Annales Mathematicae et Informaticae
57 (2023) pp. 107–123
DOI: https://doi.org/10.33039/ami.2023.03.001
URL: https://ami.uni-eszterhazy.hu

Building machine reading comprehension model from scratch

Zijian Győző Yang, Noémi Ligeti-Nagy

Hungarian Research Centre for Linguistics
{yang.zijian.gyozo,ligeti-nagy.noemi}@nytud.hu

Abstract. In this paper, we introduce a machine reading comprehension model and how we built this model from scratch. Reading comprehension is a crucial requisite for artificial intelligence applications, such as Question-Answering systems, chatbots, virtual assistants etc. Reading comprehension task requires the highest complexity of natural language processing methods. In recent years, the transformer neural architecture could achieve the ability to solve high complexity tasks. To make these applications available in Hungarian as well it is inevitable to design a Hungarian corpus of reading comprehension so that the pretrained models can be fine-tuned on this dataset.

In our research, we have created the HuRC (Hungarian Reading Comprehension) corpus, which is the first dataset in Hungarian aiming to train, test and evaluate language models on a reading comprehension task. We built such a dataset based on the English ReCoRD corpus. This is a dataset of 120,000 examples consisting of news articles containing a passage and a close-style query, where a named entity is masked and the reference answer has to be found in a list.

Using the evaluated dataset and transformers' question-answering library, we have built the first neural machine reading comprehension models in commonsense reasoning task for Hungarian.

1. Introduction

Machine (Reading) Comprehension is the field of NLP where we teach machines to understand and answer questions using unstructured text. Reading comprehension (RC)—in contrast to information retrieval—requires integrating information and reasoning about events, entities, and their relations across a full document. Question answering is conventionally used to assess RC ability.

Submitted: April 8, 2022
Accepted: March 2, 2023
Published online: March 10, 2023

For English, there are many reading comprehension datasets, many of them included in benchmark collections (ReCoRD and MultiRC in SuperGLUE, for example, [24]) or used as a standalone benchmark dataset (SQuAD, [20]). Models trained on these datasets approximate, or sometimes even surpass human performance.

With a slight delay, but the pre-training of the transformer-based architectures on Hungarian data has begun [5, 14]. Some multilingual models, such as XLM-RoBERTa [2] and mT5 [25] also contain Hungarian data. In the future, it is expected that more models will be taught in Hungarian, and it will be necessary to measure and compare the comprehension of these models as well.

On the other hand, we still lack Hungarian datasets to train and test these models. Recently, a Hungarian benchmark kit has been developed [12] containing 4 datasets at the time of submitting this paper. Here we present one of those datasets, HuRC, which is a large-scale, partly automatically, but partly manually annotated dataset aiming to test machine reading comprehension. We trained three different models on the dataset and evaluated their performance on many ways to illustrate the difficulty of this task in Hungarian. Furthermore, using ensemble method, we could combine the advantages of our models to achieve the highest performance.

2. Related work

Current English datasets often frame the task of question answering as reading comprehension: the question is about a paragraph or a document and the answer is a span in the document.

Dzendzik et al. [4] provides a deep summary of English machine reading comprehension (MRC) datasets. Based on the answer type, they differentiate cloze answer (the question is a sentence with a missing word which has to be inserted, e.g. ReCoRD [28]), selective or multiple choice (a number of options is given, and the correct one(s) should be selected, e.g. MultiRC [9]), boolean (a yes/no answer is expected, e.g. BoolQ [1]) extractive or span extractive (the answer is a substring of the passage, e.g. SQuAD [20]) and generative or free form answer (the answer has to be generated, e.g. NarrativeQA [10]).

The DeepMind Q&A datasets [7] consist of documents from news articles from CNN and Daily Mail, 90k and 197k documents with 380k and 879k questions, respectively. News portals have begun to add summary points with each news piece in recent years, apparently to accommodate online readers' short attention spans. These summary points are not simply text extractions from the article, but rather summary points that can be used to automatically create inquiries that may require comprehension of the news story to answer. The query is built by removing an entity from the statement and asking the reader to fill in the most relevant entity from the text. In pre-processing, entities are detected and coreferenced, and the text is completely masked. This is done to avoid the model relying on external knowledge about the entities when deciding on an answer, instead relying only on

its understanding of the context.

A collection of children's books was assembled from the Project Gutenberg archives for the Children's Book Test at Facebook [8]. Each question is made up of 20 consecutive sentences from the book text, with the 21st sentence serving as the query statement. A word from the query is selected and masked. The reader has to decide which word from the text (of the chosen kind) should be used to fill the placeholder in the query. Here not merely entities are masked: named entities, common nouns, verbs and prepositions may be placeholders.

StanfordNLP created the SQuAD (Stanford Question Answering Dataset) in 2016 [20], which included over 100,000 question-answer pairs derived from Wikipedia articles. The task was to build a machine learning model to answer questions using a contextual document as input. The model would return the subset of the text most likely to answer the query when given a contextual document (free form text) and a question. The answers do not have to be entities, and no sets of candidate answers are offered. SQuAD is the first large-scale QA dataset in which answers are text spans that must be identified without any extra information. Human annotators achieved an exact match score of 82.304% and a F1-score of 91.221%. No model has been able to surpass the human results on SQuAD for 2 years. In 2018, BERT was introduced [3], and the original BERT model achieved an exact match score of 85.083% and a F1-score of 91.835%.

MultiRC (Multi-Sentence Reading Comprehension) [9] is a dataset of short paragraphs and multi-sentence questions, which are questions that may be solved by combining information from numerous paragraph phrases. The dataset was created with three main objectives in mind: i) for each question, the number of right response possibilities is not pre-determined. This eliminates the model's reliance on answer possibilities and forces them to judge the validity of each answer independently of the others; ii) It is not necessary for the correct answer(s) to be a span in the text; iii) The texts come from a variety of sources, including news, fiction, and historical documents, thus ensuring diversity across domains.

BoolQ contains 15942 examples with naturally occurring questions [1]. Each example consists of a question, a passage and an answer. The authors sampled questions from a distribution of information-seeking queries. They assume this method results in significantly more challenging examples compared to existing datasets where the text pairs (the questions or the answers) were constructed by annotators.

Kočiský et al. [10] states that existing RC datasets do not test the essential integrative aspect of reading comprehension as their questions can be solved relying upon superficial information, such as local context similarity or global term frequency. They present a novel dataset to tackle this problem. In these tasks the reader must answer questions about stories by reading entire books or movie scripts. A successful answer requires understanding the underlying narrative. There are two tasks proposed in the paper: "summaries only" and "stories only", depending on whether the human-generated summary or the full story text is used to answer the question. NarrativeQA still proves to be challenging for language models: the

SOTA result is that of Masque [15]: a Rouge score of 59.87.

Zhang et al. [28] extracted their examples (more than 120 000 entries) from the CNN/Daily Mail[1] corpus to create the Reading Comprehension with Commonsense Reasoning (ReCoRD) dataset. These news articles were divided into multiple units: passage, cloze-style query (containing the masked entity) and the reference answer. The last paragraph must contain the reference answer, a proper noun which can be found in the passage. As a reading comprehension task, this named entity is masked and the model must predict the masked entity from a list of possible entities in the provided passage, where the same entity may be expressed with multiple different surface forms, which are all correct. ReCoRD is part of the SuperGLUE benchmark [24]. The results are evaluated with max (over all mentions) token-level F1 and accuracy. The best result so far on the ReCoRD dataset is an F-score of 96.4% and an accuracy of 95.9% of the Turing NLR v5 model submitted in 2021.

Most recently, ESTER was introduced [6], which is an MRC dataset for Event Semantic Relation Reasoning. The dataset contains natural language queries to reason about the five most common event semantic relations. The current SOTA systems achieve 22.1%, 63.3%, and 83.5% for token-based exact-match, F1, and event-based HIT@1 scores, which are all significantly below human performances (36.0%, 79.6%, 100% respectively).

Natural language processing has seen spectacular progress with the application of neural network technology, in particular, the Transformer model [23]. Tasks like machine reading comprehension, can be solved with high performance, if a pre-trained language model is fine-tuned. The first breakthrough model based on transformer architecture was the BERT (abbreviation of Bidirectional Encoder Representations from Transformer) model [3]. The BERT model is pre-trained on two language modeling tasks: word masking and next sentence prediction. The first native BERT model in Hungarian was published by Nemeskey [14], named as huBERT, which is the state of the art neural language model for Hungarian.

Cross-Language Understanding (XLU) is key challenge and serves as an accelerator to the development of multilingual models. In 2020, the Facebook AI team published an article presenting XLM-RoBERTa (abbreviated as XLM-R as well) [2], which is a transformer-based multilingual masked language model. XLM-R outperforms mBERT (multilingual BERT) on cross-lingual classification in the case of languages with moderate resources available. XLM-R contains Hungarian language knowledge.

T5 (Text-To-Text Transfer Transformer) [19] is a model and framework developed by the Google research team, which offers a new perspective to solve natural language processing tasks. The T5 project applies transfer learning principles in the context of the sequence-to-sequence approach. The initial idea was that all language processing tasks (translation, question answering, classification) should be considered as a text-to-text issue, therefore the input is a text and the output will be another text. mT5 [25] extends the T5 to several languages that including Hungarian. In our research, huBERT, XLM-R and mT5 models were fine-tuned

[1] https://github.com/abisee/cnn-dailymail

for the RC task.

Generative Pre-Training (GPT) designates the concept of pre-training a language model on large datasets. The application of the GPT paradigm can foster significant advancements in natural language processing, especially in the area of classification, question-answering and investigation of semantic similarity. GPT models use a Transformer Decoder architecture. A key question behind GPT experimentation is how training on large datasets can improve the performance of language models. GPT-2 achieved significant performance in several tasks already in a zero-shot setting [18]. For Hungarian, Yang trained the first GPT-2 language models [26].

Tajti proved that using ensemble approach could achieve higher system performance [22]. He defined new voting function variants for ensemble learner committee machine algorithms which can be used as competitors of the well-known voting functions. In our research, we used the GPT-2 model as language model to combine our different fine-tuned RC models to gain higher system performance.

3. Building the HuRC Corpus

Passage

(CNN) -- A lawsuit has been filed claiming that the iconic Led Zeppelin song "Stairway to Heaven" was far from original. The suit, filed on May 31 in the United States District Court Eastern District of Pennsylvania, was brought by the estate of the late musician Randy California against the surviving members of Led Zeppelin and their record label. The copyright infringement case alleges that the Zeppelin song was taken from the single "Taurus" by the 1960s band Spirit, for whom California served as lead guitarist. "Late in 1968, a then new band named Led Zeppelin began touring in the United States, opening for Spirit," the suit states. "It was during this time that Jimmy Page, Led Zeppelin's guitarist, grew familiar with 'Taurus' and the rest of Spirit's catalog. Page stated in interviews that he found Spirit to be 'very good' and that the band's performances struck him 'on an emotional level.' "

- Suit claims similarities between two songs
- Randy California was guitarist for the group Spirit
- Jimmy Page has called the accusation "ridiculous"

(Cloze-style) Query

According to claims in the suit, "Parts of 'Stairway to Heaven,' instantly recognizable to the music fans across the world, sound almost identical to significant portions of '**X**.'"

Reference Answers

Taurus

Passage

"1968 lehetett, amikor először találkoztunk, gyakorlatilag váltottuk egymást az Omega együttesben. Tamás akkor indult el az artista pályán, miközben zenélt is. Az Omegában csak néhányszor játszottunk együtt, miután én beléptem, ő éveket töltött külföldön artistaként, aztán összefutottunk az LGT-ben, ennek már 43 éve" - idézte fel Presser Gábor.

Mint kifejtette, Somló Tamás színpadi jelenléte nagy húzóerőt jelentett a zenekar számára és zenészi képességeit mutatta az is, hogy amikor Frenreisz Károly helyett belépett az LGT-be, néhány hét alatt megtanult basszusgitározni.

A Locomotiv GT utoljára 2013 augusztusában lépett színpadra, az alsóörsi LGT-fesztiválon.

(Lead) Somló Tamás nagyszerű egyénisége, énekhangja és éneklési stílusa egészen egyedülálló volt - fogalmazott Presser Gábor, az LGT vezetője a zenész halála kapcsán.

(Cloze-style) Query

Nem ismerek olyan embert, aki Tamásra haragudott volna. Életét úgy fejezte be, ahogyan élt: utolsó fellépésére, amely talán egy hónappal ezelőtt lehetett, már nagyon nehezen tudott csak elmenni, de nem mondta le, mert Pécsett egy jótékonysági koncerten játszott beteg gyerekeknek - mondta [MASK].

Reference Answers

PER: Presser Gábor

Figure 1. A ReCoRD [28] and a HuRC sample.

We created HuRC based on ReCoRD. To create the Hungarian counterpart of

ReCoRD, we used the daily news articles from Népszabadság Online[2] that had titles and summaries as well, in addition to the main text (396 886 articles). If a component was missing from an article, it was discarded. We then selected articles consisting of 3-6 paragraphs. An important criterion was that both the main text and the query (the last paragraph) contained a proper noun.

We trained a NER model using huBERT [14] for detecting proper nouns. For training NER models, the largest Hungarian NER corpus, the NYTK-NerKor (NerKor) corpus [21] was used. NerKor contains 67,524 segments, 1,028,114 tokens and 128,168 type. To fine-tune the models, we used the code provided by huggingface transformers token classification library[3]. The following modified parameters were used: learning rate = 1e-4, batch size: 4, max sequence length: 128. As for the evaluation, the IOB-based seqeval [13] method and F-score were used. In our experiments, we trained the models with 5 epoch number. At each epoch, we have saved a checkpoint and evaluated it. Our model (the checkpoint at epoch 1) achieved an F-score of **90.18** on the test set.

As a final step, we looked for proper names which are present both in the main article and the summary. Several pairs of proper names could occur in one article. In our example (see the example on the right in Figure 1), *Presser Gábor* and *Tamás* are present in both the question and the main text. In such cases, a given article is included in the database several times, with different proper name pairs. Thus, a total of 49 782 articles of different types were selected, of which a total of 88 655 instances constitute our dataset due to the phenomenon of multiple proper name pairs. Table 1 summarizes the quantitative properties of our corpus.

Table 1. Characteristics of the corpora.

	nol.hu	Silver	Gold
Segments	396,886	88,655	80,614
Segment type	-	49,782	47,199
Token	146,816,535	27,703,631	25,218,760
Type	4,361,301	1,115,260	1,078,467
Passage avg. length (word)	(article) 330.09	249.42	215.53
Query avg. length (word)	-	63.07	63.28

Our NER model did not handle some cases as expected: Table 2 shows the phenomena we corrected. Hungarian is an agglutinative language, where the majority of syntactic relations is expressed with suffixes. Most of the incorrect cases of NER were due to the fact that the model separated the suffixes from the proper name. These had to be re-attached to the proper name afterwards. In many cases, the word had a punctuation mark attached to it, but these had to be separated from the named entity. In this sense, 6 different groups of errors were distinguished. The

[2] http://nol.hu

[3] https://github.com/huggingface/transformers/tree/master/examples/pytorch/token-classification

first group was called "all", where there was no punctuation mark on the proper noun, and the tokens in question had to be combined into one. The other cases are where some punctuation mark was either before the word ("front") or after the word ("back"). There could be more than one of these punctuation marks (1,2). In addition to problems with punctuation, there were also cases, such as *NAME-[MASK]* in Table 2, where hyphenated proper nouns were split into several parts.

Table 2. Some examples for the errors of the NER corrected manually afterwards.

	examples for NER errors		Modified
"all"	[MASK]-ak [MASK]ában Észak-[MASK]	→	[MASK]
front-1	„[MASK]tel ([MASK]mal +[MASK]nak	→	„[MASK] ([MASK] +[MASK]
back-1	[MASK]-vel, [MASK]ban) [MASK]áról:	→	[MASK], [MASK]) [MASK]:
back-2	[MASK]ához.) [MASK]ban!" [MASK]ának),	→	[MASK].) [MASK]!" [MASK]),
front-1 back-1	([MASK]ban) „[MASK]t, „[MASK]ban"	→	([MASK]) „[MASK], „[MASK]"
front-1 back-2	([MASK]ában),	→	([MASK]),

In general, the main issue was caused by the feature of our NER model; namely that it marks strictly the lemma of the named entities, however, the suffixes are also integral parts of the words in Hungarian. Furthermore, in the surface form of the words, punctuation marks may be attached to the words as well. In this task, we needed the entire named entity with suffixes, but without the punctuation marks. Thus, we had to include the suffixes in the masked words, and to detach the punctuation marks from them. We could separate the following cases:

- no punctuation mark on the word (all),
- one punctuation mark before the word (front-1),
- one punctuation mark after the word (back-1),
- two punctuation marks after the word (back-2),

- one punctuation mark before the word and one punctuation mark after the word (front-1 back-1),

- one punctuation mark before the word and two punctuation marks after the word (front-1 back-2).

We then made a few small improvements to the corpus we created. The resulting corrected dataset was checked by one annotator per 100 units. For the annotation process, we provided a self-made demo interface. The automatic masking had to be validated against the following criteria: i) whether the named entity recognition and masking was correct (i.e. *Pope Francis* was masked and not just *Francis*, and *Gödöllőre* 'Gödöllő.SUB' was masked as [MASK] instead of [MASK]re), and ii) whether the masked proper name was also present in the previous parts of the article.[4] As a result of the validation, 80 614 automatically generated, manually validated text units are in the database. The dataset is already splitted into training, validation and test sets (64 614, 8 000 and 8 000 instances, respectively).[5]

3.1. The test set

Many studies reported that a small flaw in the test set may result in very biased models and may ruin the evaluation easily (see for example [16]). As HuRC was created mainly automatically, the chance of erroneous labels or masking is certainly high. We aimed to provide a test set as clean and accurate as possible, therefore the 8 000 instances of the test set were manually validated again against the following criteria: i) whether the named entity recognition and masking was correct,[6] ii) whether each and every named entity in the passage is listed in the list of named entities found by the NER model. This manual validation required >100 work hours of an annotator.[7]

4. Training models and experiments

There are two approaches to train reading comprehension models: extractive and abstractive. In the case of extractive reading comprehension task, the model identifies the answer to a given question from a document context by 'extracting' the corresponding correct answer. This approach can only produce answers which occur in the given document. But in our task, the masked phrase could be different from the found answer in grammatical form. Thus, this method, in certain cases could only give an approximate answer and may not produce the appropriate accurate answer that fit the masked token. The second approach, the abstractive

[4] A total of 12 annotators worked on the corpus.

[5] https://github.com/nytud/HuRC, https://huggingface.co/datasets/NYTK/HuRC

[6] This is only a double-check of the first annotation process. Two erroneous masking were found in the 8 000 instances of the test set.

[7] By the time this article is submitted, 50% of the test set has been validated.

method, can solve this problem. The abstractive model, based on the given document context, can generate answer from scratch, which could fit exactly to the masked token.

The extractive model learns the start and the end indices of the answers. It calculates the probability of word i being the start/end of the answer span as a dot product between *ith* input token and *start/end vector* followed by a softmax over all of the words in the paragraph. The training objective is the log-likelihood of the correct start and end position. For this task an encoder-only transformer architecture is enough to solve the problem. It is important that the model has to be equipped with Hungarian language knowledge. Thus, in our experiment, we used the state of the art Hungarian huBERT and the XLM-RoBERTa multilingual models.

The abstractive model needs text generation feature, hence an encoder-decoder transformer architecture should be applied. The task can be solved as a text-to-text task, where the input text is the concatenation of document context and question with masked token, the output text is the answer with the correct grammatical form. Since there is no Hungarian fully pre-trained encoder-decoder model, in our experiment, we used the mT5 [25] multilingual model that contains Hungarian knowledge.

To fine-tune our models, first, we have converted our collected data into format SQuAD [20], then, for training models, we used the Question answering libraries[8] that were provided by Hugging Face.

For the extractive experiments, we used 4 x GeForce GTX 1080Ti GPU (11 GB) cards and for the abstractive experiments, we used 4 x NVIDIA A100 GPU (80 GB) cards.

We have trained three different transformer models for the neural reading comprehension (NRC) task, with the following modified hyperparameters:

- Extractive Models:
  - **huBERT** (fine-tuned huBERT model): max_seq_length=512; doc_stride=5; max_answer_length=16; learning_rate=2e-5; epoch=10; batch_size=10;
  - **XLM-R** (Fine-tuned XLM-RoBERTa base model): max_seq_length=512; doc_stride=5; max_answer_length=16; learning_rate=2e-5; epoch=10; batch_size=4;
- Abstractive Model:
  - **mT5** (Fine-tuned mT5 base model): max_seq_length=1024; doc_stride=2; max_answer_length=16; learning_rate=2e-5; epoch=10; batch_size=4;

[8] https://github.com/huggingface/transformers/tree/master/examples/pytorch/question-answering

- Ensemble Model: Using the two extractive and the abstractive models, we combined them to achieve higher output results. In this experiment, in the query, we replaced the [MASK] with the predicted answers that were generated by our NRC models, then using a Hungarian GPT-2 model, we counted the perplexity values of the different queries. The final output is the query which has the lowest perplexity. For this task we used the NYTK/text-generation-news-gpt2-small-hungarian [27] model.

5. Results and evaluation

To evaluate our models, we used different kinds of approach. First, we used the official SQuAD evaluation metrics [20], *exact match* (Match) and (macro-averaged) *F1 score* (F1) respectively. Secondly, we have used the chrF-3 and chrf-6 that are commonly used in machine translation experiments [17]. In the case of Hungarian RC task, the answer could be different only in the suffices of the word, thus a character based evaluation metric could present the more accurate performance of the models.

Table 3. Results.

	Match	F1	chrF-3/chrF-6
	Extractive		
huBERT	64.50	69.03	73.12/72.43
XLM-R	58.98	63.59	67.19/66.04
	Abstractive		
mT5	69.51	76.26	82.96/83.28
ensemble	74.04	77.57	80.54/79.97

In Table 3, you can see the results of the models. As expected, mT5 could gain higher performance than the extractive method, because the abstractive method can formulate an answer in the appropriate grammatical form as opposed to the extractive. Furthermore, using the ensemble method, we could achieve the highest exact match and F1-score results by exploiting the advantages of all models. As for the chrF values the mT5 gained the highest performance, it may be because the abstractive method can generate longer answers, resulting in higher matches at the character level, but lower efficiency at the word level. The ensemble approach could keep control this "over-generation" feature of the abstractive method.

In the case of the test set of 8000 instances, 46.35% of the results were predicted correctly (exact match) by all models at the same time and 17.34% were predicted falsely. In the remaining cases at least one model could predict correctly. In Hungarian the masked entity may differ in grammatical form from the reference names entity in the context, thus for instance, in the case of the extractive method we could not expect that the model gives an exact matched answer. Therefore a

deeper evaluation method and error analysis is needed for the erroneously predicted answers.

5.1. Special evaluation method

To understand the complexity of this task for Hungarian, first we have to understand ReCoRD's original evaluation method (as it is applied in SuperGLUE, [24]). As can be seen in Figure 2, multiple reference answers are provided for one masked named entity: these are the named entities that were found in the passage and refer to the same entity. For example, if *Manchester United*, *United* and *Manchester* are found in the text of the passage, and *United* is the masked entity in the query, all the appearances of the three named entities are listed as answers.[9] In SuperGLUE, models' performance is evaluated with max (over all mentions) token-level F1 and exact match (EM).

```
ReCoRD
 ├─ version: (string) the version of ReCoRD.
 └─ data: (list) ReCoRD examples.
    Each example has the following structure.
     ├─ id: (string) the example ID.
     ├─ source: (string) the original news source of this example.
     ├─ passage: (dict) the passage of this example.
     │   ├─ text: (string) the passage text.
     │   └─ entities: (list) the named entities in the passage.
     │      Each named entity has the following structure.
     │       ├─ start: (int) the start char index of the entity.
     │       └─ end: (int) the end char index (inclusive) of the entity.
     └─ qas: (list) queries for the corresponding passage
        Each query has the following structure.
         ├─ id: (string) the query ID.
         ├─ query: (string) the query text (the missing text span is indicated by "@placeholder").
         └─ answers: (list) the reference answers.
            Each answer has the following structure.
             ├─ start: (int) the start char index of the answer in the passage.
             ├─ end: (int) the end char index (inclusive) of the answer in the passage.
             └─ text: (string) the answer text
```

Figure 2. Format of the ReCoRD dataset.

But if we try to adapt this to Hungarian data, we face a serious problem: the masked named entity may appear earlier in the text referring to the same entity of the word, but it is very likely to have a different surface form depending on the given syntactic function it bears in the query's sentence. Staying with the previous example, *Manchester United* may appear in the passage in multiple various forms, such as *Manchester Unitedet* 'Manchester United.Acc', *Manchester Unitedről* 'Manchester United-Del' etc, and the same goes for *United* (*Unitednek* 'United.Dat', for example) and *Manchester* as well. On top of that, in the query, *United* may appear in a form that was not present in the passage, *Unitedban* 'United.Ine', for example. If we expect the models to give back a list of entities derived from the list of named entities in the passage, the list would look like *Manchester Unitedet* 'Manchester

[9]Only if they refer to the football club in the given context: if *Manchester* is present in the text as the city itself, that occurrence will not be listed among the answers.

United.Acc', *Unitednek* 'United.Dat' etc., which means word forms that definitely do not fit into the sentence in the place of the masked entity.

On the other hand, it may be quite difficult for a language model that is not inherently a generative one to pick the correct lemmas and conjugate correctly at the same time. To overcome this difficulty raised by the grammatical complexity of Hungarian, we decided to insert two lists into the instances. The first one is similar to the answer list of the ReCoRD dataset: it contains the surface forms of the named entities of the passage that refer to the same entity as the masked one in the query. However, they are only listed once: if a given surface form appears more than once in the passage, it still gets into the list once. The second list contains all the lemmas of these surface forms the suffix of the masked entity applied to them: they all fit into the sentence correctly, but are not necessarily present in the passage in their current form. We call the first list "MATCH", and the second one "MATCH_SUFFIX". We evaluate the models on both lists with F-score: this way we reward correct answers and punish incorrect ones, but a non-generative model may also have a chance to perform well on this task (on the MATCH list).

To experiment further with the evaluation options and the capabilities of the models, we have also compiled a merged list of the two lists mentioned above. By the time this paper is submitted, 25% of the test set (2000 instances) is supplied with these lists. The evaluation presented below is based on this test set of 2000 instances.

6. Discussion

As can be seen in Table 4, "MATCH" list, where the reference answers are all word forms appearing in the passage, seems to be easier for huBERT and XLM-Roberta, while mT5 and the ensemble model perform better on the more advanced list, where the word forms have to fit into the masked place perfectly (thus have to be conjugated). The best overall result is that of the ensemble model, 79.58% F-score on the "MATCH_SUFFIX" list. huBERT has the best result on the MATCH list, 76.59% F-score, which is not significantly better than the ensemble model's result on this list (76.19%).

If we look at the merged list, which is really permissive, each model's performance is better than its performance on the other two lists. The ensemble model is again better than the other 3, with an F-score of 81.82%. However, huBERT beats the abstractive mT5 on this merged list (78.09%).

For half of the instances of the test set each model could predict the correct answer. These seem to be "easy" questions for them. In these cases the surface form of the masked entity is almost always suffixless (it is the nominative form of the lemma, without any case suffix on it), and if not, the given surface form appears in the passage as well.

On the other hand, in 19.2% of the cases, none of the models could predict a correct answer (on the MATCH list – this rate is 15.15% for the MATCH_SUFFIX

Table 4. Results of the special evaluation.

	MATCH	MATCH SUFFIX	MERGED
huBERT (F1)	**76.59**	71.88	78.09
XLM-R (F1)	69.99	65.82	71.46
mT5 (F1)	71.08	76.29	77.34
ensemble (F1)	76.19	**79.58**	**81.82**
each model	49.30%	49.70%	51.05%
none of the models	19.2%	15.15%	12.75%
only huBERT	5.90%	5.55%	-
only XLM-R	2.75%	2.15%	-
only mT5	4.85%	10.35%	-

list and 12.75% for the merged list). Table 5 shows some examples with the reference answers (of the merged list) and the answers of the models.

Table 5. Some examples for wrong prediction.

Reference	huBERT	XLM-R	mT5
Kissen 'Kiss.SUP', Kiss-sel 'Kiss.INS', Kiss	Alekszandrovna	Alekszandrovna	Aleks
Balogh Levente	Varga Zoltán	Varga Zoltán	Varggh Levente
Neuer	Thiago	Dante	Ribeer
MVM	MFB	MFB	MFB
Juhászék 'Juhász.FAMPL' Juhász	Juhász kérés 'Juhász question'	Lázár János már 'Lázár János already'	Tuászsék 'Tuászs.FAMPL'[10]
Washington	Washingtonnak 'Washington.DAT'	Washingtonnak 'Washington.DAT'	Washingtonban 'Washington.INE'
Indexnek 'Index.DAT'	Index	Eximbank	Index
Törökország, Törökországnak 'Turkey.DAT', Törökországból 'Turkey.ELA'	Törökország közötti 'Turkey in.between'	Törökország közötti 'Turkey in.between'	Törökországba

In the first half of the table examples (see Table 5) show cases when models have erroneously predicted a named entity regardless of the suffixes. These cases can be seen as complete mistakes. The second half of the table shows some mixed cases: the models often hallucinate, either by adding extra (common) nouns to the

[10] *Tuászs* is not a valid Hungarian proper name.

proper name, or adding some adverbs or other function words, or by generating non-existing lemmas.

As mentioned earlier, the dataset may contain an article more than once with different named entities masked in the query. We examined the articles in the test set that appear multiple times. Models are able to predict the correct answer in the different appearances of an article. Table 6 shows cases where the article has 3 different instances in the test set with different masked named entities, and the majority of the models happen to predict the correct one in all of the cases. It is quite interesting that in the case of the last example, in two instances *Pence* / *Mike Pence* is the masked entity, and in one case the models predict it well (except for XLM-RoBERTa, which happens to insist on *Putyin*). In the other case, mT5 also hallucinates an answer (*Put Pence*). For some reasons, in one case, the models rely on the surname of the politician (*Pence*), and in the other, they all use the first name of him as well (*Mike*, and *Put* can be seen as a hallucinated first name in the case of mT5).

Table 6. Some examples for the results on articles appearing three times in the test set with different masked named entities in their query.

reference	XLM-R	huBERT	mT5	ensemble
Napi Gazdaság	Magyar Nemzet	Napi Gazdaság	Magyar Gazdaság	Napi Gazdaság
Fidesz	Magyar Nemzet	Fidesz	Fidesz	Fidesz
Fidesz	Magyar Nemzet	Fidesz	Fidesz	Fidesz
Trump	Trump	Trump	Donald	Trump
Pence, Mike Pence	Putyin	Mike Pence	Put Pence	Mike Pence
Pence, Mike Pence	Putyin	Pence	Pence	Pence

As for the important role of cloze questions in NLP, one has to mention the research of Lewis et al. [11]. Their paper is a nice and clear presentation of how cloze-stlye query databases may be exploited for a broader range of studies. First they trained a model to create cloze questions from sample documents. Afterwards, they trained a standard extractive QA model on their generated data. Their results demonstrate that self-supervised extractive QA is achievable with highly competitive results. As their training data is automatically generated, the method makes the creation of extractive QA models possible for other languages and more domains as well.

7. Conclusion

In this paper we presented the first neural machine reading comprehension models in commonsense reasoning task for Hungarian. We trained the multilingual models XLM-R and mT5, and the Hungarian model huBERT on a reading comprehension dataset (HuRC) designed based on the ReCoRD dataset. We tested to extractive

(hubERT and XLM-R) and an abstractive (mT5) model to be able to compare their performance with regard to their different architectures as well. We also implemented an ensemble method by using a Hungarian GPT-2 model to count the perplexity values of the different queries built up by the predictions of the three models. We applied a complex and thorough evaluation methodology. Our result show that the reading comprehension task in Hungarian is still challenging for the different models. Extractive models seemed to be perform better in giving back already seen surface forms of the masked named entities, but the abstractive model, mt5 beats them in conjugating the words correctly. The ensemble model reached promising results in all evaluation configurations. We hope that our results will advance neural models trained for reading comprehension task for Hungarian.

References

[1] C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, K. Toutanova: *BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions*, in: NAACL, 2019.

[2] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov: *Unsupervised Cross-lingual Representation Learning at Scale*, CoRR abs/1911.02116 (2019), arXiv: 1911.02116, URL: http://arxiv.org/abs/1911.02116.

[3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova: *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), Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186, DOI: https://doi.org/10.18653/v1/N19-1423, URL: https://aclanthology.org/N19-1423.

[4] D. Dzendzik, C. Vogel, J. Foster: *English Machine Reading Comprehension Datasets: A Survey*, in: EMNLP, 2021.

[5] Á. Feldmann, R. Hajdu, B. Indig, B. Sass, M. Makrai, I. Mittelholcz, D. Halász, Z. G. Yang, T. Váradi: *HILBERT, magyar nyelvű BERT-large modell tanítása felhő környezetben*, in: XVII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Magyarország: Szegedi Tudományegyetem, Informatikai Intézet, 2021, pp. 29–36.

[6] R. Han, I.-H. Hsu, J. Sun, J. Baylon, Q. Ning, D. Roth, N. Peng: *ESTER: A Machine Reading Comprehension Dataset for Event Semantic Relation Reasoning*, 2021, DOI: https://doi.org/10.48550/ARXIV.2104.08350, URL: https://arxiv.org/abs/2104.08350.

[7] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom: *Teaching Machines to Read and Comprehend*, in: Advances in Neural Information Processing Systems (NIPS), 2015, URL: http://arxiv.org/abs/1506.03340.

[8] F. Hill, A. Bordes, S. Chopra, J. Weston: *The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations*, CoRR abs/1511.02301 (2016).

[9] D. Khashabi, S. Chaturvedi, M. Roth, S. Upadhyay, D. Roth: *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), New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 252–262, DOI: https://doi.org/10.18653/v1/N18-1023, URL: https://aclanthology.org/N18-1023.

[10] T. Kočiský, J. Schwarz, P. Blunsom, C. Dyer, K. M. Hermann, G. Melis, E. Grefenstette: *The NarrativeQA Reading Comprehension Challenge*, Transactions of the Association for Computational Linguistics TBD (2018), TBD, URL: https://TBD.

[11] P. Lewis, L. Denoyer, S. Riedel: *Unsupervised Question Answering by Cloze Translation*, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4896–4910, doi: https://doi.org/10.18653/v1/P19-1484, url: https://aclanthology.org/P19-1484.

[12] N. Ligeti-Nagy, G. Ferenczi, E. Héja, K. Jelencsik-Mátyus, L. J. Laki, N. Vadász, Z. G. Yang, T. Váradi: *HuLU: magyar nyelvű benchmark adatbázis kiépítése a neurális nyelvmodellek kiértékelése céljából*, in: XVIII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged: JATEPress, 2022, pp. 431–446.

[13] H. Nakayama: *seqeval: A Python framework for sequence labeling evaluation*, Software available from https://github.com/chakki-works/seqeval, 2018, url: https://github.com/chakki-works/seqeval.

[14] D. M. Nemeskey: *Introducing huBERT*, in: XVII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Magyarország: Szegedi Tudományegyetem, Informatikai Intézet, 2021, pp. 3–14.

[15] K. Nishida, I. Saito, K. Nishida, K. Shinoda, A. Otsuka, H. Asano, J. Tomita: *Multi-style Generative Reading Comprehension*, 2019, arXiv: 1901.02262 [cs.CL].

[16] C. G. Northcutt, A. Athalye, J. Mueller: *Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks*, ArXiv abs/2103.14749 (2021).

[17] M. Popović: *chrF: character n-gram F-score for automatic MT evaluation*, in: Proceedings of the Tenth Workshop on Statistical Machine Translation, Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 392–395, doi: https://doi.org/10.18653/v1/W15-3049, url: https://aclanthology.org/W15-3049.

[18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever: *Language Models are Unsupervised Multitask Learners* (2019).

[19] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu: *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*, Journal of Machine Learning Research 21.140 (2020), pp. 1–67.

[20] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang: *SQuAD: 100,000+ Questions for Machine Comprehension of Text*, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392, doi: https://doi.org/10.18653/v1/D16-1264, url: https://aclanthology.org/D16-1264.

[21] E. Simon, N. Vadász: *Introducing NYTK-NerKor, A Gold Standard Hungarian Named Entity Annotated Corpus*, in: Text, Speech, and Dialogue - 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6-9, 2021, Proceedings, ed. by K. Ekstein, F. Pártl, M. Konopík, vol. 12848, Lecture Notes in Computer Science, Springer, 2021, pp. 222–234.

[22] T. G. Tajti: *New voting functions for neural network algorithms*, Annales Mathematicae et Informaticae 52 (2020), pp. 229–242.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin: *Attention is All you Need*, in: Advances in Neural Information Processing Systems 30, ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett, Curran Associates, Inc., 2017, pp. 5998–6008.

[24] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, S. R. Bowman: *SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems*, 2020, arXiv: 1905.00537 [cs.CL].

[25] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, C. Raffel: *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, Online: Association for Computational Linguistics, June 2021, pp. 483–498, doi: https://doi.org/10.18653/v1/2021.naacl-main.41, url: https://aclanthology.org/2021.naacl-main.41.

[26] Z. G. Yang: *"Az invazív medvék nem tolerálják a suzukis agressziót" - Magyar GPT-2 kísérleti modell*, in: XVIII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Magyarország: Szegedi Tudományegyetem, Informatikai Intézet, 2022, pp. 463–476.

[27] Yang Zijian Győző: *"Az invazív medvék nem tolerálják a suzukis agressziót" - Magyar GPT-2 kísérleti modell*, in: XVIII. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, Magyarország: Szegedi Tudományegyetem, Informatikai Intézet, 2022, pp. 463–476.

[28] S. Zhang, X. Liu, J. Liu, J. Gao, K. Duh, B. V. Durme: *ReCoRD: Bridging the Gap between Human and Machine Commonsense Reading Comprehension*, 2018, arXiv: 1810.12885 [cs.CL].