Modeling Syntactic Knowledge With Neuro-Symbolic Computation

Hilton Alers-Valentín[1], Sandiway Fong[2] and J. Fernando Vega-Riveros[3]
[1]*Linguistics and Cognitive Science, University of Puerto Rico-Mayagüez, Puerto Rico*
[2]*Department of Linguistics, University of Arizona-Tucson, U.S.A.*
[3]*Electrical and Computer Engineering, University of Puerto Rico-Mayagüez, Puerto Rico*
{*hilton.alers, jfernando.vega*}*@upr.edu, sandiway@arizona.edu*

Keywords: Minimalist Syntax, Parser, Lexicon, Structural Ambiguity, Cognitive Modeling, Computational Linguistics, Natural Language Processing, Symbolic computation, Neural Networks, Explainable Artificial Intelligence.

Abstract: To overcome the limitations of prevailing NLP methods, a Hybrid-Architecture Symbolic Parser and Neural Lexicon system is proposed to detect structural ambiguity by producing as many syntactic representations as there are interpretations for an utterance. HASPNeL comprises a symbolic AI, feature-unification parser, a lexicon generated using manual classification and machine learning, and a neural network encoder which tags each lexical item in a synthetic corpus and estimates likelihoods for each utterance's interpretation with respect to the corpus. Language variation is accounted for by lexical adjustments in feature specifications and minimal parameter settings. Contrary to pure probabilistic system, HASPNeL's neuro-symbolic architecture will perform grammaticality judgements of utterances that do not correspond to rankings of probabilistic systems; have a greater degree of system stability as it is not susceptible to perturbations in the training data; detect lexical and structural ambiguity by producing all possible grammatical representations regardless of their presence in the training data; eliminate the effects of diminishing returns, as it does not require massive amounts of annotated data, unavailable for underrepresented languages; avoid overparameterization and potential overfitting; test current syntactic theory by implementing a Minimalist grammar formalism; and model human language competence by satisfying conditions of learnability, evolvability, and universality.

1 INTRODUCTION

The human language faculty allows speakers to associate thoughts and concepts into mental linguistic representations, which are subsequently externalized as speech, text or signs. These mental representations are hierarchical in nature, but because of constraints of nature, the externalization is linear. Therefore, speech and text consist only of strings of words as leaves or terminal nodes of the whole syntactic structure, and so information about constituents, classes and categories is literally lost in externalization. An important consequence of this fundamental property of language is structural ambiguity, or the fact that a single utterance or string of words can be interpreted in more than one way by our mental grammars. For example, the utterance 'They can fish' can be interpreted in two different ways: as meaning that they are able to fish or that they put fish in cans. This sentence is ambiguous because our internal language system can assign two different structure representations to the same string. Ambiguity may be problematic for efficient communication as it leads to misunderstandings, yet it is pervasive in language use.

To address the linguistic problem of ambiguity, we propose a Hybrid-Architecture Symbolic Parser with Neural Lexicon (HASPNeL) system that combines the effectiveness of probabilistic systems with the accuracy of syntactic representation of symbolic parsing. By encoding the syntactic rules from natural language to create a generalizable tagging system, this interdisciplinary approach represents a paradigmatic departure from traditional attempts to identify ambiguity in natural language, such as statistical methods based on machine learning and applications following machine-learning-guided rule-based derivations (Petkevič, 2014). HASPNeL would be able to not only parse grammatically acceptable novel strings and represent structural and lexical ambiguity, but would also be able to identify those strings that are not grammatically acceptable, effectively approximating the performative effectiveness of the grammaticality judgments of native speakers of a given language—flexible enough to accept novel input, yet

Alers-Valentín, H., Fong, S. and Vega-Riveros, J.
Modeling Syntactic Knowledge With Neuro-Symbolic Computation.
In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023) - Volume 3*, pages 608-616
ISBN: 978-989-758-623-1; ISSN: 2184-433X

strict enough to be able to identify the acceptability of such input, even when this input is novel.

2 ARCHITECTURE

HASPNeL uses a hybrid architecture consisting of three major components: (1) a *probabilistic encoder*, (2) a *symbolic decoder*, and (3) a *lexicon*.

2.1 Probabilistic Encoder

This component is implemented using an RNN, or self-attentive neural network, as two strong alternatives for this stage. Neural parsers can be visualized as composed of two stages: encoder and decoder. The encoder takes an input string and assigns syntactic categories to the words in that string. The decoder takes the tagged lexical items and builds the constituent structures of the sentence. This architecture is denoted encoder-decoder (Aggarwal, 2018). The output vectors should provide information about the potential categories of the lexical items together with their probabilities inferred from the learning algorithm. The syntactic categories produced by the encoder component can be a vector or set of vectors. These vectors correspond to the syntactic categories of the lexical items in the input string. Lexical items can have more than one syntactic category, though (e.g., 'fish' can be a noun or a verb). The assignment of category depends on the context where the word is used. The decoder uses these vectors to incrementally build up a labeled parse tree (Kitaev and Klein, 2018).

Encoders have been built using fixed-window-size feed-forward NNs (Durrett and Klein, 2015), but have been displaced by Recurrent Neural Networks (RNN) in part due to RNNs' ability to capture global context in sentences with variable lengths. Nevertheless, RNNs have one major limitation: the long-term memory problem (Aggarwal, 2018), i.e., RNNs are not able memorize data for long time and begin to forget their previous inputs as the learning time passes. Two implementations that compensate for the long-term memory problem of RNNs are the Long-Short-Term Memory networks (LSTM) (Hochreiter and Schmidhuber, 1997), and the Gated Recurrent Unit (GRU) (Cho et al., 2014). (Kitaev and Klein, 2018) propose the use of a self-attention encoder which makes explicit the manner in which information is transferred between different locations in the sentence. They use this approach to study the relative importance of different kinds of context to the parsing task. The locations in the sentence attend to each other based on their positions, but also based on their contents.

2.2 Symbolic Decoder

The decoder will produce the different structural analysis based on the syntactic categories produced by the neural encoder. Symbolic systems are characterized by (i) the use of a set of symbols as knowledge representations, (ii) a specific formal code (metalanguage) to formulate the symbol-handling system, and (iii) autonomy between the syntactic component (which sets the conditions for structural well-formedness) and the semantic component (which computes meaning from well-formed expressions).

The symbol-handling component of our proposed system encodes the formalisms of Minimalist Grammars (MGs) ((Stabler, 1997), (Stabler, 2011), (Collins and Stabler, 2016)) as a formalization of Minimalist syntax ((Chomsky, 1995), (Chomsky, 2001); (Chomsky, 2008)). The mathematical rigor makes it possible to address questions about the generative power and explanatory adequacy of this formalism for natural language (Graf, 2021). Moreover, by putting Minimalism on a mathematical foundation, it can be linked to existing work on parsing and learnability. This approach not only strengthens the connection between theoretical syntax and psycholinguistics, but it also opens up the gate to large-scale applications in modern language technology. As Graf points out, if Minimalist ideas can be shown to be useful for practical applications, that is mutually beneficial for all involved fields (Graf, 2021).

Linguistic theories are generative models of the human language faculty. Broadly speaking, two factors of generative models pertain to the construction of parsing models: 1) Binary Merge, the primitive operation at the heart of modern theories, is a bottom-up operation that constructs larger phrases from smaller one. However, parsing models generally operate from left to right, this is termed online, and results in structure being filled in incrementally as parsing proceeds. Therefore, it is a research challenge to re-interpret Merge as predictive parsing. 2) Merge is word-order free, in other words, core operations of grammar construct dependencies, e.g. agreement, binding, control or movement chain, between phrases based on hierarchical structure only. Syntactic objects built by Merge must be linearized during Externalization. It is a challenge to reconstruct or reverse this process during parsing.

Recent work in the Minimalist Program has highlighted the role of locally deterministic computations in the construction of syntactic representation as part of a shift in the structure of linguistic theories of narrow syntax from abstract systems of declarative rules and principles, (Chomsky, 1981), to systems

where design specifications call for efficient computation within the human language faculty (Fong, 2005). Case agreement is reanalyzed in terms of a system of probes, e.g., functional heads that target and agree with goals, e.g., referential and expletive nominals, within their c-command domain. In this system, probe-goal agreement can be long-distance and need not trigger movement.

The proposed system represents a development of Fong's implementation of the probe-goal account. He also sustains that "efficient assembly, i.e., locally deterministic computation, from a generative perspective with respect to (bottom-up) MERGE does not guarantee that parsing with probes and goals will also be similarly efficient. By locally deterministic computation, we mean that the choice of operation to apply to properly continue the derivation is clear and apparent at each step of the computation" (Fong, 2005). Therefore, following Fong's system, instead of MERGE and MOVE as the primitive combinatory operations for the assembly of phrase structure, the proposed system will also be driven by elementary tree composition with respect to a range of heads in the extended verb projection (v*, V, c, and T). Elementary tree composition is an operation that is a basic component of Tree-Adjoining Grammars (TAG) (Joshi and Shabes, 1997). The system will also be on-line in the sense that once an input element has fulfilled its function, it is discarded, i.e., no longer referenced. To minimize search, there is neither lookahead nor lookback in the sense of being able to examine or search the derivational history, but Fong's two novel devices with well-defined properties: a Move Box that encodes the residual properties of CHAINs and theta theory, and a single or current Probe Box to encode structural Case assignment and to approximate the notion of (strong) Phase boundaries. In particular, the restriction to a single Probe Box means that probes cannot "see" past another probe; thereby emulating the Phase Impenetrability Condition (PIC). Limiting the Move Box to operate as a stack will allow nesting but not overlapping movement. A consequence of this is that extraction through the edge of a strong Phase is no longer possible. Examples of parses will be used to illustrate the empirical properties of these computational elements. The system is also incremental in the sense that a partial parse is available at all stages of processing (Fong, 2005).

In more recent work, e.g. (Fong and Ginsburg, 2019), many dependency relations and phenomena across different languages (English, Arabic, Japanese and Persian) have been directly implemented in the generative framework using a Minimalist Machine. Our plan is to adapt this machinery for parsing.

2.3 Lexicon

The lexicon is the module that contains the grammatical information about all lexical items in the sentences to be analyzed by the parser. Following the Chomsky-Borer hypothesis, MGs situate all language-specific variation in the lexicon. Hence every MG is just a finite set of lexical items. Each lexical item takes the form A :: α, where A is the item's phonetic exponent and α its string of features (Graf, 2021).

As "the heart of the implemented system", the lexicon will contain every fully inflected word-form appearing in a corpus of 2000 manually-tagged sentences that were constructed for validation purposes. Lexical items are entered as a string of literals, and features are indicated by means of different data types. All lexical items are labeled with a syntactic category; additionally, each category requires a specific subset of valued features and lexical properties, which at least contains the syntactic category, subcategorization frames and relevant grammatical features (such as case, c-selectional and phi-features) for each lexical item. Since it is necessary to determine if a certain combination of words is licensed or grammatical in the language, the lexicon should include every possible entry for each ambiguous lexical item. (Alers-Valentín et al., 2019). The property of selection and uninterpretable feature matching will drive the parsing process. In the course of computation, unintepretable features belonging to analyzed constituents will be eliminated through probe-goal agreement. A (valid) parse is a phrase structure that obeys the selectional properties of the individual lexical items, covers the entire input, and has all uninterpretable features properly valued (Fong, 2005).

3 ENCODING AND ESTIMATING AMBIGUITY

Lexically ambiguous items will have as many lexical entries as meanings and/or feature bundles are identified and tagged for that item in the corpus. In those cases, the RNN encoder will produce as many outputs as there are entries for said item. For example, let us say that for the word "can" there should be (at least) three outputs: (MD 0.7 can), (VB 0.1 can), and (NN 0.2 can), where the number $n, 0 \leq n \leq 1$, corresponds to the likelihood of each category. The likelihood of a category is calculated within the corpus. The sum of the likelihood of each category should be exactly 1. On the other hand, if the item were "cans", the output would have at least options like (cans 0.2 VBZ) and (cans 0.8 NNS). In the case of the lexi-

cal verb ”cans”, it has 3rd person singular phi- (gender/person/number) features and non-preterit tense feature. So, an utterance like ”I can fish” is ambiguous, yet ”He cans fish” is not. For this last utterance, the symbolic component should be able to discard the NNS category, in spite of having a higher likelihood.

4 PROBLEMS WITH CURRENT PARSING TRENDS

Probabilistic parsers produce the most likely parse for a given word string, regardless of the acceptability or ambiguity of the string. They use a statistical model of the syntactic structure of a language, e. g., probabilistic context free grammar (PCFG). Although probabilistic parsers are widely used in NLP applications, they always output a structure even if the word sequence is ungrammatical or unacceptable for native speakers. They also require a manually annotated corpus and a statistical learning algorithm. Although these parsers are particularly good in identifying syntactic categories and have a desirable cost-benefit relation between accuracy and speed, they have been found rather ineffective in the representation of sentences containing long-distance relations among constituents (Alers-Valentín et al., 2019).

(Bernardy and Chatzikyriakidis, 2019) point out that symbolic systems can be very precise, yet they break easily in the presence of new data. Symbolic systems in NLP tasks have been criticized on the fact of their “brittleness”, i.e., that these systems tend to easily break down once they are moved to open domains. Neural Network (NN) models are currently the most used in all sorts of NLP applications since these systems, at first sight, do not seem to suffer from the brittleness problem that characterize symbolic approaches. In spite of their apparent success, (Bernardy and Chatzikyriakidis, 2019) recognize that recent studies show that NLP applications of NN, such as state-of-the-art natural language inference (NLI) systems, are rather brittle in the sense that they “fail to generalize outside individual datasets and are, furthermore, unable to capture certain NLI patterns at all” and therefore argue that, with respect to symbolic approaches, “the NLP community has been probably too hasty in dismissing them.”

There is recent literature regarding hybrid parsing systems ((Gaddy et al., 2018); (Stanojević and Stabler, 2018); (Torr et al., 2019)), like the A* neural parser developed by a research team in the University of Edinburgh. This particular system is an implementation of a minimalist grammar that uses the A* search algorithm. This system produces accurate syntactic representation in many cases, including complex structures as in across-the-board movement; however, the results are not always consistent, and the system does not account for any kind of ambiguity.

4.1 Grammaticality Judgements

In our hybrid approach, a neural lexicon handles the multiple syntactic categories of words and lexical items, while the rule-based component attempts to match those syntactic categories with well-formed phrases according to a set of grammar rules. If the rule-based component cannot fit the syntactic categories into a well-formed structure, the string is deemed non grammatical. Machine learning-based parsers are trained with sentences from a corpus. This approach infers rules from a limited set of examples, however large the set may be. To logically infer a rule describing every member of a set, the system must have information about every member of that set. According to the No Free Lunch Theorem for machine learning, every classification algorithm, when averaged over all possible data-generating distributions, has the same error rate when classifying previously unobserved points (Wolpert and Macready, 1997). “In some sense, no machine learning algorithm is universally any better than any other” (Goodfellow et al., 2016). Moreover, most ML-based parsers are trained with grammatical sentences. Even if a ML-based parser were trained including non-grammatical sentences, it is biased by the proportion of non-grammatical utterances in the corpus. During testing, both false positive and false negative non-grammatical results get buried together with other types of parsing errors.

Probabilistic systems do not appear to show any correlation between grammaticality and the rankings in a list of examples (Fong, 2022). In experiments performed by (Pereira, 2002) and (Fong, 2022), grammatical examples are not ranked highly enough to make an appearance within the 10-best list. In fact, the grammatical example ”colorless green ideas sleep furiously” only ranks 23rd out of the list of the 120 possible permutations of those five words. The results show a clear lack of discrimination between the grammatical and the ungrammatical, and Chomsky’s observation still holds: “there is no significant correlation between order of approximation and grammaticalness. If we order the strings of a given length in terms of order of approximation to English, we shall find both grammatical and ungrammatical strings scattered throughout the list, from top to bottom. Hence the notion of statistical approximation appears to be irrelevant to grammar” (Chomsky, 1956).

4.2 System Stability

The experiment by (Fong, 2022) also shows that statistical systems are not as stable as could be presumed. For example, the grammatical sentence colorless green ideas sleep furiously ranks higher than the ungrammatical *furiously sleep ideas green colorless when 34,000-40,000 treebank sentences are used in training. However, when trained with about 15,000-32,000 sentences, not only does the ungrammatical sentence rank higher, but it achieves a top-10 score for this interval, a score not achieved by the grammatical sentence at any stage of the experiment. This calls into question the stability of the statistical system (Fong, 2022). Further experimentation confirms the observed instability.

The probabilistic context free grammar (CFG) system is also surprisingly sensitive to perturbation in the training data. Another experiment by (Fong and Berwick, 2008) confirms this problem, despite the many thousands of treebank sentences available for training. Prepositional phrase (PP) attachment ambiguity is an important task for any syntactic parser, with either high attachment to the VP or low attachment to the NP, as in [Herman [VP [VP mixed [NP the milk]] [PP with the water]]] (PP high-attachment) versus [Herman [VP drink [NP the milk [PP with the water]]]] (PP low-attachment).

For this sentence, the system produced a low attachment representation. A single training example was enough to account for the low attachment. To confirm this, the relevant PP was deleted from the training example and the parser was retrained, resulting in an output with high attachment in both sentences (Fong and Berwick, 2008). The reason for this extreme sensitivity to perturbation in the training data is that there are millions of parameters that need to be estimated, and this particular parser makes use of nearly every statistical event (recorded during training), even if those events occur only once (Fong, 2022). Since symbolic systems do not make use of any statistical event, they cannot experience any degree of perturbation.

4.3 Ambiguity Detection

In contrast to probabilistic parsers, symbolic parsing systems perform very well in handling syntactic ambiguity, as they do not depend on training data (Alers-Valentín et al., 2019). It is enough to specify in the lexicon the categorial selection of lexical items like drink (one NP internal argument) and mix (one NP and one PP internal arguments). In this case, the symbolic system will always produce a PP high-attachment in clauses whose predicate requires a PP internal argument (e.g. with mix as main verb), but both PP high- and low-attachment in clauses whose predicate does not require it (as with the verb drink).

4.4 Data Requirements

To produce structural representations, symbolic systems require a (manually) annotated lexicon containing an array of lexical items with the grammatical features and properties used by the parser. The size of the lexicon is determined by the number of lexical entries required to characterize the target language, which is finite by nature, probably to a maximum in the order of 10^5 entries. On the other hand, current probabilistic parsing systems require massive amounts of good quality data. Since machine-generated data is low quality, it leads to poor performance, while good quality data, which is manually annotated, makes it extremely expensive. In machine learning approaches, a fraction of the instances is used to build and tune the training model. The remaining instances, referred to as the held-out instances, are used for testing. The accuracy of predicting the labels of the held-out instances is then reported as the accuracy of the model. The fraction used to build the model is further divided in two sets: training and validation. Strictly speaking, the validation data is also a part of the training data, because it influences the final model. For very large labeled data sets, only a modest number of examples to estimate accuracy is needed. There are two options for training the model. One is to hold-out the validation set. The other is to use cross-validation, which can closely estimate the true accuracy under certain circumstances. However, cross-validation can result computationally expensive (Aggarwal, 2018).

Besides, huge amounts of data for general-purpose NLP tasks, albeit low quality, is available for only a relatively small number of languages. For example, GPT-2 was trained on the WebText corpus, containing about 40 GB of text data. In the case of English, 40 GB is not particularly burdensome, but in the case of under-represented languages, large amounts of training data may never become available (Fong, 2022). Diminishing returns are another (expected) negative factor: “to halve the error rate, you can expect to need more than 500 times the computational resources” (Thomson et al., 2021). The enormous resources required, both in terms of energy and exposure to large amount of data, means that these probabilistic systems, independent of their potential achievements or promise of their biologically-inspired architecture, cannot possibly meet the austere learning conditions met by nature (Fong, 2022).

4.5 Model Size and Parameters

There exists a serious number-of-degrees-of-freedom problem (Fong, 2022) with (probabilistic) Context-free grammars (CFGs) -the most commonly implemented parsing formalism- as they are too unconstrained; in principle, all combinations of phrases, both exo- and endo-centric, are possible in this framework. From the point of view of empirical coverage, CFGs are too broad. At the same time, CFGs are a poor choice of formalism for encoding many types of structurally-determined relations, e.g., displacement, control, long-distance agreement or pronominal binding. Bikel observes that "...it may come as a surprise that the [parser] needs to access more than 219 million probabilities during the course of parsing the 1,917 sentences of Section 00 [of the Penn Treebank: SF]" (Bikel, 2004).

Recently, general-purpose deep neural networks have been adopted that contain vastly more parameters than the statistical CFG models. For example, the well-known GPT-2 neural net model has 1.5B parameters, and the next-generation GPT-3 model has 175B parameters (Brown et al., 2020). However, it is not clear whether these systems do anything more with the upscaled parameter size other than simply memorize more. A substantial downside of these scaled-up systems is in terms of the computational resources required to perform the training (Fong, 2022). Large model sizes (more than 100 million parameters) make it computationally expensive to train separate models for each language (Kitaev et al., 2018). In fact, GPT-3 is reputed to have cost around 4.6 million dollars to train. This has resulted in a curious admission in the case of GPT-3 (by the authors): "unfortunately, a bug resulted in only partial removal of all detected overlaps from the training data. Due to the cost of training, it wasn't feasible to retrain the model" (Brown et al., 2020).

4.6 Explanatory Adequacy

Different from current probabilistic models, HASPNeL aims to model a generative grammar, i.e., a theory that seeks to explain the properties of the I-language and the system of externalization possessed by the language user. At a deeper level, the theory of the shared language faculty, Universal Grammar (UG) in modern terms, is concerned with the innate factors that make language acquisition possible — factors that distinguish humans from all other organisms. One achieves a genuine explanation of some linguistic phenomenon only if it keeps to mechanisms that satisfy the joint conditions of learnability, evolvability, and universality, which appear to be at odds (Chomsky, 2021).

Models based on information-theoretic and machine-learning ideas have been successful in a variety of language processing tasks in which what is sought is a decision among a finite set of alternatives, or a ranking of alternatives (Pereira, 2002). In each case, the task can be formalized as learning a mapping from spoken or written material to a choice or ranking among alternatives. However, a potential weakness of such task-directed learning procedures is that they ignore regularities that are not relevant to the task, even though those regularities may be highly informative about other questions. This is in sharp contrast with human learners who are general learners and as such sensitive to regularities observed beyond those relevant to a specific task. "Furthermore, one may reasonably argue that a task-oriented learner does not really 'understand' language, since it can accurately decide just one question, while our intuitions about understanding suggest that a competent language user can accurately decide many questions pertaining to any discourse it processes. For instance, a competent language user should be able to reliably answer 'who did what to whom' questions pertaining to each clause in the discourse" (Pereira, 2002). We do not claim that HASPNeL will 'understand' language, yet it may resemble Searle's (1980) Chinese room, able to efficiently perform operations on symbolic representations to produce correct descriptions without having to choose or rank among alternatives.

4.7 Cognitive Plausibility

CFGs also pose an acquisition problem that contrasts with the human experience. Unlike the case of the cognitively-unrealistic treebank containing already-parsed sentences, hierarchical structure is not explicitly represented in primary linguistic data (Fong, 2022). General-purpose systems (GPS) are attractive to the engineering community; advantages include flexibility across problem sets and (non-language) domains. There is also an intuitive appeal in assuming setup simplicity in the language domain as if "nothing necessarily particular to language is hardcoded ahead of time" (Fong, 2022). One can regard these GPS as a continuation of the behaviorist conception, as in Bloomfield's description of language as "a matter of training and habit" (Chomsky, 2021). "However, with so many parameters, the chief downsides are that a lot of training data is required, much more than what seems to be cognitively plausible, and that there are burdensome requirements in terms of computational

resources (for training). The term overparameterization is used when a model has many more parameters than data points, like in GPT-2's case, potentially leading to overfitting, i.e., memorization of the training data, rather than true generalization" (Fong, 2022).

"While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only a few examples [...] —something which current NLP systems still largely struggle to do" (Brown et al., 2020).

We agree with Pereira's conclusion that "although statistical learning theory and its computational extensions can help us ask better questions and rule out seductive non sequiturs, their quantitative results are still too coarse to narrow significantly the field of possible acquisition mechanisms. However, some of the most successful recent advances in machine learning arose from theoretical analysis (Cortes & Vapnik, 1995; Freund & Schapire, 1997), and theory is also helping to sharpen our understanding of the power and limitations of informally-designed learning algorithms" (Pereira, 2002). On the other hand, information-theoretic and computational ideas are also playing an increasing role in the scientific understanding of language. We envision our proposed hybrid system as a step towards bringing together the best of these seemingly irreconcilable perspectives of formal linguistics and information theory.

5 EVALUATION AND ASSESSMENT

Some evaluation methods commonly used among the NLP community are not suitable for HASPNeL. That does not mean that the system cannot be evaluated, but rather that evaluation must be grounded in linguistic principles and formal computational methods.

Since we claim that the HASPNeL system overcomes some of the disadvantages of statistical parsers, it would seem reasonable at first to attempt to evaluate our system's output against that of other statistical systems like the Stanford Parser. However, there are two reasons why this attempt would be futile. In the first place, it is not possible to compare the representations produced by a symbolic system with those of a probabilistic one, since, by design, parses produced by symbolic systems have to be grammatical, yet parses by probabilistic systems do not have any guarantee or presumption of grammaticality. Symbolic parsers as HASPNeL only produce trees that can be generated by the grammar procedures and restrictions (external and internal Merge, unification, locality constraints) that is modeled by the system, On the other hand, probabilistic systems always parse any string of words, regardless of its grammaticality. The parser documentation of the (Group, 2022) states that "this parser is in the space of modern statistical parsers whose goal is to give the most likely sentence analysis to a list of words. It does not attempt to determine grammaticality, though it will normally prefer a "grammatical" parse for a sentence if one exists." In answering why a parse tree assigned to a sentence may be wrong, they give as a possible explanation that "it may be because the parser made a mistake. While our goal is to improve the parser when we can, we can't fix individual examples. The parser is just choosing the highest probability analysis according to its grammar." Evaluating the grammaticality of HASPNeL's performance against a parser like Stanford's will be advantageous to our assessment, but in the end it would not say much about our system.

Another problem with comparing HASPNeL parsing trees against those of another statistical system is that there is no match between the structural descriptions produced by the two different systems. At the core of the HASPNeL system there is a minimalist grammar, following contemporary linguistic theory. Among many other things, minimalist trees are strictly binary and endocentric (every phrase or projection has to have a head of the same category), while statistical systems still use PCFG, with unrestricted rules that allow for multiple branching nodes and exocentric representations. Also, the differences in labeling conventions are beyond comparison. Evaluation methods sometimes applied to probabilistic parsers, such as measuring the accuracy of a structural description by counting and comparing nodes and labels in trees, are not linguistically plausible. Structural representations are grammar-dependent, so they do not have an absolute or "fixed" number of nodes and branches. Likewise, trees may have the same number of the same labels although they were describing different structures. These kinds of comparisons may be somewhat useful between systems using the same grammar, but otherwise they do not produce a valid assessment. Unlike HASPNeL, since typically statistical parsers only choose "the highest probability analysis according to its grammar", they are not particularly well suited to detect ambiguity, either lexical or structural.

From the arguments outlined above, we conclude that the best evaluation of the results obtained from a symbolic, knowledge-based system can only be done by experts who, in this case, have to be human. A

descriptive adequacy assessment methodology similar to that presented by (Gomez-Marco, 2015) functions as a suitable benchmark to assess the correctness of the HASPNeL parsing model. Expert evaluators will be given a number of sentences with their corresponding structural representations produced by the system. Each evaluator assesses syntactic criteria per sentence, such as, (1) the clause's immediate constituents, (2) each constituent's internal structure, (3) argument structure, (4) identification of categories and projections, and (5) detection of structural ambiguities by representations that succeed in criteria 1-4. Assessment of each criterion can be Boolean or using a scale. Evaluators may write comments about their judgements and observations, which shall be used to fix bugs in the system's theory modeling.

Since we are working with a synthetic corpus of a manageable size, at a later stage of the project, we may be able to measure by hand the cases of lexical ambiguity in the annotated synthetic corpus and calculate the likelihood of structural ambiguity in sentences with those lexical units that are ambiguous with respect to the corpus. To assess the system's ambiguity estimation, these measurements may be compared with both the results of the system by detecting possible ambiguity and the likelihood estimates of each interpretation in those cases.

6 CONCLUSIONS

Although machine learning systems have the advantage of a relatively fast and easier training, they fail to acquire the capacity to detect structural ambiguity that gives rise to semantic ambiguity. Symbolic systems, on the other hand, do account for structural ambiguities and are suitable for the construction of a knowledge base as a model of human language cognition. The system we propose exploits the advantages of both strategies, as current literature suggests that NLP implementations are improved by combining resources from both probabilistic and symbolic AI to perform the specific tasks to which they are best. Syntactic formalisms of minimalist grammars and tree-adjoining grammars will be implemented in the system, which can be used as a computational model of language knowledge and acquisition, as well as to test current syntactic theory. This system may also serve as foundation to applications in education, text editing, and the development of other human language technologies, particularly for underrepresented languages which cannot benefit from big data approaches.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. 2219712 and 2219713. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Springer.

Alers-Valentín, H., Rivera-Velázquez, C. G., Vega-Riveros, J. F., and Santiago, N. G. (2019). Towards a principled computational system of syntactic ambiguity detection and representation. In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence - NLPinAI,*, volume 2, pages 980–987. INSTICC, SciTePress.

Bernardy, J.-P. and Chatzikyriakidis, S. (2019). What kind of natural language inference are nlp systems learning: Is this enough? In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence - NLPinAI,*, volume 2, pages 919–931. INSTICC, SciTePress.

Bikel, D. M. (2004). Intricacies of collins' parsing model. *Computational Linguistics*, 30:479–511.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners.

Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. ACL.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, 2:113–124.

Chomsky, N. (1981). *Lectures on Government and Binding*. Number 9 in Studies in generative grammar. Foris, Dordrecht.

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

Chomsky, N. (2001). Derivation by phase (mitopl 18). In *Ken Hale: A Life is Language*, pages 1–52. MIT Press.

Chomsky, N. (2008). On phases. In *Foundational Issues in Linguistic Theory: Essays in Honor of Jean-Roger Vergnaud*, pages 133–166. MIT Press.

Chomsky, N. (2021). Minimalism: Where are we now, and where can we hope to go. *Gengo Kenkyu*, 160:1–41.

Collins, C. and Stabler, E. (2016). A formalization of minimalist syntax. *Syntax*, 19(1):43–78.

Durrett, G. and Klein, D. (2015). Neural CRF parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 302–312, Beijing, China. ACL.

Fong, S. (2005). Computation with probes and goals. In *UG and External Systems: Language, Brain and Computation*, pages 311–334. John Benjamins, Amsterdam.

Fong, S. (2022). Simple models: Computational and linguistic perspectives. *Journal of the Institute for Research in English Language and Literature*, 46:1–48.

Fong, S. and Berwick, R. (2008). Treebank parsing and knowledge of language: A cognitive perspective. In *Proceedings of the Annual Conference of the Cognitive Science Society*, volume 30.

Fong, S. and Ginsburg, J. (2019). Towards a minimalist machine. In *Minimalist Parsing*, pages 16–38. Oxford University Press.

Gaddy, D., Stern, M., and Klein, D. (2018). What's going on in neural constituency parsers? an analysis. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT) 2018*, volume 1, pages 999–1010.

Gomez-Marco, O. (2015). *Towards an X-bar Parser: a Model of English Syntactic Performance*. PhD thesis, University of Puerto Rico Mayagüez.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT, Cambridge, MA.

Graf, T. (2021). Minimalism and computational linguistics.

Group, S. N. L. P. (2022). Stanford parser faq.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Joshi, A. and Shabes, Y. (1997). Tree-adjoining grammars. In Rozenberg and Salomaa, editors, *Handbook of Formal Languages*. Springer-Verlag, Berlin.

Kitaev, N., Cao, S., and Klein, D. (2018). Multilingual constituency parsing with self-attention and pre-training.

Kitaev, N. and Klein, D. (2018). Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne. ACL.

Pereira, F. (2002). Formal grammar and information theory: Together again? In Nevin, B. E. and Johnson, S. B., editors, *The Legacy of Zellig Harris: Language and Information into the 21st Century. Volume 2: Mathematics and Computability of Language*, pages 13–32. John Benjamins, Amsterdam.

Petkevič, V. (2014). Ambiguity, language structures and corpora. *La linguistique*, 50(2):63–82.

Stabler, E. (1997). Derivational minimalism. In Retoré, C., editor, *Logical Aspects of Computational Linguistics*, pages 68–95, Berlin, Heidelberg. Springer Berlin Heidelberg.

Stabler, E. (2011). Computational perspectives on minimalism. In Boeckx, C., editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–643. Oxford University Press.

Stanojević, M. and Stabler, E. (2018). A sound and complete left-corner parsing for Minimalist Grammars. In *Proceedings of the Eight Workshop on Cognitive Aspects of Computational Language Learning and Processing*, pages 65–74, Melbourne. ACL.

Thomson, N. C., Greenwald, K., Lee, K., and Manso, G. F. (2021). Deep learning's diminishing returns: The cost of improvement is becoming unsustainable. *IEEE Spectrum*.

Torr, J., Stanojević, M., Steedman, M., and Cohen, S. B. (2019). Wide-coverage neural A* parsing for Minimalist Grammars. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2486–2505, Florence, Italy. ACL.

Wolpert, D. and Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1:67.