Received: 13 January 2026 | Revised: 26 February 2026 | Accepted: 13 March 2026 | Published online: 31 March 2026

RESEARCH ARTICLE

Journal of Computational Law and Legal Technology
2026, Vol. 00(00) 1–7
DOI: 10.47852/bonviewJCLLT62029092

Catch the Platypus! Negated Conditionals as a Challenge for Machine Translation from Natural Language into Logical Formalisms Using Large Language Models

Bianca Steffes[1,*] **and Diogo Sasdelli**[2]

[1] *Saarland University, Germany*

[2] *University for Continuing Education Krems, Austria*

Abstract: One of the most promising applications of large language models in the legal domain concerns the automated conversion of natural language legal texts into logical formalisms, that is, automated formalization. Major challenges to these approaches emerge from the semantic fuzziness of natural language, which leads to sentences that are particularly difficult to formalize—we call these sentences "platypus sentences." For example, the natural negation of a sentence in natural language may have different, context-dependent meanings, which often do not correspond to the logical negation of a respective formalization of said sentence. In other words, natural negations and formalized negations often diverge from one another. This problem is further intensified when natural language conditionals (i.e., negated "if... then..." sentences) are negated. The paper at hand investigates how current large language models (GPT-5, Llama, and LogicLinguist) deal with automated formalization of negated conditionals. Our results indicate that these systems still cannot reliably deliver correct formalizations, although results can be enhanced, for example, by prompt engineering.

Keywords: automated formalization, large language models (LLMs), platypus sentences, legal AI

1. Introduction

Many applications in the wide area of LegalTech—in particular those related to automated compliance—require an adequate formal representation of legal norms, mostly by means of some logical formalism like deontic logic, predicate logic, or directly through programming languages (cf., e.g., [1, 2]). For example, to comply with traffic rules—and to be able to reliably verify this compliance—an autonomous vehicle should include some compliance checking mechanism containing a formal representation of the respective traffic rules. However, formalizing legal norms can be a rather laborious task, involving many steps and requiring cooperation among specialists of various disciplines [3]. Hence, legal tech in general would enormously benefit from approaches that could at least partially automate the process of formalizing legal norms.

Large language models (LLMs) offer a convenient method for translating natural language texts into logical formalisms, that is, a method for automated formalization. With an adequate prompt, entire arguments in natural language can be quickly translated into a logical formalism. However, depending on the specific scenario, the quality of these translations may vary substantially. For example, translations into more complicated logical formalisms (e.g., systems of reified modal logic, that is, systems combining modalities with quantification theory) tend to be more prone to errors than translations into simpler formalisms such as propositional logic. Furthermore, an at least equally challenging problem lies in the complexity of the input language, that is, natural language.

The paper at hand introduces the notion of "platypus sentences"—that is, sentences, which, due to the intrinsic ambiguity and context-dependence of natural language, might lead to errors when formalizing them, especially when automated methods are employed—and investigates how LLMs perform when given the task of formalizing dialogues containing a specific kind of such sentences, namely, negated conditionals. The main contributions of this paper are:

1) We introduce the problem of *platypus sentences* as a cause for translation errors for natural language into logical formulae.
2) We devise four dialogues employing different kinds of platypus sentences for research on the translation of platypus sentences.
3) We conduct a first exploratory analysis of the capabilities of three state-of-the-art LLMs for translating these dialogues to logical formulae.

***Corresponding author:** Bianca Steffes, Saarland University, Germany. Email: bianca.steffes@uni-saarland.de

The remainder of this paper is structured as follows: Section 2 first briefly discusses related work on logical formalisms for legal norms, on methodologies for formalizing legal norms, and on machine translation from natural language into logical formalisms, in particular when LLMs are employed and especially in the context of legal texts. Section 3 then presents the employed methodology, including a brief discussion of platypus sentences and a description of four dialogues developed to test how LLMs fare when formalizing such sentences. Test results are presented in Section 4 and discussed in Section 5. Section 6 concludes and refers to future work. Overall, our results indicate that current LLMs are unable to provide reliable formalizations of platypus sentences.

2. Related Work

Logical formalisms for normative language in general and for the law in particular have been developed based on modern mathematical logic and related approaches since the 1920s (for systematic and historic overviews of these developments, cf., e.g., [4–6]). These rather theoretical knowledge representation methods are sometimes also implemented, for example, within programming languages, such as PROLOG [7] and PROLEG [8], or within theorem provers [9–11].

Specific challenges for the formalization of natural language emerging within the legal domain have been investigated since the origins of modern logic of norms. This includes so-called *paradoxes of deontic logic* (cf., e.g., [4]) and more general problems related to the semantic ambiguity of legal language. Formalization methodologies for dealing with these challenges have been intensively investigated since the 1970s and 1980s by pioneers of legal informatics, for example, by Ilmar Tammelo [12], Leo Reisinger [13], and Laymen Allen [14]. The problems discussed in the paper at hand are directly inspired by previous work by Ron Klinger [15]. More recent approaches focus, for example, on combining manual and automated methods to enhance the efficiency of generated formalizations [16] and on using legal visualization methods to improve interdisciplinary communication among lawyers, logicians, computer scientists, and engineers [3].

Previous research has analyzed the general viability of machine translation from natural language reasoning into logical formalisms [17], including approaches based on LLMs [18, 19]. LogicLinguist, for example, is a specialized model for translating natural language statements into first-order logic [20]. In particular, the capability of such systems to detect logical fallacies has been analyzed, with some results indicating that LLMs generally perform rather poorly on these tasks [21], while better results can be achieved by more streamlined systems adopting neurosymbolic approaches [22]. In the legal domain, machine translation methodologies using LLMs have been developed for outputs in PROLEG [23] and in PROLOG [24], the latter particularly stressing the value of also employing controlled natural languages [25] as an intermediary step between natural language and logic.

While considerable previous research on the translation of natural language into logical formalisms is available, there is still little work on translation paradoxes emerging from natural language ambiguities. Even *LOGIC* [21], a dataset especially designed for logical fallacy detection, mainly focuses on fallacies based on logically incorrect inferences (e.g., *ad populum* fallacies like "Everyone should like coffee: 95% of teachers do!"). Current research is lacking an analysis of the translation of natural language sentences, which are logically consistent but ambiguous in their translation into logical formalisms.

3. Methodology

In this section, we first introduce the notion of *platypus sentences*, that is, natural language sentences, which are particularly difficult to formalize due to their semantic ambiguity (Section 3.1). Then, we describe short natural language dialogues containing platypus sentences consisting of negated natural language conditionals. Although all these sentences have the same syntactic structure, they have fundamentally different meanings, so that their proper formalizations in propositional logic strongly differ from one another. To test the capabilities of current LLMs with respect to formalizing these sentences, we select different models to prompt for translating these dialogues into logical formulae (Section 3.3). Finally, since this task, as indicated by previous related work [21], is to be considered as a challenging task, we decided to use different prompts to thoroughly grasp the models' capabilities (Section 3.4).

3.1. Platypus sentences

Ideally, formal logic should correspond to Leibniz's notion of a *characteristica universalis*, that is, a universal system of symbols capable of univocally representing every conceivable object, so that concepts and relations among these objects could be mapped to sets of ordered n-tuples among the respective symbols (for details, cf., e.g., [26]). As a *characteristica universalis*, formal logic is generally not context-dependent: the meaning of logical symbols does not change depending on the logical formulae in which they occur. In this sense, semantics is embedded within syntax. This is, of course, not the case with natural language, which is rich in ambiguities, synonyms, homonyms, idioms, metaphors, figures of speech, etc.

This semantic discrepancy between natural language and logical formalisms is the source of many logical riddles, paradoxes, paralogisms, and jokes. It is the root of odd deductions such as "all thieves are people; therefore: all *good* thieves are *good* people" and of "self-annihilating sentences" like "I am a firm believer in optimism; because if you don't have optimism, what else is there?" (for this and similar sentences cf., e.g., Saul Gorn's famous *Compendium of Rarely Used Cliches* [27]). Beyond serving as a source for intellectual entertainment, this semantic ambiguity of natural language can also lead to formalization errors, especially when automated machine translation is employed. A proper formalization should, for example, distinguish between the property of "being a cup of tea" in a sentence such as "this is a cup of coffee, not a cup of tea" and in a sentence such as "as a lawyer, formal logic is not my cup of tea." This distinction, however, presupposes an adequate consideration of the context in which these sentences are employed—a consequence of the fact that natural language is not context-independent and therefore not a *lingua characteristica* in the sense of Leibniz.

Inspired by the egg-laying, duck-billed, beaver-tailed, venomous, semiaquatic monotreme mammal, we call such context-dependent natural language sentences, which, due to their semantic ambiguity, could lead to formalization errors (in particular when machine translation is used), *platypus sentences*. The class of platypus sentences is a wide and rather heterogeneous class, which, as such, allows for several further sub-classifications. It is nonetheless relevant to analyze these sentences as a whole. Among them are not only sentences containing ambiguous properties (e.g., "being a cup of tea" or being "good") and sentences based on figurative language but also sentences building on implicit assumptions derived from the specific context in

which they appear (including, but not limited to, sentences with pronouns or relative adverbs such as "yesterday" or "behind").

Platypus sentences emerge not only from ambiguous names, adjectives, adverbs, and other individually meaningful terms (i.e., from so-called *categorems*) but also from connectors, conjunctions, etc. (i.e., from so-called *syncategorems*), which are meaningless on their own. For example, a particularly rich source of platypus sentences is conditionals, that is, sentences with an "if... then..." structure. These sentences encompass various different semantics, for example, the notion of material or strict implication, of causality, of conditional norms (sometimes called "commitment" [28]), etc. This semantic ambiguity is all the more intensified when combining conditionals with negations, a further good syncategorematic source of platypus sentences. In our current investigation, we focus on syncategorematic platypus sentences. For this reason, the dialogues we prepare for testing include platypus sentences based on negated conditionals.

3.2. Dialogues

Drawing inspiration from previous related work, we develop four different dialogues containing platypus sentences consisting of negated conditionals, each leading to a different formalization in propositional logic. The dialogue structure is important because it provides the necessary context required to determine the concrete meaning of the platypus sentences.

Dialogue 1: The unwanted poem

Bob: It is Eve's birthday; do you think that if I write him a poem, he will be happy?
Alice: No, of course not!
Bob: Why not?
Alice: Eve hates poems.

The first dialogue contains the natural language conditional question of whether Eve will be happy (Q) if Bob writes Eve a poem (P). As a general thesis, this conditional could, in principle, be formalized as a material implication, that is, by the formula $P \rightarrow Q$. However, in the particular discursive context of the dialogue, Alice's negative answer does not merely correspond to a logical negation of this formula (i.e., to the formula $\neg(P \rightarrow Q)$). Instead, she claims that Bob writing Eve a poem would actually lead to the opposite result from what Bob originally intended—that is, Eve would not be happy; after all, he does not like poems. In other words, Alice's negative answer does not correspond to the negation of a material implication but to a material implication with the same premise but a negated conclusion. The correct logical formula should therefore be $P \rightarrow \neg Q$.

Dialogue 2: The counsel's dilemma

Attorney: I do recognize that my client was at the scene of the crime. However, does being at the crime scene imply guilt?
Judge: No, I do not think so.
Attorney: In other words, it is not the case that: 'if my client was at the crime scene, then he is guilty'?
Judge: I agree.
Attorney: Then, it follows logically that my client is innocent.

The second dialogue shows a hypothetical discussion between an attorney and a judge. It traces back to a paper by Ron Klinger [15] and is centered around a proposed conditional relationship between a person being at the scene of a crime (P) and that person being guilty of the crime committed at the crime scene (Q). Again, as a general thesis, this could, in principle, be formalized by the material implication $P \rightarrow Q$. In the dialogue, the attorney tries to exploit the fact that the negation of this material implication (i.e., $\neg(P \rightarrow Q)$) is logically equivalent to the conjunction $P \wedge \neg Q$ to argue for the innocence of her client. However, in the specific context of this dialogue, the judge's denial of the natural language conditional does not amount to stating the logical negation of the general thesis (i.e., to claiming the negated material implication $\neg(P \rightarrow Q)$). Instead, the judge is merely stating that the general thesis is not being accepted, without committing to any verifunctional relationship between P and Q. In other words, the negation amounts to claiming that the propositions involved are, *ceteris paribus*, logically independent from one another. Thus, the most adequate formalization of the judge's denial of the natural language conditional (if indeed any is required at all) would be the tautological formula $(P \wedge Q) \vee (\neg P \wedge Q) \vee (P \wedge \neg Q) \vee (\neg P \wedge \neg Q)$,[1] which, as such, is irrelevant for the rest of the argument proposed by the attorney.

Dialogue 3: Tweety, the penguin

Alice: My friend's dog was barking at my bird Tweety yesterday, and Tweety was very scared.
Bob: Oh, why did it not just fly away?
Alice: Tweety cannot fly!
Bob: Oh, I thought if Tweety is a bird, then it must be able to fly?
Alice: No, this is not true! Tweety is a penguin!

The third dialogue picks up the classic example of Tweety, the penguin (introduced by J. Pearl [29]), and is based on the natural language conditional of whether Tweety's "birdness" (its property of being a bird (P)), implies it being able to fly (Q). Once again, as a general thesis, this conditional could, in principle, be formalized by the material implication $P \rightarrow Q$. In this case, however, the negation of the natural language conditional is indeed adequately represented by the negation of a material implication: since Tweety is indeed both a flightless penguin and a bird at the same time, Tweety is an adequate counterexample to the general thesis expressed by $P \rightarrow Q$. Thus, the proper formalization of Alice's negation of the natural language conditional is $\neg(P \rightarrow Q)$. In theory, this dialogue should be easier to formalize than the previous ones because, in this case, the negation of the natural language conditional actually corresponds to the logical negation of a material implication, that is, the formalization that syntactically most closely resembles the structure of the platypus sentence. In this sense, this dialogue (and the next one, which, as described below, is based on the same structure as this one) can be taken as a control test.

Dialogue 4: The mysterious platypus

Alice: Yesterday at the zoo I saw a platypus! It is a mammal that lays eggs!
Bob: But if it's a mammal, it can't lay eggs, can it?
Alice: No, that's not true!

The last dialogue is a variant of the previous dialogue and addresses the conditional of whether a platypus, being a mammal (P), is not able to lay eggs ($\neg Q$). Like in the previous dialogue, the correct formalization for the negation of this conditional is a negated material implication, albeit here with a negated consequence, that is, $\neg(P \rightarrow \neg Q)$. We choose to include this variant of the previous dialogue to assess whether the LLMs' likely previous

[1] This formula basically states that any combination of truth-values for P and Q is satisfiable; that is, they can be both true or both false, or one of them can be true, while the other is false.

Table 1
Overview of the employed dialogues

Dialogue	Platypus sentence	Correct formalization
The unwanted poem	Denial of "you think if I write him a poem, he will be happy?"	$P \rightarrow \neg Q$
The counsel's dilemma	Denial of "does being at the crime scene imply guilt?"	$(P \wedge Q) \vee (\neg P \wedge Q) \vee (P \wedge \neg Q) \vee (\neg P \wedge \neg Q)$
Tweety, the penguin	Denial of "if Tweety is a bird, then it must be able to fly?"	$\neg(P \rightarrow Q)$
The mysterious platypus	Denial of "But if it's a mammal, it can't lay eggs, can it?"	$\neg(P \rightarrow \neg Q)$

Table 2
Employed prompts

Simple	Extract logical formulae (propositional logic) from the following dialogue.
Extended	Extract logical formulae (propositional logic) from the following dialogue. Pay special attention to the negations of implications. Check their validity very carefully.

information on the famous Tweety problem would manifest itself in the quality of the results delivered, the expectation being that, if the results' quality indeed diverges, the models should perform better with Dialogue 3 than with Dialogue 4.

An overview of all four dialogues is provided in Table 1.

3.3. Models

To grasp the capabilities of current state-of-the-art LLMs in translating these dialogues to propositional logic, we perform tests on three different LLMs. As the most well-known and widely used LLM, we chose ChatGPT for the first model. We use GPT-5, which is a GPT version promoted to be especially well-suited for reasoning tasks [30]. As the second model, we use Llama 3.3 [31]—another well-known, widely used model competing with ChatGPT. The last model we use is LogicLinguist, which is specialized for translating natural language statements into first-order logic [20]. Although it also employs GPT-4o for chat interactions, it uses *Z3 Theorem Prover* for logic solving. We interacted with all of these models via their online chat interfaces.

3.4. Prompts

We use two different prompts to evaluate the models' capabilities to translate the chosen dialogues to logical formulae. The first prompt ("simple") is a simple prompt only telling the models to extract logical formulae for propositional logic for the given dialogue. This prompt is meant for evaluating a naïve approach users might take when trying to convert natural language to logical formalisms. Users might want a swift reply without a lot of prompt engineering to get results for their task.

The second prompt ("extended") is devised to help the models find the possible pitfalls in translating the given dialogues. The difficulty of translating a negated conditional is pointed out to the models, so it can be taken into account. Additionally, the models are tasked with checking the validity of the generated formulae for these negated implications to test whether the models can properly identify the specific difficulties in the dialogues.

The prompts are shown in Table 2.

4. Results

We now describe in detail how the models translated each dialogue into logical formulae. Overall, the tested models returned results of varying quality when prompted with the dialogues and prompts described above. Furthermore, models tended to perform better when the extended prompt was used. An overview of the results is provided in Table 3.

4.1. Dialogue 1: The unwanted poem

For the simple prompt, none of the models was able to generate the correct result ($P \rightarrow \neg Q$). GPT-5 and LogicLinguist both formalized the negated conditional as $\neg(P \rightarrow Q)$, while Llama rather oddly translated it to the tautology $\neg(Q \wedge \neg Q)$. When using the extended prompt, Llama still failed to correctly formalize the platypus sentence in the dialogue, translating it to $\neg(P \rightarrow Q)$. With the extended prompt, the other two models

Table 3
Formalization results for tested models; correct results in blue bold font

	Prompt	GPT-5	Llama	LogicLinguist
Dialogue 1	simple	¬(P → Q)	¬(Q ∧ ¬Q)	¬(P → Q)
	extended	**P → ¬Q**	¬(P → Q)	**P → ¬Q**
Dialogue 2	simple	¬(P → Q)	¬(P → Q)	¬(P → Q)
	extended	¬(P → Q)	¬(P → Q)	¬(P → Q)
Dialogue 3	simple	**¬(P → Q)**	¬Q	¬P
	extended	**¬(P → Q)**	**¬(P → Q)**	**¬(P → Q)**
Dialogue 4	simple	**¬(P → ¬Q)**	**¬(P → ¬Q)**	**¬(P → ¬Q)**
	extended	**¬(P → ¬Q)**	**¬(P → ¬Q)**	**¬(P → ¬Q)**

correctly identified three possible formalizations for the negated natural language conditional—including the correct solution. Both also correctly concluded that, among the options, $P \rightarrow \neg Q$ is the most fitting translation for the platypus sentence in this dialogue.

4.2. Dialogue 2: The counsel's dilemma

The platypus sentence in the second dialogue was not formalized correctly by any model for any prompt. All models derived $\neg(P \rightarrow Q)$, which is not the correct translation. However, GPT-5 and LogicLinguist (which, we recall, employs GPT-4o) noted that the judge's negation of the natural language conditional would not, from a pragmatic point of view, amount to stating $\neg(P \rightarrow Q)$, which would thus hinder the attorney from validly deducing $\neg Q$. While this is indeed the proper reasoning that should lead to the correct formalization, the systems were unable to deliver it. This could be explained either by (1) the fact that the formalization is trivial for the argument, (2) the fact that the formalization bears almost no resemblance to the respective sentence, or (3) the fact that the dialogue, through the attorney's final claim, might be taken as stressing the acceptance of $\neg(P \rightarrow Q)$ by the judge. Overall, the systems were unable to adequately grasp the absurdity of the attorney's argument.

4.3. Dialogue 3: Tweety, the penguin

For this dialogue, only GPT-5 returned the correct translation $\neg(P \rightarrow Q)$ when using the simple prompt. LogicLinguist rather oddly negated only the premise ($\neg P$), while Llama negated the consequence ($\neg Q$), which comes close to the correct answer (we recall that $\neg(P \rightarrow Q)$ is equivalent to $P \wedge \neg Q$). When using the extended prompt, all three models translated the platypus sentence in the dialogue correctly and gave a proper explanation for this formalization. While all models derived the formula based on the background knowledge that penguins are birds and that penguins cannot fly, only GPT-5 rather briefly referred to the problem as a known problem by calling it a "classic."

4.4. Dialogue 4: The mysterious platypus

For the previous dialogue, which contains a more well-known version of the same underlying problem, Llama and LogicLinguist failed when using the simple prompt. In contrast, for this dialogue, all models returned the correct result (i.e., $\neg(P \rightarrow \neg Q)$) with all prompts. Only Llama showed a slight inaccuracy when using the simple prompt by being inconsistent with its usage of symbols. More precisely, it formalized "mammals cannot lay eggs" sometimes as Q and sometimes as $\neg Q$ and thus suggested $\neg(P \rightarrow Q)$ as a formalization for the negated conditional, although it made clear that it had taken Q to mean "mammals cannot lay eggs." These results subvert our expectations, as we had assumed that models would fare better when formalizing Dialogue 3 because of it dealing with the widely known Tweety problem.

5. Discussion

Overall, among the three models, GPT-5 and LogicLinguist (which, we recall, employs GPT-4o) performed noticeably better than Llama, which was only able to deliver correct results for Dialogues 3 and 4. As described above, these tasks can be seen as *easier* tasks, since the correct formalization very closely resembles the syntactical structure of the respective platypus sentence.

As expected, the extended prompt consistently leads to better results. This is promising because the extended prompt is still *general* in the sense that it does not contain any specific information on the respective dialogues. This shows that it is possible to get better results using more detailed prompts even without having to address concrete details of specific cases.

The fact that no performance advantage could be observed when comparing the results for Dialogues 3 and 4 is interesting. It indicates that, for some reason, previous information on the Tweety problem, which the systems are likely to have had easy access to, did not play any significant role in the formalization. Indeed, one can conjecture whether this information could be counterproductive, as the Tweety problem is actually related to a different formalization problem—namely, that of defeasible reasoning—which, however, does not play any significant role in the formalization of Dialogue 3. This can be observed in GPT-5's results with the extended prompt, in which it is stated that this is a "classic where a counterexample to a universal implication shows up."

This issue is further highlighted by the fact that the reasoning presented by the models, while often leading to the correct results, was not, in its entirety, coherent. This is most clear in the results delivered by LogicLinguist, which, for both Dialogues 1 and 4, starts its answer by stressing that the dialogue illustrates that one ought not to confuse the negation of an implication with another implication. This is only true for Dialogue 4, not for Dialogue 1, in which, in fact, the negation of the conditional does correspond to a "different implication" (i.e., an implication with the same premise but a different conclusion). Incidentally, this underlying problem of confusing the negation of a conditional with another conditional is at the core of the famous *Barbershop Paradox*, a material implication paradox formulated by Lewis Carroll at the end of the 19th century [32].

This incoherence in the underlying reasoning displayed by the systems could be the cause of the odd results delivered by Llama (Dialogues 1 and 3) and by LogicLinguist (Dialogue 3) when using the simple prompt. Moreover, these results indicate that merely providing more information to the systems might not lead to better results, as the information may have the effect of misleading the systems away from the right formalization.

Finally, another factor that might substantially influence the quality of automated formalization methods is precisely the logical formalism underlying the respective outputs. In the examples discussed here, we used *classic propositional language* as the output formalism. While this formalism is usually seen as too simple to adequately represent normativity (cf., e.g., the discussions in [6]), most widely used logical formalisms for law and similar domains are based on extensions of propositional logic (e.g., modal logic, predicate logic; cf. [6]), which, however, generally retain the same structure concerning the relationship among propositions (e.g., among conditions) within their semantics. In this sense, it seems reasonable to expect that the challenges presented here should also emerge in any other formalisms—even more complex ones—as long as they expand classic propositional logic while retaining its basic structure. This, however, does not necessarily apply to formalisms that adopt a *different basic structure*, in particular concerning the semantics of negation and of conditionals. A good example would be the so-called *logic of relevant implication*, which adopts different semantics for conditionals [33].

Thus, it seems reasonable to conjecture that improving the formalization results for tasks like the ones discussed here would

involve *both* (1) optimizing the underlying general reasoning of the models, which is necessary to grasp the meaning of the natural language sentences to be formalized, and (2) improving the strictly logical reasoning, which is to be determined with respect to the specific logical formalism to be employed. In particular, it would seem desirable that models recognize the semantic limitations of the formalisms they are employing if the respective sentences in natural language are too complex to be adequately represented within them.

6. Conclusion and Future Work

Translating natural language to logical formulae is a challenging task for LLMs. In this paper, we showed that there are some kinds of sentences which may be especially hard to translate to logical formulae ("platypus sentences"). Inspecting one type of such difficult sentences—negated conditionals—we tested current state-of-the-art LLMs' capabilities in translating these sentences to logical formulae in propositional logic. Our experiments showed that using only a simple prompt often results in incorrect translations, while even a more intricate prompt does not guarantee a correct solution.

As platypus sentences are a legitimate difficulty when translating natural language to logical formulae, current and future systems devised for such a purpose should be evaluated on such sentences to gauge their capabilities. Further research is needed to analyze the performance of state-of-the-art LLMs when confronted with other types of platypus sentences. Additionally, given the small number of examples addressed in this paper, more examples that may vary in length or complexity need to be considered for further investigations. Other topics for future work include investigating the formalization of platypus sentences in formal argumentation models (e.g., so-called quantitative bipolar argumentation frameworks or QBAFs [34–37]) and whether visualization techniques (in particular legal visualization methods [38–41]) could be employed to enable clearer representations of the intrinsic ambiguity involved in these sentences.

Overall, the fact that the models largely failed to deliver proper results in the most difficult cases of Dialogues 1 and 2—with all three systems failing in the case of Dialogue 2 and Llama and LogicLinguist failing even in the easier Dialogues 3 and 4 with the simple prompt—indicates that LLM-based machine formalization is still far from being a reliable source for adequate formalizations of natural language, especially when ambiguous sentences are involved. Hence, if they are to ever catch the platypus, LLMs ought better to start putting their shoulder to the wheel—ideally without losing time searching for a wheel or trying to develop an actual shoulder.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Author Contribution Statement

Bianca Steffes: Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Visualization, Software. **Diogo Sasdelli:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Visualization, Validation.

References

[1] Kampik, T., Mansour, A., Boissier, O., Kirrane, S., Padget, J., Payne, T.R., . . . , & Zimmermann, A. (2022). Governance of autonomous agents on the web: Challenges and opportunities. *ACM Transactions on Internet Technology*, *22*(4), 1–31. https://doi.org/10.1145/3507910

[2] Mowbray, A., Chung, P., & Greenleaf, G. (2023). Representing legislative rules as code: Reducing the problems of 'scaling up'. *Computer Law & Security Review*, *48*, 105772. https://doi.org/10.1016/j.clsr.2022.105772

[3] Sasdelli, D., Steffes, B., Herrmann, M., Chitashvili, M., & Wüst, C. (2025). A normal form for representing legal norms and its visualisation through normative diagrams. In *Annual International Conference on Digital Government Research 26*. https://doi.org/10.59490/dgo.2025.1036

[4] McNamara, P., & Van De Putte, F. (2022). Deontic logic. In E. N. Zalta & U. Nodelman (Eds.), *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University. Fall 2022

[5] Parent, X., & van der Torre, L. (2018). *Introduction to deontic logic and normative systems*, UK: College Publications.

[6] Sasdelli, D. (2025). *Können Maschinen Rechtsfälle entscheiden?* German: Felix Meiner Verlag. https://doi.org/10.28937/978-3-7873-4900-5

[7] Sergot, M.J., Sadri, F., Kowalski, R.A., Kriwaczek, F., Hammond, P., & Cory, H.T. (1986). The British nationality act as a logic program. *Communications of the ACM*, *29*(5), 370–386. https://doi.org/10.1145/5689.5920

[8] Satoh, K. (2023). PROLEG: Practical legal reasoning system. In D. S. Warren, V. Dahl, T. Eiter, M. V. Hermenegildo, R. Kowalski, & F. Rossi (Eds.), *Prolog: The next 50 years* (pp. 277–283). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-35254-6_23

[9] Benzmüller, C., & Fuenmayor, D. (2021). Value-Oriented Legal Argumentation in Isabelle/HOL. *12th International Conference on Interactive Theorem Proving (ITP 2021). Leibniz International Proceedings in Informatics (LIPIcs)*, *193*, 7:1–7:20. https://doi.org/10.4230/LIPIcs.ITP.2021.7

[10] Benzmüller, C., Farjami, A., Fuenmayor, D., Meder, P., Parent, X., Steen, A., . . . , & Zahoransky, V. (2020). Logikey workbench: Deontic logics, logic combinations and expressive ethical and legal reasoning (Isabelle/HOL dataset). *Data in Brief*, *33*, 106409. https://doi.org/10.1016/j.dib.2020.106409

[11] Steen, A., & Benzmüller, C. (2021). Extensional higher-order paramodulation in Leo-III. *Journal of Automated Reasoning*, *65*(6), 775–807. https://doi.org/10.1007/s10817-021-09588-x

[12] Tammelo, I. (1978). *Modern logic in the service of law*. Austria: Springer.

[13] Reisinger, L. (1975). Probleme der Symbolisierung und Formalisierung im Recht. In G. Winkler (Ed.), *Rechtstheorie und Rechtsinformatik* (pp. 22–50). Springer.

[14] Allen, L. (1974). Formalizing Hohfeldian analysis to clarify the multiple senses of 'legal right': A powerful lens for the electronic age. *Southern California Law Review*, *48*, 428–487.

[15] Klinger, R. (1971). The paradox of counter-conditional and its dissolution. *Jurimetrics Journal*, *11*(4), 189–193. http://www.jstor.org/stable/29761216

[16] Witt, A., Huggins, A., Governatori, G., & Buckley, J. (2024). Encoding legislation: A methodology for enhancing technical validation, legal alignment and interdisciplinarity. *Artificial Intelligence and Law*, *32*(2), 293–324. https://doi.org/10.1007/s10506-023-09350-1

[17] Angeli, G., & Manning, C.D. (2014). NaturalLI: Natural logic inference for common sense reasoning. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 534–545). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1059

[18] Ryu, H., Kim, G., Lee, H.S., & Yang, E. (2025). Divide and translate: Compositional first-order logic translation and verification for complex logical reasoning. *arXiv Preprint: 2410.08047*

[19] Yang, Y., Xiong, S., Payani, A., Shareghi, E., & Fekri, F. (2023). Harnessing the power of large language models for natural language to first-order logic translation. *arXiv Preprint: 2305.15541*

[20] YesChat. ai. (2025). *Logic Linguist – logic translation tool*. https://www.yeschat.ai/gpts-9t557oyNbN2-Logic-Linguist

[21] Jin, Z., Lalwani, A., Vaidhya, T., Shen, X., Ding, Y., Lyu, Z., . . . , & Schoelkopf, B. (2022). Logical fallacy detection. In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022* (pp. 7180–7198). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-emnlp.532

[22] Lalwani, A., Kim, T., Chopra, L., Hahn, C., Jin, Z., & Sachan, M. (2025). *Autoformalizing natural language to first-order logic: A case study in logical fallacy detection*. *arXiv Preprint: 2405.02318*

[23] Nguyen, H.T., Wachara, F., Nishino, F., & Satoh, K. (2022). A multi-step approach in translating natural language into logical formula. In E. Francesconi, G. Borges, & C. Sorge (Eds.), *Legal knowledge and information systems. Frontiers in Artificial Intelligence and Applications*. IOS Press. https://doi.org/10.3233/FAIA220453

[24] Zin, M., Borges, G., Satoh, K., & Fungwacharakorn, W. (2025). Towards machine-readable traffic laws: Formalizing traffic rules into prolog using LLMs. In J. Maranhão (Ed.), *Proceedings of ICAIL2025*, 317–326. https://doi.org/10.1145/3769126.3769204

[25] Kowalski, R., & Datoo, A. (2022). Logical English meets legal English for swaps and derivatives. *Artificial Intelligence and Law*, *30*(2), 163–197. https://doi.org/10.1007/s10506-021-09295-3

[26] Lenzen, W. (2018). Leibniz and the calculus ratiocinator. In S. Hansson (Ed.), *Philosophy of Engineering and Technology*, (pp. 47–78). Springer, https://doi.org/10.1007/978-3-319-93779-3_4

[27] Gorn, S. (1992). *Self-annihilating sentences: Saul Gorn's compendium of rarely used cliches*. *Tech. rep.,* USA: University of Pennsylvania. https://repository.upenn.edu/entities/publication/3758e6d1-f02b-4b65-9615-6946fbe4e412

[28] von Wright, G.H. (1951). I. Deontic logic. *Mind*, *60*(237), 1–15. https://doi.org/10.1093/mind/lx.237.1

[29] Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. USA: Morgan Kaufmann Publishers Inc.

[30] OpenAI: Introducing gpt-5. (2025). https://openai.com/index/introducing-gpt-5/

[31] Meta, AI. (2025). *Llama 3.3 model cards and prompt formats*. https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/

[32] Carroll, L. (1894). A logical paradox. *Mind*, *3*(11), 436–438.

[33] Anderson, A.R., & Belnap, N.D. (1976). *Entailment: The logic of relevance and necessity, Vol. I*. USA: Princeton University Press.

[34] Brännström, A., Dignum, V., & Nieves, J.C. (2024). Goal-hiding information-seeking dialogues: A formal framework. *International Journal of Approximate Reasoning*, *177*, 109325. https://doi.org/10.1016/j.ijar.2024.109325

[35] Cayrol, C., Cohen, A., & Lagasquie-Schiex, M.C. (2021). Higher-order interactions (bipolar or not) in abstract argumentation: A state of the art. *Journal of Applied Logics—IfCoLog Journal of Logics and their Applications*, *8*(6), 1339–1436.

[36] Potyka, N., & Booth, R. (2024). An empirical study of quantitative bipolar argumentation frameworks for truth discovery. *Frontiers in Artificial Intelligence and Applications*, *388*, 205–216. https://doi.org/10.3233/FAIA240322

[37] Yin, X., Potyka, N., & Toni, F. (2023). Argument attribution explanations in quantitative bipolar argumentation frameworks. *Frontiers in Artificial Intelligence and Applications*, *372*, 2898–2905. https://doi.org/10.3233/FAIA230603

[38] Cyras, V., & Lachmayer, F. (2023). *Essays on the visualisation of legal informatics*. USA: Springer.

[39] Lachmayer, F., & Cyras, V. (2021). Visualization of legal informatics. In E. Schweighofer (Ed.), *JURIX2021: The Thirty-fourth Annual Conference, Vilnius, Lithuania*, 3–10. https://doi.org/10.3233/FAIA210310

[40] Sasdelli, D., & Trivisonno, A. (2023). Normative diagrams as a tool for representing legal systems. *The Review of Socionetwork Strategies*, *17*, 217–231. https://doi.org/10.1007/s12626-023-00144-0

[41] Sasdelli, D., & Steffes, B. (2025). Validity diagrams. In *Legal Knowledge and Information Systems - Proceedings of the 38th International Conference on Legal Knowledge and Information Systems – JURIX 2025*, 180–191. https://doi.org/10.3233/FAIA251587

How to Cite: Steffes, B., & Sasdelli, D. (2026). Catch the Platypus! Negated Conditionals as a Challenge for Machine Translation from Natural Language into Logical Formalisms Using Large Language Models. *Journal of Computational Law and Legal Technology*. https://doi.org/10.47852/bonviewJCLLT62029092