Natural Language Engineering (2021), **27**, pp. 647–689
doi:10.1017/S1351324920000327

CAMBRIDGE UNIVERSITY PRESS

SURVEY PAPER

A systematic review of unsupervised approaches to grammar induction

Vigneshwaran Muralidaran[1], Irena Spasić[2] and Dawn Knight[1]*

[1]School of English, Communication and Philosophy, Cardiff University, John Percival Building, Colum Drive, Cardiff, UK and [2]School of Computer Science and Informatics, Cardiff University, Queen's Buildings, The Parade, Cardiff, UK
*Corresponding author. E-mail: knightd5@cardiff.ac.uk

(Received 17 April 2019; revised 7 May 2020; accepted 8 May 2020; first published online 27 October 2020)

Abstract

This study systematically reviews existing approaches to unsupervised grammar induction in terms of their theoretical underpinnings, practical implementations and evaluation. Our motivation is to identify the influence of functional-cognitive schools of grammar on language processing models in computational linguistics. This is an effort to fill any gap between the theoretical school and the computational processing models of grammar induction. Specifically, the review aims to answer the following research questions: Which types of grammar theories have been the subjects of grammar induction? Which methods have been employed to support grammar induction? Which features have been used by these methods for learning? How were these methods evaluated? Finally, in terms of performance, how do these methods compare to one another? Forty-three studies were identified for systematic review out of which 33 described original implementations of grammar induction; three provided surveys and seven focused on theories and experiments related to acquisition and processing of grammar in humans. The data extracted from the 33 implementations were stratified into 7 different aspects of analysis: theory of grammar; output representation; how grammatical productivity is processed; how grammatical productivity is represented; features used for learning; evaluation strategy and implementation methodology. In most of the implementations considered, grammar was treated as a generative-formal system, autonomous and independent of meaning. The parser decoding was done in a non-incremental, head-driven fashion by assuming that all words are available for the parsing model and the output representation of the grammar learnt was hierarchical, typically a dependency or a constituency tree. However, the theoretical and experimental studies considered suggest that a usage-based, incremental, sequential system of grammar is more appropriate than the formal, non-incremental, hierarchical view of grammar. This gap between the theoretical as well as experimental studies on one hand and the computational implementations on the other hand should be addressed to enable further progress in computational grammar induction research.

Keywords: Natural language processing; Formal grammar; Usage-based grammar; Grammar induction; Parsing

1. Introduction

In the context of natural languages, grammar refers to a system that underlies the ability or capacity of human beings to use natural languages. Different theoretical frameworks have been proposed to formalise the principles of grammar. In the mid-1950s, Noam Chomsky developed the theoretical foundations of generative grammar (Chomsky 1957, 1965), an autonomous, formal system of rules that defines and constrains how lexical items are arranged to create well-formed sentences. This system of rules of well-formedness called syntax was central to generative theory of grammar. The generative view provided an alternative to the behaviourist theories of grammar (Bloomfield 1962; Bloom, Hood, and Lightbown 1974; Skinner 2014), which were prevalent

at the time. There are different schools of the generative grammar tradition such as transformational grammar (Chomsky 1965, 1968; Jackendoff 1977; Radford 1981), generalised phrase structure grammar (Gazdar *et al.* 1985), lexical functional grammar (Dalrymple 2001; Falk 2011), head-driven phrase structure grammar (Pollard and Sag 1994; Levine and Meurers 2006). These approaches differ in the types of rules and representations that they use to predict grammaticality, but the study of syntactic well-formedness based on certain formal rules of arrangement is central to all of them. They share the view that only syntactic well-formedness is directly accessible for analysis and that meaning or semantics can only be studied insofar as it constitutes a compositional homomorph of syntax. Thus, syntax is treated as an autonomous system independent of meaning. Another idea related to the generative view of language is that the differences found in natural languages are just parametric variations of the universal grammatical principles that are genetically encoded in the human brain. This means that there is an innate, universal grammar hardwired in the brain with abstract properties such as distinguishing a noun from a verb, a content word from a function word, and so on (Chomsky 2014, pp. 28–32). Vocabulary, word order and many other language-specific properties are parameters that will be set during language acquisition. In this paper, we refer to these diverse approaches to grammar as *generative-formal* school of thought.

In contrast, more recently introduced theories of grammars are based on an idea that structure or syntax cannot be analysed independently of meaning or semantics. Functional and cognitive linguistics are proponents of this view. In functional theories of grammar, sentence structures are understood in terms of their functions, which can be semantic (agent, patient, etc.), pragmatic (theme and rheme, topic and focus, etc.), syntactic (subject, object, etc.) or discursive (references, cohesion, etc.) (Dik 1987, 1991). These theories explain grammatical structures by grounding their analysis in the communicative situation (Bates and McWhinney 1982; Givón 1983; Nichols 1984; Dik 1987, 1991; Matthiessen and Halliday 2009). Cognitive linguistics argues that all knowledge of linguistic phenomena is conceptual in nature and that grammar is not an independent mental faculty but connected to all other general cognitive processes and structures (Evans 2006).

While generative theories imply the existence of a universal grammar, cognitive approaches to grammar treat linguistic structures as cognitive schemas or mappings between form and function that are inductively learnt through real-life language use. Cognitive grammar (Langacker 1987, 2008, 2009) and construction grammar (Goldberg 2003; Östman and Fried 2005; Król-Markefka 2014) are examples of usage-based approaches to describing human linguistic ability. Cognitive grammar argues that all linguistic units from morphemes, grammatical categories to syntactic relations are meaningful symbolic units which evoke different aspects of conceptualisation in the user's mind during language processing (Langacker 1987, 2008). Cognitive grammar is different from generative grammars in three ways: in its centrality of meaning, meaningfulness of grammar and usage-based nature of grammar (Król-Markefka 2014). Construction grammar is a theory of grammar where the primary units of linguistic analysis are *constructions* that integrate *form* and *content*. *Form* refers to any combination of phonological, morphological or syntactic patterns or templates and *content* broadly refers to the meaning derived from semantics, pragmatics and discourse structure which are analysed in terms of conceptual structures such as image schemas, frames, conceptual metaphors, mental spaces (Lakoff and Johnson 1980; Lakoff 1988; Fauconnier 1994; Hampe and Grady 2005). We refer to these different approaches to grammar as *functional-cognitive* school of thought.

Out of the two schools of thought, *generative-formal* school has been highly influential and dominant in theoretical linguistics (TL). The formal grammars based on generative tradition have found many practical applications as well. Perhaps most prevalently, they are used to describe the syntax of programming languages and compile and interpret code written in such languages (Harrison 1978; Moshier 1988). They have also been successfully applied in natural language processing (NLP) to describe and process the syntax of natural languages. Traditionally, grammars used in this context were defined manually, for example, using context-free grammar (CFG)

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

rules, which are then extended for computational implementation. Given the complexity of natural languages, such rules are not exhaustive, and this obviously creates a knowledge engineering bottleneck. In order to address this problem, data-driven methods have been used since 1990s to extrapolate a grammar from large corpora by exploiting their statistical properties (Briscoe and Waegner 1992; Leech 1993; Schabes, Roth, and Osborne 1993). The most prominent subclass of such methods, supervised machine learning, requires syntactic categories and relations to be annotated manually beforehand. Although the task of manual annotation is much simpler than that of defining the grammar rules, this involves the development of large annotated treebanks containing millions of words and thousands of sentences (for instance, Penn treebank has 3 million words of skeletally parsed text (Taylor, Marcus, and Santorini 2003)). The sheer volume of training data that should be annotated for supervised learning to perform well still presents a considerable bottleneck for knowledge engineering. Other supervised learning applications where annotations do not require specialised expertise have successfully resolved this problem through crowdsourcing (Cocos *et al*. 2015). Unfortunately, linguistic expertise is not readily available to attempt a crowdsourcing approach for creating large treebanks, so alternative approaches need to be considered. Unsupervised machine learning methods, which draw inferences from raw or unlabelled data, have started to find applications in grammar induction from text corpora (Klein and Manning 2004).

While the evolution of NLP and computational linguistics (CL) thus proceeded hand-in-hand with the evolution of generative-formal linguistic theories, computational linguists have often emphasised the divergence of aims between TL and CL, sometimes even questioning the relevance of linguistic theories to CL (Paillet 1973; Jones 2007). This is due to two reasons. Firstly, the generative linguistic theories identify linguistic classes and describe the structural units purely based on their formal properties without functional motivations. Typically, these theories provide a *non-process, descriptive* account of the overall structural properties of language. However, CL is interested in modelling a *process* account of how linguistic data can be manipulated in specified ways to yield particular results. For instance, in computational linguistics, mechanisms for accessing and deriving phrase structure rules require additional computational modules which are quite distinct and divorced from the core competence grammar modules described by the generative grammar frameworks. A more straightforward view of grammar, where processing is directly related to linguistic structures and meaning, is preferable. Secondly, with the rise of statistical methods and their usefulness in various computational linguistic tasks, a theory of grammar which is empirically grounded and compatible with statistical learning from linguistic usage is preferred.

Formal linguistic theories and statistical approaches in CL also differ in their views of ambiguity resolution. Linguistic theories focus on the human ability to recognise and form grammatical sentences. They state the formal principles that characterise the human linguistic capacity as a system and thus do not concern themselves with resolving any grammatical ambiguities. However, statistical approaches aim to assign the most probable structure out of all possible grammatical structures for a given utterance. Thus, ambiguity resolution is at the heart of CL. In this context, the functional-cognitive school has the advantage of mapping the linguistic structures to meaning directly. It emphasises usage-based learning, maintains the centrality of meaning in linguistic analysis, treats linguistic structures as form-function mappings called constructions and approaches syntactic well-formedness as successful symbolic assembly of form-function pairs. It holds that it is the meaning that can be accessed directly, and the syntax is learnt inductively through real-life language use. This has implications for grammar induction, which is defined as the process of learning the formal rules from a set of grammatical sentences with or without structural annotations (D'Ulizia, Ferri, and Grifoni 2011). In a cognitive view of grammar, grammatical categories and relations are not available beforehand but are themselves grounded in patterns of usage and conceptualisations associated with them. According to cognitive grammar, the essence of a grammar lies in conceptualisation whereby a symbolic link is construed between a linguistic form and its meaning. Induction of a sentence structure becomes the task of learning a composite

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

structure as a form-meaning assembly of the components of the sentence. Inducing such form-meaning pairs from raw text can be challenging due to having access to words as symbols. The functional-cognitive school has not successfully explained how exactly this usage-based induction of grammar can be computationally modelled. A possible approach can be to identify primitive form-meaning pairs that can be encoded from basic cognitive profiles or if they can be induced from local statistical dependencies. Unsupervised grammar induction then becomes the task of inducing grammatical categories and relations by identifying the basic word-meaning pairs and learning the patterns of their assembly.

The influence of *generative-formal* school of thought on language processing is evident from the earlier rule-based parsers to the later supervised models of parsing. A systematic study of *functional-cognitive* influences on unsupervised approaches to grammar induction has the potential to highlight any gap between the grammatical theories and the computational processing models of grammar. While surveying the parser implementations, we look for the theoretical underpinnings of these studies, their evaluation methodologies, identified baselines of evaluation and their relative strengths and weaknesses. Apart from informing us of the state-of-the-art methods and baselines, a thorough literature review can also help us identify domain-independent computational methods that might be usefully adapted to usage-based grammar induction.

This study takes the linguistic perspective to grammar induction. There is a statistical perspective of grammar induction that is taken in machine learning. It focuses on defining the model as the set of parameters and the ways they are linked together to determine the probability of a grammatical sentence. Here, an objective function is defined to allow selection of a single estimate of the parameters using a search approach that performs such estimation efficiently. These aspects are outside of the scope of this article. Instead, we concentrate on the structural aspects of defining the model through the use of grammar. The paper is organised as follows. Section 2 explains the methodology and protocol for conducting this review. It also contains various subsections which present the following details: our research questions, search strategy, criteria to include and exclude the results retrieved by our search query, quality assessment of the selected studies, data extraction, the information synthesised from the extracted data, how the data synthesised were stratified into various aspects namely: theory of grammar; output representation; how grammatical productivity is processed; how grammatical productivity is represented; scale of training; features used for learning; evaluation strategy and implementation methodology. Section 3 presents the insights obtained from theoretical, experimental and survey papers. Section 4 summarises the findings from all the studies and identifies the specific insights obtained from two important studies on unsupervised construction grammar induction. Section 5 discusses the implications of the findings for future research and concludes the review.

2. Methodology

Systematic reviews aim to identify, critically appraise, interpret and summarise all currently available evidence in relation to a given research question. Systematic reviews are common in medical and healthcare literature, but they are also becoming increasingly useful for other fields (Petticrew 2001). They follow strict scientific protocols based on explicit and reproducible methods designed to limit bias and random errors. Such protocols include multiple steps, typically: (1) identifying a set of well-defined research questions, (2) defining strict inclusion and exclusion criteria, (3) searching relevant literature databases using a carefully developed set of search terms, (4) assessing the quality of studies, (5) systematic extraction, abstraction and synthesis of evidence by multiple investigators independently. Consequently, they provide reliable conclusions and often identify research gaps to guide future research. We followed the systematic literature review methodology proposed by Kitchenham and Charters (2007). It involves all the five steps of protocols mentioned above. Figure 1 shows the steps involved in the systematic review.

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

Table 1. Research questions

ID	Research question
RQ1	Which types of grammar theories have been the subjects of grammar induction?
RQ2	Which methods have been employed to support grammar induction?
RQ3	Which features have been used by these methods for learning?
RQ4	How were these methods evaluated?
RQ5	In terms of performance, how do these methods compare to one another?

RQ1 is concerned with the types of grammars that are amenable to automated induction. Particularly, we are interested in the ways in which these grammars are represented.

RQ2 aims to identify a range of methods and techniques used to implement grammar induction approaches. Furthermore, we want to examine how these approaches address the notion of grammatical productivity, which is defined as the human capacity to keep creating new grammatical expressions by manipulating a finite set of linguistic resources.

RQ3 focuses on the types of features used by these methods and their utility for grammar induction. We also want to explore the ways in which such features can be extracted together with the associated costs, for example, in terms of manual effort involved (e.g., for annotation) and the volume of data needed to train the methods.

RQ4 is concerned with the performance of existing grammar induction methods. In particular, unsupervised methods are known to be notoriously difficult to evaluate. To that end, we want to identify which evaluation measures have been used in practice and whether they are transferable across different methods thereby allowing them to be compared and ultimately establish the baseline performance as part of RQ5.

RQ5 aims to consolidate the findings from various studies and interpret the results obtained by comparing their relative strengths and weaknesses. It can help us understand the implications of these results for future research.

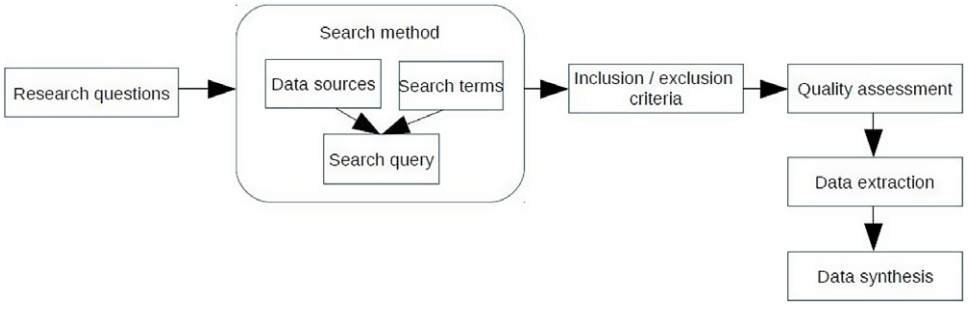

Figure 1. Systematic review protocol.

2.1 Research questions

This review aims to determine the influences of functional-cognitive school on unsupervised approaches to grammar induction in NLP. It tries to achieve that by conducting a systematic literature review on the state of the art in CL and NLP that lie at the intersection of the following domains: usage-based theories of grammar, unsupervised approaches to computational grammar induction and grammar representation. The intersection ensures that the study could be about any of the following – unsupervised parser implementation, a computational study or experimental study related to functional-cognitive school of thought, and studies related to representation of grammar. The research questions (RQs) listed in Table 1 are addressed while synthesising information obtained from unsupervised parser implementations.

2.2 Search strategy

In order to efficiently identify a set of articles relevant to the given research questions, we compiled a list of appropriate search terms. First, we identified a set of relevant domains and then

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

compiled a list of keywords related to each domain. Search queries based on these keywords were tested against relevant literature databases, which include ACM Digital Library, Cardiff University Library Search, DBLP, Google Scholar and IEEExplore. The abstracts and keywords from the retrieved results were screened to check their relevance and identify other pertinent search terms including synonyms and spelling variations. The list of search terms was refined iteratively in this manner until no significant changes could be made. Table 2 provides the finalised list of search terms, where a wildcard character was used to address inflection and derivation. Finally, a Boolean OR operator was used to combine the search terms for each domain. These subqueries were then combined using a Boolean AND operator.

2.3 Selection criteria

The retrieved articles were manually curated with respect to their relevance to the given set of research questions. To formalise the curation process, we defined a set of inclusion and exclusion criteria.

Inclusion criteria

1. The article has to contribute to the field of NLP or CL.
2. The article has to focus on grammar induction.

Exclusion criteria

1. Psycholinguistic studies which look at neurobiological factors of linguistic phenomena.
2. Studies which are language-specific or construction-specific.
3. Articles that were not peer-reviewed.
4. Certain types of publications such as editorials, informal articles, tutorials and posters.
5. Articles written in a language other than English.

Apart from this, we also followed up the citations from the studies included in our list and if any of those cited studies passed our inclusion criteria, we updated our selection list with these studies as well.

2.4 Quality assessment

All selected articles underwent quality assessment based on the following criteria:

- The research goals, methodology and contribution to the field are clearly defined.
- Data used in experiments are described with sufficient detail.
- The results are reported using theoretically sound evaluation metrics and compared, where appropriate, against a relevant baseline.
- Limitations of the study are carefully analysed.

The results returned by the top 10 pages only were chosen and exclusion criteria were applied on them which resulted in 198 articles. A total of 190 articles that remained after removing the duplicates (i.e., the same study retrieved from multiple data sources) were assessed according to inclusion criteria in Section 2.3, which reduced the number of articles to 42. All the 42 studies described their goals, methodology, experiments, results and analysis according to the quality assessment criteria listed above. This process of searching and selecting the appropriate papers for systematic review is shown in Figure 2. One study, which was known to be relevant, was not retrieved, but was added manually, thus providing a total of 43 studies reviewed here.

2.5 Data extraction

To answer our research questions, evidence was extracted from the final set of 43 articles to review and formatted according to predefined data extraction cards. The cards populated with

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

Table 2. Search query construction

Domain	Search terms
Usage-based theories of grammar	gramma*
	gramma* cat*
	syntax
	cognitive
	construction
	usage-based
	form-meaning
	construal
	schema*
	linguistic
	functionalist
	pattern
Computational grammar induction	induction
	inference
	acquisition
	parsing
	learning
	processing
	analysis
	synta* struct*
	parse
	chunking
	unsupervised
Representation of grammatical structure	representation
	formalism
	model
	framework
	finite-state
	automata
	incremental
	trees
	networks
	assembl*
	neural net*
	graphs
	shallow pars*

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

Table 2. Continued

Domain	Search terms
Review papers on grammar induction	survey
	review
	bibliographical study
	meta-analysis

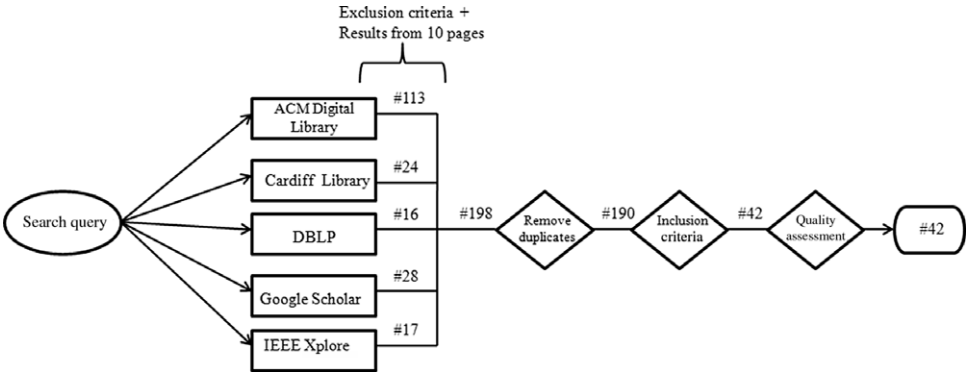

Figure 2. Search and selection of papers for systematic review.

information from articles describing practical implementations had the following fields: title, grammar theory, grammar representation, computational model, methodology, features, dataset, evaluation metrics, results and summary. The cards populated with information from articles describing theoretical and experimental studies had the following fields: title, aim, method, experiment, conditions, result, insights and summary. This systematic data extraction allowed us to compare the evidence obtained from various studies and support generalisation from the observed evidence. The studies considered in our review and their references can be accessed from Appendix.

2.6 Data synthesis

Table 3 provides a brief summary of the extracted data. Towards generalising the findings, we stratified the articles across different aspects considered in Table 3. They help us quantitatively determine the influence of functional-cognitive school on the studies retrieved from the data extraction step. Each study can be from one of these three categories: (a) a computational study or an experimental study on grammar induction or processing, (b) an implementation of unsupervised grammar induction and (c) a survey or evaluation paper on grammar induction. Each category of study offers different kinds of information. Data obtained from experimental studies on human subjects are *goal, methodology, experiment, result, insights* and *summary*. Data obtained from survey or evaluation papers are *a list of findings from the survey*. Data obtained from an implementation studies are *theory of grammar, representation of grammatical productivity, processing grammatical productivity, output representation, evaluation strategy, scale of training, features used for grammar induction,* and *approach or methodology behind the implementation*. The studies used for data extraction and a brief list of findings synthesised from them are included in the Appendix.

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

Table 3. Criteria and their values synthesised from 43 studies

Criteria	Classification	Total
Type of study	Implementation of grammar induction	33
	Theoretical or experimental studies	7
	Survey or evaluation paper	3
Theory of grammar	Generative-formal	22
	Functional-cognitive	3
	Theory-neutral	8
Representation of grammatical productivity	Hierarchical	31
	Non-hierarchical	2
Processing grammatical productivity	Incremental	1
	Non-incremental	32
Output representation	Constituency trees	14
	Dependency trees	11
	A directed multigraph of patterns learnt	4
	Construction slots or templates	2
	Transient structure	1
	Common cover link set	1
Evaluation strategy*	Comparing against gold standard treebank annotation	25
	Performance of the learner in grammaticality judgement tasks	2
	Agreement between two highly constrained models on disjoint corpora	1
	Qualitative evaluation	1
	Maximum coverage, minimum size and stability measures	3
	Agreement with CFG grammar that generated the learning data	2
Features used**	POS tags	18
	Valence and direction of attachment	10
	Distributed representation of words	5
	Word alignments	1
	Chunks or bracketed sequences	1
	Orthographic cues	2
	Heuristic rules	4
	Construction association measures	2
Grammar induction approaches***	Top-down dependency grammar models and their variations	10
	Exemplar-based models and their variations	2
	Distribution-based	3

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

Table 3. Continued

Criteria	Classification	Total
	Clustering approaches	2
	Heuristic approaches	2
	Automatic Distillation of Structure (ADIOS)	4
	Probabilistic context-free grammar (PCFG)	3
	Chunkers and their extensions	1
	Data-oriented parsing and variations (DOP)	4
	Construction grammar induction	2

*Discrepancy in total count. Study number 26 in the appendix uses two different evaluation methods.
**Discrepancy in total count. Different studies use multiple features.
***Study numbers 10 and 32 in the appendix use more than one approach for grammar induction.

The synthesis of data analysed along these aspects can answer the research questions mentioned in Section 2.1. While most of these aspects are self-explanatory, the rationale behind identifying these aspects needs a discussion before we present them in detail later. *Theory of grammar* aspect tries to understand the linguistic school of thought an implementation is influenced by. *Representation of grammatical productivity* and *Processing grammatical productivity* are two aspects which reveal how a study approaches the syntactic productivity computationally. To treat productive structures in natural languages with recursive, hierarchical application of productive formal computational operations can indicate the influence of generative-formal thought on the model of parsing. Through the two aspects said above, we see if the approach to grammatical productivity in a given study is purely in a generative-formal sense or if it explicitly or implicitly is amenable to a functional-cognitive standpoint. *Representation of grammatical productivity* is understood from how the study treats the productive elements of the language in its output: a hierarchical arrangement of productive linguistic units in the form of constituency or dependency trees shows a generative-formal influence. Any computational representation that does not implicitly assume recursion and hierarchy but reveals productivity of language through meaningful assembly of usage patterns indicates functional-cognitive influence. *Processing grammatical productivity* is understood from how the decoding is done in the parsing model. An incremental decoding indicates that grammatical productivity is processed incrementally, even partial non-sentences can have their parse, and not all words of the sentence need to be available before parsing begins. Such computational models are compatible with cognitive grammar ideas.

The aspect *Output representation* reveals the various types of grammar outputs found in the data. These output representations can sometimes be formally similar or in some cases intrinsically tied to the grammar itself. For example, the construction slots as discussed in Dunn (2017a, 2017b) are filled with syntactic phrases and are formally equivalent to a phrase structure in a constituency tree. However, given the overall scheme of the paper, which tries to learn an optimal construction grammar by allowing various levels of representation from lexical to semantic, we see that the slot could potentially be filled with any entity even entirely functional ones. Thus, we treat the construction slots as distinct kind of output representation. Similarly, the directed multigraph learnt by the Automatic Distillation of Structures (ADIOS) implementation (Solan *et al.* 2004) is directly equivalent to the CFG learnt by the system. However, we see that the method itself is amenable to extend learning form-function pairs that it is useful to recognise it as different from a regular constituency or dependency tree. *Evaluation strategy* identifies how the performance of the system is judged. *Scale of training* indicates if the learning is fully unsupervised or semi-supervised. Finally, the *approach or methodology* is an aspect that identifies what is the computational method used for grammar induction in a given study.

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

Another point to note is that many studies can fit in more than one of the categories of the relevant aspect. For example, the same study can show evidence for multiple types of evaluation strategies in different types of experiments or the same study can be analysed as adopting multiple approaches to grammar induction. In all such cases, we include the study in all the relevant categories and the total counts shown in Table 3 that reflect. The details of the actual data extracted can be found in Appendix.

These different aspects of synthesis are discussed in the following sub-subsections.

2.6.1 Theory of grammar

In Section 1, we reviewed different theories used to formalise the notion of a grammar. We classified all studies describing practical implementation of grammar induction with respect to the underlying grammatical theory. We could identify three classes of theoretical views underlying the implementation. They are

A. *Formal-generative* grammar implementations which explicitly learn a system of rules or constraints that describe how lexical items are arranged in order to form a grammatical sentence.
B. *Functional-cognitive* grammar implementations in which the linguistic capacity is explicitly modelled as an inductive process based on language use in practice.
C. *Theory-neutral* implementations are amenable to be adapted to incorporate functional-cognitive insights.

We distinguished between *functional-cognitive* and *theory-neutral* implementations when the latter did not necessarily learn a usage-based representation of grammar as the output. We found that although domain-independent bottom-up methods can be used to extract usage-based schemas, most of these studies learnt a formal grammar with their output represented by the likes of constituency or dependency trees. Recognising such theory-neutral implementations thus becomes important because they can be adapted to learn fully usage-based grammar in an unsupervised fashion.

We found that the vast majority (22 out of 33) of the studies showed the influence of formal-generative school of thought in their parsing models. A top-down dependency grammar model for unsupervised parsing exemplifies this view. Given a sequence of words, the model starts with a head, attaches a sequence of arguments to the left or right and generates a dependency tree in a top-down manner. Here, the dependencies are formal syntactic relations which are defined *a priori* by linguistic theories and are learned statistically. Klein and Manning's (2004) corpus-based induction of syntactic structure laid the foundation for a statistical generative model to unsupervised dependency parsing. Overall, we found that there were 11 studies (out of 22) that implemented the top-down dependency model of grammar or its variations (Klein and Manning 2004; Headden, Johnson, and McClosky 2009; Sangati 2010; Spitkovsky, Alshawi, and Jurafsky 2010, 2011, 2012; Boonkwan and Steedman 2011; Dominguez and Infante-Lopez 2011; Gillenwater *et al.* 2011; Mareček and Žabokrtský 2012a, 2012b). The remaining 11 implementations (out of 22) adopted different methods to learn formal grammar. They are described below.

Snyder, Naseem, and Barzilay (2009) learn constituency trees using an unordered tree alignment model where word alignments from bilingual corpora with their parts of speech (POS) are used as features to loosely bind the parallel trees from different languages. An exemplar-based approach to unsupervised parsing was proposed by Dennis (2005) where the parse tree of the target sentence is obtained by aligning it with nearest-neighbour exemplar sentences and choosing a constituency tree with minimum cost for alignment. There were three studies that used distributed representations of words in deriving the parsed trees. One was by Brooks (2006) where distributional representation of words was used to segment text into constituents and heuristics were then applied to reduce the number of possible candidates (Brooks 2006). Seginer's algorithm

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

(2007) captures the skewness of syntactic trees in its syntactic representation, restricts the search space by processing utterances incrementally (like humans do) and relies on the Zipfian distribution of words to guide its parsing decisions. The other study was Søgaard's (2011) implementation which presents a very different approach to unsupervised dependency parsing. They explore the view that dependency structure can be treated as a partial order on the nodes in terms of saliency or centrality. Their implementation assigns a dependency structure to a sequence of n words in two stages. In the first stage, they decorate n-nodes with word forms and distributional clusters, construct a directed acyclic graph in $O(n^2)$ and rank the nodes using iterative graph-based ranking (Brin and Page 1998). In the second stage, a parse tree is constructed from the ranked list of words (Søgaard 2011). Another implementation of Ponvert, Baldridge, and Erk (2011) starts with learning chunks using standard probabilistic finite state models and then cascades this chunker to achieve constituent parsing. Two studies used heuristic models (Araujo and Santamaría 2010; Santamaria and Araujo 2010). One study used clustering approach and explicit formal syntactic features to learn constituency trees (Reichart and Rappoport 2008). Reichart and Rappoport (2008) propose a labelled grammar induction by starting from POS tags, followed by the induction of initial brackets, subsequently labelling them and finally clustering the label outputs using syntactic features. There were two studies (Adriaans, Trautwein, and Vervoort 2000; Jin *et al.* 2018) that treat parsing as a task of learning the probabilistic CFGs (PCFGs). In summary, these 22 studies exhibit the *formal-generative* view either implicitly or explicitly.

Only three studies explicitly considered grammar as a usage-based system and modelled the grammar induction process from this theoretical perspective. The use of a construction grammar induction algorithm is an example of this type of implementation (Dunn 2017a, 2017b). Given a corpus of sentences, this method statistically allows all potential linguistic generalisations from the observed sentences and chooses the optimal inventory of construction slots that can generalise them. In one study, Dunn (2017a) demonstrates the learnability and falsifiability of construction grammar. Learnability is the degree to which the optimum set of constructions can be consistently selected from the large set of potential constructions; falsifiability is the ability to make testable predictions about the constructions present in a dataset. The study evaluates these two by performing an induction task. In another study, the same author describes in detail how a construction grammar learner can be implemented and evaluated (Dunn 2017b). An algorithm that achieves this using frequency and association measures of co-occurrence at multiple levels of analysis (lexical, syntactic and semantic levels) is proposed. Marques and Beuls (2016) propose proof-of-concept evaluation strategies for computational construction grammars. They take inspiration from existing measures in semantic parsing and machine translation and propose two new metrics for this evaluation task. These three studies complete discussion of grammar induction or evaluation from an explicit usage-based view thus showing their compatibility with *functional-cognitive* school.

Finally, the remaining eight studies proposed methods that can learn significant usage patterns bottom up, but they were explicitly not framed to learn usage-based grammars. These methods were used to learn constituency trees and CFG rules and evaluated against formal syntactic annotated trees. The ideas themselves need not be tied to learning formal syntactic trees and can be easily adapted to incorporate ideas from functional-cognitive theories. We call such studies *theory-neutral*. In our review, we found two major implementation methods that were theory-neutral. First method was ADIOS algorithm, which incrementally learns a model of morphosyntax from raw input by distilling structural regularities and contextual cues. At the end of learning, a directed multigraph containing an abstraction of patterns, which can in turn be represented as context-free writing rules, is produced. The second method was data-oriented parsing (DOP) where an unsupervised DOP model allows all possible binary trees to be built bottom up and computes the most probable tree from the shortest derivations of sentences. This model can be used to explain both rule-based and exemplar-based properties of a natural language.

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

There were four studies which used the approach of ADIOS to learn significant syntactic rules from data without any annotation whatsoever (not even POS tags) (Solan *et al.* 2004; Edelman *et al.* 2005; Berant *et al.* 2007; Brodsky and Waterfall 2007). One of these implementations by Brodsky and Waterfall (2007) proposed a method for evaluating the grammar learned by the ADIOS system even in the absence of any gold treebank for quantitative evaluation, or human judgement for qualitative evaluation. Their observation is that when two highly constrained models, which are trained on disjoint corpora, agree very closely on the acceptability of a given test sentence, it cannot be coincidental. An agreement between two such models indicates that the scoring of acceptability based on this agreement is correct. This was the basis of their evaluation method.

For DOP and its variations, we identified four implementations in our review (Bod 2006, 2009; Zuidema 2006; Post and Gildea 2013). The authors point out that their implementation methodology is compatible with the usage-based theoretical views such as construction grammar, but they have used the method to learn constituency trees and evaluate them against a gold treebank. By allowing all possible binary tree substructures that combine to form the final parse tree through a formal operation called labelled substitution, DOP obtains the optimal parse by computing the most probable tree from among the shortest derivations of sentences. We consider these eight studies as theory-neutral implementations.

2.6.2 Representing grammatical productivity

We have previously introduced the notion of grammatical productivity as the speaker's capacity to produce novel utterances of virtually limitless length using a finite number of linguistic resources at hand. In our review, we noticed that there were two ways to represent grammatical productivity in all studies considered. They are

A. Hierarchical arrangement of productive linguistic units, for example, constituency and dependency trees.
B. Non-hierarchical representation of productive units, for example, sequence of construction slots.

We found that 31 out of 33 studies represent grammatical productivity hierarchically, whereas the remaining two studies do it non-hierarchically (Table 3). Here, we noticed that the 22 studies that take the generative-formal view of grammar represent this productive capacity of natural language as a constituency tree or a directed dependency tree with either phrasal hierarchy or the head-dependent relations. The three studies on construction grammar implementation and evaluation represent their output as a sequence of construction slots, which is non-hierarchical. The study itself fills the construction slots with phrase structure heads which are indeed hierarchical. This might raise the question as to why this output is considered non-hierarchical. The rationale is that the study indeed allows all types of information, from lexical to semantic, to fill its construction slots and demonstrates the feasibility of learning construction grammar. In the larger scheme of the paper, we recognise that the authors learn the construction grammar with various co-occurrence metrics and therefore the fact that phrase structures are filled as entries in the construction slots does not make it central to the idea of construction slot itself. The construction slot representation is perfectly compatible with any type of assembly of symbols which are non-hierarchical. It is not inherently motivated to be filled with formal phrasal heads, and hence we identify the three studies as showing non-hierarchical approach to representing grammatical productivity. Finally, the eight studies that are theory-neutral in the sense defined in the previous Section 2.6.1 represent their output hierarchically. The four DOP implementations learned how phrases should be attached in a constituency hierarchy. In the four ADIOS implementations, the structural patterns distilled from the data are equivalent to the recursive phrase structure rules.

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

2.6.3 Processing grammatical productivity

The next aspect that we assessed was how the processing of grammatical structure was treated in each study. We identified two approaches:

A. Incremental processing
B. Non-incremental processing

In this study, by incremental processing of grammatical productivity, we mean that the analysis of grammatical structure of a sentence is incrementally updated when new sequences of words are observed; not a model which starts with all the *n* words w1, w2. . . w*n* and which treats parsing as establishing or filling the grammatical relations between those *n* words. As discussed in Section 2.6, in incremental decoding even partial non-sentences can have their parse, not all words of the sentence need be available before parsing begins. When the decoding itself is incremental in an implementation, we call the processing of grammatical productivity incremental. Otherwise, it is non-incremental. Incremental parsing is psycholinguistically motivated. Data synthesis revealed that several experimental and theoretical studies deemed incremental parsing suitable for grammar induction (Cramer 2007; Frank, Bod, and Christiansen 2012). According to functional-cognitive school, grammatical structure of a sentence is almost entirely overt and it does not conceal a deeper level of grammatical organisation (Langacker 1987 pp.46–47). A sharp distinction between competence (speaker's mental grammar) and performance (sentence production and processing) is also not maintained (Jensen 2014). Cognitive grammar literature also proposes incremental contribution of components to the holistic conceptual structure of a text (Harrison *et al.* 2014 pp. 22,100). These properties make functional-cognitive school compatible with the psycholinguistic literature. Although generative grammars can also be used to parse a sentence incrementally, the grammar formalism and derivation/parsing strategy are divorced. As mentioned in Section 1, 'mechanisms for accessing and deriving phrase structure rules require additional computational modules which are quite distinct and divorced from the core competence grammar modules described by the generative grammar frameworks'.

In our review, we found that 32 out of 33 studies were non-incremental in processing the grammatical productivity, whereas one study learned grammar incrementally. The reason to distinguish these two types is to see if an implementation can be extended to incorporate functional-cognitive insights. The incremental parsing algorithm by Seginer (2007) uses a new link representation for syntactic structure which allows a prefix of an utterance to be parsed before the full utterance has been read.

It should be noted that there could be studies where the unsupervised model that learns significant grammatical patterns could be incrementally trained. Corpus-level incremental training is an important consideration for any scheme of grammar learner. However, we explicitly did not consider it a factor of analysis in this section because its importance is equally relevant for grammar learners of any theoretical persuasion. We wanted to understand what aspects can reveal the influence of one grammatical school or the other. For example, ADIOS method extracts statistical patterns incrementally from a corpus of sentences. When new sentences are observed in the corpus, the algorithm learns new patterns and updates the directed multigraph. However, to decode the parse of a target sentence, the entire sequence of words of the target sentence is considered non-incrementally. We do not count such studies as processing the grammatical productivity in an incremental fashion.

2.6.4 Output representation

The next aspect of analysis is the output representation of the grammatical structure of a sentence. We identified six types of output representations from the 33 studies. They are

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

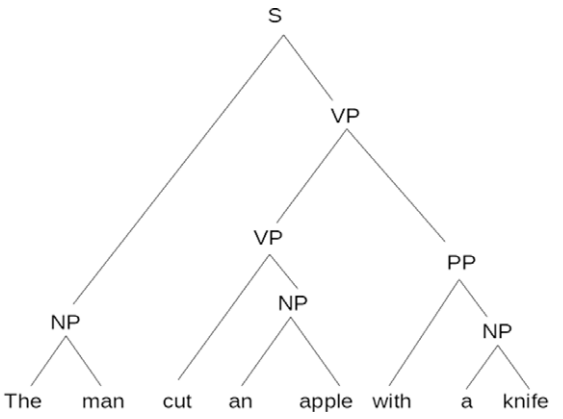

Figure 3. Constituency representation.

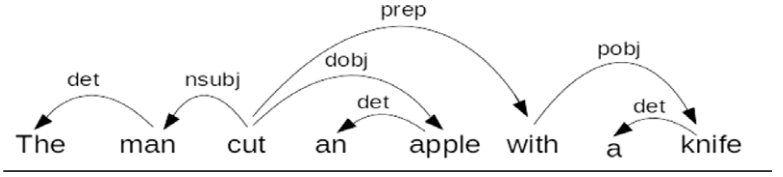

Figure 4. Dependency representation.

A. Constituency trees
B. Dependency trees
C. A directed multigraph
D. Construction slots or templates
E. Transient structure
F. Common cover link set

Let us consider an example sentence and illustrate these output representations. For a sentence 'The man cut an apple with a knife', the various output representations after grammar induction are illustrated in Figures 3–6.

A constituency tree shows the phrase structures that make up the sentence. In this analysis, every sentence is broken down into smaller phrases which combine with other such phrases to form a larger phrase until the entire sentence is analysed as well formed according to the phrase structure rules. This is shown in Figure 3.

In dependency analysis, a sentence is analysed in terms of the words (constituents) and their relationships. A dependency tree is a directed graph where each node is a word (constituent), child nodes are marked as dependents of parent nodes and the edges between the dependents and heads indicate their syntactic relationship. This is shown in Figure 4.

ADIOS implementations learn a directed multigraph where incremental update of usage patterns is done based on structural similarities and statistical information present in the text. In this technique, frequent strings of similar structure are treated as significant patterns and these patterns are treated as 'Equivalent Classes' which means members of the same equivalent class are valid alternatives in a usage pattern. ADIOS algorithm repeatedly applies its pattern recognition algorithm on all sentences in a text and outputs one directed multigraph showing both patterns and equivalent classes. A sample directed multigraph with just three sentences is shown below in

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

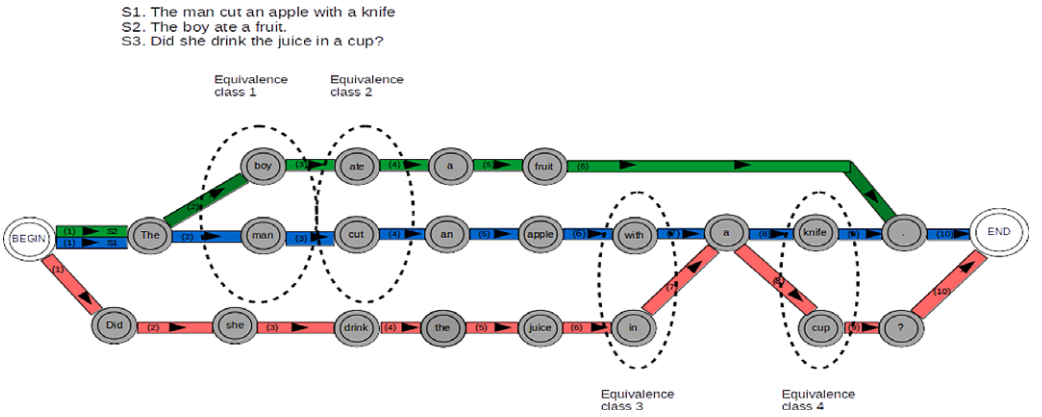

Figure 5. An initial directed multigraph for a simple corpus of three sentences.

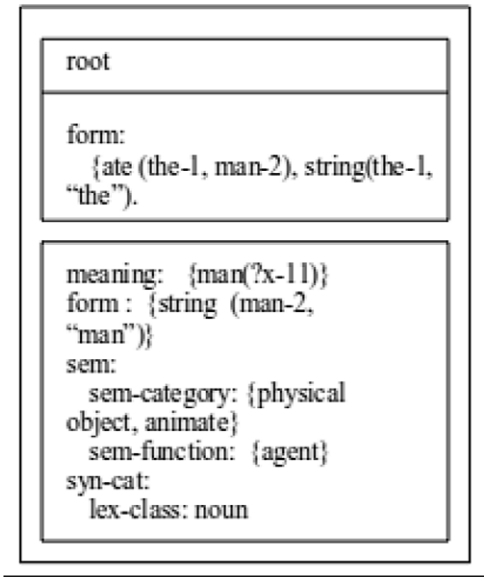

Figure 6. Transient structure.

Figure 5. More details can be obtained from Solan *et al.* 2004, Edelman *et al.* 2005, Berant *et al.* 2007 and Brodsky and Waterfall 2007.

Constructions are productive and schematic form-meaning mappings that differ in their size, level of schematicity and their internal complexity. When it comes to construction grammar, the difficulty is not so much about the representation of constructions as much it is about learning and evaluation of these constructions. One representation that we identified in our review is a simple sequence of construction slots. For the given example, a construction slot sequence could be [NN PHRASE – VB PHRASE – NN PHRASE – PREP PHRASE] or [NP < ANIMATE> – VB < TRANSFER> – NP < ANIMATE> – NP] or [NN – "give" – NP < ANIMATE> – "a piece of" – NP < ANIMATE> – "mind"]. It should be noted that the construction slot can be filled with phrasal heads, semantic information and idioms, lexical information and so on. It is not tied to any particular type of formal linguistic category. These slots can be filled with any type of symbolic assembly with a meaningful form-function mapping. In our review, we found that the

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

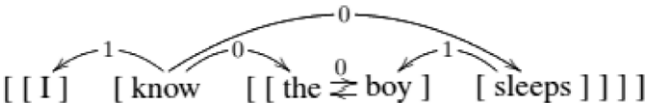

Figure 7. Shortest common cover link set.

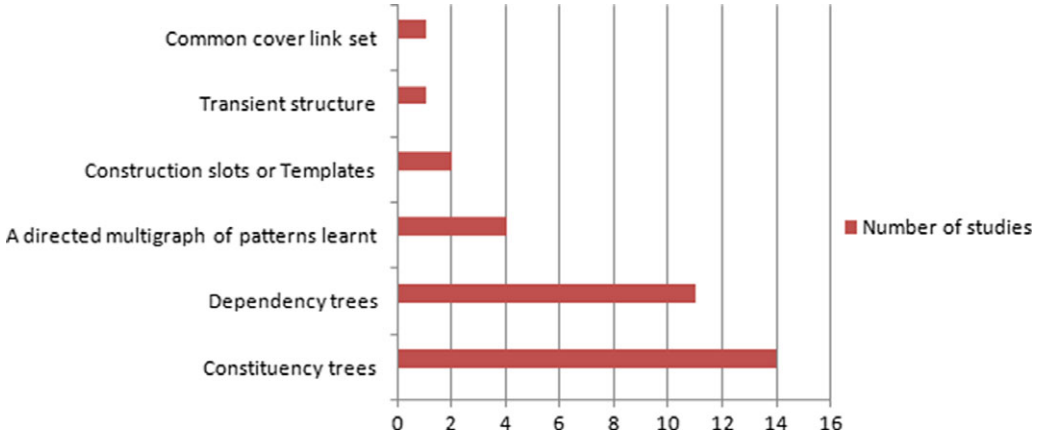

Figure 8. Output representations.

implementations adopting this output structure are compatible with functional-cognitive school of thought even though the authors themselves have used syntactic and semantic parse information to demonstrate that construction grammars are indeed learnable. More about the learning method and its relevance for future research is discussed in Section 4.

In computational linguistics, it is common to view both linguistic production and parsing in terms of a chain of consecutive operations over the linguistic structure. This chain of operations is called a transient structure in fluid construction grammar. There was one study in our review which used transient structure as its output representation after parsing (Marques and Beuls 2016). An example of a transient structure is shown in Figure 6.

Seginer (2007) introduced a representation of syntactic structure that is similar to dependency structure but suitable for incremental parsing. It is based on links between a pair of words and it defines the shortest common cover link sets for a given utterance and its bracketing. An example of the shortest common cover link set for a sentence is given in Figure 7. Although it looks like a dependency structure, there are two differences. The first difference is in the linking of the noun phrase 'the boy' where the link goes back and forth between the two words, which is not the case in a dependency tree. The second difference is based on the property called adjacency, which makes connection between 'know' and 'the'. Such a connection does not occur in a standard dependency tree.

The distribution of the output representations for the 33 studies is shown in Figure 8 and Table 3. It can be seen from this distribution that overwhelmingly 25 studies out of 33 represent their outputs in the form of constituency or dependency trees.

2.6.5 Evaluation strategies

The following evaluation strategies have been extrapolated from the 33 studies considered:

A. Comparison against the gold standard.
B. Performance of the learner in grammaticality judgement tasks.

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

C. Comparison of models trained on disjoint corpora with respect to sentence acceptability.
D. Qualitative evaluation.
E. Maximum coverage, minimum size and stability measures for a usage-based grammar.
F. Output is compared against an artificial grammar that generated the test corpus.

Comparing the system output against the gold standard is a widely used method for evaluating performance of NLP systems. In the context of formal grammars, this strategy requires a treebank, a parsed corpus in which each sentence is manually annotated with syntactic relations to be used as the gold standard. Here, the treebank annotations need to be compatible with the system output, for example, if the system learns a directed dependency tree as its output representation, then the gold standard also needs to be annotated with the directed dependencies. Manual annotation involves a variety of activities such as defining an annotation schema, writing annotation guidelines, training experts for the annotation task and achieving consensus (Neves and Ševa 2019). Consequently, this time-consuming and labour-intensive process may create a bottleneck in this evaluation strategy when application-specific annotations or data are required. In addition, given the inductive nature of usage-based grammars, the use of a gold standard, which inevitably reflects the theoretical commitments of human annotators, seems counter-intuitive (Clark and Lappin 2010). Nonetheless, we observed that the majority of practical implementations (25 out of 33) were evaluated against the gold standard treebank. This was facilitated by the fact that most methods considered represented their outputs as either constituency or dependency trees for which the existing treebanks such as Penn Treebank or Prague English Dependency Treebank can be re-used.

Alternatively, a grammar induction system can be evaluated by testing how well it performs in grammaticality judgement tasks. For example, Goteborg multiple choice test consists of 100 sentences, each containing an open slot. There would be three choices for each question and one of them should be filled in the open slot for the sentence to make the sentence grammatical. If a system has learned the grammar of a language, the open slot would be filled with the correct option. Solan *et al.* (2004) and Edelman *et al.* (2005) subjected their systems to such tasks in the form of multiple choice questions used in English as Second Language (ESL) classes. The authors demonstrated that as the number of sentences processed by the system increased so did the proportion of multiple choice questions answered correctly. The performance of the system proposed by Solan *et al.* (2004) was also compared against that of a bigram language model, that is, a language model where probability distribution of every two-word sequences in a text is calculated. The ADIOS model outperformed the bigram model by answering 60% of questions correctly which is equivalent to the average score of 9th grade ESL students. The bigram precision benchmark is 45%. This is an extrinsic evaluation of the grammar induction system by looking at its performance in the context of its application. Unlike the gold standard evaluation, this pragmatic approach does not constrain the system by conformance to the existing theoretical frameworks and in that respect seems to be better aligned with the usage-based nature of grammars considered.

Similarly, in an attempt to remove the constraint of *a priori* imposed grammatical structures, Brodsky and Waterfall (2007) base their evaluation on an assumption that it is highly unlikely for two highly constrained models trained on disjoint corpora to coincidentally agree on whether a sentence is grammatically acceptable or not. This idea is similar to that of inter-annotator agreement, where, in the absence of ground truth, high reliability implies validity. In other words, if two independently produced outputs agree, then it is considered to be valid. The authors have proposed a precision-testing scheme based on the above observation. We, however, suggest making use of existing inter-annotator agreement measures, which take into account chance agreement.

Whereas all of the above evaluation strategies use quantitative measures of performance, Dunn (2017b) proposes a qualitative assessment of representative examples of constructions from various subsets of corpus in addition to the quantitative evaluation metrics. Representative examples of constructions learnt by the system could be as follows: *[Wh-Determiner] + [Modal] + "be" +*

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

[Past-Participle] is a productive schematic representation generalised from multiple usages in the corpus such as 'that will be provided', 'that can be played', 'which will be represented', 'that should be made'. Qualitatively, one can verify that this schema covers relative clauses with passive verbs that generalises to many complementisers and modal verbs. However, as we can see, the above schema does not cover relative clauses with various tense forms. In this manner, many other schematic representations, such as *"to"* + *[Verb]* + *[Determiner]* + *[Noun]*, *[Noun]* + *[Preposition]* + *[Determiner]* + <*religion*>, *[Preposition]* + *"the"* + <*location*>, can be analysed from various subsets of the corpus to see how good are they close to speaker intuitions and what type of generalisations these schemas represent. The problem with qualitative evaluation is obviously the human effort involved in manually checking the representative construction patterns from various subsets of corpus. Also, ascertaining what constitutes a reasonably good number of representative examples for evaluation poses a problem. The question of the psychological validity of the schemas arises when an individual evaluates the constructions.

Subsequently, the same author presented their quantitative evaluation metrics to evaluate the construction learner (Dunn 2017a, 2017b). They proposed 14 multi-unit association measures whose frequency and association strength can be used to quantify a model of constructions. From many potential construction grammars that could be learnt, they used the degree of coverage and the size of grammar to choose an optimal grammar.

Finally, a Context Free Grammar (CFG) learnt by the system can be compared against the original CFG used to generate the test sentences. Edelman *et al.* (2005) evaluated their system using this method. A handcrafted CFG was used to generate a corpus of sentences, a large portion of which is used for training and the other smaller portion is used for testing. The system learns a set of patterns which can be equivalently represented as context-free rules. The rules learnt by the system are compared against the manually defined rules. This approach suffers from a similar drawback as the gold standard approach. The idea of developing a grammar to generate a large gold standard automatically avoids the drawbacks of generating gold standard annotations and can measure the degree to which unsupervised approaches can recreate complex models. However, any such grammar will always be an approximation and as such may not be representative of a language studied. Even more so, knowing that natural languages are mildly context-sensitive (Joshi 1985), the use of CFGs for this purpose may not be sufficient.

Comparison of systems based on their performance should consider the differences in datasets used for evaluation. There are distinctions between evaluations done on artificial datasets produced with handcrafted CFG rules, sentences used in English as a Second Language (ESL) multiple choice questions, Child Language Data Exchange System (CHILDES) datasets and broad coverage datasets. For grammar induction, this determines how hard the task may be. Table 4 shows the factors that are relevant to apply an evaluation strategy and marks which factors are applicable to which methods. The distribution of the number of papers adopting these different methods of evaluation in their work is shown in Figure 9.

2.6.6 Features used for learning

A feature in machine learning is defined as some characteristic of the problem studied. In linguistic applications of machine learning, features can be words themselves or their properties, which can be provided *a priori* (e.g., gold standard annotations or lexica), engineered by applying domain knowledge (e.g., named entity formation patterns) or inferred automatically (e.g., stem, embedding, etc.). Different studies used various features to learn the grammatical structure. As the focus of this review is on unsupervised approaches, we are interested primarily in those features that can be extracted from the raw data automatically, possibly by external tools, and subsequently utilised by a learning algorithm. For example, POS tags were frequently provided as features for the grammar inducers before learning (Klein and Manning 2004; Dennis 2005; Bod 2006; Zuidema 2006; Reichart and Rappoport 2008; Bod 2009; Headden *et al.* 2009; Snyder *et al.* 2009; Araujo and

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

Table 4. Factors relevant for evaluation

	Evaluation strategies					
Property	E1	E2	E3	E4	E5	E6
Intrinsic evaluation	√			√	√	√
Extrinsic evaluation		√	√			√
Manual labour	√			x		√
Multiple correct solutions		√	√	√	√	

E1 – Comparing against gold standard treebank annotation.
E2 – Performance of the learner in grammaticality judgement tasks.
E3 – Train two highly constrained models on disjoint corpora and evaluate how they agree or disagree on sentence acceptability.
E4 – Qualitative evaluation.
E5 – Maximum coverage, minimum size and stability measures for a usage-based grammar.
E6 – Agreement with CFG grammar that generated the learning data.

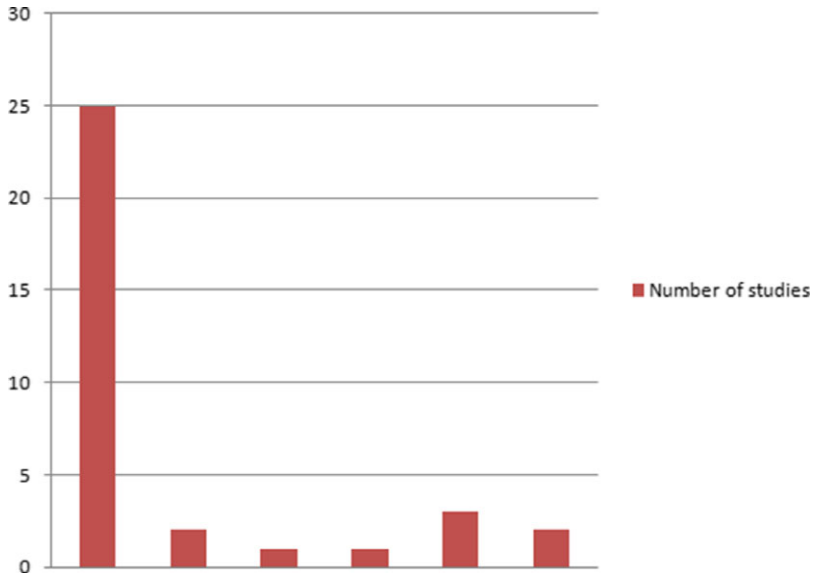

Figure 9. Evaluation strategies.

Santamaría 2010; Sangati 2010; Santamaria and Araujo 2010; Spitkovsky *et al.* 2010; Boonkwan and Steedman 2011; Dominguez and Infante-Lopez 2011; Gillenwater *et al.* 2011; Mareček and Žabokrtský 2012b; Dunn 2017a; Dunn 2017b). Rarely are such features extracted by the induction model itself. For instance, Jin *et al.* (2018) described a Bayesian Dirichlet model of depth-bounded PCFG induction where the model induces the categories for constituents and tree structures from the raw text provided as input.

Of course, richer linguistic features are likely to result in better grammar induction models. Spitkovsky *et al.* (2011) use punctuation as a feature to improve the standard dependency model with valence (DMV) and in 2012 the same authors (Spitkovsky *et al.* 2012) propose that capitalisation cues can improve dependency grammar induction. The general idea behind these implementations is that orthographic cues and their boundaries can help improve the grammar induction performance. In this way, we have identified eight types of commonly used features:

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

Table 5. Features used in various studies

Feature	Studies which use the feature in their implementation
POS tags	Klein and Manning (2004); Dennis (2005); Bod (2006); Zuidema (2006); Reichart and Rappoport (2008); Headden *et al.* (2009); Snyder *et al.* (2009); Araujo and Santamaría (2010); Sangati (2010); Santamaria and Araujo (2010); Spitkovsky *et al.* (2010); Boonkwan and Steedman (2011); Dominguez and Infante-Lopez (2011); Gillenwater *et al.* (2011); Mareček and Žabokrtský (2012a, 2012b); Marques and Beuls (2016); Dunn (2017a, 2017b)
Distributed representation of words	Adriaans *et al.* (2000); Klein and Manning (2004); Brooks (2006); Seginer (2007); Søgaard (2011)
Head, valence and direction of attachment	Klein and Manning (2004); Reichart and Rappoport (2008); Headden *et al.* (2009); Sangati (2010); Spitkovsky *et al.* (2010); Boonkwan and Steedman (2011); Dominguez and Infante-Lopez (2011); Gillenwater *et al.* (2011); Mareček and Žabokrtský (2012a, 2012b)
Word alignments from parallel corpus	Snyder *et al.* (2009)
Chunks or bracketed structures	Ponvert *et al.* (2011)
Orthographic cues	Spitkovsky *et al.* (2011, 2012)
Heuristic rules	Brooks (2006); Araujo and Santamaría (2010); Santamaria and Araujo (2010); Marques and Beuls (2016)
Construction association measures	Dunn (2017a, 2017b)

A. POS
B. Distributional representation of words
C. Head, valence and direction of attachment
D. Word alignments in parallel corpora
E. Chunks
F. Orthographic cues
G. Heuristic rules
H. Construction association measures

Most features are straightforward and are used in various studies irrespective of other aspects of implementation. *POS tags* indicate the grammatical category of the words in a text. For example, the sentence 'The man cut an apple with a knife' can be tagged as follows: 'The/DT man/NN cut/VBD an/DT apple/NN with/IN a/DT knife/NN./.', where the POS tags DT, NN, VBD and IN indicate the grammatical categories *determiner, noun, verb* and *preposition*, respectively. These tags are taken into account when dividing a sentence into non-overlapping regions of text called *chunks*, which are typically non-recursive. For example, the same sentence can be chunked as *[The man] [cut] [an apple] [with a knife]*.

POS tags were used as features in 18 studies (shown in Table 5). Ponvert *et al.* (2011) used chunking based on a probabilistic finite state model such as Hidden Markov model (HMM). The chunker is cascaded to achieve constituent parsing. Head, valence and direction of attachment are the parameters that are used in generative models of dependency parsing introduced by Klein and Manning (2004) and used in nine other studies. Word alignment refers to the task of extracting translation equivalents of words from sentence-aligned bilingual corpora, that is, corpora where the same set of sentences in the source language are aligned with the same sentence in a target language. Word alignments are typically used as features in machine translation. In our review, we observed that this feature is used for grammar induction by Snyder *et al.* (2009).

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

Distributional representation of words gives a probability distribution of a word occurring in a context when a centre word is given. The distributed representation of words is used in five studies (Adriaans *et al.* 2000; Klein and Manning 2004; Brooks 2006; Seginer 2007; Søgaard 2011). Orthographic cues such as capitalisation and punctuation are used as features in two studies (Spitkovsky *et al.* 2011, 2012).

Features that we have listed as heuristic rules and construction association measures need explanation. There were four studies (Araujo and Santamaría 2010; Santamaria and Araujo 2010; Boonkwan and Steedman 2011; Marques and Beuls 2016) which used different features: (a) identifying a set of POS tags as separator tags and using these separator tags as features to identify phrase structure boundaries, (b) encoding prior linguistic knowledge as a small number of syntactic prototypes and using them as features and (c) use handcrafted rules to learn artificial construction grammar. We see that these different features can be generalised as heuristics exploited by the authors based on their observations of linguistic patterns. We classified these different implementations as using heuristic rules as features of grammar induction. There were two studies (Dunn 2017a, 2017b) which used 14 different association measures to identify what sequence of construction patterns qualify as actual constructions rather than being potential constructions. These two studies use construction association measures as features for their learning in addition to other features.

We noticed that POS tags, morphological and orthographic cues are very generic features that are not tied to any particular implementation methodology. Although a fully unsupervised grammar induction uses no syntactic information, POS tags are typically used in most implementations (18 out of 33 studies). Orthographic cues such as capitalisation and punctuations are easier to obtain from a large corpus which can then be used in conjunction with other features to improve the overall accuracy. However, orthographic cues obtained from English are not readily transferred to languages such as Thai which do not mark punctuations or word boundaries consistently and even sentence boundaries are not indicated by any discernible cue like full stops. Another observation is that a purely distribution-based approach to grammar induction does not generalise to effectively learn syntax from raw text. Brooks (2006) showed that using a method of attachment to form constituents is more effective than distribution analysis alone. Other features such as valence (number and types of arguments taken by a word linguistically), direction of attachment, word alignments, and chunks are again used to learn constituency and dependency trees. The heuristics such as separator tags and sub-separator tags proposed by Araujo and Santamaría (2010) can be automatically extracted from the corpus and can improve constituent parsing. Table 5 shows the list of features and the studies which use those features.

2.6.7 Methodologies

Finally, we generalise the methodologies used to support grammar induction into the following categories:

A. Top-down dependency grammar models
B. Similarity or exemplar-based models
C. Distribution-based models
D. Clustering
E. Heuristic approaches
F. ADIOS
G. PCFG
H. Chunkers and their extensions
I. DOP
J. Construction grammar induction

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

Let us first define these categories. Top-down dependency grammar models start with a head and generate a sequence of arguments left and right conditioned on the head, valence and direction of attachment. As already discussed in Section 2.6.1, 10 studies incorporated generative dependency model or its variation in their implementations (Klein and Manning 2004; Sangati 2010; Spitkovsky *et al.* 2010, 2011, 2012; Boonkwan and Steedman 2011; Dominguez and Infante-Lopez 2011; Søgaard 2011; Mareček and Žabokrtský 2012a, 2012b). One of the successful baselines for unsupervised parsing is the DMV model of Klein and Manning (2004). In exemplar-based approaches, the parse of a sentence is viewed as a set of alignments with exemplars from memory. Approximately nearest-neighbour exemplars and their parse are exploited in identifying the parse of the target sentence. Two studies followed this similarity or exemplar-based implementation methodology (Dennis 2005; Snyder *et al.* 2009).

Distribution-based models use word vectors to derive the parse. There were three studies which employed these in their implementations (Brooks 2006; Seginer 2007; Søgaard 2011). Clustering approaches in general involve dividing the data points into groups where members in each group are more similar to one another than to those in the other group. In NLP, clustering typically involves grouping of grammatical elements of a sentence into groups such as POS clusters, morph clusters, and so on. There are two studies in our review which involved clustering as a part of grammar induction (Adriaans *et al.* 2000; Reichart and Rappoport 2008).

Heuristic approaches employ practical methods which yield near-optimal solutions to a problem where a perfectly acceptable optimal solution is not available or too difficult to obtain. Examples of heuristic methods are educated guesses, intuitive judgements, rule of thumb, etc. In our review, we identified two studies which involve heuristic methods to recognise phrase boundaries to form bracketed structures (Araujo and Santamaría 2010; Santamaria and Araujo 2010). ADIOS is a method which distils structural regularities and contextual cues from raw text, identifies equivalence classes of usage patterns from these structural regularities and represents this learning as a directed multigraph. This multigraph can be equivalently represented as context-free rules which represent the grammar learnt by the system. This methodology was used in four studies in our review (Solan *et al.* 2004; Edelman *et al.* 2005; Berant *et al.* 2007; Brodsky and Waterfall 2007).

PCFGs are context-free production rules that generate valid strings in a language. It is a type of formal grammar which is learnt by three studies in our review (Adriaans *et al.* 2000; Headden *et al.* 2009; Jin *et al.* 2018). Chunkers and their extensions begin with learning bracketed structures or shallow parsing and subsequently proceed to use them to induce the full grammar. There was one study in our review that cascades chunkers with HMM to achieve constituent parsing (Ponvert *et al.* 2011). Next method is DOP which is an interesting way to approach grammar. Usually, statistical methods of grammar induction start with a *predefined* grammar and use the corpus to estimate rule probabilities. In DOP, no prior grammar is assumed but uses corpus fragments to induce grammar. DOP models sentence structures based on the observed frequencies of sentence fragments without imposing any constraint on the size of such fragments. There were four studies which implemented grammar induction bottom up using DOP method (Bod 2006, 2009; Zuidema 2006; Post and Gildea 2013).

Finally construction grammar induction is a fully usage-based approach to grammar which learns form-meaning pairings called constructions. Constructions are mappings between linguistic form and linguistic function which can be from any level of generalisation such as at the lexical level, phrase level, semantic level, and so on. Learning a construction grammar from raw corpus is difficult because constructions can be of any size and they can be internally complex. Dunn (2017a) demonstrates that construction grammars are learnable and falsifiable by providing a model of learning an optimal construction grammar from a raw corpus. The same author (Dunn 2017b) presents a detailed implementation and evaluation method of the CG implementation. The distribution of these 10 methods from our list of 33 implementations is shown in Figure 10 and Table 3.

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

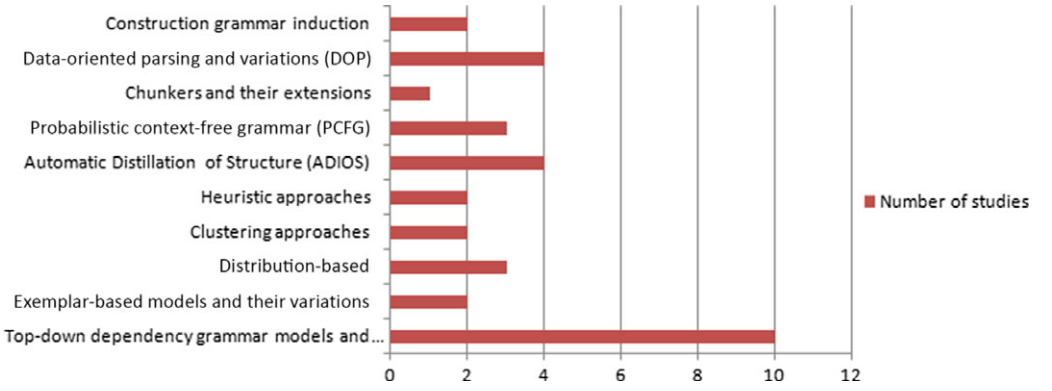

Figure 10. Grammar induction methods.

Among these methods, we identified that the top-down dependency grammar models and the PCFGs are used in multiple studies (13 out of 33) to learn formal dependency and constituency trees, respectively. As far as usage-based automatic grammar induction was concerned, we recognised four methodologies as relevant and useful: ADIOS, DOP and construction grammar induction. ADIOS is shown to be successful for inducing patterns from short sentences but exhibits limitations in parsing complex sentences. The ADIOS method can be successfully applied once the complex sentences are split into multiple simple ones. DOP works like an analogy-based pattern-matching technique which analyses a sentence by looking up the nearest-neighbour constituent structure already stored in memory. The nearest-neighbour for a given analysis is defined as the constituency tree derivation which shares the largest number of common nodes. DOP takes a corpus of sentences and allows all possible subtrees to be built from a sentence and chooses an optimal parse by computing the tree which leads to the shortest derivation. DOP is good but it is computationally expensive because it allows all possible subtrees theoretically. Although the subsequent DOP models reduced this computational space significantly in comparison to the general-case DOP, the complexity increases with sentence length. The core idea from DOP that abstract linguistic rules such as movement, agreement and discontiguous phrases can be learnt from recursive tree structures and an analogical match algorithm is powerful. However, the analogical matching that DOP employs starts with the assumption of constituent tree structures whose internal nodes can be substituted with analogous candidates. How to adapt DOP ideas without assuming constituent tree structures *a priori* is open for future exploration.

Among the studies considered, we noticed that construction grammar induction is the only method which demonstrated a fully usage-based, bottom-up approach to grammar induction successfully. The evaluation strategies are also discussed in detail. Because the outputs are not the usual tree structures, but construction slots, evaluation against a gold standard is not possible. The qualitative and quantitative evaluation measures (construction association measures) discussed in these implementations can be adopted for usage-based implementations. Apart from these 33 implementation papers, there were 7 theoretical and experimental studies and there were 3 studies that provided surveys on grammar induction or parser evaluation. We will present our findings from these studies in the next section.

3. Findings from non-implementation studies

We will now present the information and insights obtained from three non-implementation papers. D'Ulizia *et al.* (2011) present a survey of the various techniques of parsing from the literature (Lawrence, Giles, and Fong 2000). Included in the survey are 14 parser implementations with 6 computational techniques (statistical methods, evolutionary computing techniques, minimum

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

description length, heuristic methods, greedy search and clustering techniques), two presentation sets (text and informant), three types of information for learning (supervised, semi-supervised and unsupervised), three types of grammar evaluation techniques (looks-good-to-me, compare against treebank and rebuilding known grammars).

Cramer (2007) experimentally demonstrates the limitations of existing grammar induction algorithms and suggests that an incremental grammar induction is preferable to the conventional methods. A short illustration of such a system is also presented by the authors. Rimell, Clark, and Steedman (2009) bring into question the suitability of PARSEVAL measures (the standard metrics for parser performance) as reflecting the parser performance. As an alternative to the exclusive focus on incremental improvements in measures of overall accuracy such as PARSEVAL, the authors suggest a more focused parser evaluation methodology (e.g., construction-focused evaluation) as a way of improving parsing technology. These insights are strikingly significant in the light of the data that we synthesised from the 36 implementations. Most of these studies evaluated their parser output in terms of the PARSEVAL measures by comparing against gold standard output. In future, there is more scope to explore better methods and metrics for evaluation of grammar induction systems.

From the other seven theoretical studies included in our literature review, the following insights are obtained.

- Saffran (2001) conducted an experiment to verify if abstract grammatical hierarchies can be learnt using the statistical clues of local predictive dependencies. In an artificial language learning task, the adult participants were exposed to artificial language with no semantic, visual or any other clue except distribution of words and their category. The results support the hypothesis that learners can detect predictive dependencies and use it to acquire simple phrase structures. This suggests that phrase structure boundaries can be induced in an unsupervised way based on predictive dependencies bottom up with simple categories and statistics.
- Usually, discourse information (units of language beyond the level of a sentence, e.g., topic, old information, new information, focus) is not taken into consideration while modelling computational grammar induction. Kaiser and Trueswell (2004) show that the processing of non-canonical structures is facilitated by the presence of an appropriate discourse context.
- Productivity in grammar is usually treated in terms of recursion and hierarchy. Frank *et al.* (2012) suggest that grammar is better modelled sequentially and incrementally. The authors discuss evidence from multiple research fields such as cognitive neuroscience, psycholinguistics, computational models of language acquisition and argue that combining productive grammatical units need not happen hierarchically but a sequential process would suffice.
- Two studies suggested that abstract linguistic properties (subjacency constraints, movement and agreement) can be explained using analogy-based statistical models and constraints on sequential processing (Ellefson and Christiansen 2000; Bod 2007). Bod (2007) showed that learning discontiguous construction patterns, agreement and movement were possible for computational bootstrapping without a need for special, top-down universal grammar. Ellefson and Christiansen (2000) conducted two types of experiments (artificial language learning experiment and connectionist simulation) to enquire into the nature of subjacency constraints which are usually explained as part of universal grammar. Their results suggest that constraints arising from general cognitive processes, such as sequential learning and processing, are likely to play a larger role in sentence processing than has traditionally been assumed.
- Yang (2011) proposes a statistical test to check if grammar is abstract and productive or lexically specific and usage-based. Results of the statistical test show that a lexically specific, usage-based and memory-and-retrieval approach is unsupported. This result is consistent with a productive abstract grammatical system in child speech.

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

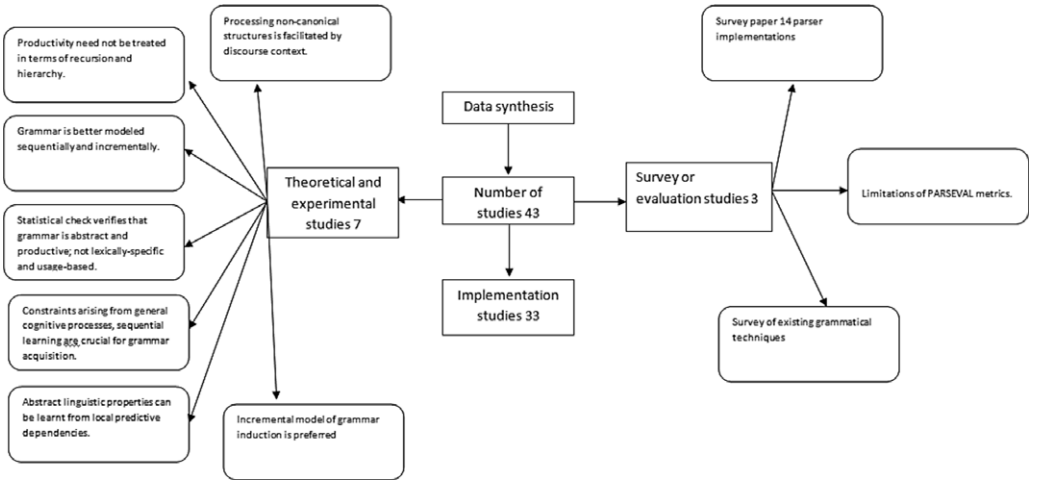

Figure 11. Overview of the findings from literature review.

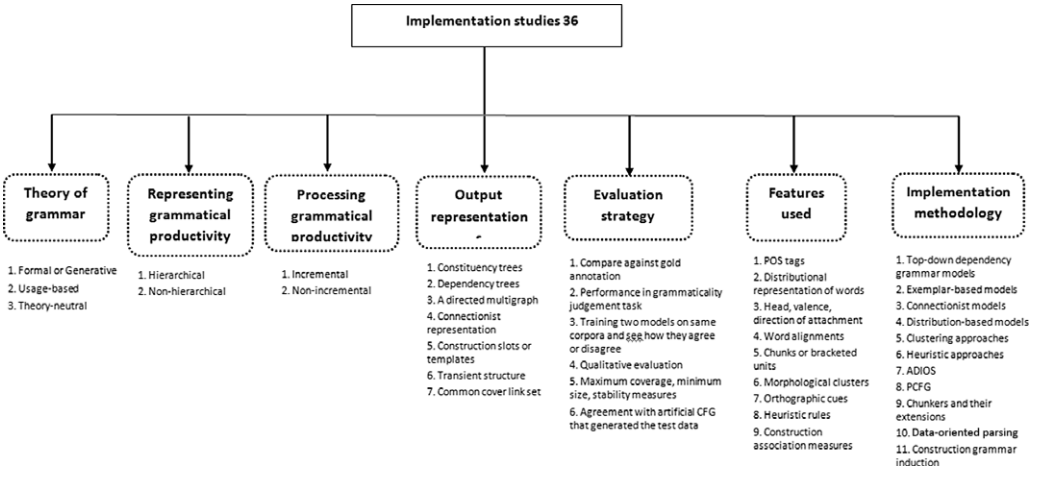

Figure 12. Findings from the implementation studies.

Six studies out of seven from our list support the view that a sequential, statistics-based and bottom-up induction of grammar explains the grammar acquisition and processing by humans. The statistical test by Yang (2011), in contrast, supported a productive, abstract grammatical system governing the child speech rather than a usage-based system. The study shows that a mere usage-based, lexically specific schema could not statistically explain the child speech data. This statistical test, however, does not resolve the innateness debate in language acquisition. It merely points out that an abstract, productive system should be in place at a very early age in the child's speech.

The summary and discussion of all these studies considered in the systematic review are presented in the next section.

4. Summary and discussion

Specific findings discussed in the previous section are summarised in Figures 11 and 12. The findings from this review suggest that many aspects of unsupervised induction of usage-based

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

grammars are yet to be fully explored. Only two studies (Dunn 2017a, 2017b) focused on computational learning of a construction grammar. They used POS tags, dependency relations and semantic labels to learn a construction grammar from a large corpus. Though these components can also be learnt in an unsupervised fashion, further research is needed to establish the impact of their accuracy on the overall grammar induced. The properties of construction slots cannot be evaluated properly by comparison against a gold standard as we explained earlier. A practical implementation of usage-based grammar induction should embed means of evaluating the grammar itself. The two studies mentioned demonstrated the feasibility of unsupervised construction grammar learning as well as its falsifiability and proposed a set of practical evaluation methods. They provide a baseline for further research into fully unsupervised parsing approaches. We will outline the approach of these studies, illustrate what challenges still remain to be addressed and underline the future directions of research.

In Dunn's (2017a, 2017b) implementations, upper-case 'Grammar' refers to the domain-independent model that learns grammatical generalisations from linguistic input. Lower-case 'grammar' refers to a particular inventory of grammatical generalisations learnt for a specific language (i.e., a large corpus of sentences). For example, given a sentence such as 'Bill gave his neighbour a piece of his mind', there are multiple potential linguistic generalisations possible. A few possibilities are shown below but there could be many other possible generalisations.

- *Sentence*: Bill gave his neighbour a piece of his mind
- *Unit-based syntactic generalisation*: [NN – VB – PRN – NN – DT – NN – PREP – PRN – NN]
- *Constituent-level syntactic generalisation*: [NP – VB – NP – NP]
- *Semantically constrained generalisation*: [NP <ANIMATE> – VB <TRANSFER> – NP <ANIMATE> – NP]
- *Lexically fixed idiomatic usage*: [NN – "give" – NP <ANIMATE> – "a piece of" – NP <ANIMATE> – "mind"]

Similarly, for all the sentences in the corpus, there are multiple potential linguistic generalisations. Of all the possible potential construction units that can generalise the sentence, one is chosen as the actual construction. An inventory of such constructions induced from each sentence in a corpus is the lower-case 'grammar' learnt from the corpus. The model which learns these constructions is the upper-case 'Grammar'. In the two studies, the authors provide a reproducible model that distinguishes the potential constructions from actual constructions. Their algorithm provides two insights: (a) constructions are meaningful symbolic units and (b) co-occurrence and distribution are indicators of meaning. An actual construction differs from all other potential constructions in terms of its being productive (high frequency) and meaningful (presence of significant co-occurrence patterns and association measures). Two types of thresholds are used to distinguish potential constructions from actual constructions: frequency threshold and thresholds on vectors of 14 independent association measures. Using the two types of threshold, the algorithm learns many possible 'grammars' from a given corpus of sentences. From these multiple 'grammars', the algorithm chooses an optimal 'grammar'. An optimal grammar is one which has larger coverage (explains maximum number of linguistic observations) but smaller size (posits as few constructions as possible). The algorithm minimises the objective function of $-log(C^2/S)$ to achieve this effect. In summary, these implementations are focused on building a computational model that learns an optimal set of mappings between individual sentences and their meaningful linguistic generalisations, with maximum coverage and minimum size.

These two studies relate to our research goal in the following manner: there are two components to 'Grammar' in the sense described earlier: (a) inventory of construction schemas that generalise the patterns of usage from a corpus of sentences and (b) an assembly of these construction schemas to form incrementally larger units. Dunn's studies were interested in the first part

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

with focus on learnability of construction grammars. The second part is a potential research area that is yet to be explored. This involves identifying a method which learns construction patterns inductively and assembles those patterns in an incremental fashion. We would like to explore how an assembly of construction schemas can be achieved in an incremental fashion and use this for unsupervised grammar induction.

The analysis and discussion made so far connect to the research questions in the following manner. The research questions 1 to 4 are answered by various subsections in the data synthesis part discussed earlier. We will now discuss how the different implementations compare with one another in terms of their performance. The implementations discussed in our review differ in their methodology, theoretical influence and various other factors. However, studies with similar evaluation strategy can be compared against one another to identify the range of performance in unsupervised parsing. We also would like to see which evaluation strategies can be adapted for parsing inspired by functional-cognitive school of thought. The most common evaluation method found in our review was to compare the system output against a gold standard annotation and report the precision, recall and *F*-measure of the system. There were 25 studies which used this method to evaluate their systems. Not all of them are comparable. One way we tried to compare the systems is if they used similar output representations and then compared their evaluation metrics. There were 14 studies which represented their output as constituency trees and evaluated against the treebank annotation. Three studies out of 14 reported their *F*-measure results by testing their output against on treebanks from multiple languages. The best performing system out of these three reports the *F*-score of 82.9, 67.0 and 47.2 for English, German and Chinese, respectively (Bod 2006). It was consistently observed that the systems performed better for English and the lowest scores were for Chinese in the other two systems. There was one system which learnt a subclass of shallow context-free languages. Here, the evaluation metrics used was the number of syntactic types and clusters learnt by the system. They reported that 8000 syntactic types and clusters were learned from a dataset of 2000 sentences. It could not be compared with any other system. The remaining 11 studies (out of 14) reported their Unlabelled Precision (UP), Unlabelled Recall (UR) and F-measure for English. They could be directly compared. The lowest accuracy was reported by Brooks (2006) with UP 33.6%, UR 14.1% and *F*-measure 19.8%. This system used a distributional approach to grammar induction and simple heuristics to reduce the number of candidate constituents using alignment patterns. The results suggest that distributional methods do not generalise enough to learn syntax effectively from raw text, but that attachment methods are more successful.

There were 11 studies which evaluated their dependency output against a treebank. Three of them evaluated their result in multiple languages. Most studies reported their results as Labelled Attachment Scores (LAS) and Unlabelled Attachment Scores (UAS). LAS were generally lower than the UAS in all the studies. The UAS reported in these systems were typically low ranging from 38.3% to 64.5%. The accuracy of these unsupervised parsers in comparison with the state-of-the art results in supervised parsing is unsurprisingly lower. For instance, Stanford supervised and semi-supervised dependency parsers report accuracy in the range of 95%–97% (Chen and Manning 2014; Kiperwasser and Goldberg 2016). The other studies in our review have different evaluation methods distinct from each other and their performances cannot be compared directly. The frequency and association measures of co-occurrence used in Dunn (2017a, 2017b) to distinguish potential constructions from actual constructions have the potential to be adapted for future research as well.

5. Conclusion

We conducted a systematic literature review of the existing approaches to unsupervised grammar induction. Most commonly, these approaches modelled a grammar as a formal, top-down system. Consequently, when such grammars were used to support parsing, its outputs were represented as

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

a constituency tree or dependency tree. In this case, the parsing results can be compared against a treebank. Here, the correct (or acceptable) parsing is determined *a priori* based on existing grammars.

Alternatively, theoretical studies suggested that an incremental, sequential and usage-based approach to grammar induction that takes discourse information and local dependencies into consideration can model the same task. It is as if the theoretical or experimental studies are the antithesis of the computational implementations included in our review. This gap between the two should be addressed to enable further progress in grammar induction and evaluation. More work on construction grammar induction should be done to test the current theories. The most challenging aspect about a truly bottom-up, usage-based approach to grammar induction is that there are usually multiple ways to generalise usage patterns bottom up. Learnability and falsifiability of usage-based patterns and evaluation strategies for the output in the absence of a gold standard are the major challenges. There were just two studies that demonstrated that unsupervised construction grammars are learnable from a raw corpus and they can be evaluated for optimality by maximising the degree of coverage and minimising the size of the grammar learnt by the system. It was, however, observed that these implementations do not learn usage patterns incrementally but rather choose sequences of construction slots as generalisations for the sequence of words in a sentence. There is scope for future research to explore how to assemble usage patterns in an incremental way and how such a system can be evaluated.

Acknowledgement. The research on which this article is based is funded by the College of Arts, Humanities and Social Sciences (AHSS) at Cardiff University, and forms part of the UK Economic and Social Research Council (ESRC) and Arts and Humanities Research Council (AHRC) *Corpws Cenedlaethol Cymraeg Cyfoes (The National Corpus of Contemporary Welsh): A community-driven approach to linguistic corpus construction* project (Grant Number ES/M011348/1).

References

Adriaans P., Trautwein M. and Vervoort M. (2000). Towards high speed grammar induction on large text corpora. In *International Conference on Current Trends in Theory and Practice of Computer Science*. Berlin, Heidelberg: Springer, pp. 173–186.

Araujo L. and Santamaría J. (2010). Evolving natural language grammars without supervision. In *Evolutionary Computation (CEC), 2010 IEEE Congress on (pp. 1–8)*. IEEE.

Bates E. and McWhinney B. (1982). Functionalist approaches to grammar.

Berant J., Gross Y., Mussel M., Sandbank B., Ruppin E. and Edelman S. (2007). Boosting unsupervised grammar induction by splitting complex sentences on function words. In *Proceedings of the Boston University Conference on Language Development*.

Bloom L., Hood L. and Lightbown P. (1974). Imitation in language development: if, when, and why. *Cognitive Psychology* **6**, 380–420.

Bloomfield L. (1962). *Language. 1933*. Holt, New York.

Bod R. (2006). An all-subtrees approach to unsupervised parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 865–872.

Bod R. (2007). A linguistic investigation into unsupervised DOP. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*. Association for Computational Linguistics, pp. 1–8.

Bod R. (2009). From exemplar to grammar: a probabilistic analogy-based model of language learning. *Cognitive Science* **33**, 752–793.

Boonkwan P. and Steedman M. (2011). Grammar induction from text using small syntactic prototypes. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 438–446.

Brin S. and Page L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems* **30**, 107–117.

Briscoe T. and Waegner N. (1992). Robust stochastic parsing using the inside-outside algorithm. In *Proc. of the AAAI Workshop on Probabilistic-Based Natural Language Processing Techniques*, pp. 39–52.

Brodsky P. and Waterfall H. (2007). Characterizing motherese: on the computational structure of child-directed language. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 29.

Brooks D.J. (2006). Unsupervised grammar induction by distribution and attachment. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pp. 117–124.

Chen D. and Christopher M. (2014). A fast and accurate dependency parser using neural networks. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chomsky N. (1957). *Syntactic Structures*. The Hague: Mouton.

Chomsky N. (1965). *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.

Chomsky N. (1968). *Remarks on Nominalization*. Linguistics Club, Indiana University.

Chomsky N. (2014). *Aspects of the Theory of Syntax*, vol. 11. MIT Press.

Clark A. and Lappin S. (2010). *Linguistic Nativism and the Poverty of the Stimulus*. John Wiley & Sons.

Cocos A., Masino A., Qian T., Pavlick E. and Callison-Burch C. (2015). Effectively crowdsourcing radiology report annotations. In *Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis*, pp. 109–114.

Cramer B. (2007). Limitations of current grammar induction algorithms. In *Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop*. Association for Computational Linguistics, pp. 43–48.

Dalrymple M. (2001). *Lexical Functional Grammar*. Brill.

Dennis S.J. (2005). An exemplar-based approach to unsupervised parsing.

Dik S. (1987). Some principles of functional grammar. *Functionalism in Linguistics* **20**, 81.

Dik S. (1991). Functional grammar. Linguistic theory and grammatical description, pp. 247–274.

D'Ulizia A., Ferri F. and Grifoni P. (2011). A survey of grammatical inference methods for natural language learning. *Artificial Intelligence Review* **36**, 1–27.

Dunn J. (2017a). Learnability and falsifiability of construction grammars. *Proceedings of the Linguistic Society of America* **2**, 1.

Dunn J. (2017b). Computational learning of construction grammars. *Language and Cognition* **9**, 254–292.

Dominguez M.A. and Infante-Lopez G. (2011). Unsupervised induction of dependency structures using Probabilistic Bilexical Grammars. In *Natural Language Processing and Knowledge Engineering (NLP-KE), 2011 7th International Conference on (pp. 314–318)*. IEEE.

Edelman S., Solan Z., Horn D. and Ruppin E. (2003). Rich syntax from a raw corpus: unsupervised does it. In *NIPS-2003 Workshop on Syntax, Semantics and Statistics*.

Edelman S., Solan Z., Horn D. and Ruppin E. (2005). Learning syntactic constructions from raw corpora. In *29th Boston University Conference on Language Development*.

Ellefson M.R. and Christiansen M.H. (2000). Subjacency constraints without universal grammar: Evidence from artificial language learning and connectionist modeling. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 22, No. 22.

Evans V. (2006). *Cognitive Linguistics*. Edinburgh University Press.

Falk Y. (2011). *Lexical-Functional Grammar*. Oxford University Press.

Fauconnier G. (1994). *Mental Spaces: Aspects of Meaning Construction in Natural Language*. Cambridge University Press.

Frank S.L., Bod R. and Christiansen M.H. (2012). How hierarchical is language use? *Proceedings of the Royal Society of London B: Biological Sciences*, p.rspb20121741.

Gazdar G., Klein E., Pullum G.K. and Sag I.A. (1985). *Generalized Phrase Structure Grammar*. Harvard University Press.

Gillenwater J., Ganchev K., Pereira F. and Taskar B. (2011). Posterior sparsity in unsupervised dependency parsing. *Journal of Machine Learning Research* **12**, 455–490.

Givón T. (1983). *Topic Continuity in Discourse: A Quantitative Cross-Language Study*, vol. 3. John Benjamins Publishing.

Goldberg A.E. (2003). Constructions: a new theoretical approach to language. *Trends in Cognitive Sciences* **7**, 219–224.

Hajic J., Hajicová E., Panevová J., Sgall P., Bojar O., Cinková S., Fucíková E., Mikulová M., Pajas P., Popelka J. and Semecký J. (2012). Announcing Prague Czech-English Dependency Treebank 2.0. In *LREC*, pp. 3153–3160.

Hampe B. and Grady J.E. (eds.) (2005). *From Perception to Meaning: Image Schemas in Cognitive Linguistics*, vol. 29. Walter de Gruyter.

Harrison C., Nuttall L., Stockwell P. and Yuan W. (2014). Introduction: cognitive grammar in literature. In *Cognitive Grammar in Literature*. John Benjamins, pp. 1–16.

Harrison M.A. (1978). *Introduction to Formal Language Theory*. Addison-Wesley Longman Publishing Co., Inc.

Headden III **W.P., Johnson M. and McClosky D.** (2009). Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 101–109.

Jackendoff R. (1977). X syntax: A study of phrase structure. Linguistic Inquiry Monographs 4. Cambridge, Mass., (2), pp. 1–249.

Jensen K.E. (2014). Performance and competence in usage-based construction grammar. In *Multidisciplinary Perspectives on Linguistic Competences*, pp. 157–188.

Jin L., Doshi-Velez F., Miller T., Schuler W. and Schwartz L. (2018). Unsupervised Grammar Induction with Depth-bounded PCFG. arXiv preprint arXiv:1802.08545.

Jones K.S. (2007). Computational linguistics: what about the linguistics? *Computational Linguistics* **33**, 437–441.

Joshi A.K. (1985). Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In Dowty D.R., Karttunen L. & Zwicky A.M. (eds), *Natural language parsing*, Cambridge University Press, pp. 206–250.

Kaiser E. and Trueswell J.C. (2004). The role of discourse context in the processing of a flexible word-order language. *Cognition* **94**, 113–147.

Kitchenham B. and Charters S. (2007). Guidelines for performing systematic literature reviews in software engineering.

Kiperwasser E. and Yoav G. (2016) Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics* **4**, 313–327.

Klein D. and Manning C.D. (2004). Corpus-based induction of syntactic structure: models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, p. 478.

Król-Markefka A. (2014). Between usage-based and meaningfully-motivated grammatical rules: a psycholinguistic basis of applied cognitive grammar. *Studia Linguistica Universitatis Iagellonicae Cracoviensis* **131**, 43.

Lakoff G. and Johnson M. (1980). Conceptual metaphor in everyday language. *The Journal of Philosophy* **77**, 453–486.

Lakoff G. (1988). Cognitive semantics. *Meaning and Mental Representations* **119**, 154.

Langacker R.W. (1987). *Foundations of Cognitive Grammar: Theoretical Prerequisites*, vol. 1. Stanford university press.

Langacker R.W. (2008). *Cognitive Grammar: A Basic Introduction*. Oxford University Press.

Langacker R.W. (2009). *Investigations in Cognitive Grammar*, vol. 42. Walter de Gruyter.

Lawrence S., Giles C.L. and Fong S. (2000). Natural language grammatical inference with recurrent neural networks. *IEEE Transactions on Knowledge and Data Engineering* **12**, 126–140.

Leech G.N. (1993). *Statistically-Driven Computer Grammars of English: The IBM/Lancaster Approach (No. 8)*. Rodopi.

Levine R.D. and Meurers W.D. (2006). Head-driven phrase structure grammar. *Encyclopedia of Language and Linguistics* **5**, 237–252.

Mareček D. and Žabokrtský Z. (2012a). Unsupervised dependency parsing using reducibility and fertility features. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*. Association for Computational Linguistics, pp. 84–89.

Mareček D. and Žabokrtský Z. (2012b). Exploiting reducibility in unsupervised dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, pp. 297–307.

Marques T. and Beuls K. (2016). Evaluation strategies for computational construction grammars. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1137–1146.

Matthiessen C.M. and Halliday M.A.K. (2009). Systemic functional grammar: a first step into the theory.

Moshier M. (1988). Extensions to unification grammar for the description of programming languages.

Neves M. and Ševa J. (2019). An extensive review of tools for manual annotation of documents. *Briefings in Bioinformatics*.

Nichols J. (1984). Functional theories of grammar. *Annual Review of Anthropology* **13**, 97–117.

Östman J.O. and Fried M. (eds) (2005). *Construction Grammars: Cognitive Grounding and Theoretical Extensions*, vol. 3. John Benjamins Publishing.

Paillet J.P. (1973). Computational linguistics and linguistic theory. In *Proceedings of the 5th Conference on Computational Linguistics*, vol. 2. Association for Computational Linguistics, pp. 357–366.

Petticrew M. 2001. Systematic reviews from astronomy to zoology: myths and misconceptions. *British Medical Journal* **322**, 98–101.

Pollard C. and Sag I.A. (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Ponvert E., Baldridge J. and Erk K. (2011). Simple unsupervised grammar induction from raw text with cascaded finite state models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. **1**. Association for Computational Linguistics, pp. 1077–1086.

Post M. and Gildea D. (2013). Bayesian tree substitution grammars as a usage-based approach. *Language and Speech* **56**, 291–308.

Radford A. (1981). *Transformational Syntax: A Student's Guide to Chomsky's Extended Standard Theory*. Cambridge University Press.

Reichart R. and Rappoport A. (2008). Unsupervised induction of labeled parse trees by clustering with syntactic features. In *Proceedings of the 22nd International Conference on Computational Linguistics*, vol. 1. Association for Computational Linguistics, pp. 721–728.

Rimell L., Clark S. and Steedman M. (2009). Unbounded dependency recovery for parser evaluation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, vol. 2. Association for Computational Linguistics, pp. 813–821.

Roche E. and Schabes Y. (eds) (1997). *Finite-State Language Processing*. MIT press.

Saffran J.R. (2001). The use of predictive dependencies in language learning. *Journal of Memory and Language* **44**, 493–515.

Sangati F. (2010). A probabilistic generative model for an intermediate constituency-dependency representation. In *Proceedings of the ACL 2010 Student Research Workshop*. Association for Computational Linguistics, pp. 19–24.

Santamaria J. and Araujo L. (2010). Identifying patterns for unsupervised grammar induction. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pp. 38–45.

Schabes Y., **Roth M. and Osborne R.** (1993). Parsing the Wall Street Journal with the inside-outside algorithm. In *Proceedings of the sixth conference on European chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 341–347.

Seginer Y. (2007). Fast unsupervised incremental parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 384–391.

Skinner B.F. (2014). *Verbal Behavior*. BF Skinner Foundation.

Snyder B., **Naseem T. and Barzilay R.** (2009). Unsupervised multilingual grammar induction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, vol. **1**. Association for Computational Linguistics, pp. 73–81.

Solan Z., **Horn D.**, **Ruppin E. and Edelman S.** (2004). Unsupervised context sensitive language acquisition from a large corpus. In *Advances in Neural Information Processing Systems*, pp. 961–968.

Søgaard A. (2011). From ranked words to dependency trees: two-stage unsupervised non-projective dependency parsing. In *Proceedings of TextGraphs-6: Graph-based Methods for Natural Language Processing*. Association for Computational Linguistics, pp. 60–68.

Spitkovsky V.I., **Alshawi H. and Jurafsky D.** (2010). From baby steps to leapfrog: How less is more in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 751–759.

Spitkovsky V.I., **Alshawi H. and Jurafsky D.** (2011). Punctuation: Making a point in unsupervised dependency parsing. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pp. 19–28.

Spitkovsky V.I., **Alshawi H. and Jurafsky D.** (2012). Capitalization cues improve dependency grammar induction. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*. Association for Computational Linguistics, pp. 16–22.

Taylor A., **Marcus M. and Santorini B.** (2003). The Penn treebank: an overview. In *Treebanks*. Dordrecht: Springer, pp. 5–22.

Yang C. (2011). A statistical test for grammar. In *Proceedings of the 2nd workshop on Cognitive Modeling and Computational Linguistics*. Association for Computational Linguistics, pp. 30–38.

Zuidema W. (2006). What are the productive units of natural language grammar?: a DOP approach to the automatic identification of constructions. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pp. 29–36.

Appendix

S. no.	Title	Findings	References
1.	Corpus-Based Induction of Syntactic Structure: Models of Dependency and Constituency	*Theory:* Formal grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Dependency tree *Evaluation strategy:* Comparison against gold standard *Features:* Tokens, Head, valency, direction of attachment and word distribution clusters *Implementation methodology:* Top-down dependency grammar model	Klein and Manning (2004)

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

S. no.	Title	Findings	References
2.	Unsupervised Multilingual Grammar Induction	*Theory:* Formal grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Constituency tree *Evaluation strategy:* Comparison against gold standard *Features:* POS tags and word alignments *Implementation methodology:* Similarity or exemplar-based models	Snyder *et al.* (2009)
3.	Unsupervised Context-Sensitive Language Acquisition from a Large Corpus	*Theory:* Theory-neutral *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Directed multigraph *Evaluation strategy:* Grammaticality judgement task *Features:* Tokens *Implementation methodology:* Automatic Distillation of Structures (ADIOS)	Solan *et al.* (2004)
4.	An All-Subtrees Approach to Unsupervised Parsing	*Theory:* Theory-neutral *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Constituency tree *Evaluation strategy:* Comparison against gold standard *Features:* Tokens and POS tags *Implementation methodology:* Data-oriented parsing (DOP)	Bod (2006)
5.	From Baby Steps to Leapfrog: How "Less is More" in Unsupervised Dependency Parsing	*Theory:* Formal grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Dependency tree *Evaluation strategy:* Comparison against gold standard *Features:* Tokens, POS tags, head, direction of attachment and valence *Implementation methodology:* Top-down dependency grammar model	Spitkovsky *et al.* (2010)
6.	Unsupervised Grammar Induction by Distribution and Attachment	*Theory:* Formal grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Constituency tree *Evaluation strategy:* Comparison against gold standard *Features:* Heuristics and distributed representation of words *Implementation methodology:* Distribution-based model	Brooks (2006)

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

S. no.	Title	Findings	References
7.	What are the Productive Units of Natural Language Grammar? A DOP Approach to the Automatic Identification of Constructions	*Theory:* Theory-neutral *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Constituency tree *Evaluation strategy:* Comparison against gold standard *Features:* Tokens and POS tags *Implementation methodology:* Data-oriented parsing (DOP)	Zuidema (2006)
8.	Simple Unsupervised Grammar Induction from Raw Text with Cascaded Finite State Models	*Theory:* Formal grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Constituency tree *Evaluation strategy:* Comparison against gold standard *Features:* Tokens *Implementation methodology:* Chunker and its extension	Ponvert *et al.* (2011)
9.	Characterizing Motherese: On the Computational Structure of Child-Directed Language	*Theory:* Theory-neutral *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Directed multigraph *Evaluation strategy:* Comparison of models trained on disjoint corpora with respect to sentence acceptability. *Features:* Tokens *Implementation methodology:* Automatic Distillation of Structures (ADIOS)	Brodsky and Waterfall (2007)
10.	Unsupervised Induction of labelled Parse Trees by Clustering with Syntactic Features	*Theory:* Formal grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Constituency tree *Evaluation strategy:* Comparison against gold standard *Features:* POS and chunks *Implementation methodology:* Chunk-based approach and clustering approach	Reichart and Rappoport (2008)
11.	A Survey of Grammatical Inference Methods for Natural Language Learning	*Findings:* Fourteen studies, six computational techniques (statistical methods, evolutionary computing techniques, minimum description length, heuristic methods, greedy search and clustering techniques), two presentation sets (text and informant), three types of information for learning (supervised, unsupervised and semi-supervised) and three types of evaluation methods (looks-good-to-me, compare against treebank and rebuilding known grammars).	D'Ulizia *et al.* (2011)

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

S. no.	Title	Findings	References
12.	A Probabilistic Generative Model for an Intermediate Constituency-Dependency Representation	*Theory:* Formal grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Dependency tree *Evaluation strategy:* Comparison against gold standard *Features:* Tokens, POS tags, head, direction of attachment and valence *Implementation methodology:* Top-down dependency grammar model	Sangati (2010)
13.	Identifying Patterns for Unsupervised Grammar Induction	*Theory:* Formal grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Constituency tree *Evaluation strategy:* Comparison against gold standard *Features:* Tokens, POS tags and heuristics *Implementation methodology:* Heuristics	Santamaria and Araujo (2010)
14.	From Ranked Words to Dependency Trees: Two-Stage Unsupervised Non-Projective Dependency Parsing	*Theory:* Formal grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Dependency tree *Evaluation strategy:* Comparison against gold standard *Features:* Tokens and distributed representation of words *Implementation methodology:* Distribution-based model	Søgaard (2011)
15.	Posterior Sparsity in Unsupervised Dependency Parsing	*Theory:* Formal grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Dependency tree *Evaluation strategy:* Comparison against gold standard *Features:* Tokens, POS tags, head, direction of attachment and valence *Implementation methodology:* Top-down dependency grammar model	Gillenwater *et al.* (2011)
16.	Capitalization Cues Improve Dependency Grammar Induction	*Theory:* Formal grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Dependency tree *Evaluation strategy:* Comparison against gold standard *Features:* Tokens, orthographic cues, head, valence and direction of attachment *Implementation methodology:* Top-down dependency grammar model	Spitkovsky *et al.* (2012)

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

S. no.	Title	Findings	References
17.	Exploiting Reducibility in Unsupervised Dependency Parsing	*Theory:* Formal grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Dependency tree *Evaluation strategy:* Comparison against gold standard *Features:* Tokens, POS tags, head, valence and direction of attachment *Implementation methodology:* Top-down dependency grammar model	Mareček and Žabokrtský (2012b)
18.	Unsupervised Dependency Parsing using Reducibility and Fertility features	*Theory:* Formal grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Dependency tree *Evaluation strategy:* Comparison against gold standard *Features:* Tokens, head, valence and direction of attachment *Implementation methodology:* Top-down dependency grammar model	Mareček and Žabokrtský (2012a)
19.	Learnability and falsifiability of Construction Grammars	*Theory:* Usage-based grammar *Representing productivity:* Non-hierarchical *Processing productivity:* Non-incremental *Output:* Construction slots *Evaluation strategy:* Maximum coverage, minimum size and stability measures for a usage-based grammar. *Features:* Tokens, POS tags, and construction association measures *Implementation methodology:* Construction grammar induction	Dunn (2017a)
20.	Boosting Unsupervised Grammar Induction by Splitting Complex Sentences on Function Words	*Theory:* Theory-neutral *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Directed multigraph *Evaluation strategy:* Comparison against gold standard and output is compared against an artificial grammar that generated the test corpus. *Features:* Tokens *Implementation methodology:* Automatic Distillation of Structures (ADIOS)	Berant *et al.* (2007)
21.	Evaluation Strategies for Computational Construction Grammars	*Theory:* Usage-based *Representing productivity:* Non-hierarchical *Processing productivity:* Non-incremental *Output:* Transient structure *Evaluation strategy:* Comparison against gold standard. *Features:* POS and heuristics *Implementation methodology:* Construction grammar induction	Marques and Beuls (2016)

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

S. no.	Title	Findings	References
22.	Learning Syntactic Constructions from Raw Corpora	*Theory:* Theory-neutral *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Directed multigraph *Evaluation strategy:* Comparison against gold standard and output is compared against an artificial grammar that generated the test corpus. *Features:* Tokens *Implementation methodology:* Automatic Distillation of Structures (ADIOS)	Edelman *et al.* (2005)
23.	Punctuation: Making a Point in Unsupervised Dependency Parsing	*Theory:* Formal grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Dependency tree *Evaluation strategy:* Comparison against gold standard. *Features:* Tokens and orthographic cues *Implementation methodology:* Top-down dependency grammar model	Spitkovsky *et al.* (2011)
24.	An Exemplar-based Approach to Unsupervised Parsing	*Theory:* Theory-neutral *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Constituency tree *Evaluation strategy:* Comparison against gold standard. *Features:* Tokens and POS tags *Implementation methodology:* Similarity or exemplar-based model	Dennis (2005)
25.	Bayesian Tree Substitution Grammars as a Usage-based approach	*Theory:* Theory-neutral *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Constituency tree *Evaluation strategy:* Comparison against gold standard. *Features:* Tokens and POS tags *Implementation methodology:* Data-oriented parsing	Post and Gildea (2013)
26.	Computational Learning of Construction Grammars	*Theory:* Usage-based *Representing productivity:* Non-hierarchical *Processing productivity:* Non-incremental *Output:* Construction slots *Evaluation strategy:* Qualitative, maximum coverage and minimum size of grammar *Features:* Tokens, POS tags, semantic tags and construction association measures *Implementation methodology:* Construction grammar learning	Dunn (2017b)

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

S. no.	Title	Findings	References
27.	Unsupervised Grammar Induction with Depth-bounded PCFG	*Theory:* Formal grammar or generative grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Constituency tree *Evaluation strategy:* Comparison against gold standard. *Features:* Tokens *Implementation methodology:* Probabilistic context-free grammar (PCFG)	Jin *et al.* (2018)
28.	From Exemplar to Grammar: A Probabilistic Analogy-Based Model of Language Learning	*Theory:* Theory-neutral *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Constituency tree *Evaluation strategy:* Comparison against gold standard. *Features:* Tokens and POS tags *Implementation methodology:* Data-oriented parsing (DOP)	Bod (2009)
29.	Grammar Induction from Text Using Small Syntactic Prototypes	*Theory:* Formal or generative grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Dependency tree *Evaluation strategy:* Comparison against gold standard. *Features:* Tokens, POS tags, head, valence and direction of attachment *Implementation methodology:* Top-down dependency grammar model	Boonkwan and Steedman (2011)
30.	Improving Unsupervised Dependency Parsing with Richer Contexts and Smoothing	*Theory:* Formal or generative grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Dependency tree *Evaluation strategy:* Comparison against gold standard. *Features:* Tokens, POS tags, head, valence and direction of attachment *Implementation methodology:* Top-down dependency grammar model	Headden *et al.* (2009)
31.	Limitations of Current Grammar Induction Algorithms	*Findings:* Survey paper and different grammar induction algorithms such as EMILE, ADIOS and ABL are discussed. Tested on Eindhoven corpus. Two experiments were conducted on various grammar induction (GI) approaches. The results from the experiments show that the current grammar induction systems like EMILE, ADIOS and ABL have severe shortcomings in deriving meaningful structure from language as complicated as Eindhoven corpus. An incremental grammar induction strategy is suggested as preferable and a short illustration of how the system should be is presented at the end.	Cramer (2007)

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

S. no.	Title	Findings	References
32.	Towards High Speed Grammar Induction on Large Text Corpora	*Theory:* Formal or generative grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Constituency tree *Evaluation strategy:* Comparison against gold standard. *Features:* Distributional representation of words *Implementation methodology:* Chunker and its extension and probabilistic context-free grammar (PCFG)	Adriaans *et al.* (2000)
33.	Unbounded Dependency Recovery for Parser Evaluation	*Findings:* Parser evaluation paper, five different parsers are compared. The five parsers are C&C, Enju, DCU, Rasp and Stanford. The suitability of PARSEVAL metrics as a measure of the performance of parsers is called into question in the paper. By pointing out that the parser performance on recovering unbounded dependencies is bad, the study motivates the necessity of better parser evaluation metrics.	Rimell *et al.* (2009)
34.	Evolving Natural Language Grammars without Supervision	*Theory:* Formal or generative grammar *Representing productivity:* Hierarchical *Processing productivity:* Incremental *Output:* Constituency tree *Evaluation strategy:* Comparison against gold standard. *Features:* POS and heuristic rules *Implementation methodology:* Heuristic approach	Araujo and Santamaría (2010)
35.	Unsupervised Induction of Dependency Structures Using Probabilistic Bilexical Grammars	*Theory:* Formal or generative grammar *Representing productivity:* Hierarchical *Processing productivity:* Non-incremental *Output:* Dependency tree *Evaluation strategy:* Comparison against gold standard. *Features:* Tokens, POS, head, valence and direction of attachment, *Implementation methodology:* Top-down dependency grammar model	Dominguez and Infante-Lopez (2011)
36.	Fast Unsupervised Incremental Parsing	*Theory:* Formal or generative grammar *Representing productivity:* Hierarchical *Processing productivity:* Incremental *Output:* Shortest common cover link sets *Evaluation strategy:* Comparison against gold standard. *Features:* Words and distribution *Implementation methodology:* Distribution-based	Seginer (2007)

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

S. no.	Title	Findings	References
37.	Use of Predictive Dependencies in Language Learning	*Goal:* To verify experimentally if abstract grammatical hierarchies can be learnt using the statistical cues of local predictive dependencies. *Method:* Artificial language learning task by humans. *Experiment:* Adult participants exposed to sentences from artificial language. They had no semantic, visual or any other clue except the distribution of words and their category. *Conditions:* Groups: intentional group, incidental group and control group. *Tests:* Rule test and fragment test. *Result:* Experimental groups (intentional and incidental) significantly outperformed control group on both the tests under various conditions. *Insights:* Predictive dependencies on the naturally occurring text can lead to simple phrase structure acquisition. We can exploit this for unsupervised learning. *Summary:* The results support the hypothesis that learners can detect predictive dependencies in the service of acquiring simple phrase structure.	Saffran (2001)
38.	The Role of Discourse Context in the Processing of a Flexible Word-order Language	*Goal:* To find out if the processing difficulty associated with non-canonical word order in a language is reduced in an appropriate discourse context. *Method:* Conducting experiments on speakers of Finnish by presenting them with non-canonical word order sentences to see the effect. *Experiment:* Two experiments: 1. Self-paced reading task 2. Eye-tracking while hearing the spoken description of scenes *Conditions:* Forty Finnish speakers, given 20 sentences in 4 patterns. Eye-tracking during listening *Result:* Reading time in the first experiment and eye movements in the second experiment support the idea that people use word order patterns to predict upcoming referents on the basis of discourse status. *Insights:* Discourse is not taken into consideration while modelling computational grammar induction. The study provides the theoretical motivation for such an approach. *Summary:* The processing of non-canonical structures is facilitated by the presence of an appropriate discourse context.	Kaiser and Trueswell (2004)

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

S. no.	Title	Findings	References
39.	A Linguistic Investigation into Unsupervised DOP	*Goal:* Study if a computational bootstrapping model of language can explain the abstract linguistic properties of natural languages without assuming a universal grammar. *Method:* An unsupervised DOP model which computes the most probable tree from among the shortest derivations of sentences. Use this model to explain both rule-based and exemplar-based properties of a natural language. *Result:* Learning discontiguous construction patterns, agreement and movement were all shown to be possible for computational bootstrapping without a need for special, top-down abstract grammar. *Summary:* Recursive tree structure and an analogical matching algorithm can help us learn various syntactic phenomena that usually appeal to an abstract, universal grammar that generates sentences top-down.	Bod (2007)
40.	A Statistical Test for Grammar	*Goal:* To see if the children acquire language through a productive grammatical system (typically expressed as generative grammar) or usage-based schematic patterns. *Method:* A statistical test is proposed to check if grammar is abstract and productive or lexically specific and usage-based. *Experiment:* Through case studies on the children's speech data, a measure of overlap between theoretical predictions (productive grammar vs. lexical-specific usage based) is measured. *Conditions:* Productivity and usage-based *Result:* Results of the statistical test show that a lexically specific, usage-based and memory-and-retrieval approach is unsupported. The results are consistent with a productive abstract grammatical system in child speech. *Summary:* These results do not resolve the innateness debate in language acquisition: they only point to the very early availability of an abstract and productive grammar.	Yang (2011)
41.	How Hierarchical is Language Use?	*Goal:* It is assumed that hierarchical phrase structure rules play an important role in natural language processing? Is it true? Can sequential model predict statistical regularities in language? *Method:* Comparison of emerging ideas in various neurophysiology, behaviour and computational studies suggest that sequential sentential structure has considerable explanatory power. *Task:* Discussion of evidences from multiple research fields: cognitive neuroscience, psycholinguistics and computational models of language acquisition	Frank *et al.* (2012)

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

S. no.	Title	Findings	References
		Insights: Combining productive grammatical units need not happen hierarchically but a sequential process would suffice. *Summary:* Such a model of language processing has implications for the fields of linguistics, ethology, psychology and computer science/natural language processing.	
42.	Rich Syntax from a Raw Corpus: Unsupervised Does It	*Goal:* Compare the theoretical and computational properties of ADIOS system to some recent works in computational linguistics and in grammar theory *Method:* Discussing the computational principles behind the ADIOS model, by comparing it to select approaches from computational and formal linguistics. *Results:* Principles behind ADIOS: (a) pattern significance is learnt probabilistically, (b) Patterns are context-sensitive and (c) Patterns are hierarchical *Insights:* Comparison of ADIOS with grammar and computational theories is done. Similar to construction grammar in its general philosophy of linguistic representations and tree adjoining grammar in its computational capacity. *Summary:* Evaluation strategy for a purely empirical, usage-based approach to grammar induction should be found.	Edelman *et al.* (2003)
43.	Subjacency Constraints without Universal Grammar: Evidence from Artificial Language Learning and Connectionist Modeling	*Goal:* To experimentally verify if subjacency constraints which usually appeal to universal grammar can be acquired through limitations on sequential learning. *Method:* Two types of experiments were conducted to enquire about the nature of subjacency constraints. Artificial language learning experiments and connectionist simulations using the same data were also made. *Experiment:* Two experiments: 1. Created two artificial languages, natural (NAT) and unnatural (UNNAT). Each artificial language consisted of a set of letter strings, each letter representing a specific grammatical class. Subjects were randomly assigned to one of three conditions (NAT, UNNAT and CONTROL). NAT and UNNAT were trained using the natural and unnatural languages, respectively. The CONTROL group completed only the test session. During training, individual letter strings were presented briefly on a computer. After each presentation, participants were prompted to enter the letter string using the keyboard. Training consisted of 2 blocks of the 30 items, presented randomly. During the test session, participants decided if the test items were created by the same (grammatical) or different (ungrammatical) rules as the training items.	Ellefson and Christiansen (2000)

https://doi.org/10.1017/S1351324920000327 Published online by Cambridge University Press

S. no.	Title	Findings	References
		2. Simple recurrent networks (SRNs) were used to show the connectionist simulations of the same human data discussed earlier in Experiment 1. *Conditions:* 1. Sixty undergraduates were recruited from an introductory psychology class at Southern Illinois University for the Experiment 1 2. Standard feed-forward neural networks equipped with an extra layer of context units. *Result:* An overall *t*-test indicated that NAT (59%) learned the language significantly better than UNNAT (54%). This result indicates that the UNNAT was more difficult to learn than the NAT. *Insights:* The results therefore corroborate the hypothesis that constraints on the learning and processing of sequential structure can explain why subjacency violations tend to be avoided: they were weeded out because they made the sequential structure of language too difficult to learn. *Summary:* The results suggest that constraints arising from general cognitive processes, such as sequential learning and processing, are likely to play a larger role in sentence processing than has traditionally been assumed. This means that what we observe today as linguistic universals may be stable states that have emerged through an extended process of linguistic evolution.	

Cite this article: Muralidaran V, Spasić I and Knight D (2021). A systematic review of unsupervised approaches to grammar induction. *Natural Language Engineering* **27**, 647–689. https://doi.org/10.1017/S1351324920000327