TRAINING AN LLM FOR GRAMMATICALITY JUDGEMENTS IN ROMANIAN

ȘERBAN HARTULAR

Universitatea din București

hartular@gmail.com

Keywords: large language models, grammaticality judgements, language acquisition, grammatical agreement, Romanian language

Cuvinte-cheie: modele ale limbajului, gramaticalitate, achiziția limbajului, acordul gramatical, limba română

1. INTRODUCTION

Large language models (hereafter LLMs) have evolved to the point where they are capable of producing cohesive, grammatical text (Reinhart et al. 2025). The question this paper proposes to examine is whether this ability to generate grammatically correct text translates into the ability to discriminate between grammatical and ungrammatical sentences – in other words, whether an LLM is able to render grammaticality judgements. This question is of interest from two perspectives. The first is that the answer would provide some insight into the (generally opaque) internal workings of an LLM. A positive answer would indicate that a large language model subsumes, in some form, a model of the grammar of the natural language it outputs text in, while a negative answer, coupled with the observation that LLMs "prefer specific grammatical structures and struggle to match the stylistic variation present in human communication" (Reinhart et al. 2025:1), would imply that LLMs' model of natural language is partial, incomplete. The second is that grammaticality judgements are, for humans at least, the basis for a range of meta-linguistic abilities – the study of a given language's grammar (or structure more generally) begins with being able to discern what is allowed and what is disallowed in that language. A positive answer would indicate that similar abilities are not impossible in the case of LLMs.

However, eliciting grammaticality judgements from LLMs is not, as we will show, entirely straightforward. To circumvent the difficulties that commonly arise in interrogating LLMs as to the grammaticality of a sentence, we carried out an experiment where an existing LLM was re-trained (at the fine-tuning level) to

SCL, LXXVII, 2026, nr. 1, Bucureşti, p. 77–98

DOI: 10.59277/SCL.2026.1.05

respond solely with information regarding the grammaticality of that sentence, in the Romanian language. Out of practical considerations, this experiment only targeted grammatical agreement as a source of ungrammaticality. The fine-tuning took two forms, based on the expected output from the model: *labelers* are trained to respond with "1" if the input sentence is grammatical and with "0" if the input sentence is ungrammatical, and *correctors* are trained to echo the input sentence if it is grammatical but return a modified (corrected) sentence if the input sentence is ungrammatical.

The performance of these two types of derived LLMs proved to be very different: the *labelers* were incapable of reliably distinguishing between grammatical and ungrammatical sentences, while the *correctors* exhibited very respectable performance by machine model standards (F-scores greater than 0.9), although we will argue that this performance is not impressive compared to that of a human proofreader.

This paper is structured as follows: Section 2 will describe some of the challenges faced in obtaining grammaticality judgements from LLMs, and the interactional models we will use to circumvent these challenges. Section 3 will consider the computational power of LLMs, and whether they can reasonably be expected to recognize that a sentence belongs to a natural language. Our conclusion is that, despite their theoretically low computational power, they have the potential to do so for a useful subset of the sentences belonging to a natural language. Section 4 describes how the training and evaluation data was generated. Section 5 describes the performance of the various versions of derived LLMs in distinguishing between grammatical and ungrammatical sentences. It will also examine the patterns that the errors of the best-performing model we generated fall into. Section 6 will present the conclusions of the paper and discuss some inherent limitations of LLMs in terms of language acquisition.

2. ELICITING GRAMMATICALITY JUDGEMENTS FROM LLMS

The simplest way of obtaining a grammaticality judgement from an LLM (here and hereafter we will be discussing *transformer* architecture LLMs, such as ChatGPT, Llama, etc.) is to simply ask it whether a sentence is grammatical or not – that is, to treat it like an informant. This has some disadvantages. Studies on more than a few hundred samples (e.g. Dentella et al. 2023) report rather poor performance and a bias in favor of answering "yes". Also, obtaining the best results requires prompt engineering (Liu et al. 2023).

It also appears that an LLM's answers will depend on its interpretation of the term "grammatical." *OpenLLM-Ro/RoLlama3.1-8b-Instruct*, a Romanian-language LLM used in this experiment, was asked whether a sentence that contained a

disagreement of gender at one point is correct from a grammatical point of view[1]. The LLM answered as follows:

> *Propoziția este corectă din punct de vedere gramatical, dar are unele erori gramaticale și erori de sintaxă.*
> 'The sentence is correct from a grammatical point of view, but has some grammatical errors and errors of syntax.'

It then proposed three changes to the sentence, one that actually addressed the ungrammaticality and two that were stylistic re-writes (replacing a word or expression with another one). This response betrays a fundamental misunderstanding of the notion of grammaticality. Grammaticality is, ideally, a binary judgement reflecting the well-formedness of a syntactic structure, irrespective of its semantic content (Chomsky 1957), let alone stylistic considerations.

These problems that arise when interrogating an LLM regarding the grammaticality of a sentence are analogous to the problems field linguists face when interviewing speakers: the interviewers want their informants to provide answers based on their innate linguistic abilities, while the informants are tempted to answer based on what school or society has taught them is proper speech. Similarly, in this case, we do not wish to know what the LLM understands by "grammatical." Rather, the question is whether or not an LLM's capacity to generate grammatical output translates into the ability to discriminate between grammatical and ungrammatical sentences.

To circumvent this problem, one should consider that an LLM is the result of two different training stages: the pretraining and the (fine)tuning (Jurafsky & Martin 2026:158-160). The first training process, the *pretraining*, generates a base form of the LLM, that is, a probabilistic model of the language, derived from enormous amounts of text. This training stage is unsupervised: the parameters of the model are adjusted based on the way words are distributed in the training corpus in order to obtain the best possible prediction of what the next word in a sequence of words would be, with no indication as to the correctness or incorrectness of the input data.

The figure below illustrates the behavior of a base LLM (*OpenLLM-Ro/RoLlama2-7b-Base*) – the product of the first stage of training only, trained on Romanian language data. Given an initial context *Mihai Eminescu a fost* 'Mihai Eminescu was', the model will output what it considers the most probable next word. This word is appended to the existing context, and the LLM outputs the next word based on the updated context, resulting in the final output *Mihai Eminescu a*

[1] The original prompt read: *Este corect din punct de vedere gramatical următorul enunț? "Întemeiat pe vaga doctrină a drepturilor omului, detestabilul pașoptism interzice orice atingere al individului numai pentru motivul că are aparență de om, fără a se întreba dacă aparține unei corporații tinere."* The ungrammaticality lies in the form of the maculine special genitival particle *al*, which should have had the feminine form *a*.

fost poetul național al României 'Mihai Eminescu was the national poet of Romania'.

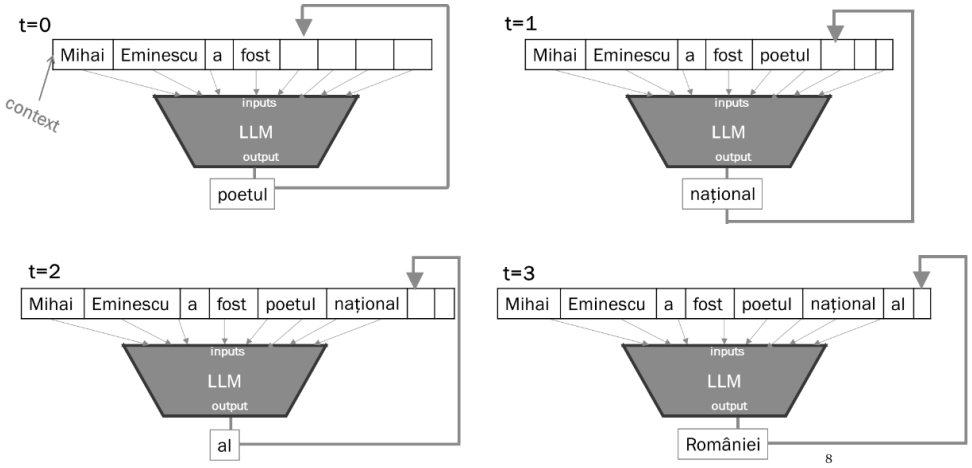

Figure 1. Base LLM Behavior

The second stage is the *fine-tuning*, whose purpose is to shape the LLM's interaction with the user. This training stage is supervised: the training data consists of pairs of prompts and expected responses. The LLM adjusts a subset of its parameters until it mimics the expected responses provided in the training data[2]. The earlier, confusing answer regarding the grammaticality of a sentence, with its three "helpful" suggestions, is the product of the LLM's fine-tuning.

To avoid this type of response and obtain grammaticality judgements that are based on the model's fundamental representation of language, the current experiment consisted of re-fine-tuning an existing LLM to generate two new LLMs whose response to an input is determined strictly by the grammaticality of the input. The first is a *labeler*, which answers "0" if the sentence is ungrammatical and "1" if the sentence is grammatical. The second is a *corrector*, which returns a corrected sentence when it detects an error of grammar in the input, and returns an output identical to the input if the input is grammatically correct.

Both these proposed LLMs are therefore *language acceptors*, that is, automata that recognize whether a string belongs to a language. The *labeler* explicitly reports whether the input belongs to the language or not. The *corrector* is an implicit acceptor: its grammaticality judgement is obtained by comparing the input to the output. For the purpose of comparing the performance of these models, we will consider the output of a *corrector* to be "1" (grammatical) if the output is

[2] The tuning stage may comprise more than one type of fine-tuning. However, unlike the pretraining, all these training processes are supervised.

identical to the input and "0" (ungrammatical) if the output is different from the input, whether or not the output consists of an appropriate correction in the latter case.

3. THE COMPUTATIONAL POWER OF LLMS

Since these proposed LLMs are re-fine-tuned to function as automata that accept a language, whether or not they have the computational power necessary to parse natural language is a relevant question. In Chomsky's hierarchy of grammars, the grammar of natural languages is at least context-free, and some languages have features that can only be described by context-sensitive grammars (Shieber 1985). Since each type of grammar in Chomsky's hierarchy is equivalent to a type of abstract automaton (Hopcroft et al. 2006), the question of whether an LLM has the necessary computational power to parse natural language can be answered by determining the LLM's position in the hierarchy of equivalent abstract automata.

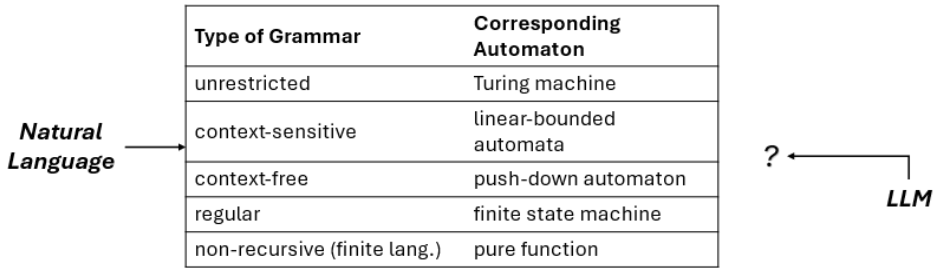

Figure 2. Grammar and Automaton Hierarchies

In addition to the standard grammar-automaton hierarchy, we included non-recursive grammars at the bottom of this hierarchy of grammars, that is, grammars that whose production rules don't allow recursion. These grammars can only generate finite languages (Sipser 2012:106), i.e., languages comprising a finite number of sentences, and, consequently, there is an upper bound to the length of a sentence belonging to that language. The simplest automaton that can recognize a finite language is a pure function – a function that has no state, takes a fixed number of inputs (equal to the maximum length of the strings belonging to the language), and generates an output based on solely these inputs. Since pure functions can recognize finite languages but cannot recognize higher order, non-finite languages, we consider the pure function the equivalent automaton to non-recursive grammars[3].

[3] See §3.2 below for the construction of a pure function based on a finite language.

It should be noted that every higher order automaton has a pure function at its core – the transfer function – that dictates the automaton's behavior with regard to the various forms of persistent storage it uses (its state, stack, readable/writeable tape, etc.).

1.1. LLMS AND CHOMSKY'S HIERARCHY

Opinion regarding transformer-architecture LLMs' position in the hierarchy of automata is not unanimous. Hahn (2020) offers a proof that LLMs cannot process regular languages. Delétang et al. (2023) conducted experiments with formal languages that showed LLMs fail at regular tasks when inputs are long, and that increasing the training data and size of the LLM is "insufficient for it to 'climb the Chomsky hierarchy'". Li et al. (2024) show that LLMs can process context-free formal languages if a stack is attached to them, while Schuurmans et al. (2024) show that LLMs are equivalent to Turing machines if they can read from and write to a memory. All of this suggests that LLMs are in fact pure functions, which can behave like more powerful automata only if they interact with other forms of storage. Pérez et al. (2021) offer a proof that LLMs are Turing-complete if the inputs have arbitrary numeric precision, which is not however a realistic assumption.

We do not propose make any claims here about transformer-architecture LLMs in general. But, in the particular case of the LLMs in this experiment, we will claim that they are, in fact, pure functions.

An LLM works with a *context* that consists of a succession of tokens. Tokens are words, parts of words, symbols, etc. The alphabet of tokens that a model uses is finite, and the model processes words that do not map to a single token (e.g. long, unusual or previously unencountered words) by splitting the word up into several tokens. The length of the context is also finite and, in the case of the models used in this experiment, can be specified when the model is trained. The model uses an encoder to translate each token in the context to a numeric representation. The model then uses these numeric representations to calculate the numeric representation of what it considers to be the next appropriate token in the given sequence. The corresponding token is written into the next empty slot in the context. This updated context becomes the input for generating another token, and so on. The model will keep adding tokens to the context until it either outputs an end-of-output token, or it runs out of space in the context (Jurafsky & Martin 2026).

The figures below illustrate the expected behavior of each type of automaton given the ungrammatical input *Copiii merge veseli* 'Children goes happily' (correct version *Copiii merg veseli* 'Children go happily'). The *labeler* clearly behaves like a pure function: it takes an input of finite maximum width (the context) and outputs one value based on it: "0" if the input is ungrammatical, "1" if the input is grammatical.

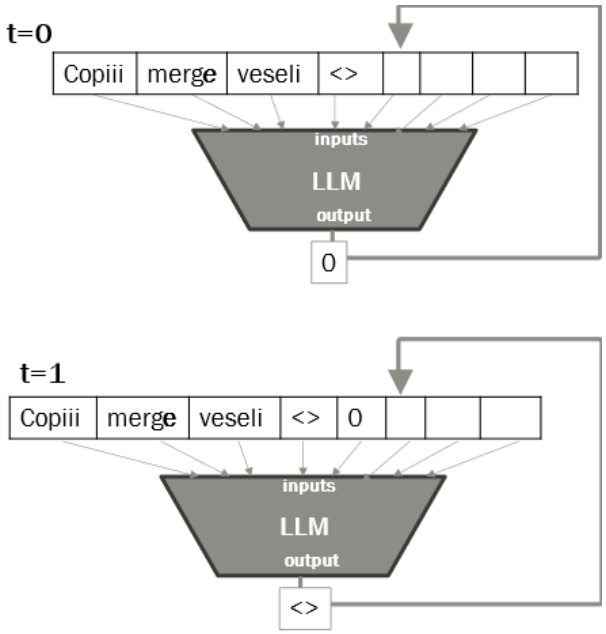

Figure 3. Labeler behavior

The *corrector* goes through several iterations before arriving at its final state.

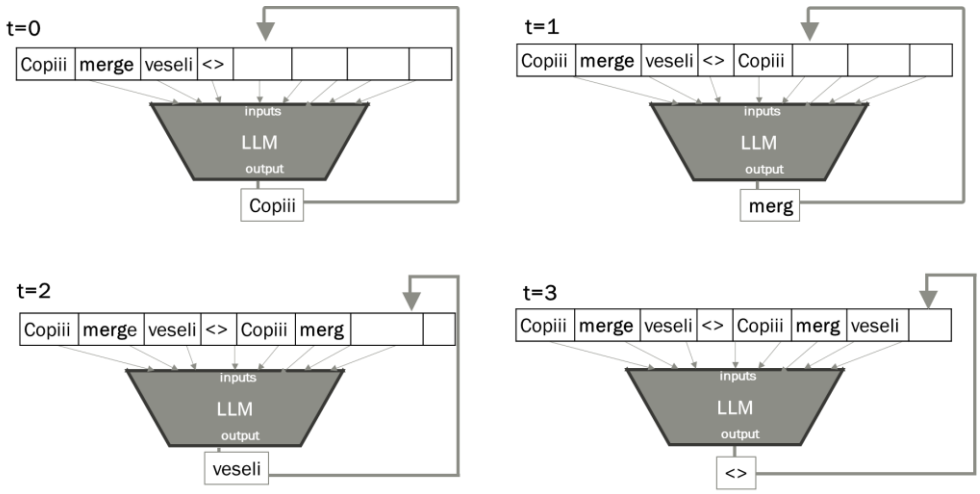

Figure 4. Corrector behavior

However, it accepts no new inputs while doing so, and this is its crucial limitation. Since the only source of input is the initial context, and the context is

finite, the corrector is not powerful enough to process regular or higher order languages (which do not necessarily have an upper bound on the length of the strings belonging to them). The corrector can therefore be modeled as a pure function that takes the initial context as its input and generates the final context as its output.[4]

1.2. LLMS AS AUTOMATA THAT CAN RECOGNIZE A USEFUL SUBSET OF A LANGUAGE

Given that the LLMs used here are pure functions, which lie at the bottom of the Chomsky grammar/automata hierarchy, it would appear that one cannot, *a priori*, expect them to function as language acceptors and render grammaticality judgements for natural language.

However, one *can* expect them to process a potentially useful subset of natural language, since simpler automata can simulate the behavior of more powerful ones within certain limits. For instance, a push-down automaton can be transformed into a state machine if the stack is finite, i.e. if one sets a limit to the maximum depth of the stack (e.g. Nederhof 2000:32). Such an automaton is able to process, e.g., inputs with nested parentheses, as long as the number of nested parentheses does not exceed the depth of the stack. Likewise, a recursive grammar can be transformed into a non-recursive grammar if one limits the number of times a production can occur (Ginsburg & Spanier 1968), and an automaton based on this grammar will be able to recognize a subset of the original language.

More generally, and of relevance to our case, a pure function can recognize the subset of strings that do not exceed a certain length belonging to any language. This can shown through brute-force construction. Since the alphabet of input tokens is finite and there is an upper limit to the length of the strings, it is possible to enumerate every possible string that does not exceed this limit, and label each enumerated string as belonging or not belonging to the language. This truth table is the automaton.

Therefore, it is *possible* for the LLM to recognize the subset of strings belonging to a context-free or context-sensitive language that do not exceed a certain length. Since the length of the context of the models used in this experiment is great enough to accommodate the vast majority of the sentences used for training and evaluation data, LLMs may be capable of learning to render grammaticality judgements on this data. The question this experiment explores is to what extent they do so.

[4] Since the finiteness of the context is the crucial limitation, it should be noted that, when Pérez et al. (2021) posited arbitrary precision for the LLM's inputs (see above §3.1), they were essentially allowing the context to be arbitrarily long, since an infinite-precision input can represent an arbitrary amount of data. This may explain why, under this assumption, LLMs are Turing-complete.

2. EXPERIMENT SETUP

2.1. TARGETED UNGRAMMATICALITIES

The present experiment does not attempt to cover every possible type of ungrammaticality. For practical reasons, it targets ungrammaticalities caused by violating the rules of agreement in gender, number and person. Agreement is an important means of expressing syntactic subordination in Romanian (GALR II:18), it is ubiquitous and, therefore, we considered that distinguishing between sentences where the rules of agreement are respected and those where they are violated requires an accurate interpretation of a sentence's syntactic structure.

- Within an NP, we targeted the following types of agreement:
  - *gender and number agreement of **determiners**, **numerals**, **adjectives** with the **noun** they determine*
  - *gender and number agreement of the **special genitival particle** **al/a/ai/ale** that determines a possessor with the **possessed noun**.*

The latter type of agreement, where the particle agrees not with the noun it determines but rather with the noun's possessor, can occur over longer distances than the former, and can stymie even native speakers (GBLR: 126-131).

- Within a VP, we target the following types of agreement:
  - *number and person agreement of the **verb** with the **subject***
  - *number and gender agreement of a **predicative adjective** with the **subject** (in copulative constructions or when it is a predicative object)*
  - *number and gender agreement of a **participle** with the **subject** in passive constructions*

The training data consisted of correct sentences and versions of these sentences where one of the aforementioned agreements is violated.

2.2. GENERATING UNGRAMMATICAL SENTENCES

Initial attempts showed that re-fine-tuning a model to generate the desired type of outputs requires dataset sizes on the order of ten thousand or more. Since manually generating ungrammatical sentences on this scale was impractical, the training datasets were generated automatically. This process requires lexical, morphological, and syntactic information about the words in a sentence. The syntactic information allows the generation script to identify words that are in agreement. This being done, the lexical and morphological characteristics of one of the words that participates in the grammatical agreement were extracted, and one

of the morphological features that participate in the grammatical agreement is changed. The lemma of the word and the altered morphological information are then fed into a lexicon that outputs the corresponding form of the word. This altered form was substituted for the original to generate an ungrammatical sentence.

The following figure illustrates how this process yields a sentence that violates subject-verb agreement (*Copiii mergeau/*mergea la școală* 'The children were going/*was going to school'), by replacing the plural form of the verb with the singular.

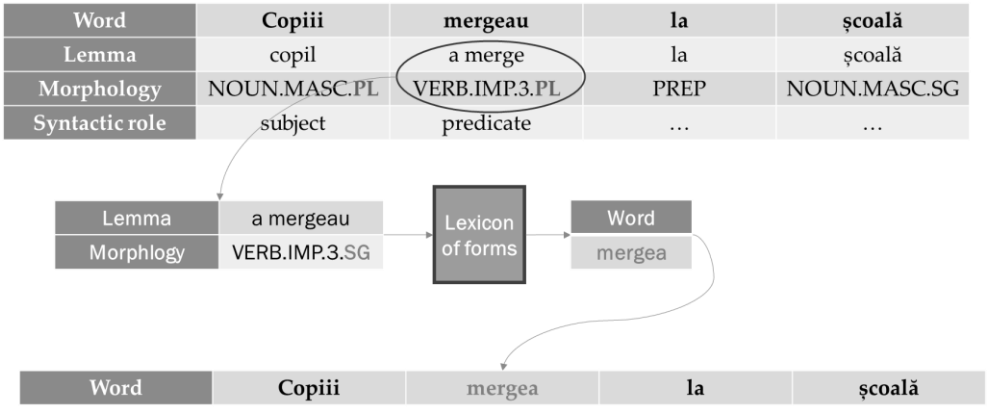

Figure 5. Generating an ungrammatical sentence

2.3. SOURCES OF TRAINING DATA

The first source for grammatical sentences was the syntactically annotated Romanian language corpus RoRefTrees (Barbu Mititelu et al. 2016), hosted by the Universal Dependencies project[5]. The syntactic annotation for this corpus contains all the necessary information for generating ungrammatical sentences. However, since this corpus proved too small (9524 sentences) to generate sufficient training data, a number of articles were scraped from online publications and automatically parsed using the *stanza* parser (Qi et al. 2020). The sources for articles were: *dilema.ro* (1592 articles), a cultural magazine whose contributors use grammatically correct and often complex sentences; and *cotidianul.ro* (1482 articles), a mainstream news site.

For generating the altered forms of words, we used the Romanian language lexicon RoLEX (Lőrincz et al. 2023), which contains over 330,000 morphologically annotated word forms.

[5] https://universaldependencies.org/

2.4. TRAINING AND EVALUATION DATASETS

For evaluation, 4,347 sentences were drawn randomly from the RoRefTrees corpus. Based on these grammatical sentences, we generated 28,592 ungrammatical sentences. We only used sentences from RoRefTrees to generate the evaluation dataset because the automatic parse inevitably introduces some errors in the annotation.

For training, we used the remaining 5,177 sentences from RoRefTrees and another 72,630 sentences from the scraped articles to obtain a total of 77,807 grammatical sentences. Based on these sentences, we generated 519,622 ungrammatical sentences.

To avoid using unbalanced datasets for training, we adopted two solutions:

i) Undersampling the ungrammatical sentences: for each grammatical sentence, we included one randomly selected ungrammatical version of that sentence, to obtain a balanced dataset of 155,614 unique sentences, half grammatical, half ungrammatical. This dataset will be labelled *ds_unique*.

ii) Oversampling the grammatical sentences: for each ungrammatical sentence we included its grammatical (original) version, to obtain a balanced dataset of 1,039,244 sentences – the ungrammatical sentences unique, the grammatical sentences not unique. This dataset will be labeled *ds_ovrsmpl*.

Both datasets were used to train models, with a view to observing which dataset yields the better results.

2.5. THE ORIGINAL MODEL

The model used for fine-tuned was *RoLlama3.1-8b-Instruct* (Masala et al. 2024). This model was generated by OpenLLM-Ro, an online community dedicated to building Romanian language models. It is based on Meta AI's *Llama-3.1* (a multi-lingual model) and fine-tuned for generating text in Romanian. At 8 billion parameters, it does not belong to the largest class of models. However, experimentation with it showed that it generally generates correct and complete sentences in Romanian.

3. EXPERIMENT RESULTS

3.1. OVERALL PERFORMANCE

The following table presents the performance of the various models. It should be noted that the evaluation sample is not evenly divided: it contains approximately seven times more ungrammatical than grammatical sentences, since it is possible to generate several ungrammatical sentences based on one grammatical sentence. Consequently, simply providing the overall accuracy for a model would be

misleading. For instance, a model that always declares any sentence to be ungrammatical exhibits no power of discrimination – nevertheless, due to the unbalanced nature of the evaluation dataset, its accuracy would be a seemingly respectable 87%. The table presents the performance on the grammatical and ungrammatical inputs separately, and then an average of the two F-scores as a measure to gauge overall performance. For grammatical inputs, PPV is the positive predictive value (i.e., the accuracy in recognizing grammatical inputs as being grammatical) and the F-score calculated in terms of true positives. For ungrammatical inputs, NPV is the negative predictive value (i.e., the accuracy in recognizing ungrammatical inputs as being ungrammatical) and the F-score is calculated in terms of true negatives.

Since the difference in number between the grammatical and ungrammatical inputs skews the individual F-scores, the average F-score is reported as the overall measure of a model's performance. An average F-score of 0.5 indicates no power of discrimination on the part of the model – it is the expected score of models that always answer 0, always answer 1, or answer at random. An average F-score below 0.5 indicates that a model can be expected to be wrong more often than a model with no power of discrimination, and an average F-score above 0.5 indicates that the model can be expected to be correct more often than a model with no power of discrimination. The average F-scores below indicate that, with the exception of the correctors, the models exhibit little power of discrimination. The correctors, however, perform well.

Table 1

Performance of the various models

Model Type	**Training Set**	**Epochs**	**Grammatical inputs**		**Ungrammatical inputs**		**Overall**
			PPV	***F* (TP)**	**NPV**	***F* (TN)**	***F* avg**
Informant	n/a	n/a	35%	0.198	67%	0.759	0.478
Labeler	ds_unique	1	1%	0.027	99%	0.923	0.475
Labeler	ds_unique	2	23%	0.181	80%	0.836	0.508
Labeler	ds_unique	3	90%	0.234	12%	0.212	0.223
Labeler	ds_ovrsmpl	1	20%	0.188	85%	0.864	0.526
Labeler	ds_ovrsmpl	2	5%	0.078	95%	0.91	0.494
Labeler	ds_ovrsmpl	3	82%	0.289	41%	0.573	0.431
Corrector	ds_unique	1	95%	0.851	96%	0.974	**0.913**
Corrector	ds_ovrsmpl	1	90%	0.858	97%	0.977	**0.917**
Number of eval. samples			*4347*		*28592*		*32939*

The informant represents the performance of the original model, *OpenLLM-Ro/RoLlama3.1-8b-Instruct*. The original model was queried regarding the grammaticality of each sentence in the evaluation dataset, with the requirement that its responses take a "yes" or "no" form. Thus, it was possible to automatically label the responses and calculate its performance. Its performance is mediocre, and it exhibits a "no" bias – a tendency to over-correct.

The *labelers* were trained on both datasets for several epochs – i.e., once a model was trained and its performance was evaluated, it was trained again a second time on the same dataset, and then a third time. Training for several epochs is a common practice for helping models converge, that is, arrive at the point where they correctly mimic the expected output. The labelers, however, failed to converge. Instead, their behavior appears to oscillate – after two epochs where they overcorrect (severely in the first epoch), on the third epoch they undercorrect.

The internal workings of an LLM are opaque, but the superior performance of the correctors can be explained by the fact that their configuration exploits an LLM's basic function as a generator of the likeliest next token. One may suppose that the corrector copies each token in the original input but, if it encounters a token that is unlikely due to its ungrammaticality, it replaces it with a semantically similar but grammatically more probable alternative.

Of the two corrector models, one trained on the oversampled dataset *ds_ovrsmpl* exhibits a tendency to overcorrect relative to the preceding one. Therefore, we will consider the corrector trained on the dataset of unique sentences the best model, for its even performance on both grammatical and ungrammatical inputs, even if its F-score is marginally lower.[6] This is the model whose performance will be examined in detail in the following sections.

3.2. RESULTS FOR GRAMMATICAL INPUTS

To understand the performance of the *corrector*, it is useful to go beyond a simple score and examine the individual sentences where it made mistakes. For the situations where the model reported that a grammatical input is ungrammatical, there is a subset of cases where its response is justifiable. (In the examples that follow, we will show a change made by the model on an input as follows: **{original_word→ model_replacement_for_it }**.)

- The original input actually *was* ungrammatical (generally as the result of a minor typographic error in the original corpus).
- The input was too long, the model did not have room in the context to complete its output
- The input contains obsolete forms or expressions (e.g. *văz* instead of *văd* 'I see')

[6] If the grammatical and ungrammatical data were weighted equally, its F-score would actually be the higher one, at 0.95.

In those cases where there is no such justification for the model's behavior, most of the time the model replaced a grammatical input with an output that was also grammatical, but different. These changes follow certain patterns:

- Situations where the rules of agreement are not clean-cut.

For instance, disjunctively coordinated compound subjects may take verbs either in the singular or in the plural, depending on whether the reading of the subject is inclusive or exclusive (v. GALR II:386-387).

...comerciantul sau reprezentantul său trebuie ***{informați → informat}*** *în scris...*
'...the merchant or his representative must be **{informed.PL→informed.SG}** in writing…'

- Situations where changing the gender or number of an adjective makes it select a different regent, usually a noun in an adjacent prepositional phrase

Înhumări ... în mormane mai mici de piatră ***{rotunde →rotundă}****...*
'Burials… in small mounds.PL of stone.SG **{round.PL →round.SG}** ...

In the original input, the adjective *rotunde* 'round.PL' refers to the noun *mormane* 'mounds'. The model changed the adjective to *rotundă* 'round.SG', which agrees with the noun *piartă* 'stone'. The model's output is both grammatically correct and semantically plausible, but has a different meaning than the original does.

- Changes to certain deictic morphological traits (e.g. person, tense).

The model exhibits a tendency of replacing the perfect simple tense with the present, and the first and second persons with the third, reflecting characteristics of the training dataset: the perfect simple is a relatively uncommon tense in the standard language, and the third person is far more common in writing (as opposed to oral interaction) than the first and second persons. In the absence of the context, the model's changes yield acceptable (but semantically different) sentences.

După câteva minute ***{se umplu →se umple}*** *și masa fetei.*
'After a few moments, the girl's table **{filled.PERF →fills.PRES}** up.'

Câți oameni ***{ai tăiat → a tăiat}*** *cu cuțitul acela?*
'How many men have **{cut.2ND →cut.3RD}** with that knife?'

- Replacing certain words with more common semantic (near-)equivalents

Somnul e asimilat morții, inerției, opririi ***{sterpe → sterile}*** *a vieții.*
'Sleep is likened to death, inertia, the **{barren → sterile}** ceasing of life'

*Când Gaittany **{aminti → i-a spus}** lui Ioanide că a doua zi...*
'When Gaittany **{reminded →told}** Ioanide that the next day…'

These changes reflect the LLM's probabilistic nature: it replaces certain tokens with tokens that are similar but more common according to its training data.

Some changes, however, are nonsensical. For instance:

*Frunzele acestor pomi au pete **{cafenii →cafenele}**.*
'The leaves of these trees have spots [that are] **{brown →coffee shops}**.

The following table summarizes the numeric prevalence of these phenomena within the results.

Table 2

Results for grammatical inputs

Result	Count	Percent.
The model reported the input is grammatical (correct response)	**4119**	**94.8%**
The model reported the input is ungrammatical, but		
the input actually *was* ungrammatical (mistake in original)	16	0.4%
the input is too long	42	1.0%
the input contains obsolete word forms/expressions	14	0.3%
The model was unquestionably wrong		**3.6%**
it offered a grammatically correct alternative	83	1.9%
it offered a grammatically incorrect alternative	44	1.0%
it made spacing / punctuation modifications	29	0.7%
Total	**4347**	**100%**

3.3. RESULTS FOR UNGRAMMATICAL INPUTS

In those situations where an ungrammatical input was reported as being grammatical, there are a very few instances where, as in the previous section, the original actually *was* ungrammatical (due to a typographic error), and the input provided as being ungrammatical was, in fact grammatical. Leaving these instances aside, there are situations where the supposedly ungrammatical input is, strictly speaking, grammatical, albeit not always acceptable semantically. These are situations that should be avoided in the future when automatically generating ungrammatical sentences.

The most common situation where this occurs involves a noun followed by two modifiers, the first a prepositional phrase, the second an adjective or a genitive

noun. Changing the gender or number of the adjective or of the genitival particle can change the regent of the second modifier. For instance, in *negocierile de aderare a României* 'the negotiations for Romania's adhesion', changing the number of the genitival particle *a/ale* yields *negocierile de aderare ale României* 'Romania's negotiations for adhesion'. The corrector labelled the latter construction as being grammatical which, strictly speaking, it is, even though the former construction was more appropriate in the context it appeared in.

A similar source of accidentally grammatical sentences is the homonymy, Romanian, of adverbs and masculine singular adjectives. For instance, the word *natural* may be the masculine singular adjective 'natural', or the adverb 'naturally'. In the noun phrase *rezervația naturală protejată* 'the protected natural reservation', changing the gender of the adjective *naturală* from feminine to masculine yields a grammatical noun phrase where the word *natural* is interpreted as an adverb determining the adjective *protejată* 'protected': *rezervația natural protejată* 'the naturally protected reservation'. Again, the corrector labelled the latter construction as being grammatical which, strictly speaking, it is, even though it made little sense in the context it appeared in.

As the table below shows, these situations represent a very small proportion (0.5%) of the ungrammatical evaluation data. Nevertheless, this kind of data should be avoided in future experiments of this type.

In many cases where the model mislabels an ungrammatical sentence, the error is glaring, and the model's judgement is inexplicable.

*Și totuși, patrulele nu **conta**.*
'Nevertheless, the patrols.PL didn't **matter.SG**.'

*Îi spuse **bune** ziua doamnei Parsons și se îndreptă spre ușă.*
'He said **good.PL** day.SG to Mrs. Parsons and walked to the door.'

Table 3

Results for ungrammatical inputs

Result	Count	Percent.
Model reported the input is ungrammatical (correct response)	**27381**	**95.8%**
Model reported the input is grammatical, but		
the original was incorrect, the "ungrammatical" input was correct	9	0.03%
the input could be grammatical (depending on the interpretation)	151	0.5%
the model was unquestionably wrong	**1051**	**3.5%**
Total	**28592**	**100%**

3.4. RESULTS FOR NESTED CLAUSES

Besides measuring the model's performance on the evaluation dataset, we tested the model on a small dataset to evaluate whether the model is capable of recognizing ungrammaticalities across nested clauses. Since nested clauses are associated with context-free grammars, the capacity to process them would indicate that the model has successfully "climbed" the Chomsky hierarchy.

The sentences were generated such that an initial noun had to agree with a final adjective, with one or more nested clauses separating them, where each nested clause ended with a disruptor – a noun of a different gender than the initial noun. For instance (the noun and the adjective that must agree are bold, and the disruptor(s) are underlined):

> ***Mâncarea*** *din care a mâncat* *băiatul* *era* ***bună / *bun***.
> 'The **food.FEM** of which ate the *boy.MASC* was **good.FEM / *good.MASC**.'

> **Mâncarea** din care a mâncat *băiatul* pe care l-a certat *bucătarul* era **bună / *bun**.
> 'The **food.FEM** of which ate the boy.MASC whom scolded the cook.MASC was **good.FEM / *good.MASC**.'

The model was tested with sentences containing up to 5 successive nested clauses, for both gender and number agreement. It correctly distinguished between grammatical and ungrammatical sentences. While the test sample was too small for the results to be conclusive, they do suggest that the model has "climbed" the Chomsky hierarchy and is capable of recognizing at least context-free languages.

4. CONCLUSIONS AND DISCUSSION

We suggested earlier that the superior performance of the *corrector* is due to the fact that it exploits the LLMs basic functionality as an automaton for generating the likeliest next word. We also speculated that the mechanism the corrector uses to correct a sentence is that, if it encounters a token that is unlikely due to its ungrammaticality, it replaces it with a semantically similar but grammatically more probable alternative. The examination of the situations where the *corrector* mistakenly labels grammatical sentences as being ungrammatical supports the above supposition. In these situations, the *corrector* frequently changed a morphological trait (or entire word) for one that was more frequent in its training data.

The performance of the *corrector*, with a ~95% accuracy for detecting both grammatical and ungrammatical sentences and an F-score that is above 0.9, is respectable by machine learning standards. However, by human standards, its

performance is not necessarily impressive. One can think of the *corrector* as a proofreader. An accuracy of 95% translates to an error rate of 5%, or 1 out of 20. This, then, is a proofreader that not only misses one out of twenty mistakes, but also randomly inserts mistakes into the text, at an average rate of one for every twenty good sentences. By human standards, it is a poor proofreader.

If exploiting the intended functionality of an LLM's architecture is what allows us to obtain the best results, this same architecture is the source of its limitations. As Bryant et al. (2023: 664) point out, "ultimately, all LM-based approaches suffer from the limitation that probability is not grammaticality." This limitation was on display when the *corrector* changed correct forms or words with forms that are likelier based on its training data.

However, while probability is not grammaticality, grammaticality *is* one of the factors that determines probability. The fact that the model has generalized, based on the training data, the requirement to enforce gender, number and person agreement strongly suggests that it does have an underlying representation of grammar, which allows it to recognize the syntactic relationships between words. Therefore, with regards to the questions posed at the beginning of the paper, we believe the results support the idea that an LLM subsumes a model of the language's grammar, and that the mistakes the *corrector* makes are largely due to its inability to distinguish between improbability due to ungrammaticality and improbability due to other (e.g. semantic) factors.

The fact that the *corrector*'s performance is nevertheless inferior to that of a human proofreader raises the question of what explains this gap in linguistic competence. After all, human babies do not acquire language in ideal conditions: they acquire language, including its grammar, based on what they hear around them, including grammatically erroneous sentences, and they are not told which sentences are grammatically correct and which are not (Lasnik & Lidz 2016). Similarly, LLMs' pretraining is unsupervised: it is based on vast quantities of text, some of which will contain errors, and the LLM under training receives no indication which sentences are correct and which are incorrect. Despite these shared limitations, humans make better proofreaders than LLMs trained to that specific purpose. We propose an explanation based on Saussure's notion of a *sign*.

A *sign* comprises a *signifier* and a *signified* that the former refers to (Saussure 2012[1916]:65-67). An LLM works with incomplete signs: it deduces information about a *signifier* based on its association with other signifiers, but it has no access to the *signified* – it may know that a trumpet produces sound because this information appears, explicitly and implicitly, in its training corpus, but it has never heard a sound or seen a trumpet. Humans, on the other hand, can access the signified through their senses. They can relate an utterance to a perceptible state of fact. This allows them to deduce that some utterances are aberrant in ways that LLMs cannot.

For instance, if one uses the plural with reference to a single object, e.g. "What a pretty doll this *are*" while holding a single doll, this results in a mismatch both between the grammatical number of the subject and the verb, and between the number of dolls implied by the verb (greater than one) and the actual number of dolls one is referring to. A human being can note that a grammatical incongruence is associated with a communicative incongruence and conclude that this sentence is, in some respect, an aberration, a data point that can be excluded when making deductions about language. An LLM has no basis for concluding that a sentence in its training dataset is an aberration. It will take an incorrect sentence into account just as much as it does a correct sentence. In other words, a human's learning process is not, in fact, as unsupervised as an LLM's. Humans may not be told explicitly whether a sentence is grammatical or not, but the process of relating an utterance to a state of facts gives them a reference for checking the content of the utterance against that is unavailable to an LLM.

BIBLIOGRAPHY

GALR = Guțu Romalo, Valeria (ed.). 2008. *Gramatica limbii române*, vol. I-II. București: Editura Academiei Române.

GBLR = Pană Dindelegan, Gabriela (ed.). 2010. *Gramatica de bază a limbii române*. București: Editura Univers Enciclopedic Gold.

Barbu Mititelu, Verginica, Radu Ion, Radu Simionescu, Elena Irimia, Cenel-Augusto Perez. 2016. "The Romanian Treebank Annotated According to Universal Dependencies", in *Proceedings of HrTAL2016, Dubrovnik, Croatia*, 29 September – 1 October 2016.

Bryant, Christopher, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, Ted Briscoe. 2023. "Grammatical Error Correction: A Survey of the State of the Art". *Computational Linguistics 2023*; 49 (3): 643–701.

Chomsky, Noam. 1957. *Syntactic Structures*. The Hague/Paris: Mouton.

Dale Schuurmans, Hanjun Dai, Francesco Zanini. 2024. "Autoregressive Large Language Models are Computationally Universal." https://arxiv.org/abs/2410.03170

Delétang, Gregoire, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, Pedro A. Ortega. 2023. "Neural Networks and the Chomsky Hierarchy". In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dentella, V., F. Günther, E. Leivada. 2023. "Systematic testing of three Language Models reveals low language accuracy, absence of response stability, and a yes-response bias", *Proc. Natl. Acad. Sci. U.S.A.* 120 (51).

Ginsburg, Seymour, Edwin H. Spanier. 1968. "Derivation-bounded languages", *Journal of Computer and System Sciences*, 2(3) 228–250.

Hahn, Michael. 2020. "Theoretical Limitations of Self-Attention in Neural Sequence Models". *Transactions of the Association for Computational Linguistics*, 8:156–171.

Han, Daniel, Michael Han, and Unsloth Team. 2023. *Unsloth*. http://github.com/unslothai/unsloth

Hopcroft, J. E., Motwani, R., & Ullman, J. D. 2006. *Introduction to Automata Theory, Languages, and Computation*. Pearson.

Jiaoda Li, Jennifer White, Mrinmaya Sachan, and Ryan Cotterell. 2024. “A Transformer with Stack Attention”. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4318–4335, Mexico City, Mexico. Association for Computational Linguistics.

Jorge Pérez, Pablo Barceló, and Javier Marinkovic. 2021, “Attention is Turing-complete". *Journal of Machine Learning Research* 22(75), 1–35.

Jurafsky, Daniel and James H. Martin. 2026. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd edition. Online manuscript released January 6, 2026. https://web.stanford.edu/~jurafsky/slp3.

Lasnik, Howard; Lidz, Jeffrey L. 2016. “The Argument from the Poverty of the Stimulus”, in Roberts, Ian (ed.), *The Oxford Handbook of Universal Grammar*, Oxford University Press, 221–249.

Li, Jiaoda, Jennifer C. White, Mrinmaya Sachan, Ryan Cotterell. 2024. “A Transformer with Stack Attention”. *arXiv:2405.04515*.

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing”. *ACM Computing Surveys*, 55(9): 1–35.

Lőrincz, Beáta, Elena Irimia, Adriana Stan, and Verginica Barbu Mititelu. 2023. “RoLEX: The Development of an Extended Romanian Lexical Dataset and Its Evaluation at Predicting Concurrent Lexical Information.” *Natural Language Engineering*, 29(3): 720–45. https://doi.org/10.1017/S1351324922000419.

Masala, Mihai, Denis Ilie-Ablachim, Alexandru Dima, Dragos Georgian Corlatescu, Miruna-Andreea Zavelca, Ovio Olaru, Simina-Maria Terian, Andrei Terian, Marius Leordeanu, Horia Velicu, Marius Popescu, Mihai Dascalu, and Traian Rebedea. 2024. “’Vorbești Românește?’ A Recipe to Train Powerful Romanian LLMs with English Instructions”, in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 11632–11647, Miami, Florida, USA. Association for Computational Linguistics.

Michael Sipser. 2012. *Introduction to the Theory of Computation*. Cengage Learning.

Nederhof, Mark-Jan. 2000. “Practical experiments with regular approximation of context-free languages”. *Computational Linguistics* 26(1), 17–44.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton and Christopher D. Manning. 2020. “Stanza: A Python Natural Language Processing Toolkit for Many Human Languages”. In: *Association for Computational Linguistics (ACL) System Demonstrations*. 2020.

Reinhart, Alex, Ben Markey, Michael Laudenbach, Kachatad Pantusen, Ronald Yurko, Gordon Weinberg, & David West Brown. 2025. “Do LLMs write like humans? Variation in grammatical and rhetorical styles”. *Proc. Natl. Acad. Sci. U.S.A.* 122 (8).

Saussure, Ferdinand de. 2012[1916]. *Course in General Linguistics*. Columbia University Press, pp. 65–67.

Shieber, Stuart M. 1985. “Evidence Against the Context-Freeness of Natural Language”. In: Savitch, W.J., Bach, E., Marsh, W., Safran-Naveh, G. (eds) *The Formal Complexity of Natural Language. Studies in Linguistics and Philosophy*, vol. 33. Springer, Dordrecht. https://www.eecs.harvard.edu/shieber/Biblio/Papers/shieber85.pdf

APPENDIX – DATASET AND MODEL INFORMATION

The datasets used in this experiment are hosted at https://huggingface.co/

Dataset Name	Split	Training Data
hartular/grammatical_errors_rrt_press-v2	train	ds_oversmpl
hartular/grammatical_errors_rrt_press-v2	train.1per	ds_unique

The evaluation dataset and the model responses can be found in

Dataset Name	Split
hartular/eval_GrammarAgree_rrt-v2-final	test

The models are also hosted at https://huggingface.co/:

Model Name	Model Type	Training Data	Epochs
OpenLLM-Ro/RoLlama3.1-8b-Instruct	Informant	n/a	n/a
hartular/GrammarAgreeLabeler-X7-EP1-v2-all_per	Labeler	ds_ovrsmpl	1
hartular/GrammarAgreeLabeler-X7-EP2-v2-all_per	Labeler	ds_ovrsmpl	2
hartular/GrammarAgreeLabeler-X7-EP3-v2-all_per	Labeler	ds_ovrsmpl	3
hartular/GrammarAgreeLabeler-X7-EP1-v2-1per	Labeler	ds_unique	1
hartular/GrammarAgreeLabeler-X7-EP2-v2-1per	Labeler	ds_unique	2
hartular/GrammarAgreeLabeler-X7-EP3-v2-1per	Labeler	ds_unique	3
hartular/GrammarAgreeCorrector-X7-EP1-v2-all_per	Corrector	ds_ovrsmpl	1
hartular/GrammarAgreeCorrector-X7-EP1-v2-1per	Corrector	ds_unique	1

The models were trained using the Unsloth library (Han et al. 2023), with the following model parameters:

max_seq_length = 1024, lora_rank = 16;

and the following training arguments:

learning_rate=3e-4, lr_scheduler_type="linear",
per_device_train_batch_size=8, gradient_accumulation_steps=2,
num_train_epochs=NUM_EPOCHS, fp16=not
unsloth.is_bfloat16_supported(), bf16=unsloth.is_bfloat16_supported(),
logging_steps=1, optim="adamw_8bit", weight_decay=0.01,
warmup_steps=10, output_dir=out_model_name, seed=0.

TEACHING AN LLM GRAMMATICALITY JUDGEMENTS IN ROMANIAN

(Abstract)

This paper investigates whether large language models (LLMs) can perform grammaticality judgments in Romanian. To address the limitations of prompting LLMs directly, the study fine-tunes a Romanian LLM into two specialized models: a labeler, which classifies sentences as grammatical or ungrammatical, and a corrector, which modifies ungrammatical inputs. The experiment focuses on agreement errors (gender, number, person), using automatically generated datasets. Results show that labelers fail to reliably discriminate between grammatical and ungrammatical sentences, while correctors achieve high performance (F-scores above 0.9). However, correctors may unnecessarily replace correct forms with more probable alternatives, reflecting probabilistic rather than rule-based behavior. The findings suggest that LLMs subsume a model of the grammar but struggle to distinguish grammaticality from likelihood, highlighting both their potential and their limitations in modeling natural language structure.

ANTRENAREA UNUI LLM PENTRU JUDECĂȚI PRIVIND GRAMATICALITATEA ÎN LIMBA ROMÂNĂ

(Rezumat)

Lucrarea investighează dacă modelele limbajului de mari dimensiuni (LLM) pot discrimina între enunțuri gramaticale și negramaticale în limba română. Pentru a evita limitațiile interogării directe, un LLM românesc este re-antrenat în două variante: un etichetator (*labeler*), care clasifică propozițiile ca fiind gramaticale sau negramaticale, și un corector (*corrector*), care modifică propozițiile eronate. Studiul se concentrează pe erori de acord (gen, număr, persoană), folosind seturi de date generate automat. Rezultatele arată că etichetatorii nu disting fiabil între propoziții corecte și incorecte, în timp ce corectorii obțin performanțe net superioare (scoruri F de peste 0,9). Totuși, aceștia tind să înlocuiască forme corecte cu variante mai probabile statistic. Concluziile sugerează că LLM-urile conțin o reprezentare a gramaticii limbii, dar confundă gramaticalitatea cu probabilitatea, evidențiind atât potențialul, cât și limitele lor.