LLMs' morphological analyses of complex FST-generated Finnish words

Anssi Moisio[1], **Mathias Creutz**[2], and **Mikko Kurimo**[1]
[1]Department of Information and Communications Engineering, Aalto University, Finland
[2]Department of Digital Humanities, University of Helsinki, Finland
anssi.moisio@aalto.fi, mathias.creutz@helsinki.fi, mikko.kurimo@aalto.fi

Abstract

Rule-based language processing systems have been overshadowed by neural systems in terms of utility, but it remains unclear whether neural NLP systems, in practice, learn the grammar rules that humans use. This work aims to shed light on the issue by evaluating state-of-the-art LLMs in a task of morphological analysis of complex Finnish noun forms. We generate the forms using an FST tool, and they are unlikely to have occurred in the training sets of the LLMs, therefore requiring morphological generalisation capacity. We find that GPT-4-turbo has some difficulties in the task while GPT-3.5-turbo struggles and smaller models Llama2-70B and Poro-34B fail nearly completely.

1 Do neural networks learn grammar?

The debate on whether neural networks (NNs) can be accurate models of human language often revolves around the question whether NNs learn similar grammar rules as children do. In a famous instance of the debate, Rumelhart and McClelland (1986) argued that a NN can capture the implicit rules that govern how English verbs are inflected in the past tense. In a response, Pinker and Prince (1988) counter that explicit rules are indispensable to explain how children learn past tenses, and more generally to explain the psychology of language.

Neural methods have gradually become more capable of modelling varied aspects of language, which could be viewed as supporting the implicit rules argument. (For updates on the past-tense debate see Kirov and Cotterell (2018); Corkery et al. (2019); Fukatsu et al. (2024).) The most recent instances of the debate are over large language models (LLMs), whose language-generation and task-solving capabilities have surprised many. The recent debate consequently concerns modelling human language more generally instead of focusing on specific phenomena such as verb inflection. Considering the success of LLMs, it is clear that they learn some implicit rule-abiding behaviour that enables them to process and generate language competently, but it is still not clear if they learn grammar similarly to humans, or if they learn and employ some other set of rules.

Assessing grammatical knowledge learned by NNs is not straightforward, but there are at least two popular approaches. Training a classifier (called a 'probe' (Alain and Bengio, 2016) or a 'diagnostic classifier' (Hupkes et al., 2018), first developed by Shi et al. (2016); Adi et al. (2017)) to classify the internal representations of NNs has been used to inspect what aspects of grammar are encoded in them. Probing studies have found various syntactical information encoded in neural NLP systems (Jawahar et al., 2019; Tenney et al., 2018; Papadimitriou et al., 2021), but interpreting the results remains contentious (Voita and Titov, 2020; Immer et al., 2022).

The other popular method is to directly inspect a neural LM's next-unit predictions, or to train a classifier NN to predict which word is most acceptable, given sequence of previous words. In an influential work by Linzen et al. (2016), knowledge of subject-verb agreement in LSTM networks was assessed this way, and it was concluded that 'LSTMs can learn to approximate structure-sensitive dependencies fairly well'. Similar *targeted syntactic evaluation* methods, inspired by methods in psycholinguistics (e.g. Crain and Fodor (1985); Stowe (1986)), have subsequently been employed to assess the knowledge of many different grammatical phenomena in NNs, for example anaphora or negative polarity items (Marvin and Linzen, 2018; Futrell et al., 2019; Jumelet and Hupkes, 2018; Hu et al., 2020). Larger test suites such as BLiMP (Warstadt et al., 2020) or SyntaxGym (Gauthier et al., 2020) are used as benchmarks to track advances in the field.

The general conclusion has not changed much since that of Linzen et al.'s: the networks are *fairly* good at acquiring the grammar rules. Sometimes results of a single study are interpreted as evidence that the NNs have acquired a syntactical rule completely (e.g. Wilcox et al. (2023)), but a closer inspection often proves such an interpretation premature (e.g. Lan et al. (2024)). Since there is no conclusive evidence that NNs learn from text the same grammar that people use, it remains an important task to delineate the instances where NNs, and LLMs in particular, adhere to and utilise grammar, and the instances where they do not.

Designing targeted syntactic evaluation tests requires careful formulation of the sequences. For example, Wilcox et al. (2023) examined the understanding of filler-gap effects by comparing the probabilities of acceptable and unacceptable continuations for sentence pairs such as ‘I know *what* the lion devoured’ and ‘I know *that* the lion devoured’. The continuation ‘yesterday’ is assumed to be acceptable for the former but not the latter sequence. However, ‘yesterday’ could be an acceptable next word even for the latter sequence: consider the sentence ‘I know that the lion devoured yesterday’s leftovers.’ This example highlights the difficulty of designing test sentences of this sort.

Instead of inspecting the next-unit predictions or training diagnostic classifiers, in this work we ask LLMs explicitly to perform a classification task, which is possible due to the flexible text generation capacity of the LLMs. This makes the evaluation relatively unambiguous. For example, asking an LLM directly ‘Is the verb “devour” transitive or intransitive?’ does not leave much room for confounding factors. The apparent limitation of this method is that even if a model fails in an explicit classification task like this, we cannot rule out the possibility that the model nevertheless encodes perfect *implicit* knowledge of the verb and how to use it in any context. However, we make the assumption in this work that if the LLMs had learned a grammar rule as perfectly as humans, they would be able to answer the explicit questions as competently as humans. This seems justified considering the type and difficulty of, and LLMs’ performance in, other tasks used to evaluate LLMs, such as academic and professional exams (OpenAI, 2023).

This approach was also taken by Weissweiler et al. (2023), who assessed the morphological competence of GPT-3.5-turbo by asking it directly to fill in past tenses of words in a sentence, and concluded that it ‘massively underperforms purpose-built systems’. Similarly, Weller-Di Marco and Fraser (2024) took a morphologically complex word W and asked GPT-3.5-turbo questions such as ‘What is the head noun of W?’.

In this work we present LLMs directly and explicitly with a classification task to investigate the knowledge of Finnish morphology in LLMs. Although Finnish has relatively few speakers worldwide (<10 million), it is not a low-resource language, having about 32B tokens of available training texts (Luukkonen et al., 2023, 2024). Consequently, the state-of-the-art (SOTA) multilingual LLMs such as GPT-4 are fluent in Finnish, and could be expected to have a good grasp of the grammar, if the LLMs are in fact good at learning grammar from text.

2 Data and methods

Previous datasets of inflected Finnish words include the MorphyNet (Batsuren et al., 2021) and UniMorph (Kirov et al., 2016; Batsuren et al., 2022) corpora. We chose not to use data from these datasets for two reasons. Firstly, complex words comprising unusually many morphemes make it possible to assess if the systems can generalise to many types of possible inflections instead of learning only the most common inflection types. The previous datasets do not include many extremely complex word forms, but these can be generated using a finite-state transducer (FST). Secondly, since the SOTA LLMs have been trained on very large datasets harvested from the Internet, it is likely that the previously published datasets are included in their training data, which would preclude fair assessment.

We use the Omorfi tools (Pirinen, 2015; Pirinen et al., 2017) that are based on finite-state morphology (Koskenniemi, 1984; Beesley and Karttunen, 2003) to generate inflected forms of Finnish nouns. The Omorfi library includes some 500k lexemes, of which about 140k are nouns. We inflect the nouns in all possible combinations of number, grammatical case, and possessive suffix (see Table 1 for examples, and Appendix A for further details), which

BASE	+PL	+INE	+SG2 / +PL1
		laitteissa	*laitteissasi* / *laitteissamme*
laite	*laitteet*	+TRA	
		laitteiksi	*laitteiksesi* / *laitteiksemme*

Table 1: Examples of inflections of the word ‘laite’ (‘device’). PL means plural, INE and TRA are case classes, and SG2/PL1 are possessive suffixes. Inflections in each column include also those in the columns to their left.

creates about 25M word forms. A random sample of 2000 inflected nouns is used as a test set in our experiments. We are unaware of any assessment of the generation accuracy of Omorfi, so we performed manual evaluation of the first 200 words in the sample and found 6 incorrectly inflected words. We therefore estimate the generation accuracy to be around 97%, which creates an upper bound for the classification accuracy of the test set. We publish the test set and all code to reproduce the results at `https://github.com/aalto-speech/llm-morph-tests`. We note, however, that once the data is published, it is subject to the same data contamination issue as the previous datasets mentioned above—the good thing is that one can always draw a new random sample from the full set of 25M forms.

Uniform sampling of lexemes creates a bias towards low-frequency types that are correlated with regularity of the inflection (Kodner et al., 2023). We note that this is the case in our data, as we took a random sample of the lexemes, and this should be kept in mind when interpreting the results; there are probably not many irregularly inflected words, which makes the task easier. This is not an issue, however, given our research question of whether the LLMs have picked up even the most *systematic* inflection types from textual data.

Prompt:
Jäsennä taivutetut substantiivit tällä tavalla: taivutusmuoto – perusmuoto, luku, sijamuoto, omistusliite vedessämme – vesi, yksikkö, inessiivi, 1. persoonan monikko kinoksiksensa – kinos, monikko, translatiivi, 3. persoona peukalostanne – peukalo, yksikkö, elatiivi, 2. persoonan monikko huurteenani – huurre, yksikkö, essiivi, 1. persoonan yksikkö sängiltäsi – sänki, monikko, ablatiivi, 2. persoonan yksikkö koivuumme – koivu, yksikkö, illatiivi, 1. persoonan monikko kaistojaan – kaista, monikko, partitiivi, 3. persoona rehtiyksiesi – rehtiys, monikko, genetiivi, 2. persoonan yksikkö laaksoillani – laakso, monikko, adessiivi, 1. persoonan yksikkö talollenne – talo, yksikkö, allatiivi, 2. persoonan monikko kansoiltanne – kansa,

Correct answer:
monikko, ablatiivi, 2. persoonan monikko

Table 2: An example 10-shot prompt. An English translation of the first two rows is: *Parse the inflected nouns in this manner: inflected form – base form, number, grammatical case, possessive suffix.* The following rows are the examples. We use n-shot prompts with $n \in \{0, 1, 5, 10\}$, and for all n we use the same n first examples. For instance, the 5-shot prompts have the *vedessämme, kinoksiksensa, peukalostanne, huurteenani*, and *sängiltäsi* example rows.

LLMs are prompted to give a morphological analysis given an inflected form and the base form. That is, the models should give the correct number, case, and possessive suffix classes of the inflected noun. The prompt, shown in Table 2, comprises a short description of the task and the desired format, after which there are 0, 1, 5, or 10 examples of the task before the test word.

We test **GPT-4-turbo**-1106-preview (Achiam et al., 2023) (which outperformed GPT-4-0613 in preliminary experiments), **GPT-3.5-turbo**-1106, **Llama2-70B** (Touvron et al., 2023) (outperformed smaller Llama2 models and chat versions), and **Poro-34B** (Luukkonen et al., 2024), which is trained on Finnish, English, and programming code.

For Poro and Llama2, we performed a coarse tuning of the temperature parameter on a validation set, and found no large differences but 0.5 to be marginally better than the others, so we used this value in the experiments with these models. For the GPT models we found a temperature of 0.0 to yield the best results, so this value is used for GPT-4-turbo and GPT-3.5-turbo. We did not tune the top_p parameter (of *nucleus sampling*) but used the default value 1.0.

Additionally, we trained simple recurrent neural network (RNN) models to also classify words (one RNN for each category: number, case, and possessive suffix), using random samples of the FST-generated word forms as training data (excluding the test set). The aim of this comparison is to give some indication of the difficulty of the task, and to see if NNs can handle the task if they are specifically trained on this small subset of Finnish morphology. We took the RNN off the shelf of the Pytorch library[1] without tuning any of its hyperparameters. It consists of three layers of size 128.

3 Results

The rightmost plot in Figure 1 shows that besides GPT-4-turbo, the models perform poorly in the task. GPT-4-turbo is not close to perfect accuracy either, and the combined 10-shot result does not reach the result achieved by simple RNNs trained with 80k words. With training set sizes of 800, 4k, 8k, 40k, and 80k words, the RNNs achieved accuracies of 0.380, 0.765, 0.774, 0.821, and 0.840, respectively.

[1] From the tutorial at `https://pytorch.org/tutorials/intermediate/char_rnn_classification_tutorial`

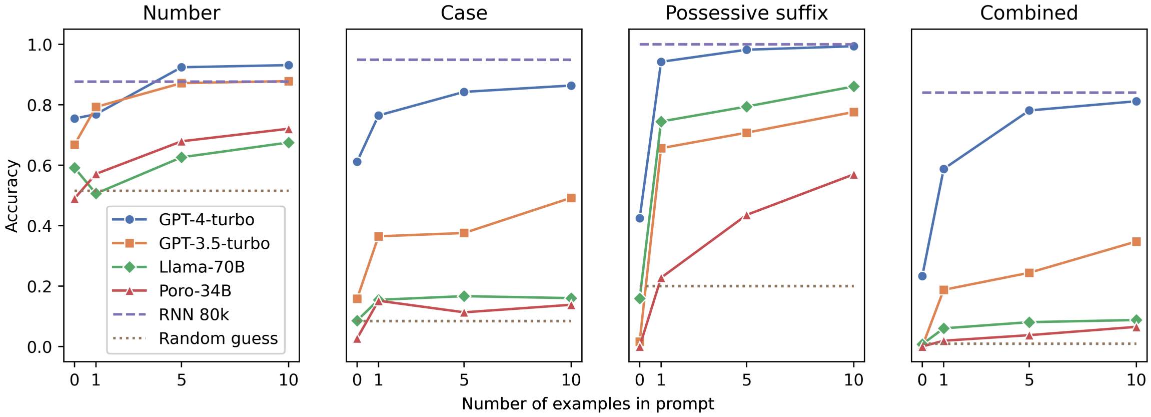
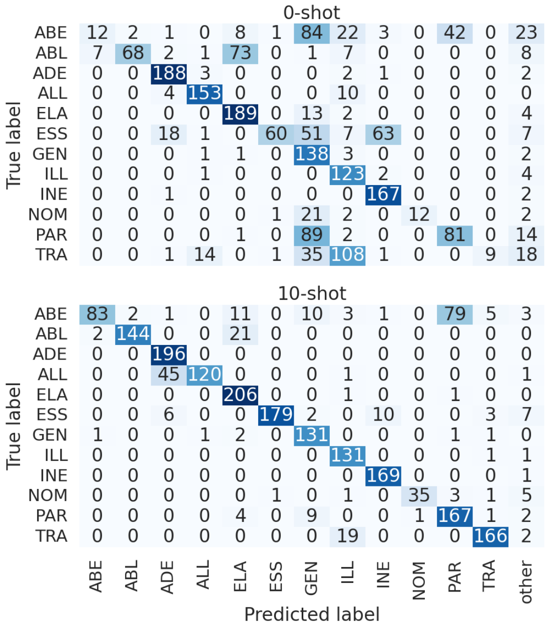


Figure 1: Results in the morphological analysis task.

The first three plots from left in Figure 1 break down the classification task into the three component classification tasks: number, case, and possessive suffix. There are some differences in the strengths of the models: Llama outperforms GPT-3.5 in the possessive suffix classification task, while GPT-3.5 performs better for other classification tasks. In number classification, Poro outperforms Llama, although Llama performs better in other tasks.

Figure 2 shows the confusion matrices for GPT-4-turbo classifications of cases for the 0-shot and 10-shot setups. From the 0-shot confusion matrix we can see that the model does predict all classes even though we did not provide it with the names of the classes we expected it to recognise. This is not surprising, since GPT-4-turbo has no difficulties if asked to inflect a Finnish word in all cases and to provide the names of the cases. It is obvious that GPT-4-turbo has a fair amount of both declarative knowledge (metalinguistic knowledge; it knows the classes) and procedural knowledge (knows how to inflect the words) of the Finnish morphology. Therefore, the challenge in this task comes presumably from the need to generalise to infrequently used, morphologically complex word forms.



0-shot

	ABE	ABL	ADE	ALL	ELA	ESS	GEN	ILL	INE	NOM	PAR	TRA	other
ABE	12	2	1	0	8	1	84	22	3	0	42	0	23
ABL	7	68	2	1	73	0	1	7	0	0	0	0	8
ADE	0	0	188	3	0	0	0	2	1	0	0	0	2
ALL	0	0	4	153	0	0	0	10	0	0	0	0	0
ELA	0	0	0	0	189	0	13	2	0	0	0	0	4
ESS	0	0	18	1	0	60	51	7	63	0	0	0	7
GEN	0	0	0	1	1	0	138	3	0	0	0	0	2
ILL	0	0	0	1	0	0	0	123	2	0	0	0	4
INE	0	0	1	0	0	0	0	0	167	0	0	0	2
NOM	0	0	0	0	0	1	21	2	0	12	0	0	2
PAR	0	0	0	0	1	0	89	2	0	0	81	0	14
TRA	0	0	1	14	0	1	35	108	1	0	0	9	18

10-shot

	ABE	ABL	ADE	ALL	ELA	ESS	GEN	ILL	INE	NOM	PAR	TRA	other
ABE	83	2	1	0	11	0	10	3	1	0	79	5	3
ABL	2	144	0	0	21	0	0	0	0	0	0	0	0
ADE	0	0	196	0	0	0	0	0	0	0	0	0	0
ALL	0	0	45	120	0	0	0	1	0	0	0	0	1
ELA	0	0	0	0	206	0	0	1	0	0	1	0	0
ESS	0	0	6	0	0	179	2	0	10	0	0	3	7
GEN	1	0	0	1	2	0	131	0	0	0	1	1	0
ILL	0	0	0	0	0	0	0	131	0	0	0	1	1
INE	0	0	0	0	0	0	0	0	169	0	0	0	1
NOM	0	0	0	0	0	1	0	1	0	35	3	1	5
PAR	0	0	0	0	4	0	9	0	0	1	167	1	2
TRA	0	0	0	0	0	0	0	19	0	0	0	166	2

Figure 2: Case label confusions of GPT-4-turbo in the 0-shot and 10-shot setups. See Appendix B for all confusion matrices.

4 Discussion

4.1 Reasons behind the errors

Most current SOTA LLMs use subword tokenisation methods such as BPE (Sennrich et al., 2016) that break down infrequent character sequences into multiple shorter tokens while keeping frequent sequences as single tokens. Intuitively, having long tokens that combine multiple morphemes into a single token could hinder the capacity to model morphology, since multiple embeddings would have to be learned for a single morpheme. Of the three model families, Poro uses the longest tokens, having an average of 3.55 characters per token in our test words, while the Llama average is 2.16 and the GPT average is 2.26. Furthermore, the average length of the last token of a word is even longer: 4.42 for Poro, 2.41 for Llama, and 2.78 for GPT. For example, the first two test words whose possessive suffix Poro classifies incorrectly and differences in the tokenisations of the different models are shown in Table 3. Both of these words have the

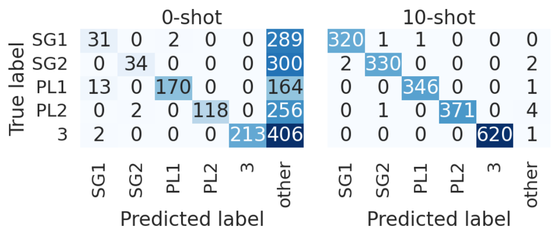


Figure 3: Possessive suffix label confusions of GPT-4-turbo in the 0-shot and 10-shot setups. See Appendix B for all confusion matrices.

Base form	*lyhty* (*lantern*)	*tarttuma* (*infection*)
Test word	*lyhtyjämme*	*tarttumassamme*
Poro tokens	ly hty jämme	t art t um assamme
Llama tokens	ly ht yj äm me	tart t um ass am me
GPT tokens	ly ht y j äm me	t art t um ass am me

Table 3: BPE tokenisations of different models.

first person plural possessive suffix, which always ends in 'me'. The possessive suffix 'me' is combined with the case morpheme (partitive 'jä' in 'lyhtyjämme' and inessive 'ssa' in 'tarttumassamme') by Poro but not by GPT or Llama. This might be one reason Poro misclassifies these words, while GPT and Llama do not, and in general why Poro lags behind the other models in the possessive suffix classification task as seen in Figure 1. The possessive suffix is simple to recognise, *if* the tokenisation is conducive to the task: a rule that checks the last two letters of the word and assigns `'ni'->SG1; 'si'->SG2; 'me'->PL1; 'ne'->PL2; else->3` would achieve 100% accuracy on our test set. Admittedly, the rule would have to be more complicated if there were also words without any possessive suffix, since these words could end in virtually any two letters: for instance, 'vesi' ('water') ends in 'si' but does not have any possessive suffix (SG2 form would be 'vetesi') as does the translative case 'vedeksi' without a possessive suffix (the translative case with SG2 suffix becomes 'vedeksesi').

Class frequencies could also explain some of the confusions. For example, GPT-4 often confuses abessive cases as partitive, seen in Figure 2. In addition to partitive being often quite similar to abessive, for example the inflected forms 'kättä' and 'kädettä' of the base 'käsi' ('hand'), partitive is also much more common than abessive: 16.2% versus 0.1% of occurrences in Kettunen (2005).

4.2 Interpretations and implications

The results suggest that despite the versatile language generation capacity of GPT-4-turbo it has not acquired the rules of Finnish morphology as completely as could be expected based on its language generation capacity. Instead, GPT-4 employs some other set of heuristics to decide the next token, although these undoubtedly overlap somewhat with grammar rules. This is hardly a surprise given the literature reviewed in Section 1, where the general conclusion tends to be that NNs rarely use grammar rules systematically, although usually fairly well.

The ineptitude of neural nets to follow grammar rules is related to systematic compositionality and inefficiency w.r.t training data set size, which are said to be weaknesses of neural nets compared to rule-based systems. Learning grammar enables *systematic compositional generalisation* (Fodor and Pylyshyn, 1988): learning a concise grammar rule such as '`the suffix -nne indicates 2nd person plural possessive form`' would enable generalising to all possible 2nd person plural forms in Finnish, obviating the need to learn word-specific associations and therefore reducing the required training corpus size. GPT-4 reaches close to 100% accuracy in this simple task of classifying possessive suffixes (RNN reaches 100%, and it is obvious that Finnish speakers would also reach 100%). However, the fact that it still sometimes classifies words ending in 'nne' as 2nd person singular instead of plural (see Figure 3) betrays its incomplete grasp of the systematic possessive suffixes in Finnish. Similar arguments apply to the other two classification tasks and the combined classification task.

5 Conclusion

We conclude that even a SOTA LLM, GPT-4-turbo, does not model Finnish morphology thoroughly enough to allow it to provide morphological analyses of rare and complex word forms with a high accuracy. Contrasting this with its impressive text generation capacity suggests that it utilises some other language processing heuristics, which clearly overlap somewhat with morphological rules since it rarely produces incorrect forms, but which preclude human-level systematic generalisation on our test set. GPT-4-turbo outperforms models such as GPT-3.5-turbo and Llama2-70B, however, by a large margin.

6 Limitations

Our experiments are limited to only one language and only four LLMs, which of course means we cannot be certain how the models perform on different languages, or how other models perform in Finnish, even though we suggest our results shed some light on general questions of grammar represented in LLMs. We also have not optimised the prompt beyond trying out a few different phrasings, so we assume some other prompt could elicit better performance especially in the 0- and 1-shot setups.

As noted in the introduction, we assess LLMs using explicit, metalinguistic questions about Finnish morphology. It is in principle possible that even if the models fail in this task, having a limited grasp of the morphological labels, they could succeed in using the words correctly in sentences and representing their meanings correctly.

7 Acknowledgements

We thank the anonymous reviewers for their insightful comments and feedback. The work was supported by the Finnish Cultural Foundation grant 00240853 and the Academy of Finland grant 337073. The computational resources were provided by Aalto ScienceIT. The use of the GPT-3.5 and GPT-4 systems via the Azure OpenAI API was provided by Aalto IT Services.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *International Conference on Learning Representations*.

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Khuyagbaatar Batsuren, Gábor Bella, and Fausto Giunchiglia. 2021. MorphyNet: a large multilingual database of derivational and inflectional morphology. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 39–48, Online. Association for Computational Linguistics.

Khuyagbaatar Batsuren, Omer Goldman, Salam Khalifa, Nizar Habash, Witold Kieraś, Gábor Bella, Brian Leonard, Garrett Nicolai, Kyle Gorman, Ryan Cotterell, Reut Tsarfaty, Ekaterina Vylomova, et al. 2022. UniMorph 4.0: Universal Morphology. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 840–855, Marseille, France. European Language Resources Association.

Kenneth R Beesley and Lauri Karttunen. 2003. Finite-state morphology: Xerox tools and techniques. *CSLI, Stanford*, pages 359–375.

Maria Corkery, Yevgen Matusevych, and Sharon Goldwater. 2019. Are we there yet? encoder-decoder neural networks as cognitive models of English past tense inflection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3868–3877, Florence, Italy. Association for Computational Linguistics.

Stephen Crain and Janet Dean Fodor. 1985. How can grammars help parsers? In *Natural Language Parsing Psychological, Computational, and Theoretical Perspectives*, pages 94–128. Cambridge University Press.

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Akiyo Fukatsu, Yuto Harada, and Yohei Oseki. 2024. Learning bidirectional morphological inflection like humans. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10249–10262, Torino, Italy. ELRA and ICCL.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Alexander Immer, Lucas Torroba Hennigen, Vincent Fortuin, and Ryan Cotterell. 2022. Probing as quantifying inductive bias. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1839–1851, Dublin, Ireland. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do language models understand anything? on the ability of LSTMs to understand negative polarity items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231, Brussels, Belgium. Association for Computational Linguistics.

Kimmo Kettunen. 2005. Sijamuodot haussa-tarvitseeko kaikkea hakutermien morfologista vaihtelua kattaa? Master's thesis.

Christo Kirov and Ryan Cotterell. 2018. Recurrent neural networks in linguistic theory: Revisiting pinker and prince (1988) and the past tense debate. *Transactions of the Association for Computational Linguistics*, 6:651–665.

Christo Kirov, John Sylak-Glassman, Roger Que, and David Yarowsky. 2016. Very-large scale parsing and normalization of Wiktionary morphological paradigms. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3121–3126, Portorož, Slovenia. European Language Resources Association (ELRA).

Jordan Kodner, Sarah Payne, Salam Khalifa, and Zoey Liu. 2023. Morphological inflection: A reality check. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6082–6101, Toronto, Canada. Association for Computational Linguistics.

Kimmo Koskenniemi. 1984. A general computational model for word-form recognition and production. In *10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics*, pages 178–181.

Nur Lan, Emmanuel Chemla, and Roni Katzir. 2024. Large language models and the argument from the poverty of the stimulus. *Linguistic Inquiry*, pages 1–56.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Risto Luukkonen, Jonathan Burdge, Elaine Zosa, Aarne Talman, Ville Komulainen, Väinö Hatanpää, Peter Sarlin, and Sampo Pyysalo. 2024. Poro 34b and the blessing of multilinguality. *arXiv preprint arXiv:2404.01856*.

Risto Luukkonen, Ville Komulainen, Jouni Luoma, Anni Eskelinen, Jenna Kanerva, Hanna-Mari Kupari, Filip Ginter, Veronika Laippala, Niklas Muennighoff, Aleksandra Piktus, Thomas Wang, Nouamane Tazi, Teven Scao, Thomas Wolf, Osma Suominen, Samuli Sairanen, Mikko Merioksa, Jyrki Heinonen, Aija Vahtola, Samuel Antao, and Sampo Pyysalo. 2023. FinGPT: Large generative models for a small language. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2710–2726, Singapore. Association for Computational Linguistics.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *arxiv 2303.08774*.

Isabel Papadimitriou, Ethan A. Chi, Richard Futrell, and Kyle Mahowald. 2021. Deep subjecthood: Higher-order grammatical features in multilingual BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2522–2532, Online. Association for Computational Linguistics.

Steven Pinker and Alan Prince. 1988. On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, 28(1-2):73–193.

Tommi A Pirinen. 2015. Development and use of computational morphology of finnish in the open source and open science era: Notes on experiences with omorfi development. *SKY Journal of Linguistics*, 28:381–393.

Tommi A Pirinen, Inari Listenmaa, Ryan Johnson, Francis M. Tyers, and Juha Kuokkala. 2017. Open morphology of finnish. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.

David E Rumelhart and James L McClelland. 1986. On learning the past tenses of English verbs. In *Parallel distributed processing: explorations in the microstructure of cognition, vol. 2: psychological and biological models*, pages 216–271. MIT Press.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.

Laurie A Stowe. 1986. Parsing wh-constructions: Evidence for on-line gap location. *Language and cognitive processes*, 1(3):227–245.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2018. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

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*.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: A benchmark of linguistic minimal pairs for English. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 409–410, New York, New York. Association for Computational Linguistics.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. Counting the bugs in ChatGPT's wugs: A multilingual investigation into the morphological capabilities of a large language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6508–6524, Singapore. Association for Computational Linguistics.

Marion Weller-Di Marco and Alexander Fraser. 2024. Analyzing the understanding of morphologically complex words in large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1009–1020, Torino, Italy. ELRA and ICCL.

Ethan Gotlieb Wilcox, Richard Futrell, and Roger Levy. 2023. Using computational models to test syntactic learnability. *Linguistic Inquiry*, pages 1–44.

A Details of the classification task

We inflect Finnish nouns in all possible combinations of number, grammatical case, and possessive suffix. Tables 4 and 5 list the classes of case and possessive suffix with examples of both singular and plural forms. We include a possessive suffix in all the forms in our test set.

Short	Name	SG e.g.	PL e.g.
ABE	abessive	talotta	taloitta
ABL	ablative	talolta	taloilta
ADE	adessive	talolla	taloilla
ALL	allative	talolle	taloille
ELA	elative	talosta	taloista
ESS	essive	talona	taloina
GEN	genitive	talon	talojen
ILL	illative	taloon	taloihin
INE	inessive	talossa	taloissa
NOM	nominative	talo	talot
PAR	partitive	taloa	taloja
TRA	translative	taloksi	taloiksi

Table 4: Finnish grammatical cases used in the experiments, with example inflections of the word 'talo' ('house'). There are three more grammatical cases in Finnish (totalling 15), but comitative and instructive are not supported by Omorfi, and accusative does not have its own unambiguous surface form, so these three are not included in our data.

Class	SG e.g. (ELA)	PL e.g. (ELA)
-	talosta	taloista
SG1	talostani	taloistani
SG2	talostasi	taloistasi
PL1	talostamme	taloistamme
PL2	talostanne	taloistanne
3	talostaan, talostansa	taloistaan, taloistansa

Table 5: Possessive suffixes in Finnish, with example inflections of the word 'talo' ('house') with the elative grammatical case 'talosta'. SG1 is 'first person singular', SG2 is 'second person singular' etc. The third person has the same forms in singular and plural, but there are synonyms such as 'talostaan' and 'talostansa'.

B Detailed results

Figures 4 through 13 show the confusion matrices of all models and in all classification tasks. Not all rows sum up to exactly to the same number: for example, in Figure 5 1-shot matrices, the SG row for Llama2 adds up to 964, whereas for Poro it adds up to 962. This is because of ambiguity in the task: for example the form 'taloni' could be singular or plural (if the case is nominative). If the a system gives one of the correct classes, the 'true label' is also assigned to that class in these confusion matrices. If the system predicts incorrectly, the 'true label' could be any of the correct classes (whichever happens to be listed last in our data).

One notable thing in the confusion matrices is that Llama2-70B does not give many nonsense answers: when one or more examples are given in the prompt, the Llama2-70B almost always gives class names, correct or incorrect, which are actual classes, leaving the 'other' column empty in Figures 5, 8, and 12. One reason that this is not the case for the GPT models is probably that GPT-4-turbo and GPT-3.5-turbo have been tuned for chat. In Microsoft Azure docs it is stated that 'Like GPT-3.5 Turbo, and older GPT-4 models, GPT-4 Turbo is optimized for chat and works well for traditional completions tasks.'[2]. GPT-4-turbo therefore often asks for clarification if it doesn't recognise the word, leading to nonsense classifications. Poro, on the other hand, is not tuned for chat, but still gives a lot of 'other' answers. This seems to be more about Poro not grasping the format that the answer should be given in, or simply not knowing which classes are possible answers.

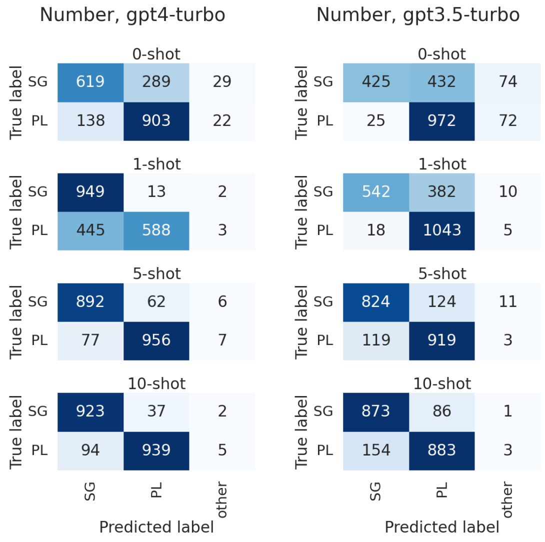


Figure 4: Confusions in the GPT-4-turbo and GPT-3.5-turbo number classification task.

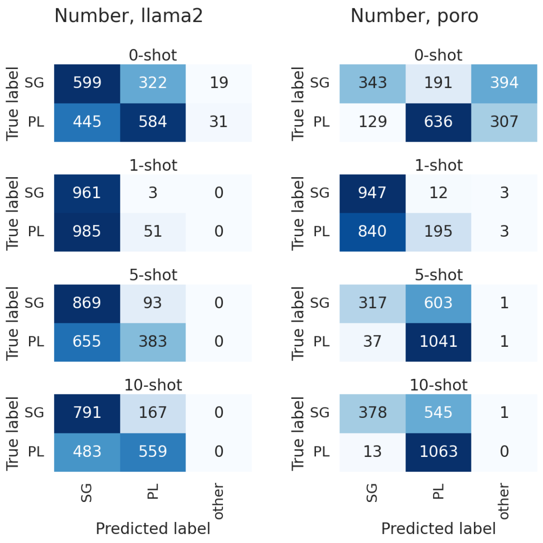
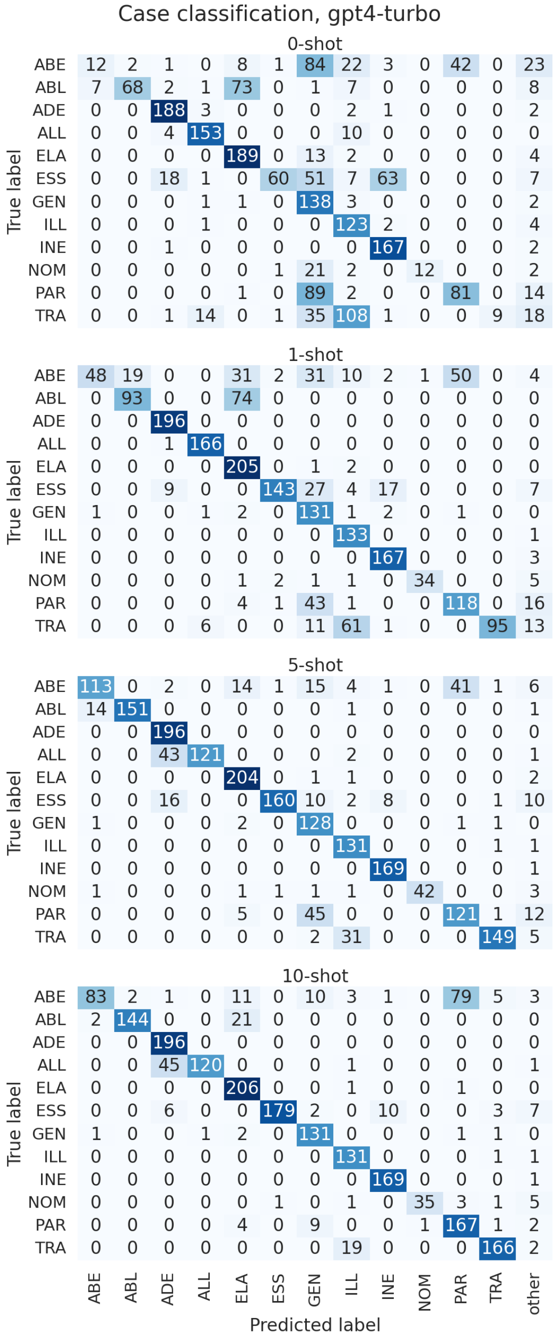


Figure 5: Confusions in the Llama2-70B and Poro-34B number classification task.

[2] https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/models



0-shot

	ABE	ABL	ADE	ALL	ELA	ESS	GEN	ILL	INE	NOM	PAR	TRA	other
ABE	12	2	1	0	8	1	84	22	3	0	42	0	23
ABL	7	68	2	1	73	0	1	7	0	0	0	0	8
ADE	0	0	188	3	0	0	0	2	1	0	0	0	2
ALL	0	0	4	153	0	0	0	10	0	0	0	0	0
ELA	0	0	0	0	189	0	13	2	0	0	0	0	4
ESS	0	0	18	1	0	60	51	7	63	0	0	0	7
GEN	0	0	0	1	1	0	138	3	0	0	0	0	2
ILL	0	0	0	1	0	0	0	123	2	0	0	0	4
INE	0	0	1	0	0	0	0	0	167	0	0	0	2
NOM	0	0	0	0	0	1	21	2	0	12	0	0	2
PAR	0	0	0	0	1	0	89	2	0	0	81	0	14
TRA	0	0	1	14	0	1	35	108	1	0	0	9	18

1-shot

	ABE	ABL	ADE	ALL	ELA	ESS	GEN	ILL	INE	NOM	PAR	TRA	other
ABE	48	19	0	0	31	2	31	10	2	1	50	0	4
ABL	0	93	0	0	74	0	0	0	0	0	0	0	0
ADE	0	0	196	0	0	0	0	0	0	0	0	0	0
ALL	0	0	1	166	0	0	0	0	0	0	0	0	0
ELA	0	0	0	0	205	0	1	2	0	0	0	0	0
ESS	0	0	9	0	0	143	27	4	17	0	0	0	7
GEN	1	0	0	1	2	0	131	1	2	0	1	0	0
ILL	0	0	0	0	0	0	0	133	0	0	0	0	1
INE	0	0	0	0	0	0	0	0	167	0	0	0	3
NOM	0	0	0	0	1	2	1	1	0	34	0	0	5
PAR	0	0	0	0	4	1	43	1	0	0	118	0	16
TRA	0	0	0	6	0	0	11	61	1	0	0	95	13

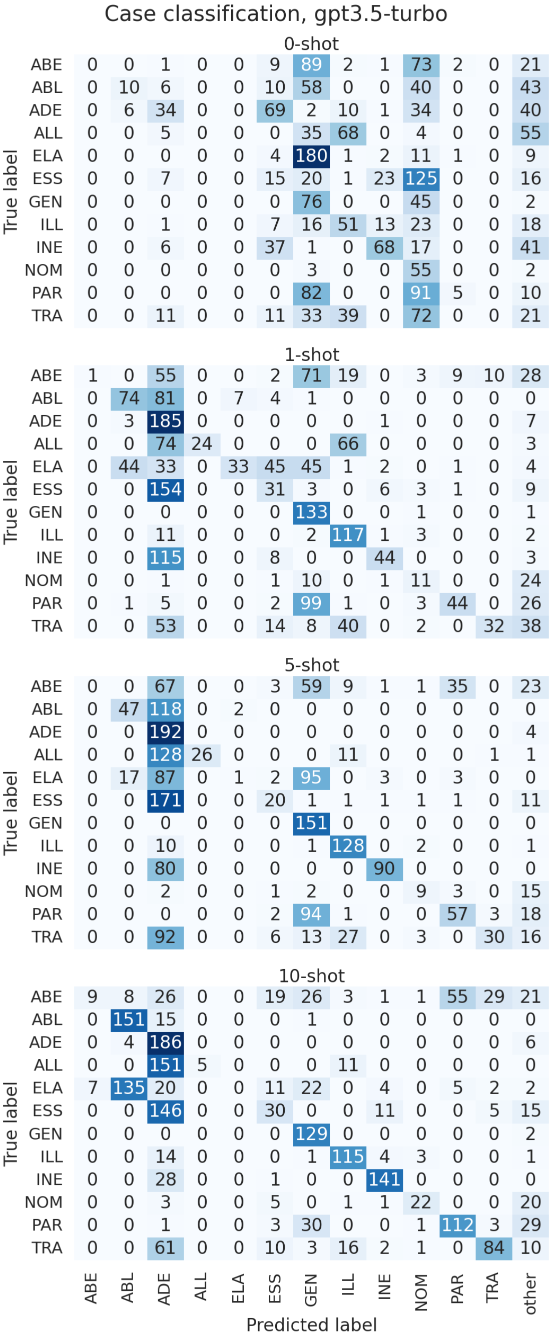
5-shot

	ABE	ABL	ADE	ALL	ELA	ESS	GEN	ILL	INE	NOM	PAR	TRA	other
ABE	113	0	2	0	14	1	15	4	1	0	41	1	6
ABL	14	151	0	0	0	0	0	1	0	0	0	0	1
ADE	0	0	196	0	0	0	0	0	0	0	0	0	0
ALL	0	0	43	121	0	0	0	2	0	0	0	0	1
ELA	0	0	0	0	204	0	1	1	0	0	0	0	2
ESS	0	0	16	0	0	160	10	2	8	0	0	1	10
GEN	1	0	0	0	2	0	128	0	0	0	1	1	0
ILL	0	0	0	0	0	0	0	131	0	0	0	1	1
INE	0	0	0	0	0	0	0	0	169	0	0	0	1
NOM	1	0	0	0	1	1	1	1	0	42	0	0	3
PAR	0	0	0	0	5	0	45	0	0	0	121	1	12
TRA	0	0	0	0	0	0	2	31	0	0	0	149	5

10-shot

	ABE	ABL	ADE	ALL	ELA	ESS	GEN	ILL	INE	NOM	PAR	TRA	other
ABE	83	2	1	0	11	0	10	3	1	0	79	5	3
ABL	2	144	0	0	21	0	0	0	0	0	0	0	0
ADE	0	0	196	0	0	0	0	0	0	0	0	0	0
ALL	0	0	45	120	0	0	0	1	0	0	0	0	1
ELA	0	0	0	0	206	0	0	1	0	0	1	0	0
ESS	0	0	6	0	0	179	2	0	10	0	0	3	7
GEN	1	0	0	1	2	0	131	0	0	0	1	1	0
ILL	0	0	0	0	0	0	0	131	0	0	0	1	1
INE	0	0	0	0	0	0	0	0	169	0	0	0	1
NOM	0	0	0	0	0	1	0	1	0	35	3	1	5
PAR	0	0	0	0	4	0	9	0	0	1	167	1	2
TRA	0	0	0	0	0	0	0	19	0	0	0	166	2

Figure 6: Confusions of GPT-4-turbo in the case classification task.



0-shot

	ABE	ABL	ADE	ALL	ELA	ESS	GEN	ILL	INE	NOM	PAR	TRA	other
ABE	0	0	1	0	0	9	89	2	1	73	2	0	21
ABL	0	10	6	0	0	10	58	0	0	40	0	0	43
ADE	0	6	34	0	0	69	2	10	1	34	0	0	40
ALL	0	0	5	0	0	0	35	68	0	4	0	0	55
ELA	0	0	0	0	0	4	180	1	2	11	1	0	9
ESS	0	0	7	0	0	15	20	1	23	125	0	0	16
GEN	0	0	0	0	0	0	76	0	0	45	0	0	2
ILL	0	0	1	0	0	7	16	51	13	23	0	0	18
INE	0	0	6	0	0	37	1	0	68	17	0	0	41
NOM	0	0	0	0	0	0	3	0	0	55	0	0	2
PAR	0	0	0	0	0	0	82	0	0	91	5	0	10
TRA	0	0	11	0	0	11	33	39	0	72	0	0	21

1-shot

	ABE	ABL	ADE	ALL	ELA	ESS	GEN	ILL	INE	NOM	PAR	TRA	other
ABE	1	0	55	0	0	2	71	19	0	3	9	10	28
ABL	0	74	81	0	7	4	1	0	0	0	0	0	0
ADE	0	3	185	0	0	0	0	0	1	0	0	0	7
ALL	0	0	74	24	0	0	0	66	0	0	0	0	3
ELA	0	44	33	0	33	45	45	1	2	0	1	0	4
ESS	0	0	154	0	0	31	3	0	6	3	1	0	9
GEN	0	0	0	0	0	0	133	0	0	1	0	0	1
ILL	0	0	11	0	0	0	2	117	1	3	0	0	2
INE	0	0	115	0	0	8	0	0	44	0	0	0	3
NOM	0	0	1	0	0	1	10	0	1	11	0	0	24
PAR	0	1	5	0	0	2	99	1	0	3	44	0	26
TRA	0	0	53	0	0	14	8	40	0	2	0	32	38

5-shot

	ABE	ABL	ADE	ALL	ELA	ESS	GEN	ILL	INE	NOM	PAR	TRA	other
ABE	0	0	67	0	0	3	59	9	1	1	35	0	23
ABL	0	47	118	0	2	0	0	0	0	0	0	0	0
ADE	0	0	192	0	0	0	0	0	0	0	0	0	4
ALL	0	0	128	26	0	0	0	11	0	0	0	1	1
ELA	0	17	87	0	1	2	95	0	3	0	3	0	0
ESS	0	0	171	0	0	20	1	1	1	1	1	0	11
GEN	0	0	0	0	0	0	151	0	0	0	0	0	0
ILL	0	0	10	0	0	0	1	128	0	2	0	0	1
INE	0	0	80	0	0	0	0	0	90	0	0	0	0
NOM	0	0	2	0	0	1	2	0	0	9	3	0	15
PAR	0	0	0	0	0	2	94	1	0	0	57	3	18
TRA	0	0	92	0	0	6	13	27	0	3	0	30	16

10-shot

	ABE	ABL	ADE	ALL	ELA	ESS	GEN	ILL	INE	NOM	PAR	TRA	other
ABE	9	8	26	0	0	19	26	3	1	1	55	29	21
ABL	0	151	15	0	0	0	1	0	0	0	0	0	0
ADE	0	4	186	0	0	0	0	0	0	0	0	0	6
ALL	0	0	151	5	0	0	0	11	0	0	0	0	0
ELA	7	135	20	0	0	11	22	0	4	0	5	2	2
ESS	0	0	146	0	0	30	0	0	11	0	0	5	15
GEN	0	0	0	0	0	0	129	0	0	0	0	0	2
ILL	0	0	14	0	0	0	1	115	4	3	0	0	1
INE	0	0	28	0	0	1	0	0	141	0	0	0	0
NOM	0	0	3	0	0	5	0	1	1	22	0	0	20
PAR	0	0	1	0	0	3	30	0	0	1	112	3	29
TRA	0	0	61	0	0	10	3	16	2	1	0	84	10

Figure 7: Confusions of GPT-3.5-turbo in the case classification task.

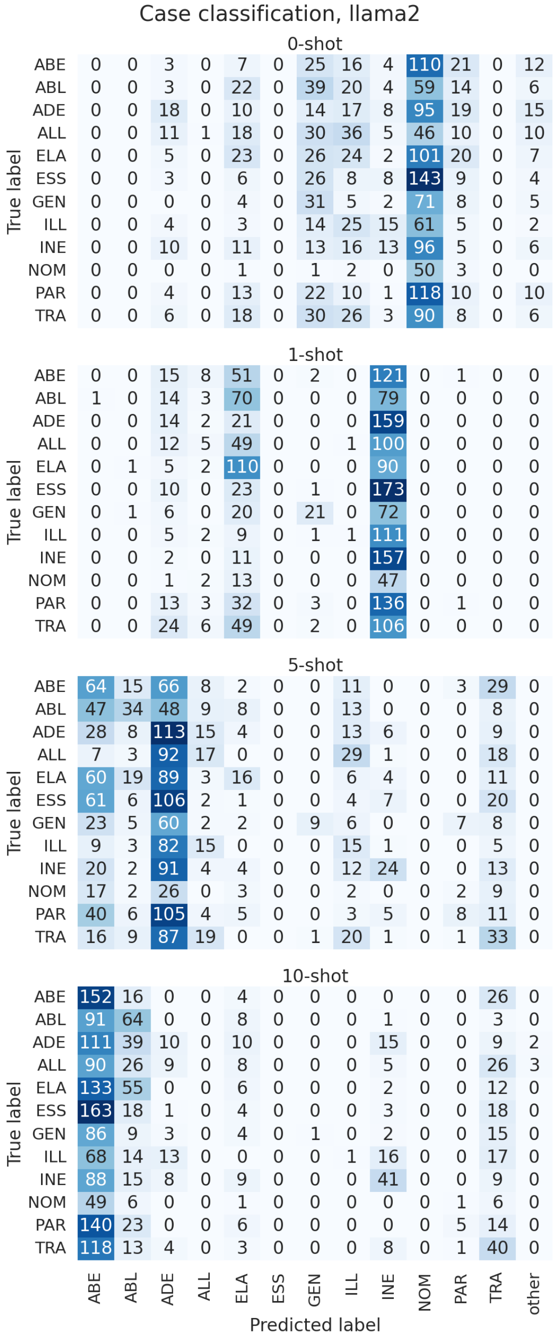


Figure 8: Confusions of Llama2-70B in the case classification task.

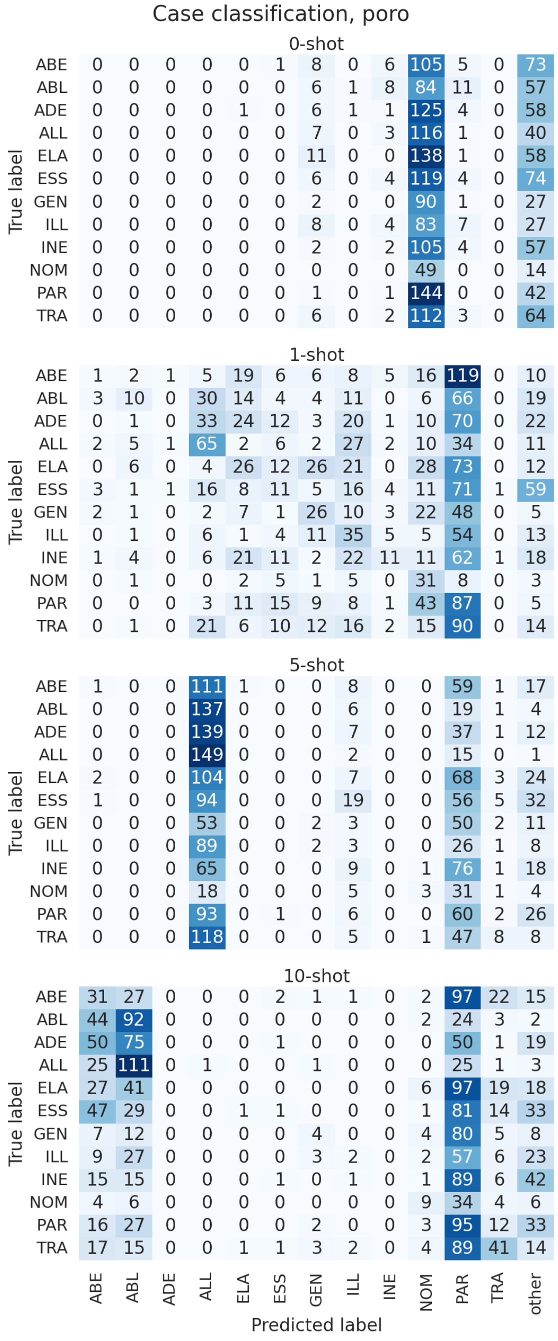
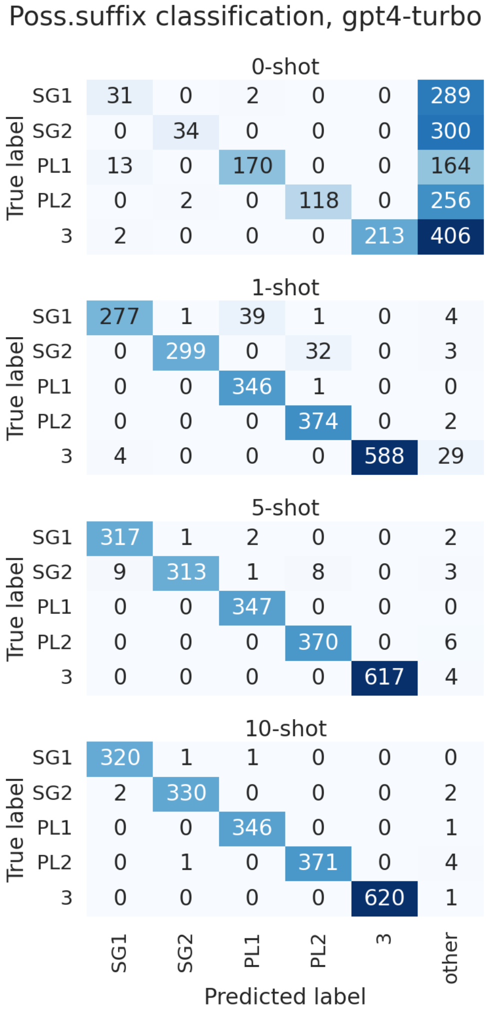


Figure 9: Confusions of Poro-34B in the case classification task.



Poss.suffix classification, gpt4-turbo

0-shot

True label \ Predicted label	SG1	SG2	PL1	PL2	3	other
SG1	31	0	2	0	0	289
SG2	0	34	0	0	0	300
PL1	13	0	170	0	0	164
PL2	0	2	0	118	0	256
3	2	0	0	0	213	406

1-shot

True label \ Predicted label	SG1	SG2	PL1	PL2	3	other
SG1	277	1	39	1	0	4
SG2	0	299	0	32	0	3
PL1	0	0	346	1	0	0
PL2	0	0	0	374	0	2
3	4	0	0	0	588	29

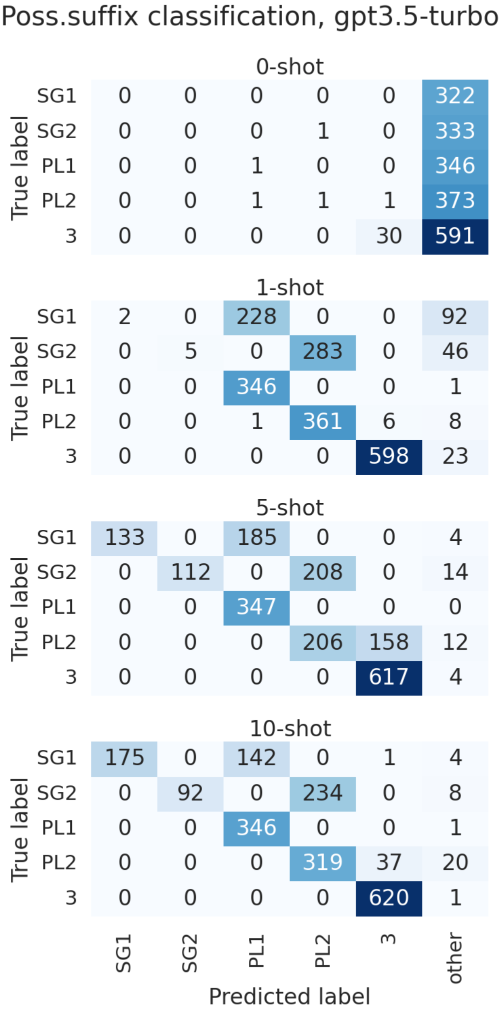
5-shot

True label \ Predicted label	SG1	SG2	PL1	PL2	3	other
SG1	317	1	2	0	0	2
SG2	9	313	1	8	0	3
PL1	0	0	347	0	0	0
PL2	0	0	0	370	0	6
3	0	0	0	0	617	4

10-shot

True label \ Predicted label	SG1	SG2	PL1	PL2	3	other
SG1	320	1	1	0	0	0
SG2	2	330	0	0	0	2
PL1	0	0	346	0	0	1
PL2	0	1	0	371	0	4
3	0	0	0	0	620	1

Figure 10: Confusions of GPT-4-turbo in the possessive suffix classification task.



Poss.suffix classification, gpt3.5-turbo

0-shot

True label \ Predicted label	SG1	SG2	PL1	PL2	3	other
SG1	0	0	0	0	0	322
SG2	0	0	0	1	0	333
PL1	0	0	1	0	0	346
PL2	0	0	1	1	1	373
3	0	0	0	0	30	591

1-shot

True label \ Predicted label	SG1	SG2	PL1	PL2	3	other
SG1	2	0	228	0	0	92
SG2	0	5	0	283	0	46
PL1	0	0	346	0	0	1
PL2	0	0	1	361	6	8
3	0	0	0	0	598	23

5-shot

True label \ Predicted label	SG1	SG2	PL1	PL2	3	other
SG1	133	0	185	0	0	4
SG2	0	112	0	208	0	14
PL1	0	0	347	0	0	0
PL2	0	0	0	206	158	12
3	0	0	0	0	617	4

10-shot

True label \ Predicted label	SG1	SG2	PL1	PL2	3	other
SG1	175	0	142	0	1	4
SG2	0	92	0	234	0	8
PL1	0	0	346	0	0	1
PL2	0	0	0	319	37	20
3	0	0	0	0	620	1

Figure 11: Confusions of GPT-3.5-turbo in the possessive suffix classification task.

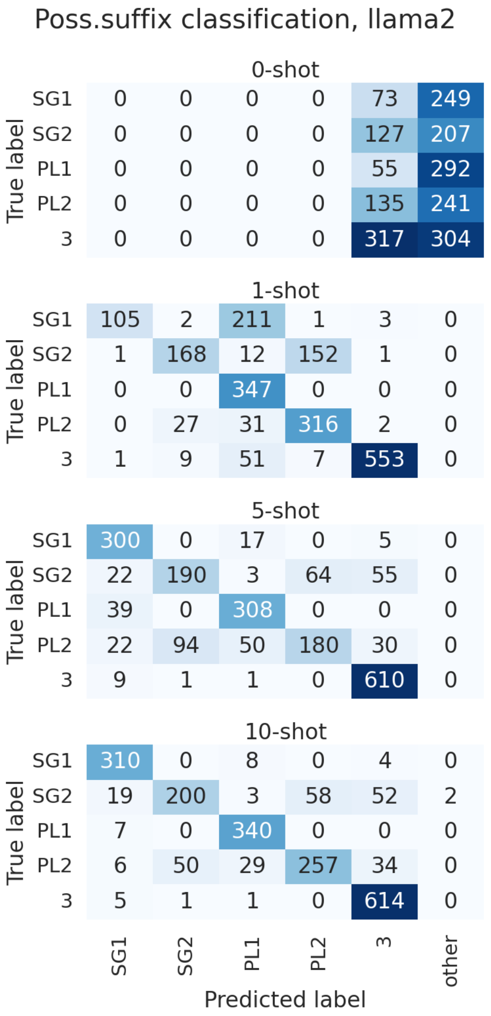


Figure 12: Confusions of Llama2-70B in the possessive suffix classification task.

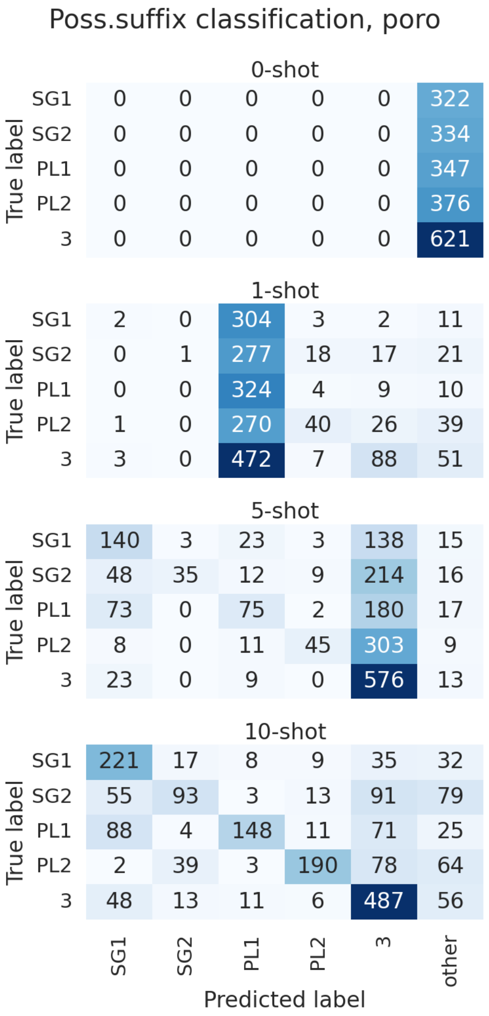


Figure 13: Confusions of Poro-34B in the possessive suffix classification task.