Inducing an Incremental Grammar for Production in First Language Acquisition

Xihao Wang (wangxihao@pku.edu.cn)[1,2], Yuan Gao (yg386@cam.ac.uk)[3,4] &
Weiwei Sun (ws390@cam.ac.uk)[3]
[1]Department of Chinese Language and Literature, Peking University
[2]Center for Chinese Linguistics, Peking University
[3]Department of Computer Science & Technology, University of Cambridge
[4]ALTA Institute

Abstract

This paper investigates how children may acquire an incremental grammar that supports word-by-word speech production via computational modeling. We formalize language acquisition as the learning of a probabilistic, bidirectional mapping between utterances and intended meanings. Our first contribution is the design of a precise incremental grammar grounded in Synchronous Hyperedge Replacement Grammar. The grammar constrains syntactico-semantic analyses to be strictly left-branching, reflecting the incremental nature of human language comprehension. We then conduct statistical production-oriented grammar induction over naturalist data. For the meaning-to-text generation task, the incremental grammar yields slightly lower semantic constituent recognition accuracy but unexpectedly higher language generation quality, compared to a hierarchical grammar. It also exhibits a small and consistent accuracy gap of semantic constituent recognition relative to a supervised oracle, an effect not observed for the hierarchical grammar. Together, these results suggest a new method for comparing alternative grammatical architectures in silico.

Keywords: computational modeling; first language acquisition; language production; incrementality

Introduction

Computational modeling has become an important tool for studying language acquisition in cognitive science, by providing a quantitative perspective on learning (Benders & Blom, 2023; MacWhinney, 2010; Pearl & Goldwater, 2016). A closely related line of work in Natural Language Processing (NLP) studies unsupervised grammar induction, which aims to learn the grammatical structure of a language from observational data alone, without explicitly annotated syntactic and/or semantic structures (Goldwater, 2011). Within cognitive modeling, Johnson (2013) frames such statistical grammar induction algorithms as a model of language acquisition, and substantial work has followed in this vein, including approaches driven by comprehension (Kim et al., 2019; Liang et al., 2013; S. Yang et al., 2021) as well as production (Gao & Sun, 2025).

A central challenge of grammar induction is to model the syntax–semantics interface, the mapping between linguistic form and meaning, as well as its rapid development in early childhood. Many researchers have argued that performance (real-time language use) plays an important role in shaping competence grammar, which encodes the abstract knowledge of language (Sag & Wasow, 2011). For example, Hawkins (2004) hypothesizes that word order preferences can be explained by processing efficiency in performance. From this view, an acquisition model that targets the syntax–semantics interface must also take the constraints imposed by processing into account.

One prominent processing constraint is incrementality. Incrementality indicates that in comprehension, we do not wait for an entire sentence to be finished before beginning to interpret it; instead, we process and integrate each word as it arrives in real time. This incremental processing has been empirically demonstrated through multiple methods (Altmann & Kamide, 1999; Kutas & Hillyard, 1980; Pallier et al., 2011). Phillips (2003) argues that constituency, a fundamental syntactic concept, should be defined with respect to incrementality in human language processing. If performance considerations largely determine the shape the architecture of competence grammar and the form of the syntax–semantics interface, a crucial question follows: how can an incremental grammar be induced by a child from linguistic input?

In this paper, we investigate production-oriented grammar induction as a computational simulation of first language acquisition (FLA). Specifically, we study the problem of inducing a probabilistic grammar from meaning representations that encode communicative intentions. The resulting grammar rules support mapping between meaning representations and linguistic utterances, thereby providing a precise model of the syntax–semantics interface. This setup mimics how a child could acquire a grammar for producing utterances, and it has been explored by our previous work (Gao & Sun, 2025).

Our first contribution is a concrete and comprehensive computational model of comprehension-oriented incremental grammar built upon Synchronous Hyperedge Replacement Grammar (SHRG; Drewes et al., 1997). SHRG is a flexible graph-rewriting formalism that represents linguistic knowledge, including lexical items, phrase-structure rules, and other constructions, as hypergraphs, and captures syntax–semantics correspondences through synchronized rewriting operations. Previous work has demonstrated the effectiveness of SHRG as a core engine for semantic parsing (Chen & Sun, 2020; Chen et al., 2018), meaning-to-text generation (Ye & Sun, 2020), and computational models of language acquisition (Gao & Sun, 2025). We define an incremental grammar tailored to online sentence comprehension. When parsing a word sequence, the grammar enforces a left-branching analysis, approximating how human comprehenders incrementally construct partial interpretations based on the input received so far. To the

In G. Boleda, I. Dautriche, M. Maldonado, C. Saldana and T. Torrent (Eds.), *Proceedings of the 48th Annual Conference of the Cognitive Science Society*

best of our knowledge, the only comparable computational grammar is that of Demberg et al. (2013), based on Tree Adjoining Grammar. However, their framework is restricted to syntactic structure, whereas our approach integrates semantic interpretation directly into the incremental process.

We follow the same practice as outlined by Gao and Sun (2025) and conduct production-oriented grammar induction on the data from Child Language Data Exchange System (CHILDES; Macwhinney, 2000). Given only the meaning graphs as training input, we induce a probabilistic syntax–semantics interface by estimating the rule probabilities with unsupervised Expectation–Maximization (EM; Dempster et al., 1977), treating derivations as latent structure. The key difference to Gao and Sun (2025) is the grammar initialization. We replace the standard hierarchical grammar with an incremental grammar, so learning starts from a derivational hypothesis space that is constrained to left-to-right incremental constructions. The performance of the learned grammar is evaluated using metrics that reflect both semantic constituent recognition and natural language generation quality. We find three non-trivial facts from our experiments:

- Although our unsupervised learning algorithm achieves remarkably high accuracy, there remains a small but consistent gap between its performance and that of a supervised oracle (which has access to ground-truth linguistic structures). By contrast, for a hierarchical grammar, the unsupervised model matches the supervised oracle with no gap.
- The automatically induced incremental grammar yields slightly worse performance on semantic constituent recognition (i.e. mapping meanings to syntactic trees) compared to a hierarchical grammar, but it produces better results in generation tasks (measured by BLEU scores for utterance output). Overall, the incremental grammar achieves comparable accuracy to the hierarchical grammar.
- An online EM training procedure achieves performance equivalent to standard batch EM in our setting. Other EM variants, such as Viterbi EM, yield similar results. This suggests that updating grammar parameters incrementally on a sentence-by-sentence basis does not harm learning outcomes, which is encouraging from a cognitive perspective, as children acquire language in an online manner.

Background

Unsupervised Grammar Induction

Grammar induction has largely aimed to recover linguistic structures, typically constituent trees or dependency trees, from raw text without explicit annotations. A foundational line of work induces probabilistic context-free grammar (PCFG) by maximizing sentence likelihood using Expectation Maximization algorithms (Carroll & Charniak, 1992; Lari & Young, 1990) and Bayesian models (Cohn & Blunsom, 2010; Johnson et al., 2007). Recent efforts attempts to integrate grammar induction algorithms with neural networks (Dyer et al., 2016; Jin et al., 2018; Kim et al., 2019; S. Yang et al., 2021). Gao and Sun (2025), on the other hand, induces a production-oriented graph grammar from semantic graphs only by generalizing the EM algorithm onto graph structures.

Incremental Parsing in NLP

In NLP, syntactic parsing aims to analyze a sequence of tokens—typically words—in order to determine its grammatical structure (Jurafsky & Martin, 2026; Manning, 1999). Parsing approaches are generally divided into two categories: knowledge-intensive methods, which use linguistically informed grammatical rules to constrain possible structures (Chen et al., 2018; Peng et al., 2015), and data-intensive methods, which learn to predict syntactic structures from large annotated corpora using machine-learning techniques (Cao et al., 2021; Dozat & Manning, 2018).

An incremental parser gradually constructs partial grammatical structures of a sentence from left to right (Costa et al., 2003; Jain & Waibel, 1989; Lane & Henderson, 2001). Neural NLP systems have achieved extremely impressive performance by exploiting the full sentential context. Nevertheless, some work continues to model the incrementality of language processing. For example, Shen et al. (2021) jointly predict the next word and its syntactic structure using an RNN with a dynamic oracle, while K. Yang and Deng (2020) maintains a partial syntactic tree and applies a graph convolutional network (GCN; Kipf and Welling 2017) to encode the structure and predict the next parsing action. These studies have incorporated incrementality into parsing algorithms, but they remain limited to hierarchical structure building; sentence processing is modeled primarily as successive merge operations that combine smaller constituents into larger ones. Following Marr's tri-level hypothesis (Marr, 1982), we argue that such approaches address incrementality mainly at the *algorithmic level*.

We formalize the notion of *Incremental Grammar* (Phillips, 2003) within the SHRG framework, with the goal of modeling incrementality at the *computational level*. We adopt a view of grammar as a finite set of rules that represents and generates linguistic expressions, where each rule belongs to a specific type of operation defined over a linguistic representation. Many theories in the Minimalist Program (MP; Chomsky 1995) propose two basic syntactic operations: *Merge*, which combines two syntactic components into a larger one, and *Move*, which repositions syntactic elements freely. In the next section, we present an operational method — distinct from previous approaches — that constructs an Incremental Grammar comprising two types of operations: *chunking* and *merge*.

English Resource Grammar

The goal of this paper is to construct a formal model of the syntax–semantics interface in developmental language. Doing so requires explicit representations of the syntactico-semantic analyses of child speech. To this end, we use the English Resource Grammar (ERG; Copestake and Flickinger 2000; Flickinger 2011), a broad-coverage, linguistically mo-

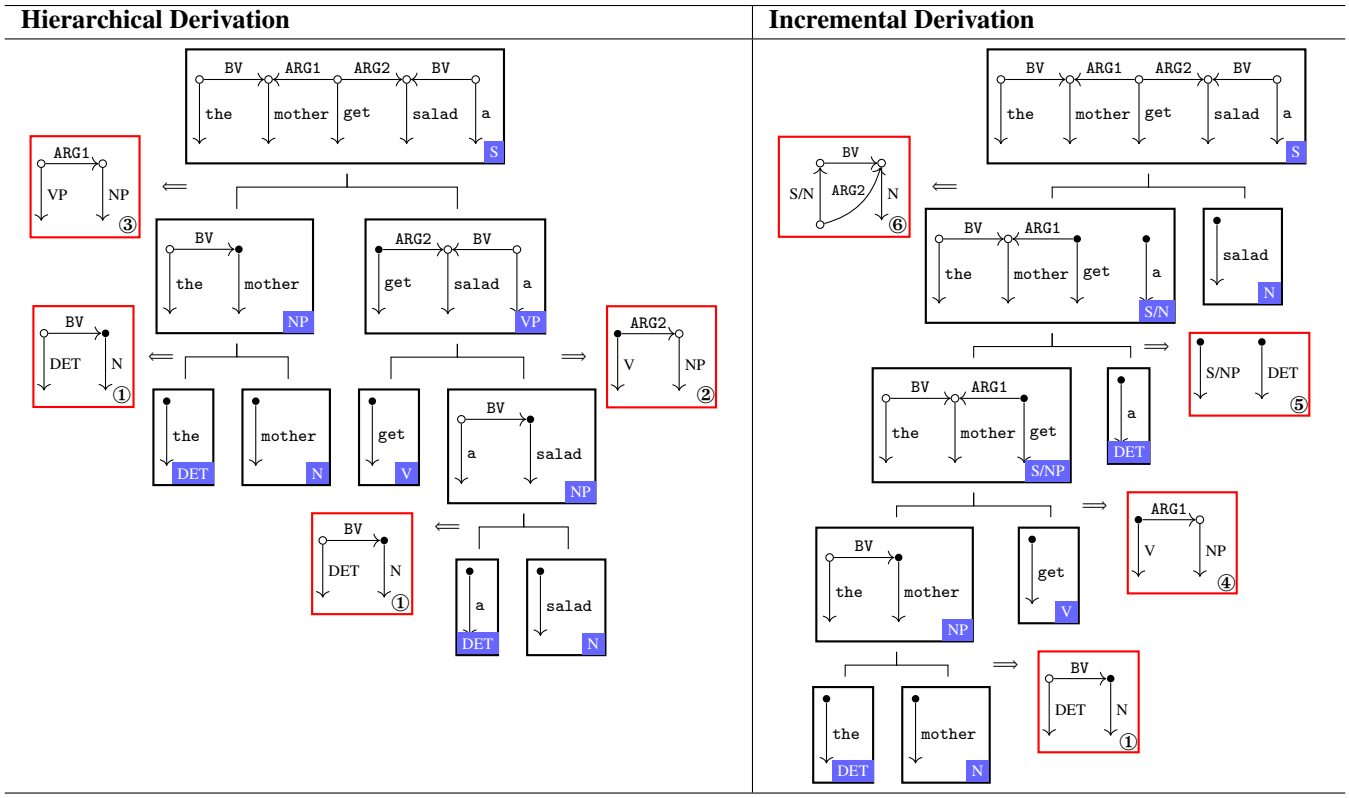

Table 1: This table illustrates the process of obtaining the semantic representation of the sentence "*the mother got a salad*" under the SHRG framework, comparing hierarchical derivation with incremental derivation. To make the illustration more intuitive and easier to understand, we omit the specific ERS tags, and instead represent lexical items using their lemmas (e.g., *got* is represented as *get*). The blue labels in the corners indicate the syntactic category of each node, and the hypergraph within each node represents the semantics of the corresponding sentence fragment. Hypergraphs enclosed in red rectangles represent rules, indicating that the semantic graph in the parent node is constructed by replacing hyperedges with the semantic graphs of its child nodes. Within each hypergraph, black nodes denote external nodes that may connect to other graphs in subsequence process, whereas white nodes denote internal nodes that do not participate in further combination.

tivated precision grammar for English, to obtain automatic analyses of child speech. The semantic representations produced by ERG are known as English Resource Semantics (ERS; Flickinger et al. 2014), whose native formalism is Minimal Recursion Semantics (MRS; Copestake 2002; Copestake et al. 2005). Although MRS is a logic-based representation, it can be systematically converted into graph-based semantic representations, such as Elementary Dependency Structures (EDS; Oepen and Lønning 2006), which reduce the full MRS logical form to a dependency-style semantic graph. In our experiments, we adopt EDS graphs as the meaning representation. Because syntactic derivations in our model are strictly left-branching, we do not make use of ERG derivation trees.

Formalizing an Incremental Grammar

Synchronous Hyperedge Replacement Grammar

Phrase structure describes how words are merged to form a complete sentence; from a top-down perspective, it can also be viewed as a process in which non-terminal symbols are rewritten as sequences of linguistic symbols. HRG (Drewes et al., 1997)) extends this intuition to graphs, where a graph can be seen as the result of a sequence of operations that rewrite labeled hyperedges into edge-labeled hypergraphs. Unlike ordinary edges, which connect exactly two nodes, a hyperedge may connect one or more than two nodes, and the graph that contains hyperedges is referred to as a hypergraph. For example, in the left column of Table 1, the graph rooted at node S is a hypergraph with five nodes and nine labeled hyperedges. This hypergraph is constructed by replacing the hyperedges labeled VP and NP in rule ③ with hypergraphs.

Due to the high structural complexity of graphs, the space of possible HRG rules is extremely large, making it impractical to manually construct a exhaustive rule set. To address this issue, we automatically extract HRG rules from a corpus by establishing an explicit correspondence between HRG rules and context-free grammar (CFG) rules, thereby constructing an SHRG (Chen et al., 2018; Peng et al., 2015)). As shown in Table 1, the determiner *the* and the noun *mother* form a noun phrase via the CFG rule "NP → DET N". By synchronously replacing the hyperedges labeled DET and N in

HRG rule ①, we obtain the semantic graph corresponding to the noun phrase *the mother*. More generally, given the full CFG derivation of a sentence, the SHRG framework allows us to construct its semantic graph by applying the HRG rule corresponding to each CFG rule in the derivation. Conversely, if a semantic graph can be decomposed into a combination of HRG rules, we can then generate a sentence according to the corresponding CFG rules.

Operations in an Incremental Grammar

We follow the algorithm proposed by Chen et al. (2018) to construct an SHRG by using a left-branching binary tree as the derivation of a sentence rather than its phrase structure. The right column of Table 1 illustrates a derivation licensed by a resulting incremental grammar. At the beginning, the lexical semantics of the first word "the" serves as the initial semantic graph. The semantic representations of the subsequent words, "mother" and "got", are then sequentially merged into the current graph, producing the structure shown at node S/NP, which indicates that an additional noun phrase is required to complete the sentence. The appearance of the determiner "a" keeps the current semantic graphs aside and introduces a separate subgraph that requires further completion. Finally, the word "salad" finishes the semantics of the noun phrase "a salad", and also merges the two parts together, yielding the complete semantic graph of the full sentence.

In our Incremental Grammar, each rule indicates the relation between the processing semantic graph and the lexical semantics of the new word, which is always a hyperedge with one node. From the illustrations above, we can see that all rules can be divided into two types of operations on the partial graph of semantics: *chunking* and *merge*.

Chunking A chunking rule, such as Rule ⑤ in Table 1, is represented as a disconnected hypergraph, in which the hyperedge introduced by the new word remains separate from the existing semantic subgraphs. This operation indicates the presence of a linguistic segment whose syntactic and semantic structure is relatively independent of the preceding expressions, suggesting that the current semantic representation is complete to some extent. Christiansen and Chater (2016) argued that language processing is subject to a "Now-or-Never" bottleneck, whereby newly incoming information may quickly obliterate earlier information due to limitations in human memory. To address this bottleneck, Christiansen and Chater (2016) proposed that both language comprehension and production rely on a "Chunk-and-Pass" mechanism, which recodes linguistic input into chunks and then passes them to a higher level of linguistic representation. Our *chunking* operation naturally divides a sentence into such chunks for subsequent semantic composition, closely aligning with the "Chunk-and-Pass" processing.

Merge A rule belonging to the merge operation combines several small semantic segments to form a large semantic graph. Different from the merge operation in the Minimalist Program, our incremental grammar distinguishes between two types of merge operations: *local merge* and *cross-span merge*. Both types of merge operations attach the lexical semantics of the next word to the currently processed semantic subgraph. In addition, the cross-span merge also integrates the just-completed semantic segment with previous segments to form a larger one. Certain merge rules can be applied in either manner, depending on the structural context. For instance, Rule ⑥ in Table 1 not only merges the lexical semantics of the words "a" and "salad" to complete the representation of the noun phrase "a salad", but also combines the subgraphs of "the mother got" and "a salad" together to form the full semantic graph of the sentence.

Experiments of Grammar Induction

Dataset and Statistics of Rules

Following the work of Gao and Sun (2025), we extract rules and do grammar induction on the same dataset, formed using child speech from Child Language Data Exchange System (CHILDES; Macwhinney, 2000). In total, 1,244,745 transcribed child utterances were extracted from 79 English corpora spanning ages from 0 to 10 years, covering the critical language acquisition years. Each utterance was automatically paired with a meaning representation by parsing the utterance with the ERG. Overall, 77.6% of the utterances were successfully parsed, resulting in 966,030 meaning-utterance pairs. For the experiments in this paper, we sample 1000 utterances from each age group (in month) and report the findings.

To obtain rules that are more interpretable for linguistic analysis, we replace the original ERS labels in derivations with common category abbreviations. For example, in Table 1, the original label of the phrase "a salad" is `sp-hd`, indicating that "a" functions as the specifier and "salad" as the head. However, when organizing the entire sentence into a left-branching binary tree from left to right, the node labeled "S/NP" cannot be assigned the label `root_strict/sp-hd`, since `sp-hd` encodes the internal relation between constituents within the noun phrase but does not capture its role in the full derivation.

Table 2 shows the number and proportion of rules for different operations in the corpus at each developmental stage.As shown in the tables, both the variety and proportion of chunking rules increase with age. Similarly, the proportion of rules involving the cross-span merge operation also increases, reflecting the fact that cross-span merge primarily combines non-adjacent chunks. In contrast, although the number of local merge rules increases over time, its proportion among all rules decreases. Taken together, these results suggest that the "Chunk-and-Pass" mechanism is relatively demanding for children, resulting in less frequent use of chunking rules at early developmental stages. With increasing age, children appear to become more proficient in this mechanism, leading to a gradual increase in chunking rule usage. Finally, rules involving both local and cross-span merges are extremely rare,

Corpus	Chunking	Merge		
		Local	Cross-span	Both
Prelinguistic	3 (10.34%)	24 (82.76%)	2 (6.90%)	0 (0.00%)
Holophrasic	49 (17.01%)	208 (72.22%)	31 (10.76%)	0 (0.00%)
Two-word	100 (18.05%)	375 (67.69%)	76 (13.72%)	3 (0.54%)
Telegraphic	355 (19.99%)	1023 (57.60%)	374 (21.06%)	24 (1.35%)
Later Multi-word	956 (22.42%)	2185 (51.23%)	1062 (24.90%)	62 (1.45%)
Mature	1635 (22.73%)	3421 (47.55%)	2038 (28.33%)	100 (1.39%)

Table 2: Number and percentage of rules belong to different operations

highlighting the clear distinction between the two types and warranting further investigation.

Production by Recursively Decomposing Graphs

So far, we have obtained a set of rules for the incremental grammar by applying the SHRG framework to sentences organized into left-branching binary trees in an incremental manner. The next question is how these rules should be used to generate sentences from meaning representations.

Under the framework of SHRG, each CFG rule is bound to an HRG rule. Therefore, after we decompose a sentence into syntactic constituents and combine them into a binary derivation tree, we can obtain the corresponding semantic graph by sequentially applying the associated HRG rules. Conversely, if we can decompose a semantic graph into semantic constituents according to HRG rules, and then apply the corresponding CFG rules to derive the derivation tree, we can finally produce a sentence.

We now assume that the set of all rules is $\Gamma = \{\gamma_1, ..., \gamma_n\}$, where each rule $\gamma : \{g_i, g_j\} \rightarrow g_k$ merges two hypergraphs into a larger one. Given a semantic graph g, let $\text{Sub}(g)$ denote the set of all its subgraphs. Consider a set $A = \{a_1, ..., a_m | a_i \in \text{Sub}(g), 1 \leq i \leq m\}$:

- For any $a_k \in A$, if there exists no $a_{k'} \in A$ with $k' \neq k$ such that $a_{k'}$ is a subgraph of a_k, then a_k is called a *terminal constituent*;
- For any $a_k \in A$,if there exist $a_i, a_j \in A$ with $i \neq j$, both being subgraphs of a_k, and there exists a rule $\gamma_l \in \Gamma$ such that $\gamma_l(a_i, a_j) = a_k$, then a_k is called a *non-terminal constituent*. It is said to be composed of a_i and a_j, written as $a_k = a_i +_{\gamma_l} a_j$, and a_i, a_j are sub-constituents of a_k.

If $g \in A$ and, for all $a_k \in A, a_k \neq g$, a_k is a sub-constituent of exactly one other constituent, then A, together with the corresponding rules $\Gamma_A \subset \Gamma$, is a *derivation* of graph g; we denote this pair as $t = (A, \Gamma_A)$. The goal of language production is then to find such a derivation for a given semantic graph.

Unsupervised Grammar Induction

In language production, one of the most challenging problems is that there may be multiple possible derivations t for a given graph g. To address this issue, Gao and Sun (2025) construct a probabilistic SHRG, in which assigning a weight to each rule turns the symbolic interface into a generative model over derivations (and thus over realizations).

Therefore, for a semantic graph g and derivation t,

$$p(g, t; \phi) = \sum_{\gamma \in P} \phi(\gamma)^{c(\phi; t, g)}, p(g; \phi) = \sum_{t} p(g, t; \phi)$$

where ϕ is a parameter vector assigning a probability to each SHRG rule γ and $c(\phi; t, g)$ is the number of usage of rule γ in derivation t for graph g. So the model represents not only what structures are allowed under the grammar, but which derivations are preferred.

Given only observed meaning representations $\{g_i\}$, we estimate ϕ by Expectation Maximization (EM; Dempster et al., 1977) to maximize the corpus likelihood $\sum_i p(g_i; \phi)$. The E-step computes expected rule counts $\mathbb{E}[c(\cdot)]$ under the posterior $p(t|g; \phi)$; the M-step re-estimates each rule by normalizing its expected count against its left-hand-side label

$$\phi^*(\gamma) = \frac{\mathbb{E}[c(\gamma)]}{\sum_{\gamma' \in P} \mathbb{E}[c(\gamma')]}$$

We report the same two metrics reported in Gao and Sun (2025) that target different performances of an induced probabilistic SHRG. *Parsing F1* measures how well the learned grammar recovers semantic constituents. We use the induced probabilistic grammar to parse semantic graphs, and the induced meaning constituents are collected and scored against the gold constituents using the exact-match F1 score.For a parsed graph g, let the gold derivation be $t = (A, \Gamma_A)$ and the parsed derivation be $t' = (A', \Gamma_{A'})$. Then, the true positive (TP) count is $|A \cap A'|$, the false negative (FN) count is $|A - A \cap A'|$, and the false positive (FP) count is $|A' - A \cap A'|$. *Generation BLEU* measures meaning-to-text generation quality. The learned grammar is used to generate an utterance from each meaning representation and compare to the original child utterance using Bilingual Evaluation Understudy (BLEU; Papineni et al., 2002).

Results and Analysis

Following Gao and Sun (2025), we constructed an oracle that learns weights for each rule in a supervised manner based on the ERG parses, enabling a comparison with the results obtained through the unsupervised EM algorithm. Since we modified the data by using common category abbreviations, which are also from the ERG Parser, instead of the original

Dev. Stage	Incremental		Hierarchical	
	Oracle	EM	Oracle	EM
Prelinguistic	99.89	99.89	99.89	99.89
Holophrasic	98.30	97.84	98.74	98.31
Two-word	97.15	96.25	97.75	97.43
Telegraphic	94.73	92.48	95.79	96.00
Later Multi-word	93.35	91.30	93.65	93.94
Mature	94.16	92.14	94.64	94.84

Table 3: F1 scores of semantic constituents recognition

labels, the experimental results for hierarchical grammar fall short of those reported by Gao and Sun (2025) in both semantic constituent recognition and language production, as shown in Table 3 and Figure 2. This may be because common categories omit some internal syntactic information, which hinders the identification of semantic structures and, in turn, reduces the effectiveness of language generation.

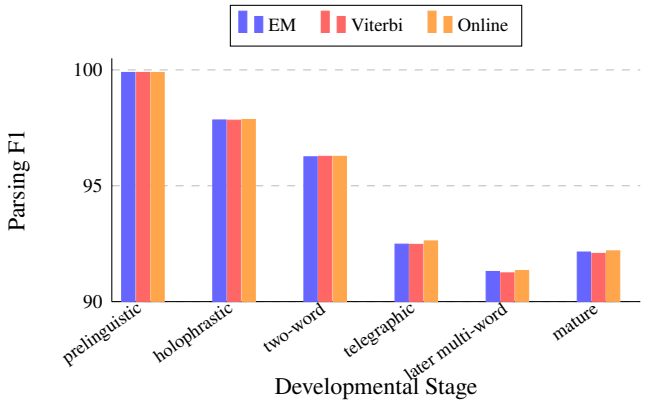

Figure 1: Parsing F1 scores of different EM model variants across developmental stages.

Both incremental grammar and hierarchical grammar perform reasonably well in semantic constituents recognition, with hierarchical grammar exhibiting a slightly better performance. However, a comparison between the results of the oracle and the EM algorithm shows that, when based on hierarchical grammar, the EM algorithm not only achieves a level comparable to the oracle but even outperforms it in certain stages by uncovering better semantic structures. In contrast, when based on incremental grammar, there remains a noticeable gap between the results of the EM algorithm and those of the oracle. This indicates that, compared to hierarchical grammar, incremental grammar is more challenging to acquire.

Figure 1 presents the F1 scores of the EM algorithm and its two variants, Viterbi EM and Online EM, on graph parsing. The standard EM algorithm computes marginal probabilities across all parses, while Viterbi EM updates only the most likely parse for each graph. The Online EM algorithm, proposed by Liang and Klein (2009), updates parameters immediately after processing each graph, with a learning rate that decays over time, largely simulating the process of human language acquisition. Although we used an incremental grammar rather than the hierarchical grammar employed by Gao and Sun (2025), we obtained similar experimental results, as shown in Figure 1, where the performance of each algorithm across the datasets is largely comparable, indicating that the choice of algorithm do not affect the outcomes.

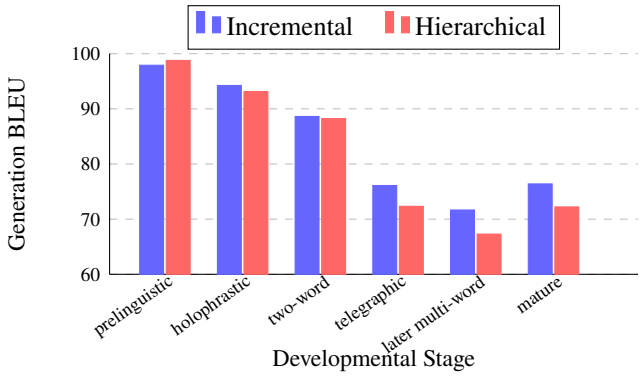

Figure 2: Generation BLEU scores (EM) across developmental stages, comparing *Incremental* vs. *Hierarchical*.

According to Figure 2, hierarchical grammar consistently outperforms incremental grammar in identifying semantic constituents, whether in the Oracle or EM algorithm results. However, for generation tasks, incremental grammar produces better outcomes than hierarchical grammar. As noted earlier, our production process organizes the CFG rules corresponding to the decomposed semantic constituents into a tree structure, which is then used to derive sentences. Despite the lower accuracy of constituents obtained via decomposition, incremental grammar still achieves superior generation results, providing strong evidence of its greater effectiveness in language production. We hypothesize that this is because algorithms based on hierarchical grammar, while more accurate in identifying semantic constituents, face greater difficulty in combining them correctly. In contrast, due to their left-to-right processing nature, incremental grammar finds it easier to combine constituents into a derivation in the correct order, thereby generating the correct sentences.

Conclusion

This paper presented a formal, comprehension-oriented incremental grammar as a model of how children may acquire real-time language production. Empirically, we showed that a grammar induced under incremental constraints achieves performance comparable to that of traditional hierarchical models in production-oriented grammar induction. More broadly, our results demonstrate how alternative grammatical assumptions can be systematically compared in silico within a unified learning and evaluation framework.

Acknowledgments

This research was supported by the Major Project of the Ministry of Education, China, titled ”Research on a Compre-

hensive Linguistic Knowledge Base for Machine Language Competence Evaluation" (Project No. 22JJD740004) and the Open Project Fund of the State Key Laboratory of Multimedia Information Processing (Project No. SKLMIP-KF-2025-01). Yuan Gao's work was supported by Cambridge University Press & Assessment.

References

Altmann, G. T., & Kamide, Y. (1999). Incremental interpretation at verbs: Restricting the domain of subsequent reference. *Cognition*, *73*(3), 247–264. https://doi.org/10.1016/s0010-0277(99)00059-1

Benders, T., & Blom, E. (2023). Computational modelling of language acquisition: An introduction. *Journal of Child Language*, *50*(6), 1287–1293.

Cao, J., Lin, Z., Sun, W., & Wan, X. (2021). Comparing knowledge-intensive and data-intensive models for English resource semantic parsing. *Computational Linguistics*, *47*(1), 43–68. https://doi.org/10.1162/coli_a_00395

Carroll, G., & Charniak, E. (1992). Two experiments on learning probabilistic dependency grammars from corpora. *Citeseer*.

Chen, Y., & Sun, W. (2020, July). Parsing into variable-in-situ logico-semantic graphs. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 6772–6782). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.605

Chen, Y., Sun, W., & Wan, X. (2018, July). Accurate SHRG-based semantic parsing. In I. Gurevych & Y. Miyao (Eds.), *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 408–418). Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1038

Chomsky, N. (1995). *The minimalist program*. MIT Press.

Christiansen, M. H., & Chater, N. (2016). The now-or-never bottleneck: A fundamental constraint on language. *Behavioral and brain sciences*, *39*, e62.

Cohn, T., & Blunsom, P. (2010, July). Blocked Inference in Bayesian Tree Substitution Grammars. In J. Hajič, S. Carberry, S. Clark, & J. Nivre (Eds.), *Proceedings of the ACL 2010 Conference Short Papers* (pp. 225–230). Association for Computational Linguistics.

Copestake, A. (2002). *Implementing typed feature structure grammars* (Vol. 110). CSLI publications Stanford.

Copestake, A., & Flickinger, D. (2000, May). An open source grammar development environment and broad-coverage English grammar using HPSG. In M. Gavrilidou, G. Carayannis, S. Markantonatou, S. Piperidis, & G. Stainhauer (Eds.), *Proceedings of the second international conference on language resources and evaluation (LREC'00)*. European Language Resources Association (ELRA). https://aclanthology.org/L00-1276/

Copestake, A., Flickinger, D., Pollard, C., & Sag, I. A. (2005). Minimal recursion semantics: An introduction. *Research on language and computation*, *3*(2), 281–332. https://doi.org/10.1007/s11168-006-6327-9

Costa, F., Frasconi, P., Lombardo, V., & Soda, G. (2003). Towards incremental parsing of natural language using recursive neural networks. *Applied Intelligence*, *19*(1), 9–25. https://doi.org/10.1023/A:1023860521975

Demberg, V., Keller, F., & Koller, A. (2013). Incremental, predictive parsing with psycholinguistically motivated Tree-Adjoining Grammar. *Computational Linguistics*, *39*(4), 1025–1066. https://doi.org/10.1162/COLI_a_00160

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(1), 1–22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

Dozat, T., & Manning, C. D. (2018, July). Simpler but more accurate semantic dependency parsing. In I. Gurevych & Y. Miyao (Eds.), *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 484–490). Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-2077

Drewes, F., Kreowski, H.-J., & Habel, A. (1997). Hyperedge replacement graph grammars. In *Handbook of graph grammars and computing by graph transformation: Volume i. foundations* (pp. 95–162). World Scientific Publishing Co., Inc.

Dyer, C., Kuncoro, A., Ballesteros, M., & Smith, N. A. (2016, June). Recurrent Neural Network Grammars. In K. Knight, A. Nenkova, & O. Rambow (Eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 199–209). Association for Computational Linguistics. https://doi.org/10.18653/v1/N16-1024

Flickinger, D. (2011). Accuracy vs. robustness in grammar engineering. *Language from a cognitive perspective: Grammar, usage, and processing*, *201*, 31–50.

Flickinger, D., Bender, E. M., & Oepen, S. (2014, May). Towards an encyclopedia of compositional semantics: Documenting the interface of the English Resource Grammar. In N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the ninth international conference on language resources and evaluation (LREC'14)* (pp. 875–881). European Language Resources Association (ELRA). https://aclanthology.org/L14-1457/

Gao, Y., & Sun, W. (2025, November). A computational simulation of language production in first language acquisition. In C. Christodoulopoulos, T. Chakraborty, C. Rose, & V. Peng (Eds.), *Proceedings of the 2025 conference on empirical methods in natural language processing* (pp. 10992–11006). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.emnlp-main.557

Goldwater, S. (2011, July). Unsupervised NLP and human language acquisition: Making connections to make progress. In O. Abend, A. Korhonen, A. Rappoport, & R. Reichart (Eds.), *Proceedings of the first workshop on unsupervised learning in NLP* (p. 1). Association for Computational Linguistics. https://aclanthology.org/W11-2201/

Hawkins, J. (2004). *Efficiency and complexity in grammars*. OUP Oxford. https://books.google.co.uk/books?id=_FITDAAAQBAJ

Jain, A., & Waibel, A. (1989). Incremental parsing by modular recurrent connectionist networks. *Advances in neural information processing systems*, *2*.

Jin, L., Doshi-Velez, F., Miller, T., Schuler, W., & Schwartz, L. (2018). Unsupervised Grammar Induction with Depth-bounded PCFG (L. Lee, M. Johnson, K. Toutanova, & B. Roark, Eds.). *Transactions of the Association for Computational Linguistics*, *6*, 211–224. https://doi.org/10.1162/tacl_a_00016

Johnson, M. (2013). Language acquisition as statistical inference. In S. R. Anderson, J. Moeschler, & F. Reboul (Eds.), *The language-cognition interface* (pp. 109–134). Libraire Droz.

Johnson, M., Griffiths, T., & Goldwater, S. (2007). Bayesian Inference for PCFGs via Markov Chain Monte Carlo. *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, 139–146.

Jurafsky, D., & Martin, J. H. (2026). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, with language models* (3rd) [Online manuscript released January 6, 2026]. https://web.stanford.edu/~jurafsky/slp3/

Kim, Y., Dyer, C., & Rush, A. (2019, July). Compound probabilistic context-free grammars for grammar induction. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th annual meeting of the association for computational linguistics*.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. https://openreview.net/forum?id=SJU4ayYgl

Kutas, M., & Hillyard, S. A. (1980). Reading senseless sentences: Brain potentials reflect semantic incongruity. *Science*, *207*(4427), 203–205. https://doi.org/10.1126/science.7350657

Lane, P. C., & Henderson, J. B. (2001). Incremental syntactic parsing of natural language corpora with simple synchrony networks. *IEEE Transactions on Knowledge and Data Engineering*, *13*(2), 219–231. https://doi.org/10.1109/69.917562

Lari, K., & Young, S. J. (1990). The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech & Language*, *4*(1), 35–56. https://doi.org/10.1016/0885-2308(90)90022-X

Liang, P., Jordan, M. I., & Klein, D. (2013). Probabilistic grammars and hierarchical dirichlet processes. In A. O'Hagan & M. West (Eds.), *The oxford handbook of applied bayesian analysis*. Oxford University Press.

Liang, P., & Klein, D. (2009). Online em for unsupervised models. *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, 611–619.

MacWhinney, B. (2010). Computational models of child language learning: An introduction. *Journal of Child Language*, *37*(3), 477–485.

Macwhinney, B. (2000). The CHILDES project: Tools for analyzing talk. *Child Language Teaching and Therapy*, *8*. https://doi.org/10.1177/026565909200800211

Manning, C. (1999). *Foundations of statistical natural language processing*. The MIT Press.

Marr, D. (1982). Vision: A computational investigation into the human representation and processing of visual information.

Oepen, S., & Lønning, J. T. (2006, May). Discriminant-based MRS banking. In N. Calzolari, K. Choukri, A. Gangemi, B. Maegaard, J. Mariani, J. Odijk, & D. Tapias (Eds.), *Proceedings of the fifth international conference on language resources and evaluation (LREC'06)*. European Language Resources Association (ELRA). https://aclanthology.org/L06-1214/

Pallier, C., Devauchelle, A.-D., & Dehaene, S. (2011). Cortical representation of the constituent structure of sentences. *Proceedings of the National Academy of Sciences*, *108*(6), 2522–2527. https://doi.org/10.1073/pnas.1018711108

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002, July). Bleu: A Method for Automatic Evaluation of Machine Translation. In P. Isabelle, E. Charniak, & D. Lin (Eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 311–318). Association for Computational Linguistics. https://doi.org/10.3115/1073083.1073135

Pearl, L., & Goldwater, S. (2016). Statistical learning, inductive bias, and bayesian inference in language acquisition. In J. Lidz, W. Williams, & J. Fuller (Eds.), *The oxford handbook of developmental linguistics* (pp. 664–695). Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199601264.013.28

Peng, X., Song, L., & Gildea, D. (2015). A synchronous hyperedge replacement grammar based approach for AMR parsing. *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, 32–41. https://doi.org/10.18653/v1/K15-1004

Phillips, C. (2003). Linear order and constituency. *Linguistic inquiry*, *34*(1), 37–90. https://doi.org/10.1162/002438903763255922

Sag, I. A., & Wasow, T. (2011). Performance-compatible competence grammar. In R. D. Borsley & K. Börjars (Eds.),

Non-transformational syntax: Formal and explicit models. Wiley-Blackwell.

Shen, Y., Tan, S., Sordoni, A., Reddy, S., & Courville, A. (2021, June). Explicitly modeling syntax in language models with incremental parsing and a dynamic oracle. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, & Y. Zhou (Eds.), *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 1660–1672). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.132

Yang, K., & Deng, J. (2020). Strongly incremental constituency parsing with graph neural networks. *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

Yang, S., Zhao, Y., & Tu, K. (2021, June). PCFGs can do better: Inducing probabilistic context-free grammars with many symbols. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, & Y. Zhou (Eds.), *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 1487–1498). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.117

Ye, Y., & Sun, W. (2020). Exact yet Efficient Graph Parsing, Bi-directional Locality and the Constructivist Hypothesis. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4100–4110. https://doi.org/10.18653/v1/2020.acl-main.377