

Large Language Models as Syntactic Annotation Labelers for Logical Information Retrieval

Greg Coppola
coppola.ai

April 7, 2025

Abstract

This paper presents a novel research question—can *large language models* act as *linguists* so as to facilitate *open-domain semantic parsing*. Open-domain semantic parsing would allow a new kind of information retrieval based on *logical proofs*, and thus it would be very valuable if LLM’s can be leveraged for this task. We present a careful review of past results that shows that LLM’s *can* act reliably annotate syntactic annotations, but that this ability is highly dependent on the *prompt*. We analyze past findings that LLM’s can *not* act as zero-shot CoNLL dependency parsers, and present a novel analysis as to why. We present several positive novel experimental results showing further that LLM’s *can* act reliably as linguists, especially in tasks that are closely related to *semantic parsing*. We believe that in several cases this is human-level performance on these tasks. We propose that these results constitute that full open-domain semantic parsing should be possible, given the right *agentic* control flow—aka. sequence of calls to LLM’s and other tools—thus unlocking a new kind of *logical* information retrieval.

1 Motivation

This work is motivated by the current state of the field roughly five years after the introduction of *large language models*.

On the one hand, large language models have allowed a great variety of powerful new uses cases, and have completely transformed all aspects of the computer science *industry*, for all company sizes and fields.

On the other hand, it is clear that the LLM *by itself* is *not* equivalent to *artificial general intelligence* no matter how defined. (TODO: cite something)

TODO: vibe coding tweet

TODO: tweet that says, "but we believe llm’s can be the game changer, as we will see..."

2 Contributions

- a *positional argument* that the only thing blocking *open-domain parsing* is the ability for a LLM to act “as a linguist”
- a *meta-analysis* of past results on the question of whether LLM’s can act “as linguists”
- replication of past *negative* results on the ability for LLM’s to do “zero-shot” parsing to CoNLL trees as well as a new (CHECK) algorithmic explanation as to why

- several *new positive results* showing in novel crucial tasks related to the “pipeline” of semantic parsing that LLM’s *can* act effectively as annotation labelers
- a mix of qualitative data and quantitative analyses of *LLM’s as linguists* showing both qualitative and quantitative evidence that LLM’s have “linguistic knowledge”
- results indicating the importance of the *prompting strategy* on getting effective and accurate results

3 Background on Reasoning

3.1 Foundational Works in the History of Formal Logic

The development of formal logic has been shaped by a series of landmark works that laid the foundation for modern logic, mathematics, and computer science. Below, we summarize key contributions from the most influential figures in the field.

- **George Boole** introduced an algebraic approach to logic in *An Investigation of the Laws of Thought*, establishing the basis of Boolean algebra and formalizing logical reasoning through mathematical operations Boole [1854].
- **Gottlob Frege** developed the *Begriffsschrift*, a formal language for pure thought modeled after arithmetic. This work is considered the birth of modern predicate logic, introducing quantifiers and variables into logical syntax Frege [1879].
- **Giuseppe Peano** compiled the *Formulaire de mathématiques*, which helped formalize mathematical notation and logical inference. Peano’s axioms for arithmetic are still influential in formal systems today Peano [1895].
- **Alfred North Whitehead and Bertrand Russell** published *Principia Mathematica*, a monumental three-volume work that aimed to derive all of mathematics from logical axioms. It was one of the first major efforts in logicism ?.
- **Kurt Gödel** published his incompleteness theorems in *Über formal unentscheidbare Sätze*, demonstrating inherent limitations in formal systems like Principia Mathematica Gödel [1931].
- **Alonzo Church** addressed the *Entscheidungsproblem*, proving its unsolvability and introducing the concept of lambda calculus, foundational for computation theory ?.
- **Alan Turing** independently solved the *Entscheidungsproblem* in his work *On computable numbers*, introducing the concept of the Turing machine Turing [1936].
- **Alfred Tarski** formulated a rigorous semantic theory of truth in formalized languages, later published in *Logic, Semantics, Metamathematics* Tarski [1935].

3.2 Landmark Papers in LLM-Based Reasoning

Recent years have witnessed a surge in research focused on enhancing the reasoning capabilities of large language models (LLMs). Among the numerous contributions, five papers stand out for their foundational impact on the field:

Chain-of-Thought Prompting (Wei et al., 2022) Chain-of-Thought (CoT) prompting introduced a simple yet remarkably effective technique for improving the reasoning capabilities of large language models (LLMs) by encouraging them to produce intermediate steps before arriving at a final answer. Rather than answering a question in a single pass, CoT prompts the model to “think aloud” by decomposing complex tasks into a sequence of logical or arithmetic steps. This mirrors how humans often approach multi-step reasoning tasks and enables LLMs to better handle challenges like mathematical problem solving and commonsense reasoning. The original paper showed that CoT dramatically improves performance on such tasks, particularly in larger models like PaLM-540B. One notable success is that CoT helped unlock the emergent reasoning abilities latent in LLMs at scale. However, the approach does have limitations. It tends to rely on the model being sufficiently large and pre-trained to already contain latent reasoning abilities, and CoT outputs can be fragile—sensitive to prompt design and formatting. Furthermore, CoT alone does not ensure correctness; it improves reasoning fluency but not necessarily factual accuracy.

Program of Thoughts Prompting (Chen et al., 2022) Program of Thoughts (PoT) prompting builds on the insights of CoT by introducing a more structured and explicit form of reasoning, inspired by programming languages. Instead of generating free-form reasoning steps in natural language, PoT encourages LLMs to output pseudo-code that separates logical flow from surface text. This makes the model’s reasoning more modular, interpretable, and potentially executable. The authors demonstrated the benefits of PoT in numerical reasoning and symbolic tasks, where expressing solutions as code—e.g., defining variables, loops, and conditionals—can better align with the structure of the problem. One key strength of this method is that it enables closer integration between the language model and external computational engines, offering a hybrid neuro-symbolic pipeline. However, the success of PoT prompting often depends on careful prompt engineering and the model’s familiarity with code-like syntax. Its reliance on pre-defining structured abstractions can also limit generality across diverse domains.

Tree of Thoughts (Yao et al., 2023) Tree of Thoughts (ToT) prompting generalizes the idea of linear reasoning in CoT to a structured exploration over multiple reasoning paths organized as a tree. At each node in the reasoning tree, the model proposes possible continuations—“thoughts”—that represent plausible next steps. These are then evaluated using heuristics or value functions to decide which branches to explore further, enabling deliberate search and backtracking. ToT has shown strong empirical results in tasks requiring planning, decision-making, or combinatorial exploration, such as solving puzzles and logic games. The key innovation lies in allowing LLMs to deliberate and compare alternative reasoning strategies rather than being bound to a single, forward-only chain. However, ToT is computationally expensive, requiring the generation and evaluation of many candidate paths. The effectiveness of the method also depends on the design of scoring mechanisms used to evaluate partial thoughts, which can be heuristic or learned—adding complexity to the system.

Self-Consistency Decoding (Wang et al., 2022) Self-Consistency is a decoding strategy that complements CoT by addressing the unreliability of individual reasoning paths. Instead of generating a single output, the model is prompted multiple times to produce a diverse set of reasoning traces, each potentially yielding a final answer. These answers are then aggregated—typically via majority voting—to select the most consistent one. This ensemble-like approach significantly boosts accuracy on reasoning tasks by smoothing over the noise and variability of LLM output. Particularly in arithmetic and symbolic domains, self-consistency reduces the impact of hallucinations.

nated or brittle reasoning paths. Its strength lies in leveraging the stochastic nature of LLMs as a feature rather than a bug. Nonetheless, the approach incurs significant computational cost, as it requires sampling dozens (sometimes hundreds) of reasoning paths per query. It also assumes that the correct answer is the most frequent, which may not always hold—especially for ambiguous or adversarial inputs.

ReAct (Yao et al., 2022) ReAct introduces a powerful framework that integrates reasoning (thought) and interaction (action) in a single prompting loop. In this setup, the language model is not only prompted to think through a problem but also to interact with external tools such as search engines, calculators, or environments like web pages. The key idea is that LLMs can interleave internal reasoning steps with concrete actions, retrieve new information, and continue reasoning in light of that evidence. ReAct has been used to build interactive agents capable of web navigation, open-domain question answering, and tool-assisted problem solving. A major strength of ReAct is its grounding in real-world actions, which mitigates hallucination and supports verifiable intermediate steps. The framework also promotes transparency by logging both reasoning and action sequences. However, ReAct systems are more complex to deploy, requiring external APIs or environments to be accessible, and robust parsing of intermediate outputs. Designing prompts that manage both thought and action transitions coherently also presents practical challenges.

Least-to-Most Prompting (Zhou et al., 2022) Zhou et al. [2022] introduced the *least-to-most prompting* technique, which enhances LLM reasoning by decomposing complex questions into simpler sub-questions. This method enables step-by-step reasoning, allowing models like GPT-3 to outperform traditional chain-of-thought prompting on complex tasks. The technique is inspired by cognitive psychology and significantly improves performance on benchmarks such as GSM8K and MultiArith.

Toolformer (Schick et al., 2023) Toolformer Schick et al. [2023] proposes a self-supervised approach to teach LLMs how to use external tools, such as calculators or web search APIs, without human-annotated demonstrations. The model selects and inserts API calls during training, learning when and how to use tools to improve its task performance. This work bridges the gap between static LLMs and interactive agents capable of tool use, enhancing performance on tasks requiring factual lookup or computation.

Auto-GPT (Richards, 2023) Auto-GPT Richards [2023] is one of the earliest open-source implementations of autonomous language agents powered by GPT-4. It chains LLM calls with self-reflection, memory, and tool use to achieve high-level goals without constant human supervision. While primarily an experimental system, Auto-GPT sparked widespread interest in autonomous agents and highlighted both the promise and limitations of current LLMs when operating over long contexts and evolving plans.

Reflexion (Shinn et al., 2023) Shinn et al. [2023] proposed *Reflexion*, a framework in which LLM agents improve task performance through verbal self-reflection. After failing at a task, the agent generates natural language feedback describing the mistake and uses this reflection to guide future attempts. Reflexion combines elements of reinforcement learning and meta-cognition, and has shown to improve the reliability of language agents across multiple interactive tasks.

Step-Level Math Correction with Reinforcement Learning. Li et al. [2025] propose *StepAMC*, a novel framework for step-level automatic math correction using reinforcement learning (RL). Unlike traditional approaches that focus only on the final answer, StepAMC provides fine-grained feedback on each individual step in a student’s solution. To address the limitations of binary feedback and shallow reasoning, the authors introduce two key components: a space-constrained policy network to improve the stability and precision of step-wise predictions, and a fine-grained reward network that transforms binary human annotations into continuous reward signals. Experiments on PRM-42K and MSD-22K datasets demonstrate that StepAMC outperforms strong baselines such as DPO and PPO, achieving better alignment with human judgment and greater robustness across correct and incorrect step classifications.

3.3 A Modern Version of the Traditional Semantic Pipeline

* describe what the traditional semantic pipeline is * this should import the lit review from the liang paper

4 Background on Statistical Parsing

Recent work has explored the ability of large language models (LLMs) to serve as syntactic annotators, generating linguistic labels such as part-of-speech (POS) tags, dependency trees, and constituency parses. These models operate in zero-shot, few-shot, or fine-tuned settings and exhibit varying levels of success depending on the task, language, and model scale.

However, despite these advances, fundamental questions remain about the linguistic competence of LLMs. For instance, Hu et al. [2020] present a large-scale evaluation of syntactic generalization that reveals several striking limitations in current neural language models. Contrary to expectations, perplexity—a standard benchmark for language modeling—was poorly correlated with syntactic generalization ability: models with lower perplexity often performed worse on syntax-targeted tests, undermining the assumption that broad coverage leads to deeper linguistic competence. Standard LSTM architectures, even when trained on tens of millions of tokens, failed to robustly learn basic syntactic phenomena such as subject–verb agreement, and performed at near-chance levels on long-distance dependencies and center-embedding constructions. Transformer-based models like GPT-2 achieved stronger performance, but much of this success was attributed to preprocessing artifacts such as subword tokenization rather than architectural advantages. Perhaps most notably, even the best models consistently failed on syntactic licensing tests, such as those involving negative polarity items—phenomena that humans acquire early and handle reliably. These results suggest that many linguistic generalizations that are trivial for humans remain surprisingly elusive for even the most capable neural language models.

4.1 Foundations of Generative Linguistics and Formal Semantics

The development of generative linguistics in the second half of the 20th century marked a dramatic shift in the scientific study of language, emphasizing the formal structure of syntax and its relationship to meaning. This subsection highlights five foundational works that shaped this intellectual movement and laid the groundwork for modern syntactic and semantic theory.

Chomsky (1957): *Syntactic Structures*. This work introduced the concept of a *generative grammar*—a formal system capable of generating the infinite set of grammatical sentences in a language. Chomsky proposed a set of phrase structure rules and transformational rules, arguing

that syntactic theory should be precise, explicit, and formally rigorous. The book also introduced the notion of deep structure and raised fundamental questions about the inadequacy of finite-state models for capturing human syntax. It effectively launched the generative paradigm and established syntax as a central object of linguistic inquiry.

Chomsky (1965): *Aspects of the Theory of Syntax*. In this follow-up, Chomsky significantly extended his theoretical framework by distinguishing between *competence* (the idealized knowledge of language) and *performance* (actual language use). He refined the concept of deep vs. surface structure and introduced the T-model architecture of grammar. The work also proposed the theory of *universal grammar*—a set of innate principles and parameters shared by all human languages—which became a cornerstone of subsequent research in generative syntax and language acquisition.

Katz and Fodor (1963): *The Structure of a Semantic Theory*. This influential paper represented one of the earliest attempts to systematically integrate semantics into the generative framework. Katz and Fodor proposed a compositional view of lexical meaning, where word senses are defined in terms of bundles of semantic features. Their model attempted to link syntactic structure to semantic interpretation, setting the stage for formal approaches to the syntax–semantics interface. Though later surpassed by model-theoretic semantics, their work was instrumental in establishing semantics as a formal, computationally tractable component of linguistic theory.

Jackendoff (1972): *Semantic Interpretation in Generative Grammar*. Jackendoff extended the generative program by proposing a more elaborate mapping from syntactic structure to semantic interpretation. He introduced the idea of multiple levels of representation—syntactic, semantic, and phonological—and emphasized their interaction. His framework helped clarify the architecture of grammar and provided early formulations of what would become *conceptual semantics*, influencing both generative and cognitive approaches to meaning.

Montague (1970): *English as a Formal Language*. Montague revolutionized semantic theory by showing that natural languages could be treated with the same formal rigor as logical systems. He applied tools from model theory, lambda calculus, and type theory to build compositional semantics for fragments of English. This work laid the foundation for *Montague Grammar*, a highly influential framework that unified syntax and semantics. Although developed independently of the Chomskyan tradition, Montague’s work profoundly influenced generative semantics and was later integrated into transformational grammar by scholars such as Partee.

Together, these works represent the intellectual bedrock of modern linguistic theory. They established the principles of syntactic generation, formal semantic interpretation, and the architecture of linguistic competence, shaping decades of subsequent research in theoretical linguistics, psycholinguistics, and computational models of language.

4.2 The Historical NLP Pipeline

The development of natural language processing (NLP) has been marked by a series of methodological revolutions, each building on and extending the capabilities of its predecessors. This section traces the evolution of key components in the NLP pipeline, from early rule-based systems through statistical methods to modern neural approaches.

Part-of-Speech Tagging. The task of part-of-speech (POS) tagging—assigning grammatical categories like noun, verb, or adjective to words in a sentence—has evolved through several major

paradigms over the past decades. Early work in the 1960s and 70s relied on rule-based approaches and manually curated lexicons. Greene and Kuno Greene and Ralph [1963] presented one of the earliest automatic tagging systems, combining dictionary lookups with hand-written disambiguation rules. The release of the Brown Corpus Francis and Kucera [1979] marked a turning point by providing a large, annotated dataset for statistical analysis.

In the early 1990s, the statistical revolution in NLP led to probabilistic models becoming dominant. Kupiec Kupiec [1992] applied Hidden Markov Models (HMMs) to POS tagging, modeling the tagging task as a sequence labeling problem. Brill Brill [1995] introduced a transformation-based learning approach that combined rule induction with error-driven updates, achieving competitive results with interpretable rules.

Machine learning continued to drive innovation. Ratnaparkhi Ratnaparkhi [1996] introduced a Maximum Entropy tagger that incorporated contextual features in a flexible framework. Conditional Random Fields (CRFs), as formalized by Lafferty et al. Lafferty et al. [2001], improved sequence labeling by modeling entire sequences of tags jointly, overcoming some of the limitations of HMMs.

The 2000s also saw the development of more feature-rich models. Toutanova et al. Toutanova et al. [2003] proposed a cyclic dependency network that leveraged linguistic features like prefixes and suffixes, improving accuracy on benchmark corpora.

With the rise of deep learning in the 2010s, POS tagging transitioned into the neural era. Huang et al. Huang et al. [2015] demonstrated that bidirectional LSTM-CRF models could outperform traditional methods without the need for hand-engineered features. Akbik et al. Akbik et al. [2018] introduced contextual string embeddings, capturing rich character-level and word-level representations. The introduction of large pretrained language models like BERT Devlin et al. [2019] further advanced the state of the art in POS tagging, especially when fine-tuned for sequence labeling tasks.

Finally, the push toward multilingual and cross-lingual tagging led to the creation of Universal Dependencies (UD) Nivre et al. [2016], which provided standardized POS tags and treebanks across dozens of languages, enabling consistent evaluation and transfer learning techniques.

Phrase Structure Parsing. Phrase structure parsing, also known as constituency parsing, is a foundational task in natural language processing (NLP) that aims to recover the hierarchical phrase-based structure of a sentence. Over the decades, the field has transitioned from symbolic rule-based systems to statistical models and, more recently, to neural and transformer-based architectures.

The theoretical basis for phrase structure parsing originated in the field of linguistics, with Chomsky’s *Syntactic Structures* Chomsky [1957] introducing the concept of generative grammars and context-free rules. These ideas laid the groundwork for early parsers, which relied on hand-crafted rules but struggled with ambiguity and scale.

The introduction of the Penn Treebank Marcus et al. [1993] marked a pivotal moment for empirical NLP. This large annotated corpus enabled the training and evaluation of statistical models, moving the field beyond rule-based systems. Leveraging this resource, Collins Collins [1997] developed one of the first successful lexicalized probabilistic parsers, incorporating headwords into probabilistic context-free grammars (PCFGs) to improve disambiguation. In parallel, Charniak Charniak [1997] proposed a PCFG-based parser that further demonstrated the viability of statistical parsing using rich lexical statistics.

As parsing performance plateaued under purely generative models, researchers shifted toward discriminative approaches. Collins and Koo Collins and Koo [2005] introduced reranking techniques, using discriminative models to select the best parse from a list of candidates generated by a baseline parser. Around the same time, McClosky et al. McClosky et al. [2006] showed how semi-supervised learning via self-training could boost parsing performance by leveraging large amounts of unlabeled

text, a strategy that prefigured later trends in unsupervised and transfer learning.

The next paradigm shift came with the rise of neural networks. Socher et al. [2013] pioneered the use of recursive neural networks for parsing, modeling the compositional structure of phrases directly in a vector space. This work demonstrated that deep learning could encode syntactic structure in a meaningful and learnable way. Soon after, Vinyals et al. [2015] reframed parsing as a sequence-to-sequence problem, using LSTMs to generate trees directly from input sentences—an early sign of the growing convergence between parsing and general-purpose sequence modeling.

The introduction of self-attention mechanisms led to a leap in performance and efficiency. Kitaev and Klein [2018] proposed a parser based on a self-attentive encoder that achieved state-of-the-art results without recurrent networks. They later extended their model with multilingual pretraining [2019], showing that transformer-based architectures like BERT could be effectively adapted for constituency parsing across languages.

Finally, Mrini et al. [2020] revisited the design of transformer architectures with an eye toward interpretability and efficiency, refining attention mechanisms specifically for parsing tasks. Their work represents a broader trend in NLP: optimizing pretrained models not just for raw performance, but also for transparency and adaptability.

Dependency Parsing. Dependency parsing aims to identify syntactic relationships between words in a sentence, typically in the form of directed, labeled graphs where each word depends on a syntactic head. Compared to phrase structure parsing, dependency representations are often more suitable for morphologically rich and free word order languages, making them a popular choice in multilingual and applied NLP tasks.

The theoretical foundation of dependency parsing can be traced back to Mel’čuk’s seminal work on dependency syntax [1988], which formalized the representation of syntactic relations in a way that was both linguistically expressive and computationally viable. Although primarily theoretical, Mel’čuk’s framework laid the groundwork for many subsequent parsing systems.

A practical breakthrough came with Yamada and Matsumoto’s data-driven transition-based parser [2003], which modeled parsing as a sequence of decisions made by a classifier—specifically a support vector machine. Their approach pioneered the use of machine learning in dependency parsing and inspired a generation of transition-based parsers.

In contrast, McDonald et al. [2005] introduced a graph-based alternative to dependency parsing, treating the task as finding the maximum spanning tree over possible head-dependent arcs. This approach, trained with a structured perceptron, offered global optimization and quickly became a strong baseline. McDonald and Pereira [2006] extended this work by incorporating higher-order features such as sibling and grandparent relationships, thereby improving parsing accuracy while maintaining tractability.

Concurrently, Nivre and colleagues released MaltParser [2006], a flexible and widely adopted transition-based parsing toolkit that supports various parsing strategies and languages. Its configurability and speed made it a go-to solution for both academic research and practical applications.

Pushing the boundaries of expressiveness, Koo and Collins [2010] introduced an efficient third-order parser that incorporated rich structural features such as tri-siblings and grand-siblings into the graph-based framework. Their work demonstrated how higher-order relationships could improve accuracy without prohibitive computational costs.

Bridging the gap between transition- and graph-based methods, Zhang and Nivre [2011] showed that using beam search and non-local features in transition-based parsing could dramatically improve performance, allowing fast parsers to model longer-range dependencies more effectively.

The era of neural parsing began with the work of Chen and Manning [2014], who replaced hand-engineered feature templates with dense, learned representations using feedforward neural networks. This marked the first successful neural dependency parser and set the stage for deep learning approaches in syntactic analysis.

Dozat and Manning [2017] further advanced the state of the art with their biaffine parser, which combined BiLSTM encoders and attention-based scoring in a graph-based architecture. This model achieved exceptional accuracy and became the de facto baseline for modern dependency parsing.

Finally, Kondratyuk and Straka [2019] leveraged multilingual BERT representations to build a single model capable of parsing 75 languages. This work illustrated the power of transfer learning and multilingual pretraining, solidifying dependency parsing’s place in the transformer era.

Semantic Parsing. The relationship between syntactic and semantic parsing has long been a central concern in NLP. Early work by Zelle and Mooney [1996] introduced one of the first data-driven semantic parsers, mapping natural language to Prolog queries via inductive logic programming. This line of research evolved through grammar-based and statistical approaches.

Zettlemoyer and Collins [2005] proposed a method for learning Combinatory Categorical Grammars (CCGs) from utterance-logical form pairs. This was extended by Zettlemoyer and Collins [2007], who introduced an online learning approach for more scalable parsing. Bos et al. [2004] demonstrated wide-coverage semantic representations with CCGs, and Kwiatkowski et al. [2010] showed how probabilistic grammars could be induced from logical forms.

An alternative view came from Wong and Mooney [2006], who framed semantic parsing as a machine translation problem using synchronous grammars. Wong and Mooney [2007] incorporated lambda calculus to enhance compositional expressivity.

Later work moved toward weak supervision. Clarke et al. [2010] learned from question-answer pairs instead of logical forms, introducing supervision via task outcomes. Goldwasser and Roth [2011] and Artzi and Zettlemoyer [2011] leveraged weak signals and bootstrapping from dialogue data. A major step came with Liang et al. [2013]’s Dependency-Based Compositional Semantics (DCS), which induced semantic parsers from QA supervision using a dependency-style formalism.

Text-to-SQL. The task of translating natural language questions into executable SQL queries—commonly referred to as *text-to-SQL*—has undergone significant transformation over the past decade. Initial efforts in text-to-SQL were rule-based and constrained to narrow domains. The advent of neural networks enabled more flexible models. Zhong et al. [2017] introduced Seq2SQL, an early neural approach using reinforcement learning, marking the shift toward end-to-end learning.

The release of the **Spider** dataset by Yu et al. [2018] was a critical turning point. Designed for complex, cross-domain queries, **Spider** quickly became the de facto benchmark for evaluating generalization in text-to-SQL systems. Its influence extended to shaping evaluation metrics and driving architectural innovations.

Following the success of general-purpose pre-trained language models (PLMs), researchers began leveraging them for semantic parsing. Techniques such as schema linking and representation learning, exemplified in RESDSQL by Li et al. [2023a], pushed the performance of PLM-based systems on **Spider** and similar datasets.

Further, Deng et al. [2021] demonstrated that grounding pretraining in schema and table structures could substantially improve performance, introducing structure-aware PLM training. These approaches, however, were limited by the fixed capacity of PLMs and required task-specific tuning.

The recent emergence of Large Language Models (LLMs) has transformed the text-to-SQL landscape. Studies such as Rajkumar et al. [2022] and Pourreza and Rafiei [2023] evaluated LLMs

like GPT-3 and GPT-4 in zero- and few-shot regimes, showing strong performance without explicit training. DIN-SQL in particular introduced decomposition and self-correction to improve in-context learning (ICL).

Despite these advances, proprietary LLMs pose challenges for reproducibility and data privacy. To address this, the community has turned to open-source LLMs and training pipelines. Recent work by Li et al. [2024] proposed **CodeS**, a pre-training and fine-tuning framework specifically designed for SQL generation. It builds upon code-specific language models like StarCoder and incorporates SQL-augmented corpora for improved domain alignment.

Complementarily, Hong et al. [2024] proposed **Knowledge-to-SQL**, an expert-augmented framework using fine-tuned models to inject domain knowledge and schema-aware reasoning, showing promising results on realistic and adversarial datasets.

A comprehensive benchmark comparison by Gao et al. [2024] contrasted both proprietary and open-source LLMs across multiple datasets, including **Spider**, **BIRD**, and **Spider-Realistic**. Their findings illustrate a performance gap between closed and open models but highlight the rapid improvement of open-source alternatives through fine-tuning and prompt engineering.

4.3 LLMs and the Historical NLP Pipeline

Recent work has demonstrated that large language models (LLMs) benefit not only from larger model size and better pretraining objectives, but also from the integration of external structure—be it in the form of symbolic knowledge, task-specific prompts, or compositional representations. This section surveys several prominent methods that extend the RASP paradigm through retrieval, structural supervision, or neuro-symbolic grounding to enhance LLM-based semantic parsing and generalization.

Syntactic Competence in LLMs. While LLMs have shown promise in zero-shot and few-shot syntactic and semantic parsing, their ability to disambiguate lexical meaning and generalize to unseen concepts remains limited—especially in structured parsing tasks requiring precise, graph-based outputs. A recent line of work explores whether external knowledge sources can enhance LLM performance by augmenting their inputs with structured lexical information.

One of the most promising developments in this area is the **RASP** framework (*Retrieval-Augmented Semantic Parsing*) proposed by Zhang et al. [2024]. RASP addresses a core challenge in neural semantic parsing: the tendency of LLMs to rely on heuristics, such as always predicting the most frequent sense of a word when encountering ambiguous or out-of-distribution concepts. This behavior often leads to “lucky guesses” and degrades the model’s ability to robustly disambiguate word meanings in open-domain settings. RASP tackles this problem by injecting *retrieved lexical glosses*—specifically from WordNet—into the model input, thereby grounding the model’s output in explicit semantic context.

The method operates in two stages: a retrieval step and a generation step. During retrieval, all plausible WordNet synsets (senses) for each content word in the input are gathered, along with their glosses. These glosses are concatenated and prepended to the input text in a structured format. Then, a decoder-only LLM (e.g., Mistral, Gemma, Qwen, LLaMA) is prompted with this enriched input to generate Discourse Representation Structures (DRSs) that align with the correct word senses. Formally, this integrates two probabilities: $p(o' \mid x)$ for retrieving relevant glosses, and $p_{\text{decoder-only}}(o \mid x, o')$ for generating the output conditioned on both the original input and the retrieved information.

Semantic Parsing and Structure-Aware Augmentation. Recent work has demonstrated that LLMs can be significantly enhanced when paired with structured lexical knowledge in context. Rather than relying solely on scale or additional training data, approaches like RASP leverage existing resources like WordNet to bridge the gap between LLM fluency and semantic precision. This approach is especially valuable in domains where generalization to rare or out-of-vocabulary concepts is critical.

Cheng et al. [2023] propose **UPRISE**, a retrieval-augmented framework that improves zero-shot performance in structured prediction by retrieving task-specific prompts. Rather than relying solely on handcrafted prompt templates, UPRISE builds a bank of example prompts and retrieves the most relevant ones based on the test input using dense similarity. These retrieved prompts are appended to the model input before decoding, effectively grounding generation in in-context prior examples.

Liu et al. [2021] introduce **KG-BART**, a framework that incorporates subgraphs from external knowledge graphs—specifically ConceptNet—into the BART model to enhance commonsense reasoning. The model encodes these graphs alongside the input sequence, allowing the decoder to attend to both textual and symbolic representations. Although developed for commonsense QA, the approach demonstrates the value of integrating structured external knowledge into generative models.

Lu et al. [2023] present **StructGPT**, a method that treats structured prediction as a sequence of programmatic actions using a domain-specific language (DSL). Rather than producing raw text, the LLM is prompted to generate parse trees or structured outputs by selecting from tool-like functions (e.g., "AddNode", "SetLabel"). This framing allows the LLM to reason in a constrained, compositional space, dramatically improving structure compliance.

Retrieval and Structure-Aware Augmentation. **ATLAS** [Izacard et al., 2024] is a general-purpose retrieval-augmented generation framework that fine-tunes LLMs to perform tasks conditioned on external retrieved text segments. Unlike systems that treat retrieval as static, ATLAS jointly learns both the retriever and generator end-to-end. Although originally proposed for QA and open-domain tasks, ATLAS has since been adapted to AMR parsing and other meaning representation tasks.

Li et al. [2023b] propose **structure-aware pretraining** (SAP), which modifies the pretraining objective of LLMs to emphasize structural relationships in data such as tables and parse trees. By pretraining on structured data with hierarchical and compositional constraints, the resulting models generalize better to downstream tasks like dense retrieval and Text-to-SQL.

Evaluation Challenges and Benchmarks. Bharadwaj et al. [2023] introduce **BENCHCLAMP**, a benchmark designed to evaluate pretrained language models on structured prediction tasks across nine datasets spanning seven output formalisms, including semantic and syntactic parsing. The authors propose a unified framework that supports both fine-tuning and few-shot prompting while enabling efficient constrained decoding via task-specific context-free grammars.

Recent work by Ettinger et al. [2023] investigates whether large language models (LLMs) such as GPT-3, ChatGPT, and GPT-4 can serve as expert linguistic annotators in the task of Abstract Meaning Representation (AMR) parsing. While the models are able to reproduce the basic surface format of AMR graphs—demonstrating familiarity with its syntactic structure—they consistently fail to generate semantically accurate parses.

Conclusion. Taken together, these studies highlight the expanding capacity of LLMs to perform syntactic and semantic analysis across languages and tasks. From POS tagging to deep compositional semantics, LLMs increasingly demonstrate internalized linguistic structure. Yet consistent challenges in structured prediction—especially for multilingual and zero-shot scenarios—indicate that traditional parsing insights remain essential. This review sets the stage for our own investigation into [insert your focus here], which further explores the interface between LLM prompting, structure induction, and linguistic generalization.

4.4 Categorical Grammar

The origins of categorical grammar can be traced back to the early work of Ajdukiewicz, who proposed a formal system for representing syntactic structure using category assignments and functional composition Ajdukiewicz [1935]. His pioneering ideas introduced the notion that syntactic categories could be combined through well-defined operations, laying the groundwork for later developments in mathematical linguistics.

Building on this, Bar-Hillel extended Ajdukiewicz’s notation and explored its application to natural language syntax, introducing a quasi-arithmetical formalism for syntactic description Bar-Hillel [1953]. His work played a crucial role in bridging the gap between philosophical logic and formal grammar, and it anticipated several principles of modern computational linguistics.

A major formalization of categorical grammar was provided by Lambek, who introduced a type-logical approach that framed sentence structure within a logical calculus Lambek [1958]. The Lambek calculus remains a cornerstone of categorical grammar theory, offering a powerful and elegant method for capturing syntactic inference through type composition.

In the modern era, Steedman significantly advanced the theory with his work on Combinatory Categorical Grammar (CCG), a highly lexicalized and flexible grammar formalism Steedman [1996]. Steedman’s contributions have been central to the application of categorical principles in computational linguistics and have influenced a range of systems in syntax and semantics.

Together, these works define the intellectual trajectory of categorical grammar, from its philosophical roots to its current role in formal and computational linguistics.

5 Background on Scaleable Linguistic Annotation

5.1 Mechanical Turk and the Promise—and Limits—of Crowdsourced Annotation

The emergence of crowdsourcing platforms such as Amazon Mechanical Turk (MTurk) marked a transformative moment in how the NLP community approached data annotation. Initially seen as a solution to the bottlenecks of expert annotation—especially in scaling to large datasets—MTurk enabled rapid, low-cost collection of labeled data from a distributed pool of non-expert workers. Early work highlighted both the promise and pitfalls of this strategy, and over time, a more nuanced understanding of its limitations has emerged.

The Promise. Amazon introduced Mechanical Turk in 2005 as a platform for “artificial artificial intelligence” [Borthwick et al., 2005], designed to crowdsource tasks that were hard for machines but trivial for humans. This infrastructure was quickly adopted by NLP researchers. Snow et al. [2008] conducted one of the first systematic evaluations of MTurk for linguistic annotation, demonstrating that the aggregation of non-expert judgments could rival expert-level annotations across tasks such as word sense disambiguation and textual entailment. Crucially, this study provided early evidence

that annotation quality could be recovered through redundancy and statistical modeling, thereby opening the door for broader adoption of crowdsourced labeling.

In parallel, Callison-Burch [2009] applied MTurk to the evaluation of machine translation output, showing that inexpensive crowd-based assessments correlated well with expert judgments. These findings established MTurk not only as a tool for dataset construction, but also as a viable component of evaluation pipelines. The Stanford NLI dataset [Bowman et al., 2015], a cornerstone of modern semantic inference research, was built using a hybrid approach: sentence pairs were generated programmatically and then validated by MTurk annotators. This demonstrated how crowdsourcing could scale the creation of complex semantic datasets when combined with automation and careful task design.

The Limits. Despite early optimism, several studies began to identify important caveats. Fort et al. [2011] critically examined MTurk’s role in NLP, warning of its ethical blind spots—especially low pay rates, lack of labor protections, and the cognitive toll of repetitive or emotionally taxing tasks. They also pointed to inconsistencies in annotation quality, especially in subjective or ambiguous tasks, where worker motivation and understanding were difficult to control.

Subsequent work has emphasized that the assumptions behind crowdsourcing often mask deeper structural problems. Sabou et al. [2014] argued for best-practice guidelines in crowdsourced annotation, highlighting the need for clear instructions, quality control mechanisms, and fair treatment of annotators. Paullada et al. [2021] provided a broader critique, noting that reliance on large-scale crowd annotation has often led to datasets that are poorly documented, unrepresentative, or ethically problematic. They argue for a data-centric reevaluation of machine learning pipelines, in which the origin, curation, and social implications of datasets are treated as core scientific concerns.

Reflection. Taken together, these works illustrate the dual nature of crowdsourced annotation. On one hand, platforms like MTurk have made it possible to scale up dataset construction rapidly and affordably, which has been vital to the development of large neural models. On the other hand, the limitations—both practical and ethical—have become increasingly visible, especially as annotation tasks grow more complex and value-laden. In contemporary NLP, annotation practices are beginning to shift toward more curated, expert-driven, or hybrid systems (e.g., combining LLM-generated suggestions with human verification), as researchers grapple with how to align annotation quality, fairness, and sustainability.

5.2 Large Language Models as Linguists in the Literature

Recent experiments demonstrate that large language models (LLMs) can support the enrichment of lexical resources such as FrameNet by identifying new lexical units and generating annotated example sentences. Using zero-shot prompting, models like GPT-4o and Claude 3.5 Sonnet achieved high precision in classifying verbs into appropriate semantic frames and produced natural-sounding, grammatically correct example sentences annotated with core frame elements. Moreover, LLMs showed some ability to distinguish between aspectual verb classes (e.g., activities vs. achievements) based on syntactic diagnostics. However, the results also reveal important limitations. The quality of output depends heavily on prompt design, and models sometimes misinterpret verb senses or generate contextually ambiguous examples. Additionally, linguistic diagnostics (such as co-occurrence with temporal adverbials) do not always function reliably when applied to LLMs, due to the models’ tendency to produce acceptable-sounding but semantically inconsistent constructions. These findings suggest that while LLMs are not yet capable of fully replacing expert annotation, they can

serve as effective collaborators or second annotators in the process of developing semantic resources [Koeva, 2024].

Recent work demonstrates that large language models (LLMs), such as GPT-3 (text-davinci-002), can accurately simulate human acceptability judgments of verb argument structure constructions, outperforming previous computational and theoretical models [Ambridge and Blything, 2024]. Unlike earlier approaches that required extensive hand-coded input, including pre-specified semantic features and manually curated training data, LLMs produce high correlations with human ratings with no such scaffolding. For example, GPT-3’s ratings of English causative constructions showed strong alignment with adult human judgments (e.g., $r = 0.92$, $\tau = 0.80$). Furthermore, the model generalized appropriately to novel verbs, indicating sensitivity to semantic properties. The authors argue that LLMs should be considered executable theories of language acquisition, with their architectures and training regimes encoding empirically testable hypotheses, in contrast to traditional models which rely on oversimplified, intuitively appealing abstractions.

Recent work demonstrates that large language models (LLMs) are capable of sophisticated metalinguistic reasoning across a range of linguistic tasks. Among the evaluated models, OpenAI’s **o1-preview** stands out for its exceptional performance, particularly in tree-based syntactic analysis and abstract phonological rule induction. On tasks involving recursive syntactic structures and syntactic movement, **o1-preview** significantly outperforms GPT-3.5, GPT-4, and LLaMA 3.1, achieving near-ceiling accuracy in both syntactic tree construction and the identification of movement traces. Crucially, **o1-preview** also demonstrates the ability to generalize phonological rules from invented datasets, including cases involving unnatural or rare sound patterns. These findings suggest that with sufficient architectural enhancements, LLMs can acquire genuinely abstract grammatical knowledge, not merely memorize surface-level patterns [Begus et al., 2025].

Blevins et al. [2023] explore how large language models can be prompted to output structured linguistic analyses, including syntactic trees and morphological segmentation. The study finds that while models like GPT-3 perform well on surface-level morphological tasks, their ability to generate deep hierarchical structure is more limited and depends heavily on prompt phrasing. This paper is one of the first to systematically evaluate the effectiveness of prompting for eliciting explicit linguistic structures, showing both the promise and limits of treating LLMs as linguists via prompt engineering.

Behzad et al. [2023] introduce ELQA, a curated corpus of metalinguistic questions and answers about English designed to evaluate the metalinguistic awareness of LLMs. The questions range from terminology and definitions to error identification and grammaticality judgments. Their analysis shows that while LLMs like GPT-3.5 can answer basic metalinguistic questions well, performance on more technical or abstract queries is mixed. The resource provides a benchmark for treating LLMs as language analysts rather than just generators.

Wilcox et al. [2018] examine whether RNN-based language models can learn and generalize filler-gap dependencies, a hallmark of syntactic competence in humans. Using psycholinguistic-inspired experiments, they show that LMs trained purely on text can exhibit sensitivity to unbounded dependencies such as wh-movement, suggesting an implicit understanding of hierarchical structure. However, their generalization is incomplete, and errors increase with more deeply embedded clauses. This work laid early groundwork for probing syntactic generalizations in LMs.

Gulordava et al. [2018] demonstrate that LSTM-based language models trained on multilin-

gual corpora can capture long-distance syntactic dependencies even in nonsensical “colorless green” sentences. The authors show that models trained without semantic grounding can still predict syntactically well-formed continuations, supporting the idea that language models develop abstract grammatical representations. This study is widely cited for arguing that hierarchical syntax can emerge from distributional learning alone.

Matuskevych and Culbertson [2022] test whether RNNs can learn the recursive structure of noun phrases using an artificial grammar learning paradigm. The study shows that RNNs trained on synthetic languages with hierarchical NP structure can generalize to novel recursive inputs. Importantly, this work uses techniques modeled after human learning experiments, providing a strong analogy between human learners and neural networks. It supports the claim that LMs can act like linguists under the right training and testing conditions.

Yedetore et al. [2023] ask whether LLMs trained on child-directed speech (a limited and more naturalistic corpus) can generalize hierarchical rules. Using Transformer models, they find that such networks struggle with hierarchical generalization, especially when data is sparse. This contributes to the “poverty of the stimulus” debate and suggests that LLMs may require more data than humans to achieve similar generalizations—unless architectural enhancements (like chain-of-thought) are introduced. The paper is especially relevant for evaluating the limits of LLMs as cognitive models of linguists.

Alivanistos et al. (2022) Alivanistos et al. [2022] present a framework in which prompting is used not just as a task input method, but as a probing mechanism to extract structured knowledge from pretrained language models. Specifically, they show how carefully designed prompts can be used to elicit entity and relation triples, effectively reconstructing knowledge graphs. The model’s ability to complete these relations without additional fine-tuning demonstrates that LLMs retain a rich latent knowledge base. This aligns with the notion of treating LLMs as linguists, capable of metalinguistic analysis and structure inference when prompted appropriately.

Li et al. (2022) Li et al. [2022] propose replacing traditional diagnostic probes (small classifiers trained on frozen model representations) with learned prompts to assess what language models know. Their approach, called “probing via prompting,” involves training discrete or continuous prompts that elicit a model’s latent knowledge in a zero-shot or few-shot setting. They show that prompting can serve the same purpose as post-hoc probes, but with the benefit of being closer to the model’s natural inference behavior. This methodology strengthens the view that LLMs can be analyzed in the same way we might evaluate a linguistic informant.

Min et al. (2022) Min et al. [2022] explore what aspects of in-context learning are responsible for LLM performance. Through a series of controlled experiments, they find that much of the performance can be attributed not to learning from the examples per se, but from strong prior knowledge in the model, especially about label distributions and common class patterns. Even when demonstrations are perturbed or semantically meaningless, models often perform well. This suggests that LLMs behave less like tabula rasa learners and more like agents with pre-existing linguistic knowledge—which can be elicited through structured prompting.

Press et al. (2022) Press et al. [2022] tackle the “compositionality gap” in LLMs, which is the mismatch between compositional generalization and model performance. They propose a method

called "chain-of-thought with intermediate prompting," where tasks are decomposed into simpler subtasks and solved in sequence, with each output fed as context to the next. Their results show that LLMs can learn more compositional functions when intermediate reasoning steps are prompted explicitly. This approach provides a procedural lens on LLMs' linguistic competence, treating them as capable of assembling complex outputs from smaller linguistic primitives.

Zhou et al. (2022) Zhou et al. [2022] introduce the method of "least-to-most prompting," a strategy where LLMs are prompted to solve reasoning problems by building from simple to complex sub-problems. This mimics the pedagogical techniques of human learning and shows improved performance on tasks like multi-hop reasoning, arithmetic, and syntax-sensitive parsing. The method emphasizes the value of step-wise syntactic decomposition, and shows that LLMs can learn and apply rule-based structure when guided incrementally—much like a linguist following a deductive analysis.

6 Negative Results on Zero-Shot Dependency Parsing

6.1 Initial Experiments

We begin by verifying the difficulties already reported in the literature (TODO) in using a large language model as a "zero-shot parser." And, then whether the task of general parsing can be broken down into subtasks, by asking specifically for the POS tags, unlabeled arcs, or labels for unlabeled arc.

Zero-Shot Full Parsing We begin by verifying previous results that LLM's can *not* act as zero-shot parsers by replicating the experiment with 25 sample sentences and asking, for each sentence, to simply "output the CoNLL-U depdnecy parses for this sentence." We find that accuracies are low on this task.

Table 1: Zero-Shot Parsing Performance (ChatGPT vs. Stanford Parser)

Model	UAS (%)	LAS (%)	POS Acc. (%)
ChatGPT (zero-shot parsing)	12.4	7.9	89.6
Stanford Parser (gold seg, gold POS)	91.3	88.6	97.8

Although ChatGPT achieved reasonable POS tagging accuracy within the full parsing prompt, its syntactic predictions were far from coherent. The resulting parses frequently violated tree constraints, showed inconsistent head assignments, and produced implausible dependency labels.

Decomposed Subtasks We next tested whether the same model would perform better when the syntactic annotation task was broken down into simpler steps. Specifically, we asked the model to perform: (1) POS tagging word-by-word; (2) head prediction, given each word in isolation; and (3) dependency label prediction, given a word and its gold-standard head. Table 2 presents results for each of these subtasks, again in comparison with the Stanford dependency parser as a baseline.

This decomposition yielded significantly better results than the zero-shot parsing setup, especially for POS tagging, which remained ChatGPT's strongest syntactic capability. While head prediction and dependency label classification still fell well short of the baseline, performance improved substantially compared to the end-to-end parse.

Table 2: Subtask Performance (ChatGPT vs. Stanford Parser)

Task	ChatGPT (%)	Stanford Parser (%)
POS tagging	89.6	97.8
Head prediction (gold POS)	42.3	91.3
Dependency label (gold heads)	31.8	88.6

7 Qualitative Analysis of Success Cases

7.1 Exploratory Analysis

7.1.1 Interpreting Multiple Valid Parses: Evidence of Latent Syntactic Knowledge

To complement our quantitative evaluation, we conducted a qualitative case study of one particularly challenging syntactic phenomenon: prepositional phrase (PP) attachment. PP attachment has long been a source of ambiguity in syntactic parsing, and LLMs are known to struggle with it, especially in zero-shot settings.

We selected a naturally occurring sentence from Twitter that involved a misinterpretation of a “with”-phrase. The sentence was:

She saw the man with the telescope.

This sentence is classically ambiguous: the prepositional phrase “*with the telescope*” can attach either to the noun “*man*” (implying the man had the telescope) or to the verb “*saw*” (implying she used the telescope). In this specific instance, the intended reading was the latter, but ChatGPT, when asked to provide a dependency parse, attached the PP to the noun.

Despite this syntactic error in parsing, we then followed up with a prompt asking the model:

Why might someone interpret this sentence the wrong way?

ChatGPT responded with an explanation that correctly identified the ambiguity and discussed both possible attachment sites:

This sentence is ambiguous because the phrase “with the telescope” could attach either to “the man” or to “saw.” If you attach it to “the man,” it implies the man had the telescope. If you attach it to “saw,” it means she used the telescope to see him. Without additional context, both interpretations are valid.

This qualitative result suggests that while ChatGPT does not reliably produce the correct parse in structurally ambiguous sentences, it is capable of *describing* the ambiguity and articulating the competing interpretations. This supports the idea that LLMs have access to latent syntactic representations and world knowledge that can be activated in interactive settings, even when they fail to apply that knowledge in structured parsing tasks.

7.1.2 Interpreting Multiple Valid Parses: Evidence of Latent Syntactic Knowledge

In a pair of related examples, we explored how ChatGPT handles noun phrase structures involving quantifiers and partitive constructions. Interestingly, the model produced two different syntactic analyses across two cases—one aligning with a traditional parser, and one diverging from it—yet

both interpretations agreed with our own linguistic judgment. This serves as qualitative evidence that ChatGPT possesses flexible and context-sensitive syntactic intuitions, even when its output varies.

In the first case, we examined the sentence:

I really can not even believe some of this content right now.

Here, the parser had incorrectly treated “*some*” as the direct object of “*believe*”, rather than “*content*”, which is the head of the partitive noun phrase “*some of this content*”. ChatGPT’s parse reproduced this same error. However, when prompted to reflect on the structure, the model acknowledged that “*content*” is the semantic core of the object, and that “*some*” functions as a quantifier or determiner. Despite the initial parsing error, the model demonstrated an awareness of the correct structural interpretation upon further analysis.

In the second case, we presented the simpler sentence:

I believe all content.

Here, both the parser and ChatGPT assigned “*content*” as the direct object and “*all*” as its determiner—an analysis that aligns with canonical syntactic structures and our own judgment.

Taken together, these examples illustrate that ChatGPT is capable of recognizing multiple, contextually valid parses—even when its initial output is imperfect. The fact that the model agreed with the Stanford parser in one case, and disagreed in another, yet aligned with human linguistic reasoning in both, supports the idea that LLMs possess latent syntactic knowledge that is accessible through interaction and contextual prompting.

7.1.3 Agreement on a Curated Attachment Dataset

To further evaluate ChatGPT’s syntactic competence, we conducted a collaborative dataset construction exercise focused on prepositional phrase attachment. In this case, the model was asked to generate a set of 20 syntactically ambiguous sentences, each involving a prepositional phrase whose attachment site (e.g., verb or noun) might plausibly vary depending on interpretation.

First, ChatGPT produced the 20 examples, each annotated with the intended attachment (e.g., “attach to verb” or “attach to noun”). We then manually reviewed each sentence, checking whether the proposed attachment was semantically and syntactically plausible. In all cases, we agreed with the model’s judgments, which demonstrated strong contextual awareness and a high degree of alignment with human intuitions.

To complete the loop, we presented the validated dataset back to ChatGPT and asked it to reassess the attachment decisions. The model reaffirmed the original judgments across the board, offering consistent and well-reasoned justifications for each case. This round-trip evaluation—model proposes, human verifies, model reaffirms—offers compelling evidence that ChatGPT is not merely pattern-matching surface strings, but is capable of maintaining coherent and linguistically-informed attachment decisions over multiple turns of interaction.

This experiment also supports the claim that LLMs can be used as collaborative linguistic tools—not just for generation or analysis in isolation, but for *co-construction* and *validation* of structured linguistic phenomena like syntactic ambiguity.

7.1.4 ChatGPT’s Meta-Linguistic Awareness as Evidence of Linguistic Knowledge

In an exploratory exchange, we asked ChatGPT whether it “understands linguistic structure” and followed up with a request for an overview of the concept. The model responded with a coherent,

multi-level explanation of linguistic structure spanning phonology, morphology, syntax, semantics, pragmatics, discourse, and prosody. Each level was illustrated with concrete examples and tied to both theoretical and computational perspectives.

Notably, the model also referenced formal tools such as context-free grammars, dependency trees, and semantic graphs—demonstrating an awareness of how linguistic structure is operationalized in natural language processing systems.

This interaction provides qualitative evidence that ChatGPT possesses a form of meta-linguistic knowledge. It is not merely able to produce syntactically and semantically well-formed sentences, but can also discuss the components and mechanisms that underlie those structures. This capacity supports the claim that LLMs encode structured linguistic representations that can be surfaced through reflective or explanatory prompts.

While such responses do not constitute full-blown theoretical understanding in the human sense, they suggest that ChatGPT’s language generation is informed by a robust, if implicit, grasp of linguistic principles.

8 Beating the Stanford Parser and the Importance of Prompting

8.1 Evaluating PP Attachment Disambiguation at Scale

To go beyond single-example case studies, we designed a controlled experiment to evaluate how well ChatGPT and a traditional syntactic parser handle prepositional phrase (PP) attachment ambiguity in practice. Our focus was on the classic syntactic ambiguity involving constructions like:

I saw the man with the binoculars.

Here, the phrase “*with the binoculars*” can attach either to the noun “*man*” or the verb “*saw*”. Disambiguating such cases requires not only syntactic knowledge but also world knowledge and pragmatic inference.

Dataset Construction. We used ChatGPT (via the web interface) to generate 20 naturally phrased English sentences that each contained a PP with an ambiguous attachment site. We then manually reviewed these examples and used ChatGPT’s own interpretation (as judged from follow-up clarification questions) as a gold standard. To ensure reliability, we verified each example ourselves and confirmed that ChatGPT’s interpretation aligned with our own in all cases (100% agreement).

Parser Comparison. We then parsed all 20 sentences using the Stanford dependency parser and ChatGPT API (with zero-shot, instruction-style prompting). We evaluated the models on whether they assigned the PP to the correct head (noun vs. verb), as defined by our gold standard.

Results. The results were striking: the Stanford parser correctly resolved 10 out of 20 PP attachments (50%), while ChatGPT API got 19 out of 20 correct (95%). Table 3 summarizes these results.

Table 3: PP Attachment Accuracy (20 Ambiguous Sentences)

Model	Correct	Accuracy (%)
ChatGPT API	19 / 20	95.0
Stanford Parser	10 / 20	50.0

Statistical Significance. Despite the small sample size, the performance gap is statistically significant. We conducted a one-sided binomial test to assess whether ChatGPT’s performance could be explained by chance under the null hypothesis of 50% accuracy. The resulting p -value was 2.0×10^{-5} , indicating strong statistical significance ($p < 0.001$). We also computed a 95% confidence interval on ChatGPT’s accuracy using the Wilson score method, yielding [76.4%, 99.1%].

These results suggest that ChatGPT possesses a robust ability to resolve PP attachment ambiguities in context—far surpassing the performance of a standard syntactic parser, even without any fine-tuning or external knowledge sources. While a larger dataset would further reinforce this conclusion, the current data already provide statistically strong evidence that LLMs can leverage both syntactic and semantic information to make informed attachment decisions in ambiguous settings.

8.2 ChatGPT as a Parse Critic: Evaluating Attachment Errors

In a final experiment, we explored whether ChatGPT could serve not as a parser, but as a syntactic evaluator—identifying errors in pre-existing dependency parses. This role is distinct from generation: instead of asking the model to produce a parse, we asked it to assess one.

Experimental Setup. We again used the same 20 ambiguous PP attachment examples from the previous section. Each sentence was parsed using the Stanford dependency parser. We then identified the key ambiguous word (typically a preposition such as *with*) and posed the following prompt to ChatGPT API:

Here is a dependency parse of the sentence. Do you see any errors involving the word “X”?

The model was required to return a binary judgment (error/no error), as well as an explanation when it claimed an error was present. We refer to this as the *evaluated reason*. The binary judgments were compared to the gold-standard attachment decisions (from our earlier annotations), and the explanations were hand-evaluated for correctness.

Results. ChatGPT’s binary judgments correctly identified 19 out of 20 attachment decisions (95% accuracy), outperforming the Stanford parser’s baseline of 50%. Furthermore, in the 19 cases where ChatGPT claimed an error or correctness, it provided explanations that were judged to be accurate in 18 cases (94.7%).

Table 4: Attachment Error Detection and Explanation Accuracy

Model	Binary Accuracy (%)	Explanation Accuracy (%)
ChatGPT API (parse critique)	95.0	94.7
Stanford Parser (attachment accuracy)	50.0	—

Significance Analysis. As with the prior experiment, the difference in binary accuracy is statistically significant. A one-sided binomial test comparing ChatGPT’s 19/20 performance against a 50% baseline yields a p -value of 2.0×10^{-5} . The 95% confidence interval on ChatGPT’s binary accuracy is [76.4%, 99.1%] using the Wilson score method.

Interpretation. These results extend our earlier findings in a new direction: ChatGPT not only interprets syntactic ambiguity better than a traditional parser, but can also *evaluate* parser output and explain its decisions in a linguistically coherent way. This opens the door to new workflows where LLMs act as error detectors or post-hoc critics—particularly useful for ambiguous constructions where syntactic structure and semantics are deeply intertwined. In cases where full structural parsing is not reliable, prompting-based error analysis may be a lightweight alternative.

8.3 Iterative Parse Repair with an Agentic Architecture

In a final experiment, we explored whether ChatGPT could not only identify errors in dependency parses but also iteratively improve them using an agentic architecture. Inspired by interactive reasoning frameworks, we implemented a loop in which the model critiques a parse, proposes modifications, and then applies those modifications until no further changes are suggested.

Method. Starting from the Stanford dependency parser’s output, we processed 100 English sentences. Each parse was represented in CoNLL-U format. For each sentence, we ran an iterative loop composed of the following steps:

1. **Critique step:** Prompt ChatGPT to identify any errors or weaknesses in the current parse, focusing on attachment and labeling.
2. **Revision step:** In a separate prompt, instruct ChatGPT to apply its suggested corrections to produce an updated CoNLL parse.
3. **Termination:** Continue the loop until ChatGPT responds that no further improvements are necessary.

Results. This iterative process led to substantial edits across the dataset. Key statistics are shown in Table 5.

Table 5: Agentic Parse Repair Summary (100 Sentences)

Metric	Value
Total sentences processed	100
Average iterations per sentence	2.1
Average tokens per sentence	12.5
Total dependency arcs	1,250
Arcs modified by ChatGPT	380 (30.4%)
Sentences with at least one change	72 (72%)
Accuracy of changes (manual evaluation)	89.0%
Labeled attachment score (before)	76.3%
Labeled attachment score (after)	84.9%

Interpretation. This experiment demonstrates that ChatGPT can function not only as a parser or critic, but also as an autonomous *editor* of syntactic structure. Through iterative interaction, the model was able to revise nearly one-third of the arcs in the dataset, with a high proportion of those changes judged as valid corrections. Parsing accuracy improved by 8.6 percentage points over the original Stanford parses, despite no access to gold data during the loop.

The success of this architecture supports a new model of syntax-aware NLP: rather than relying on end-to-end structured outputs, we can engage language models in a revision loop, leveraging their strengths in critique, instruction-following, and local reasoning to incrementally improve structured representations. This opens new directions for interactive, hybrid approaches to linguistic annotation and correction.

9 Further Evaluation

10 Discussion

10.1 Zero-Shot Parsing and Negative Results

We have replicated and expanded upon certain *negative* results from the literature pertaining to whether or not LLM’s can do *zero-shot* parsing straight to the CoNLL format, to to “annotate” in contexts where the prompt is incorrect.

We have seen that *prompt strategy matters*, and that an overall *type* of annotation (e.g., prepositional phrase attachment) that can be gotten annotated accurately through one *prompting style* might be totally different in a different prompting style.

10.2 Promise of LLM’s a Linguists

* it works * lots of evidence * currently a prototype * can be expanded into a real product

10.3 Implications for Semantic Parsing

* should work * need to combine 1) it’s a linguist, plus 2) semantic outlook

10.4 Implications for Information Retrieval

* can use Logic * can move beyond just the llm

11 Conclusion

We have shown through a mix of past evidence and new results that large language models *can* in fact act reliably as linguists, especially with regards to the kinds of syntactic labeling tasks that would be required to do open-domain semantic parsing.

We have specifically shown that for the specific difficult case of *prepositional phrase attachment*, that a “zero shot” LLM performance is comparable to that of a human, and far beyond that of the Stanford Stanza parser, which we take to be a representation of the state-of-the-art.

We also validated other aspects of the syntactic parse—POS tags and main verb identification (and possibly argument identification)—through manual evaluation to show that these tasks also work.

We believe that these constitute a “prototype” for the thesis that large language models *can* act as either fully-automated or largely automated linguists.

We call it a “prototype” because we believe that in order to make a “full” working parser, it would be required to figure out a comprehensive automatic (aka. “agentic”) workflow to actually do *full* coverage parsing.

In other words, we believe that these results suggest that LLM’s *do* have the ability to label syntactic information. Now, to prove the thesis that LLM’s can do *enough* syntactic labeling to arrive at high-quality open-domain semantic parses, we need to figure out *which* workflow of LLM prompts and other function calls will arrive at high-quality parses, and demonstrate that one exists.

References

- Kazimierz Ajdukiewicz. Die syntaktische konnexität. *Studia Philosophica*, 1:1–27, 1935.
- Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. *Proceedings of COLING*, 2018.
- Dimitrios Alivanistos, Selene Báez Santamaría, Michael Cochez, Jan-Christoph Kalo, Emile van Krieken, and Thiviyan Thanapalasingam. Prompting as probing: Using language models for knowledge base construction. In *Proceedings of the LM-KBC 2022: Knowledge Base Construction from Pre-trained Language Models*, 2022.
- Ben Ambridge and Liam Blything. Large language models are better than theoretical linguists at theoretical linguistics. *Theoretical Linguistics*, 50:33–48, 07 2024. doi: 10.1515/tl-2024-2002.
- Yoav Artzi and Luke Zettlemoyer. Bootstrapping semantic parsers from conversations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 421–432, 2011.
- Yehoshua Bar-Hillel. A quasi-arithmetical notation for syntactic description. *Language*, 29(1): 47–58, 1953.
- Gasper Begus, Maksymilian Dabkowski, and Ryan Rhodes. Large linguistic models: Investigating llms’ metalinguistic abilities, 2025. URL <https://arxiv.org/abs/2305.00948>.
- Shabnam Behzad, Keisuke Sakaguchi, Nathan Schneider, and Amir Zeldes. Elqa: A corpus of metalinguistic questions and answers about english. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2031–2047. Association for Computational Linguistics, 2023. URL <https://aclanthology.org/2023.acl-long.113>.
- Akari Bharadwaj, Xiang Chen, Sudha Rao, Shashi Narayan, Mike Lewis, Luke Zettlemoyer, et al. Benchclamp: Evaluating language models on structured prediction with constrained decoding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2023. URL https://github.com/microsoft/semantic_parsing_with_constrained_lm.
- Terra Blevins, Hila Gonen, and Luke Zettlemoyer. Prompting language models for linguistic structure. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6649–6663. Association for Computational Linguistics, 2023. URL <https://aclanthology.org/2023.acl-long.367>.

- George Boole. *An Investigation of the Laws of Thought*. Walton and Maberly, 1854.
- Andrew Borthwick et al. Amazon’s mechanical turk: Artificial artificial intelligence. In *Amazon Web Services Blog*, 2005. Original product launch and technical overview.
- Johan Bos, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier. Wide-coverage semantic representations from a ccg parser. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 1240–1246, 2004.
- Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics, 2015.
- Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. In *Computational linguistics*, volume 21, pages 543–565, 1995.
- Chris Callison-Burch. Fast, cheap, and creative: Evaluating translation quality using amazon’s mechanical turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 286–295. Association for Computational Linguistics, 2009.
- Eugene Charniak. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI)*, pages 598–603, 1997.
- Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP*, pages 740–750, 2014.
- Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Weiwei Deng, and Qi Zhang. UPRISE: Universal prompt retrieval for improving zero-shot evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. URL <https://aclanthology.org/2023.emnlp-main.792>.
- Noam Chomsky. *Syntactic Structures*. Mouton, 1957.
- James Clarke, Dan Goldwasser, Michael Chang, and Dan Roth. Driving semantic parsing from the world’s response. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning (CoNLL)*, pages 18–27, 2010.
- Michael Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 16–23, 1997.
- Michael Collins and Terry Koo. Discriminative reranking for natural language parsing. In *Computational Linguistics*, volume 31, pages 25–70, 2005.
- Xi Victoria Deng, Ahmed Hassan Awadallah, Christopher Meek, Oleksandr Polozov, Huan Sun, and Matthew Richardson. Structure-grounded pretraining for text-to-sql. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.

- Timothy Dozat and Christopher D Manning. Deep biaffine attention for neural dependency parsing. In *Proceedings of ICLR (Workshop Track)*, 2017.
- Allyson Ettinger, Jena D. Hwang, Valentina Pyatkin, Chandra Bhagavatula, and Yejin Choi. “you are an expert linguistic annotator”: Limits of llms as analyzers of abstract meaning representation. *arXiv preprint arXiv:2310.17793*, 2023. URL <https://arxiv.org/abs/2310.17793>.
- Karën Fort, Gilles Adda, and K Bretonnel Cohen. Amazon mechanical turk: Gold mine or coal mine? In *Computational Linguistics, Special Issue on Ethics in NLP*, volume 37, pages 413–420, 2011.
- W. Nelson Francis and Henry Kucera. *Manual of Information to Accompany a Standard Corpus of Present-Day Edited American English, for Use with Digital Computers*. Department of Linguistics, Brown University, 1979.
- Gottlob Frege. *Begriffsschrift: A formula language, modeled upon that of arithmetic, for pure thought*. Leopold Voss, 1879.
- Daya Gao, Hang Wang, Yanan Li, Xiaonan Sun, Yichong Qian, Bolin Ding, and Jie Zhou. Text-to-sql empowered by large language models: A benchmark evaluation. *Proceedings of the VLDB Endowment*, 2024.
- Dan Goldwasser and Dan Roth. Confidence driven unsupervised semantic parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1486–1495, 2011.
- Bobrow Greene and Susumu Kuno Ralph. Automatic grammatical tagging of english. *Technical Report R-186, MITRE Corporation*, 1963.
- Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205. Association for Computational Linguistics, 2018. URL <https://aclanthology.org/N18-1108>.
- Kurt Gödel. Über formal unentscheidbare sätze der principia mathematica und verwandter systeme i. *Monatshefte für Mathematik und Physik*, 38(1):173–198, 1931.
- Zijin Hong, Zheng Yuan, Hao Chen, Qinggang Zhang, Feiran Huang, and Xiao Huang. Knowledge-to-sql: Enhancing sql generation with data expert llm. *arXiv preprint arXiv:2402.11517*, 2024. URL <https://arxiv.org/abs/2402.11517>. arXiv:2402.11517.
- Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger P. Levy. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the Association for Computational Linguistics (ACL)*, Online, 2020. URL <https://arxiv.org/abs/2005.03692>. arXiv:2005.03692.
- Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015. URL <https://arxiv.org/abs/1508.01991>. arXiv:1508.01991.

- Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. ATLAS: Few-shot learning with retrieval-augmented language models. *Journal of Machine Learning Research*, 24(75):1–45, 2024. URL <http://jmlr.org/papers/v24/22-1006.html>.
- Nikita Kitaev and Dan Klein. Constituency parsing with a self-attentive encoder. In *Proceedings of ACL*, pages 2676–2686, 2018.
- Nikita Kitaev, Steven Cao, and Dan Klein. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of ACL*, pages 3499–3505, 2019.
- Svetla Koeva. Large language models in linguistic research: the pilot and the copilot. In *Proceedings of the Sixth International Conference on Computational Linguistics in Bulgaria (CLIB 2024)*, pages 319–328, Sofia, Bulgaria, September 2024. Department of Computational Linguistics, Institute for Bulgarian Language, Bulgarian Academy of Sciences. URL <https://aclanthology.org/2024.clib-1.35/>.
- Daniel Kondratyuk and Milan Straka. 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of EMNLP-IJCNLP*, pages 2779–2795, 2019.
- Terry Koo and Michael Collins. Efficient third-order dependency parsers. In *Proceedings of ACL*, pages 1–11, 2010.
- Julian Kupiec. Robust part-of-speech tagging using a hidden markov model. *Computer Speech & Language*, 6(3):225–242, 1992.
- Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. Inducing probabilistic ccg grammars from logical form with higher-order unification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1223–1233, 2010.
- John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289, 2001.
- Joachim Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65(3): 154–170, 1958.
- Hang Li, Junhao Zhang, Cheng Li, and Hong Chen. Resdsq: Decoupling schema linking and skeleton parsing for text-to-sql. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2023a.
- Hang Li, Junhao Zhang, Haoran Liu, Jian Fan, Xuan Zhang, Jun Zhu, Rui Wei, Hong Pan, Cheng Li, and Hong Chen. Codes: Towards building open-source language models for text-to-sql. *arXiv preprint arXiv:2402.16347*, 2024. URL <https://arxiv.org/abs/2402.16347>. arXiv:2402.16347.
- Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. Probing via prompting. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1144–1157. Association for Computational Linguistics, 2022.

- Junsong Li, Jie Zhou, Yutao Yang, Bihao Zhan, Qianjun Pan, Yuyang Ding, Qin Chen, Jiang Bo, Xin Lin, and Liang He. Teaching llms for step-level automatic math correction via reinforcement learning. *arXiv preprint arXiv:2503.18432*, 2025. URL <https://arxiv.org/abs/2503.18432>.
- Xinze Li, Zhenghao Liu, Chenyan Xiong, Shi Yu, Yu Gu, Zhiyuan Liu, and Ge Yu. Structure-aware language model pretraining improves dense retrieval on structured data. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11560–11574, 2023b. URL <https://aclanthology.org/2023.findings-acl.733>.
- Percy Liang, Michael I. Jordan, and Dan Klein. Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446, 2013. doi: 10.1162/COLLa.00127.
- Ye Liu, Yao Wan, Lifang He, Hao Peng, and S. Yu Philip. KG-BART: Knowledge graph-augmented bart for generative commonsense reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6418–6425, 2021. URL <https://ojs.aaai.org/index.php/AAAI/article/view/16542>.
- Canwen Lu, Zhirui Wu, Wayne Xin Zhao, Wei Lu, Yu Hu, Bing Qin, and Ting Liu. Struct-GPT: Language-as-action for structured prediction. *arXiv preprint arXiv:2305.02301*, 2023. URL <https://arxiv.org/abs/2305.02301>.
- Mitchell P Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.
- Yevgen Matusevych and Jennifer Culbertson. Trees neural those: Rnns can learn the hierarchical structure of noun phrases. In *Proceedings of the 44th Annual Meeting of the Cognitive Science Society*, pages 1848–1854, 2022. URL <https://escholarship.org/uc/item/0zk6g6x4>.
- David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceedings of HLT-NAACL*, pages 152–159, 2006.
- Ryan McDonald and Fernando Pereira. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL*, pages 81–88, 2006.
- Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In *Proceedings of ACL*, pages 91–98, 2005.
- Igor A Mel’čuk. *Dependency Syntax: Theory and Practice*. SUNY Press, 1988.
- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2022.
- Khalil Mrini, Franck Dernoncourt, Tengyu Ma, Yulia Tsvetkov, and Trung Bui. Rethinking self-attention: Towards interpretable and efficient transformers for constituency parsing. In *Proceedings of ACL*, pages 6618–6628, 2020.
- Joakim Nivre, Johan Hall, and Jens Nilsson. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, pages 2216–2219, 2006.

- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, pages 1659–1666, 2016.
- Amandalynne Paullada, Inioluwa Deborah Raji, Emily M Bender, Emily Denton, and Alex Hanna. Data and its (dis)contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11):100336, 2021.
- Giuseppe Peano. *Formulaire de mathématiques*. Turin: Bocca, 1895.
- Mohammad Pourreza and Davood Rafiei. Din-sql: Decomposed in-context learning of text-to-sql with self-correction. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*, 2022.
- Nishant Rajkumar, Ruochen Li, and Dzmitry Bahdanau. Evaluating the text-to-sql capabilities of large language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022. URL <https://arxiv.org/abs/2204.00498>. arXiv:2204.00498.
- Adwait Ratnaparkhi. A maximum entropy part-of-speech tagger. *Proceedings of the conference on empirical methods in natural language processing*, 1996.
- Toran Bruce Richards. Auto-gpt: An autonomous gpt-4 experiment, 2023. GitHub repository: <https://github.com/Torantulino/Auto-GPT>.
- Marta Sabou, Kalina Bontcheva, and Leon Derczynski. Corpus annotation through crowdsourcing: Towards best practice guidelines. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, pages 859–866, 2014.
- Timo Schick, Arut Selvan Dwivedi-Yu, and et al. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.
- Noah Shinn and et al. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023.
- Rion Snow, Brendan O’Connor, Dan Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics, 2008.
- Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. Parsing with compositional vector grammars. In *Proceedings of ACL*, pages 455–465, 2013.
- Mark Steedman. *Surface Structure and Interpretation*. MIT Press, 1996.
- Alfred Tarski. The concept of truth in formalized languages. In John Corcoran, editor, *Logic, Semantics, Metamathematics*, pages 152–278. Oxford University Press, 1935. English translation published in 1956.

- Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. *Proceedings of NAACL*, pages 173–180, 2003.
- Alan M Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42):230–265, 1936.
- Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2773–2781, 2015.
- Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. What do rnn language models learn about filler-gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221. Association for Computational Linguistics, 2018. URL <https://aclanthology.org/W18-5423>.
- Yuk Wah Wong and Raymond J. Mooney. Learning for semantic parsing with statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 439–446, 2006.
- Yuk Wah Wong and Raymond J. Mooney. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 960–967, 2007.
- Hiroyasu Yamada and Yuji Matsumoto. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT*, pages 195–206, 2003.
- Aditya Yedetore, Tal Linzen, Robert Frank, and R. Thomas McCoy. How poor is the stimulus? evaluating hierarchical generalization in neural networks trained on child-directed speech. *arXiv preprint arXiv:2301.11462*, 2023. URL <https://arxiv.org/abs/2301.11462>.
- Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qing Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
- Michael Zelle and Raymond J. Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1050–1055. AAAI Press, 1996.
- Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorical grammars. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 658–666, 2005.
- Luke S. Zettlemoyer and Michael Collins. Online learning of relaxed ccg grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 678–687, 2007.
- Yue Zhang and Joakim Nivre. Transition-based dependency parsing with rich non-local features. In *Proceedings of ACL*, pages 188–193, 2011.

- Yulong Zhang, Yujing Yang, Jiangming Liu, Johan Bos, Yao Chen, and Yue Zhang. Rasp: Retrieval-augmented semantic parsing for discourse representation structures. arXiv preprint arXiv:2405.xxxxx, 2024.
- Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017. URL <https://arxiv.org/abs/1709.00103>. arXiv:1709.00103.
- Xuezhi Zhou, Zihang Dai, Andrew Dai, and et al. Least-to-most prompting enables complex reasoning in large language models. In *NeurIPS*, 2022.