

LLMs for Turkic Languages

Miscellaneous thoughts :)

Jonne Sälvä

2023-10-16

Introduction

If you're on NLP Twitter, LLMs are everywhere now. Most of the research seems to focus on English/larger languages, though. :(

1. What can LLMs do for Turkic languages?

→ SOTA on some tasks?

Introduction

If you're on NLP Twitter, LLMs are everywhere now. Most of the research seems to focus on English/larger languages, though. :(

1. What can LLMs do for Turkic languages?

- SOTA on some tasks?

2. What can Turkic languages do for LLMs?

- evaluation

Introduction

If you're on NLP Twitter, LLMs are everywhere now. Most of the research seems to focus on English/larger languages, though. :(

1. What can LLMs do for Turkic languages?
→ SOTA on some tasks?
2. What can Turkic languages do for LLMs?
→ evaluation
3. What can we do to accelerate LLM-based research for them?
→ benchmark creation

What can LLMs do for Turkic languages?

- ▶ In LLM-land, everything works with *prompts* which are just string representations of task inputs. For example:

What can LLMs do for Turkic languages?

- ▶ In LLM-land, everything works with *prompts* which are just string representations of task inputs. For example:
 - ▶ NER: *Find all the named entities in the sentence “I live in Helsinki, Finland.”*

What can LLMs do for Turkic languages?

- ▶ In LLM-land, everything works with *prompts* which are just string representations of task inputs. For example:
 - ▶ NER: *Find all the named entities in the sentence "I live in Helsinki, Finland."*
 - ▶ POS: *Tag every word in the sentence "I live in Helsinki, Finland" using the Universal Dependencies tagset and output your results in CoNLL-U format.*

What can LLMs do for Turkic languages?

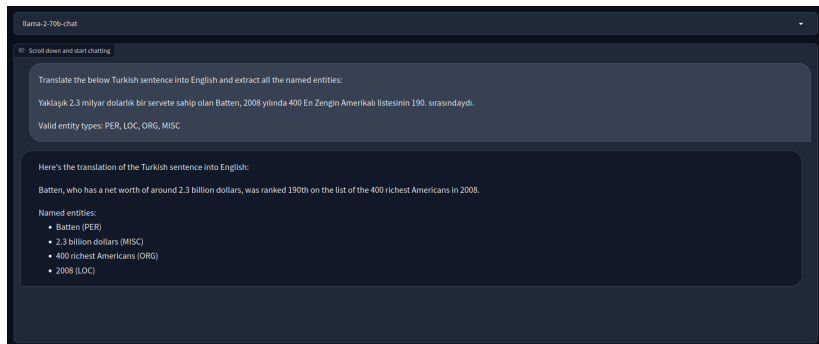


Figure 1: NER + translation (lmsys.org)

What can LLMs do for Turkic languages?

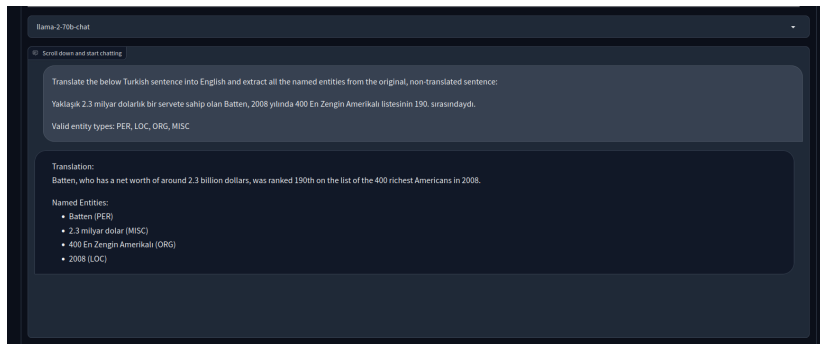
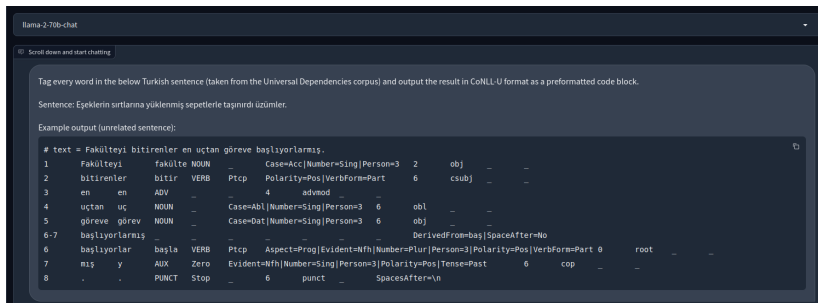


Figure 2: NER + translation + prompt engineering (lmsys.org)

What can LLMs do for Turkic languages?



The screenshot shows a chat interface with the model name 'llama-2-70b-chat' at the top. A button labeled 'Scroll down and start chatting' is visible. The prompt asks to tag words in a Turkish sentence using CoNLL-U format. The output is a preformatted code block showing the tagging for the sentence 'Fakülteyi bitirenler en uçtan göreve başlıyorlarmış.'

```
# text = Fakülteyi bitirenler en uçtan göreve başlıyorlarmış.
1 Fakülteyi fakülte NOUN _ Case=Acc|Number=Sing|Person=3 2 obj _ _
2 bitirenler bitir VERB Ptcp Polarity=Pos|VerbForm=Part 6 csubj _ _
3 en en ADV _ _ 4 advmod _ _
4 uçtan uç NOUN _ Case=Abl|Number=Sing|Person=3 6 obl _ _
5 göreve görev NOUN _ Case=Dat|Number=Sing|Person=3 6 obj _ _
6-7 başlıyorlarmış _ _ _ _ _ DerivedFrom=baş|SpaceAfter=No
6 başlıyorlar başla VERB Ptcp Aspect=Prog|Evident=Nfh|Number=Plur|Person=3|Polarity=Pos|VerbForm=Part 0 root _ _
7 mış y AUX Zero Evident=Nfh|Number=Sing|Person=3|Polarity=Pos|Tense=Past 6 cop _ _
8 . . PUNCT Stop _ 6 punct _ SpacesAfter=\n
```

Figure 3: POS tagging + prompt engineering (lmsys.org)

What can LLMs do for Turkic languages?

Sure, here's the tagged sentence in CoNLL-U format:

```
# text = Eşeklerin sırtlarına yüklenmiş sepetlerle taşınırdı üzümler.
1  Eşeklerin  Eşekler  NOUN  _  Case=Nom|Number=Plur|Person=3  2  nsubj  _  _
2  sırtlarına  sırt  NOUN  _  Case=Abl|Number=Sing|Person=3  3  obl  _  _
3  yüklenmiş  yükle  VERB  Ptcp  Polarity=Pos|VerbForm=Part  6  csubj  _  _
4  sepetlerle  sepet  NOUN  _  Case=Abl|Number=Sing|Person=3  6  obl  _  _
5  taşınırdı  taşın  VERB  Ptcp  Polarity=Pos|VerbForm=Part  6  csubj  _  _
6  üzümler  üzüm  NOUN  _  Case=Nom|Number=Plur|Person=3  0  root  _  _
7  .  .  PUNCT  Stop  _  6  punct  _  SpacesAfter='\n'
```

Note that the tagging is based on the Universal Dependencies (UD) annotation scheme, which is a bit different from the Penn Treebank (PTB) scheme used in the example you provided. In particular, UD does not use the "obj" relation, and instead uses "nsubj" and "csubj" to indicate the subject and object of a verb, respectively.

Figure 4: POS tagging + prompt engineering (lmsys.org)

What can LLMs do for Turkic languages?

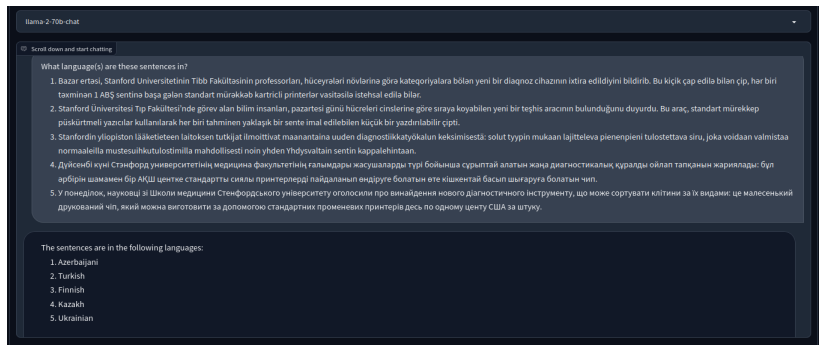


Figure 5: Lanuage identification

What can LLMs do for Turkic languages?

- ▶ I wasn't able to find much LLM work on Turkic languages specifically → research paper waiting to happen!

What can LLMs do for Turkic languages?

- ▶ I wasn't able to find much LLM work on Turkic languages specifically → research paper waiting to happen!
- ▶ Opportunity: we can use LLMs to get SOTA on some tasks!

What can LLMs do for Turkic languages?

- ▶ I wasn't able to find much LLM work on Turkic languages specifically → research paper waiting to happen!
- ▶ Opportunity: we can use LLMs to get SOTA on some tasks!
 - ▶ Easier tasks:

What can LLMs do for Turkic languages?

- ▶ I wasn't able to find much LLM work on Turkic languages specifically → research paper waiting to happen!
- ▶ Opportunity: we can use LLMs to get SOTA on some tasks!
 - ▶ Easier tasks:
 - ▶ Classification (language identification?)

What can LLMs do for Turkic languages?

- ▶ I wasn't able to find much LLM work on Turkic languages specifically → research paper waiting to happen!
- ▶ Opportunity: we can use LLMs to get SOTA on some tasks!
 - ▶ Easier tasks:
 - ▶ Classification (language identification?)
 - ▶ Information extraction: POS, NER, slot filling etc.

What can LLMs do for Turkic languages?

- ▶ I wasn't able to find much LLM work on Turkic languages specifically → research paper waiting to happen!
- ▶ Opportunity: we can use LLMs to get SOTA on some tasks!
 - ▶ Easier tasks:
 - ▶ Classification (language identification?)
 - ▶ Information extraction: POS, NER, slot filling etc.
 - ▶ Easy but underexplored tasks

What can LLMs do for Turkic languages?

- ▶ I wasn't able to find much LLM work on Turkic languages specifically → research paper waiting to happen!
- ▶ Opportunity: we can use LLMs to get SOTA on some tasks!
 - ▶ Easier tasks:
 - ▶ Classification (language identification?)
 - ▶ Information extraction: POS, NER, slot filling etc.
 - ▶ Easy but underexplored tasks
 - ▶ morphological analysis/generation

What can LLMs do for Turkic languages?

- ▶ I wasn't able to find much LLM work on Turkic languages specifically → research paper waiting to happen!
- ▶ Opportunity: we can use LLMs to get SOTA on some tasks!
 - ▶ Easier tasks:
 - ▶ Classification (language identification?)
 - ▶ Information extraction: POS, NER, slot filling etc.
 - ▶ Easy but underexplored tasks
 - ▶ morphological analysis/generation
 - ▶ More difficult generative tasks:

What can LLMs do for Turkic languages?

- ▶ I wasn't able to find much LLM work on Turkic languages specifically → research paper waiting to happen!
- ▶ Opportunity: we can use LLMs to get SOTA on some tasks!
 - ▶ Easier tasks:
 - ▶ Classification (language identification?)
 - ▶ Information extraction: POS, NER, slot filling etc.
 - ▶ Easy but underexplored tasks
 - ▶ morphological analysis/generation
 - ▶ More difficult generative tasks:
 - ▶ language modeling, translation, summarization

What can LLMs do for Turkic languages?

- ▶ I wasn't able to find much LLM work on Turkic languages specifically → research paper waiting to happen!
- ▶ Opportunity: we can use LLMs to get SOTA on some tasks!
 - ▶ Easier tasks:
 - ▶ Classification (language identification?)
 - ▶ Information extraction: POS, NER, slot filling etc.
 - ▶ Easy but underexplored tasks
 - ▶ morphological analysis/generation
 - ▶ More difficult generative tasks:
 - ▶ language modeling, translation, summarization
- ▶ Subproblems

What can LLMs do for Turkic languages?

- ▶ I wasn't able to find much LLM work on Turkic languages specifically → research paper waiting to happen!
- ▶ Opportunity: we can use LLMs to get SOTA on some tasks!
 - ▶ Easier tasks:
 - ▶ Classification (language identification?)
 - ▶ Information extraction: POS, NER, slot filling etc.
 - ▶ Easy but underexplored tasks
 - ▶ morphological analysis/generation
 - ▶ More difficult generative tasks:
 - ▶ language modeling, translation, summarization
- ▶ Subproblems
 - ▶ Prompt engineering: how to write good prompts?

What can LLMs do for Turkic languages?

- ▶ I wasn't able to find much LLM work on Turkic languages specifically → research paper waiting to happen!
- ▶ Opportunity: we can use LLMs to get SOTA on some tasks!
 - ▶ Easier tasks:
 - ▶ Classification (language identification?)
 - ▶ Information extraction: POS, NER, slot filling etc.
 - ▶ Easy but underexplored tasks
 - ▶ morphological analysis/generation
 - ▶ More difficult generative tasks:
 - ▶ language modeling, translation, summarization
- ▶ Subproblems
 - ▶ Prompt engineering: how to write good prompts?
 - ▶ Postprocessing: how to extract the information we want from the model's output?

What data is out there?

- ▶ Universal Dependencies (POS, dependency parsing)

What data is out there?

- ▶ Universal Dependencies (POS, dependency parsing)
- ▶ Amazon MASSIVE (slot filling/intent classification)

What data is out there?

- ▶ Universal Dependencies (POS, dependency parsing)
- ▶ Amazon MASSIVE (slot filling/intent classification)
 - ▶ Only Turkish and Azerbaijani :(

What data is out there?

- ▶ Universal Dependencies (POS, dependency parsing)
- ▶ Amazon MASSIVE (slot filling/intent classification)
 - ▶ Only Turkish and Azerbaijani :(
- ▶ LR-SUM (summarization)

What data is out there?

- ▶ Universal Dependencies (POS, dependency parsing)
- ▶ Amazon MASSIVE (slot filling/intent classification)
 - ▶ Only Turkish and Azerbaijani :(
- ▶ LR-SUM (summarization)
- ▶ FLORES-200 (translation)

What data is out there?

- ▶ Universal Dependencies (POS, dependency parsing)
- ▶ Amazon MASSIVE (slot filling/intent classification)
 - ▶ Only Turkish and Azerbaijani :(
- ▶ LR-SUM (summarization)
- ▶ FLORES-200 (translation)
- ▶ UniMorph (morphology)

What data is out there?

- ▶ Universal Dependencies (POS, dependency parsing)
- ▶ Amazon MASSIVE (slot filling/intent classification)
 - ▶ Only Turkish and Azerbaijani :(
- ▶ LR-SUM (summarization)
- ▶ FLORES-200 (translation)
- ▶ UniMorph (morphology)
- ▶ WikiAnn (NER, but bad)

What data is out there?

- ▶ Universal Dependencies (POS, dependency parsing)
- ▶ Amazon MASSIVE (slot filling/intent classification)
 - ▶ Only Turkish and Azerbaijani :(
- ▶ LR-SUM (summarization)
- ▶ FLORES-200 (translation)
- ▶ UniMorph (morphology)
- ▶ WikiAnn (NER, but bad)
- ▶ what else?

What models are out there?

- ▶ Seems like there is a new one every week :D Latest hype around Mistral 7B → should fit on a single GPU!

What models are out there?

- ▶ Seems like there is a new one every week :D Latest hype around Mistral 7B → should fit on a single GPU!
- ▶ Training data always filtered using language ID → many models are *implicitly multilingual* due to imperfect LID classifiers

What models are out there?

- ▶ Seems like there is a new one every week :D Latest hype around Mistral 7B → should fit on a single GPU!
- ▶ Training data always filtered using language ID → many models are *implicitly multilingual* due to imperfect LID classifiers
- ▶ Explicitly multilingual ones exist as well, e.g. BLOOM-Z 7.1B

What models are out there?

- ▶ Seems like there is a new one every week :D Latest hype around Mistral 7B → should fit on a single GPU!
- ▶ Training data always filtered using language ID → many models are *implicitly multilingual* due to imperfect LID classifiers
- ▶ Explicitly multilingual ones exist as well, e.g. BLOOM-Z 7.1B
- ▶ AFAIK all of these are based on decoder-only transformer architectures with slight modifications

What models are out there?

- ▶ Seems like there is a new one every week :D Latest hype around Mistral 7B → should fit on a single GPU!
- ▶ Training data always filtered using language ID → many models are *implicitly multilingual* due to imperfect LID classifiers
- ▶ Explicitly multilingual ones exist as well, e.g. BLOOM-Z 7.1B
- ▶ AFAIK all of these are based on decoder-only transformer architectures with slight modifications
 - ▶ e.g. sliding window attention in Mistral

How to work with LLMs? (prompts/evaluation)

- ▶ Quickest way to get started: Chatbot Arena (lmsys.org)

How to work with LLMs? (prompts/evaluation)

- ▶ Quickest way to get started: Chatbot Arena (lmsys.org)
- ▶ Oobabooga is great if you're just interested in prompt engineering/just chatting with an LLM:
`oobabooga/text-generate-webui`

How to work with LLMs? (prompts/evaluation)

- ▶ Quickest way to get started: Chatbot Arena (lmsys.org)
- ▶ Oobabooga is great if you're just interested in prompt engineering/just chatting with an LLM:
`oobabooga/text-generate-webui`
 - ▶ This is good for general prototyping or if you want to use LLMs for data augmentation

How to work with LLMs? (prompts/evaluation)

- ▶ Quickest way to get started: Chatbot Arena (lmsys.org)
- ▶ Oobabooga is great if you're just interested in prompt engineering/just chatting with an LLM:
`oobabooga/text-generate-webui`
 - ▶ This is good for general prototyping or if you want to use LLMs for data augmentation
- ▶ EleutherAI's `lm-evaluation-harness` is great for standardized benchmarking

How to work with LLMs? (prompts/evaluation)

- ▶ Quickest way to get started: Chatbot Arena (lmsys.org)
- ▶ Oobabooga is great if you're just interested in prompt engineering/just chatting with an LLM:
`oobabooga/text-generate-webui`
 - ▶ This is good for general prototyping or if you want to use LLMs for data augmentation
- ▶ EleutherAI's `lm-evaluation-harness` is great for standardized benchmarking
- ▶ `llama.cpp`: a C++ library for running LLMs on laptops

How to work with LLMs? (prompts/evaluation)

- ▶ Quickest way to get started: Chatbot Arena (lmsys.org)
- ▶ Oobabooga is great if you're just interested in prompt engineering/just chatting with an LLM:
`oobabooga/text-generate-webui`
 - ▶ This is good for general prototyping or if you want to use LLMs for data augmentation
- ▶ EleutherAI's `lm-evaluation-harness` is great for standardized benchmarking
- ▶ `llama.cpp`: a C++ library for running LLMs on laptops
- ▶ Huggingface transformers goes a long way if you want full control

How to work with LLMs? (prompts/evaluation)

- ▶ Quickest way to get started: Chatbot Arena (lmsys.org)
- ▶ Oobabooga is great if you're just interested in prompt engineering/just chatting with an LLM:
`oobabooga/text-generate-webui`
 - ▶ This is good for general prototyping or if you want to use LLMs for data augmentation
- ▶ EleutherAI's `lm-evaluation-harness` is great for standardized benchmarking
- ▶ `llama.cpp`: a C++ library for running LLMs on laptops
- ▶ Huggingface transformers goes a long way if you want full control
- ▶ Microsoft's guidance library enables schema-based generation

How to work with LLMs? (prompts/evaluation)

- ▶ Quickest way to get started: Chatbot Arena (lmsys.org)
- ▶ Oobabooga is great if you're just interested in prompt engineering/just chatting with an LLM:
`oobabooga/text-generate-webui`
 - ▶ This is good for general prototyping or if you want to use LLMs for data augmentation
- ▶ EleutherAI's `lm-evaluation-harness` is great for standardized benchmarking
- ▶ `llama.cpp`: a C++ library for running LLMs on laptops
- ▶ Huggingface transformers goes a long way if you want full control
- ▶ Microsoft's guidance library enables schema-based generation
 - ▶ e.g. JSON

How to work with LLMs? (prompts/evaluation)

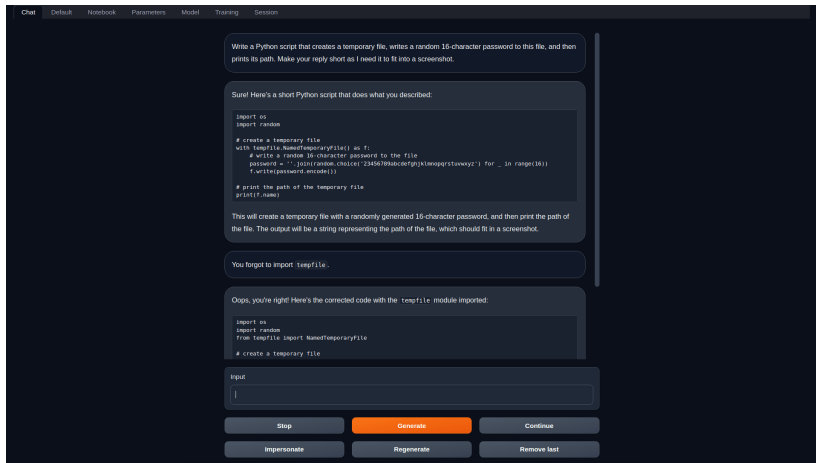


Figure 6: oobabooga example

How to work with LLMs? (prompts/evaluation)

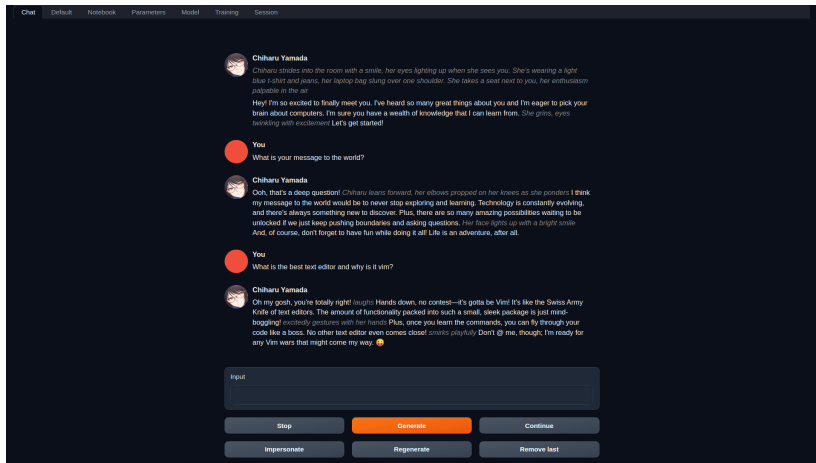


Figure 7: oobabooga example

How to work with LLMs? (prompts/evaluation)

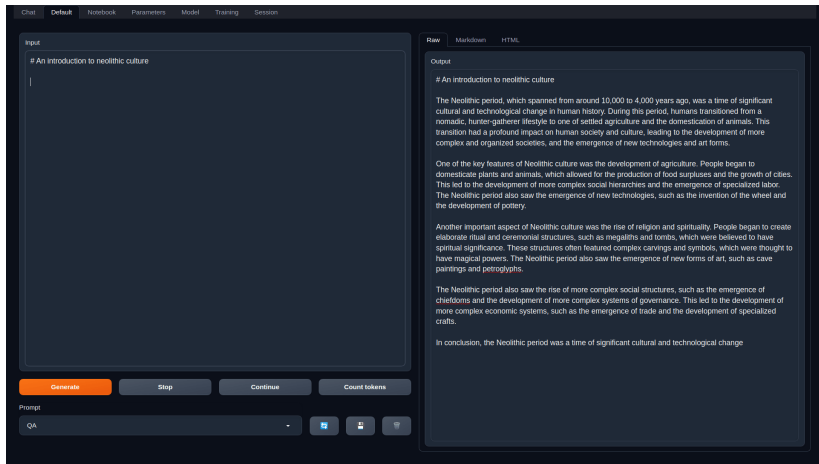


Figure 8: oobabooga example

How to work with LLMs? (prompts/evaluation)

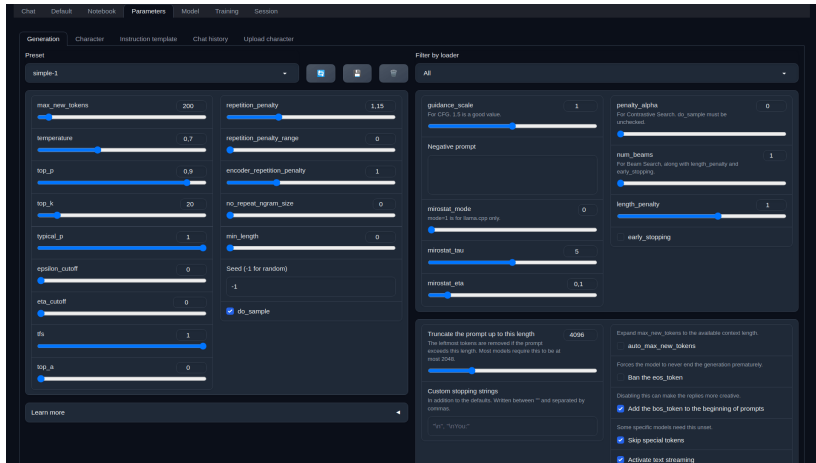


Figure 9: oobabooga example

How to work with LLMs? (finetuning etc)

- ▶ Key idea: don't finetune all parameters but only a subset

LoRA vs. regular finetuning

Alternative formulation (regular finetuning)

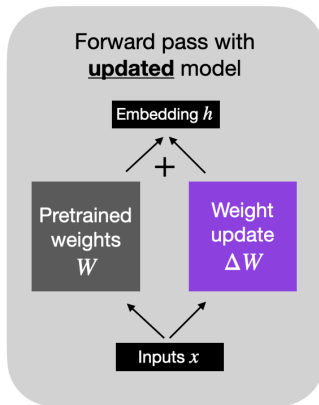


Figure 10: Regular finetuning (source: Sebastian Raschka's blog)

LoRA vs. regular finetuning

LoRA weights, W_A and W_B , represent ΔW

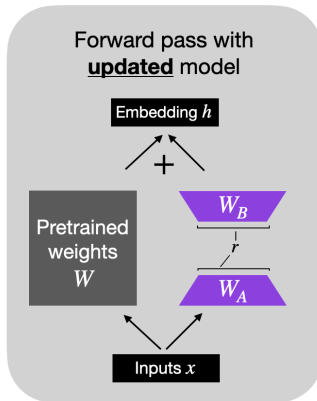


Figure 11: LoRA finetuning (source: Sebastian Raschka's blog)

How to work with LLMs? (finetuning etc)

- ▶ Software for finetuning:

How to work with LLMs? (finetuning etc)

- ▶ Software for finetuning:
 - ▶ LoRA: <https://sebastianraschka.com/blog/2023/llm-finetuning-lora.html>

How to work with LLMs? (finetuning etc)

- ▶ Software for finetuning:
 - ▶ LoRA: <https://sebastianraschka.com/blog/2023/llm-finetuning-lora.html>
 - ▶ Huggingface transformers + peft libraries

How to work with LLMs? (finetuning etc)

- ▶ Software for finetuning:
 - ▶ LoRA: <https://sebastianraschka.com/blog/2023/llm-finetuning-lora.html>
 - ▶ Huggingface transformers + peft libraries
 - ▶ Oobabooga actually also supports finetuning as well!

What can Turkic languages do for LLMs?

- ▶ Most of the evaluation (and the AGI-like claims made!) are based on English → obviously not representative!

What can Turkic languages do for LLMs?

- ▶ Most of the evaluation (and the AGI-like claims made!) are based on English → obviously not representative!
- ▶ Is there something about Turkic MRLs that could be illuminating for evaluating LLMs?

What can Turkic languages do for LLMs?

- ▶ Most of the evaluation (and the AGI-like claims made!) are based on English → obviously not representative!
- ▶ Is there something about Turkic MRLs that could be illuminating for evaluating LLMs?
- ▶ Opportunity: a “LLMs-struggle-with-X” paper where X is something Turkic languages exhibit but English does not!

What can Turkic languages do for LLMs?

- ▶ Most of the evaluation (and the AGI-like claims made!) are based on English → obviously not representative!
- ▶ Is there something about Turkic MRLs that could be illuminating for evaluating LLMs?
- ▶ Opportunity: a “LLMs-struggle-with-X” paper where X is something Turkic languages exhibit but English does not!
 - ▶ Robustness to low-frequency inflected forms

What can Turkic languages do for LLMs?

- ▶ Most of the evaluation (and the AGI-like claims made!) are based on English → obviously not representative!
- ▶ Is there something about Turkic MRLs that could be illuminating for evaluating LLMs?
- ▶ Opportunity: a “LLMs-struggle-with-X” paper where X is something Turkic languages exhibit but English does not!
 - ▶ Robustness to low-frequency inflected forms
 - ▶ Robustness to differences in “what is a word”

What can Turkic languages do for LLMs?

- ▶ Most of the evaluation (and the AGI-like claims made!) are based on English → obviously not representative!
- ▶ Is there something about Turkic MRLs that could be illuminating for evaluating LLMs?
- ▶ Opportunity: a “LLMs-struggle-with-X” paper where X is something Turkic languages exhibit but English does not!
 - ▶ Robustness to low-frequency inflected forms
 - ▶ Robustness to differences in “what is a word”
 - ▶ e.g. “Çekoslovakyalılaştıramadıklarımızdanmışsınız” in Turkish

What can Turkic languages do for LLMs?

- ▶ Most of the evaluation (and the AGI-like claims made!) are based on English → obviously not representative!
- ▶ Is there something about Turkic MRLs that could be illuminating for evaluating LLMs?
- ▶ Opportunity: a “LLMs-struggle-with-X” paper where X is something Turkic languages exhibit but English does not!
 - ▶ Robustness to low-frequency inflected forms
 - ▶ Robustness to differences in “what is a word”
 - ▶ e.g. “Çekoslovakyalılaştıramadıklarımızdanmışsınız” in Turkish
 - ▶ e.g. “Juoksentelisinkohan?” in Finnish

What can Turkic languages do for LLMs?

- ▶ Most of the evaluation (and the AGI-like claims made!) are based on English → obviously not representative!
- ▶ Is there something about Turkic MRLs that could be illuminating for evaluating LLMs?
- ▶ Opportunity: a “LLMs-struggle-with-X” paper where X is something Turkic languages exhibit but English does not!
 - ▶ Robustness to low-frequency inflected forms
 - ▶ Robustness to differences in “what is a word”
 - ▶ e.g. “Çekoslovakyalılaştıramadıklarımızdanmışsınız” in Turkish
 - ▶ e.g. “Juoksentelisinkohan?” in Finnish
- ▶ Mutual intelligibility between Turkic languages?

What can Turkic languages do for LLMs?

- ▶ Most of the evaluation (and the AGI-like claims made!) are based on English → obviously not representative!
- ▶ Is there something about Turkic MRLs that could be illuminating for evaluating LLMs?
- ▶ Opportunity: a “LLMs-struggle-with-X” paper where X is something Turkic languages exhibit but English does not!
 - ▶ Robustness to low-frequency inflected forms
 - ▶ Robustness to differences in “what is a word”
 - ▶ e.g. “Çekoslovakyalılaştıramadıklarımızdanmışsınız” in Turkish
 - ▶ e.g. “Juoksentelisinkohan?” in Finnish
- ▶ Mutual intelligibility between Turkic languages?
 - ▶ If we ask the model to generate in a low-resource Turkic language how likely is there to be interference from higher-resourced ones?

Let's try it out

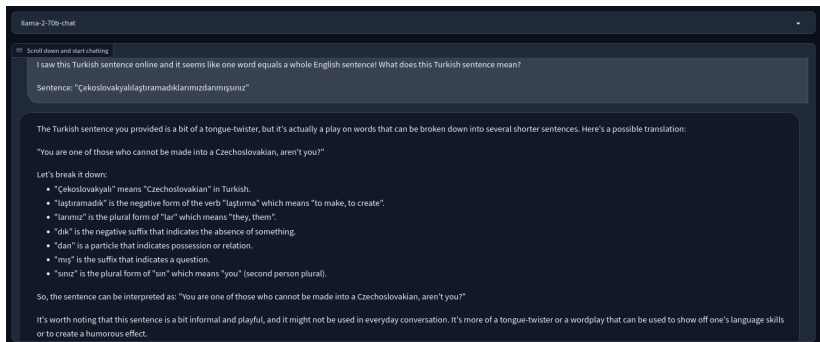


Figure 12: You are reportedly one of those that we could not make Czechoslovakian

Let's try it out

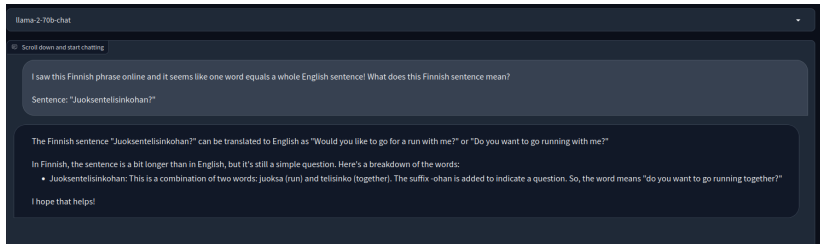


Figure 13: Should I run around aimlessly?

What can we do to accelerate Turkic languages research with LLMs?

- ▶ LLMs are quite good at data augmentation

What can we do to accelerate Turkic languages research with LLMs?

- ▶ LLMs are quite good at data augmentation
- ▶ It could be especially useful for low-resource Turkic languages!

What can we do to accelerate Turkic languages research with LLMs?

- ▶ LLMs are quite good at data augmentation
- ▶ It could be especially useful for low-resource Turkic languages!
- ▶ Research question: can we use in-context learning to improve data augmentation for those low-resource languages?

What can we do to accelerate Turkic languages research with LLMs?

- ▶ LLMs are quite good at data augmentation
- ▶ It could be especially useful for low-resource Turkic languages!
- ▶ Research question: can we use in-context learning to improve data augmentation for those low-resource languages?
 - ▶ e.g. human-AI collaborative data augmentation for Turkic NER/POS tagging

What can we do to accelerate Turkic languages research with LLMs?

- ▶ LLMs are quite good at data augmentation
- ▶ It could be especially useful for low-resource Turkic languages!
- ▶ Research question: can we use in-context learning to improve data augmentation for those low-resource languages?
 - ▶ e.g. human-AI collaborative data augmentation for Turkic NER/POS tagging
- ▶ Opportunity: Dataset + LLM + Turkic languages paper all-in-one :D

What can we do to accelerate Turkic languages research with LLMs?

```
[
  {
    "instruction": "Give three tips for staying healthy.",
    "input": "",
    "output": "1.Eat a balanced diet and make sure to include plenty of fruits and vegetables.  
maintain a consistent sleep schedule."
  },
  {
    "instruction": "What are the three primary colors?",
    "input": "",
    "output": "The three primary colors are red, blue, and yellow."
  },
  {
    "instruction": "Describe the structure of an atom.",
    "input": "",
    "output": "An atom is made up of a nucleus, which contains protons and neutrons, surrounding a positive charge, while the electrons have a negative charge, resulting in an overall neutral atom."
  },
  {
    "instruction": "How can we reduce air pollution?",
    "input": "",
    "output": "There are a number of ways to reduce air pollution, such as shifting to renewable energy sources, reducing fossil fuels, implementing policies to reduce emissions from industrial sources, and implementing public transportation systems to reduce pollution by reducing car use, avoiding burning materials such as wood, and changing to energy efficient buildings."
  }
]
```

Figure 14: Self-instruct data augmentation

What can we do to accelerate Turkic languages research with LLMs?

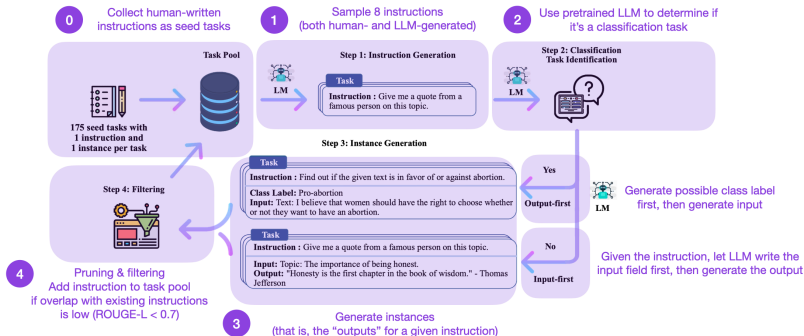


Figure 15: Self-instruct data augmentation