Evaluating the Quality of a Corpus Annotation Scheme Using Pretrained Language Models

Furkan Akkurt[*], Onur Güngör[*], Büşra Marşan[†], Tunga Güngör[*],
Balkız Öztürk Başaran[*], Arzucan Özgür[*], and Susan Üsküdarlı[*]

[*]Boğaziçi University, [†]Stanford University

Objectives

- Evaluating treebank annotation scheme quality,
- Comparing annotation schemes of two versions of the Turkish BOUN treebank [1].

Introduction

- Paradigm shift in NLP by pretrained and large language models,
- Universal Dependencies (UD) project [2] provides treebanks (i.e. sets of sentences) annotated, on a token basis, in dependency grammar for ∼150 languages,
- Annotation differences and inconsistencies due to varying linguistic theories or simple mistakes,
- Proposing a novel method using large language models to evaluate and compare treebank annotations,
- Method demonstrated by comparing two distinct versions of the Turkish BOUN treebank,

Background

- UD Turkish BOUN Treebank: ∼9k sentences from 5 domains, with 2 distinct versions: 2.8 and 2.11,
  - v2.8: semi-automated annotation, reviewed by native speakers, with limitations in expressivity due to differences between UD framework and Turkish features,
  - v2.11: manual reannotation by expert linguists, addressing representation challenges of Turkish and solving errors,
- Large Language Models (LLMs) show capabilities in tasks beyond their training, as they're able to solve unknown tasks in a zero-, one- or few-shot fashion,
- Evaluation of UD resources focuses on annotation quality and performance in downstream tasks,
- Using LLMs for evaluating treebank annotations is a relatively unexplored area.

Method

- **LLM input**: annotations for a single sentence without surface form, requesting the sentence's original text, including a one-shot example of the task,
  - Specifically lemmas, parts of speech, morphological features and dependency relations are provided in natural language,
- **LLM output**: generated text for the sentence based only on the annotations,
  - Sentence generated by the LLM is compared with the original sentence,
  - Comparison is done on both the character- and token-level.

Annotation comparison between versions

In the annotations of the sentence "Ali'den oyuna katılmasını istediler." (*They wanted Ali to join the game.*), the only difference between the versions is the feature set of the verbal noun "katılmasını" (*his/her joining*). While there are no morphological features in v2.8, v2.11 includes the feature set `Case=Acc | Number=Sing | Number[psor]=Sing | Person=3 | Person[psor]=3 | Polarity=Pos | VerbForm=Vnoun | Voice=Pass`.

Prompt for the example:

- Task explanation for the LLM
- Example input and output for the LLM
- 1st token's *lemma* is "Ali", its *part of speech* is **proper noun**, its *case* is **ablative**, its *number* is **singular number**, and its *person* is **third person**.
- 2nd token's *lemma* is "oyun", its *part of speech* is **noun**, its *case* is **dative**, its *number* is **singular number**, and its *person* is **third person**.
- 3rd token's *lemma* is "kat", its *part of speech* is verb, its *case* is **accusative**, its *number* is **singular number**, its *possessor's number* is **singular possessor**, its person is **third person**, its *possessor's person* is **third person**, it is **positive**, its *verb form* is **verbal noun**, and its *voice* is **passive voice**.
- 4th token's *lemma* is "iste", its *part of speech* is verb, its *aspect* is **perfect aspect**, its *evidentiality* is **first hand**, its *number* is **plural number**, its *person* is **third person**, it is **positive**, and its *tense* is **past tense**.
- 5th token's *lemma* is ".", and its *part of speech* is **punctuation**.
- Specifying requested format (e.g. JSON)

The output should be the sentence "Ali'den oyuna katılmasını istediler." (*They wanted Ali to join the game.*). While for v2.11, the LLM generates the sentence correctly, for v2.8, the LLM generates the sentence as "Ali'den oyuna katmak istediler." due to the missing morphological features.

Results

- Method applied to v2.8 and v2.11 on 500 random sentences, showing a consistent increase of 1.5% character-level accuracy across LLMs, using Poe API [3],
- GPT-4 [4] produces highly accurate generations, while smaller open-source models, like Llama 2 [5], lack accuracy,
  - Open-source LLMs are not trained on Turkish data,
  - Understanding Turkish linguistic features is rare in models,
- Using GPT-4:
  - Character-level accuracy (sequence matching): 90.0% for v2.8 and 91.3% for v2.11,
  - Token-level accuracy (F1): 73.8% for v2.8 and 76.9% for v2.11.

Conclusion

- Method provides insights into annotation schemes and contributes to higher quality language resources,
- Turkish BOUN treebank v2.11 shows better linguistic representation than v2.8,
- Method can be applied to other treebanks and languages,
- Code released on GitHub: `github.com/boun-tabi/eval-ud`.

References

1. Marşan, Büşra, et al. "Enhancements to the BOUN Treebank Reflecting the Agglutinative Nature of Turkish." ALTNLP. 2022.
2. de Marneffe, Marie-Catherine, et al. "Universal Dependencies." Computational Linguistics. 2021.
3. Poe API at `developer.poe.com`.
4. GPT-4 (OpenAI) at `openai.com/gpt-4`.
5. Llama 2 (Meta) at `llama.meta.com/llama2`.

Acknowledgements

This work was supported by Boğaziçi University Research Fund Grant Number 16909.

Contact Information

- `tabilab.cmpe.bogazici.edu.tr`
- `furkan.akkurt@bogazici.edu.tr`

- **Conference**: LREC-COLING 2024
- **Place**: Turin, Italy
- **Dates**: May 20-25, 2024