Becoming_aware (44)
Age (53)
Departing (60)
Coming_up_with (23)
Subjective_influence (25)
Becoming (23)
Creating (35)
Scrutiny (31)
Leadership (39)
Conduct (22)
Installing (28)
Appointing (47)
Motion (12)
Frame



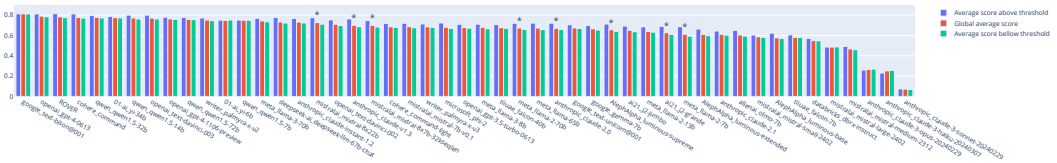


Figure 6: $f5$ complexity factor on all the examples of naturalQA

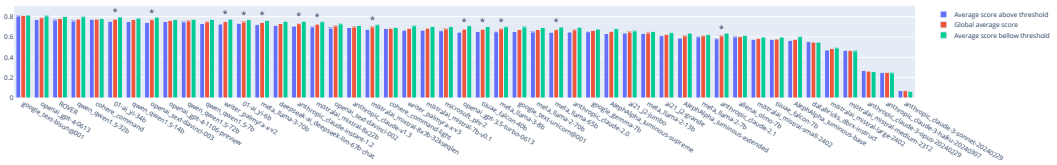


Figure 7: $f6$ complexity factor on all the examples of naturalQA