Linguistic Models at the Crossroads of Agents, Learning and Formal Languages

Leonor BECERRA-BONACHE[a]; **M. Dolores JIMÉNEZ-LÓPEZ**[b]

[a]*Laboratoire Hubert Curien, Jean Monnet University, 18 rue Benoit Lauras, 42100, Saint-Etienne, France*

[b]*Research Group on Mathematical Linguistics, Universitat Rovira i Virgili, Av. Catalunya 35, 43002 Tarragona, Spain*

KEYWORD

Agents; Learning; Formal Languages; Linguistics

ABSTRACT

This paper aims at reviewing the most relevant linguistic applications developed in the intersection between three different fields: machine learning, formal language theory and agent technologies. On the one hand, we present some of the main linguistic contributions of the intersection between machine learning and formal languages, which constitutes a well-established research area known as Grammatical Inference. On the other hand, we present an overview of the main linguistic applications of models developed in the intersection between agent technologies and formal languages, such as colonies, grammar systems and eco-grammar systems. Our goal is to show how interdisciplinary research between these three fields can contribute to better understand how natural language is acquired and processed.

1. Introduction

Currently, interdisciplinarity is a key feature to make progress and increase the rate of scientific findings in any research area. Linguistics –as the scientific study of natural language– is not an exception. Natural language is a hard problem not only for linguistics that has not yet provided universal accepted theories about how language is acquired and processed, but also for computer science that up to now has implemented natural language processing systems that are far from being satisfactory. In order to explain natural language, we need to cross traditional academic boundaries. We should attack the problem from various angles and methods, eventually across disciplines, forming a new method for understanding natural languages. Therefore, interdisciplinarity should be an essential trait of the research in language.

In this paper, we review the role that the interrelation of three areas can play in the area of linguistics. Specifically, we want to show how the interdisciplinary relation between *machine learning*, *formal languages* and *agent technologies* can contribute to the solution of one of the most persistent problems in science: the explanation of how natural language is acquired and processed.

Agent technologies is one of the most important areas emerged in Information Technology in the 90's. By implementing autonomous entities driven by beliefs, goals, capabilities, plans and agency properties, agent technologies capture essential aspects of the modeled systems. The metaphor of autonomous problem solving entities cooperating and coordinating to achieve their objectives is a natural way of conceptualizing many problems. In fact, the multi-agent system literature spans a wide range of fields.

Machine learning is one of the most active research areas within Artificial Intelligence. Its main goal is to develop techniques that allow computers to learn. Machine learning algorithms construct a model based on the inputs that they receive, and then they use that model to make predictions or decisions. Examples applications of machine learning are: spam filtering, handwriting recognition, computer vision, etc.

Formal language theory provides mathematical tools for the description of linguistic phenomena. It was born in the middle of the 20th century as a tool for modeling and investigating the syntax of natural languages. However, very soon it developed as a new research field, separated from Linguistics, with specific problems, techniques and results and, since then, it has had an important role in the field of Computer Science.

In this paper, we focus on the linguistic applications of the interrelation among those disciplines. Specifically, we consider:

- The interrelation between *machine learning and formal languages*. Understanding human learning well enough to reproduce aspects of that learning behaviour in a computer system is a worthy scientific goal. One of the less understood learning capacities of humans is their ability to acquire a natural language. In order to better understand natural language acquisition, research in formal models of language learning, within the field of machine learning, has received significant attention. The theory of formal language theory is central to the field of machine learning, since the specific subfield of grammatical inference deals with the process of learning grammars and languages from data.
- The interrelation between *agent technologies and formal languages*. While the first generation of formal languages was based on rewriting, a further development in this area has been the idea of several devices collaborating for achieving a common goal. Formal language theory has taken advantage of the idea of formalizing agent architectures where a hard task is distributed among several task-specific agents. In fact, non-standard formal language models have been proposed as grammatical models of agent systems.

The paper is organized as follows. In section 2, the intersection between machine learning and formal languages is taken into account. As we have said, this intersection constitutes a well-established research field known as *Grammatical Inference*. We review here the main linguistic applications of this field. In section 3, the interrelation between formal language theory and agent technologies is considered. We show how agent technologies can offer good solutions and alternative frameworks to classic models in the area of processing and computing languages that can be useful for the description and analysis of natural language. Section 4 presents some conclusions.

2. Learning and Formal Languages

The intersection between the field of *machine learning* and the *theory of formal languages* constitutes a well-known research field called *Grammatical Inference* (GI) (de la Higuera, 2010). GI refers to the process of learning (formal) grammars and languages from data. In a GI problem, there are two different actors involved: a teacher and a learner. The teacher provides information to the learner, and the learner (or learning algorithm), from that information, must identify the underlying language. For example, if the target language (i.e., the language to be learnt) is (a^+b), the teacher could provide to the learner strings that belong to the target language (i.e., positive data), such as ab, aab, $aaaab\ldots$ And the learner, from this information, should infer that the target language is (a^+b).

The field of GI was originated in the 60's, mainly by the work developed by E.M. Gold (Gold, 1967). Motivated by the problem of how children acquire their native language, E.M. Gold tried to investigate, from a theoretical point of view, how the ability to speak a language can be achieved in an artificial way. In order to do that, the task of acquiring a language was reduced to the task of learning a formal language. This formal approach of language acquisition takes its root in the work developed by N. Chomsky in the 50's (Chomsky, 1956). He proposed to use formal languages to describe the syntax of natural languages. A formal language is a set of strings (i.e., finite sequence of symbols) over an alphabet Σ. A formal grammar is a finite set of rules that generate a language. It consists of four components:

- A finite set N of nonterminal symbols (they can be replaced)
- A finite set T of terminal symbols (they cannot be replaced)
- A start symbol $S \in N$ (all the strings of the language derive from it)
- A finite set P of production rules (they specify which symbols may replace other symbols). These rules are used to generate strings and have the following form: $u \rightarrow v$, such that $u, w \in (N \cup T)^*$ and u contains at least one nonterminal symbol.

According to the form of their rules, Chomsky defined four types of grammars: regular, context-free, context-sensitive and recursively enumerable. This classification is known as *Chomsky hierarchy* (it is hierarchical, since one type of grammar includes the previous one).

Since Gold's pioneering work, a big amount of research has been done by researchers coming from different scientific traditions: machine learning, formal languages, computational linguistics, pattern recognition, etc. Two main approaches can be distinguished in GI:

1) *Theoretical approaches*: researchers aim to prove efficient learnability of grammars. Most of GI researchers have been focused on this approach. Their aim is to obtain formal results, for example: formal descriptions of the target languages, formal proofs about the efficiency of a learning algorithm, etc.

2) *Practical approaches*: researchers aim to develop systems that learn grammars from real data. Instead of proving the learnability of grammars, researchers focus on providing empirical systems that deal with natural language data.

Despite the original linguistic motivation of the GI studies, most of the work in this field have been focused on obtaining formal results, without exploiting the linguistic relevance of the classes that have been studied and the results obtained. We will review next some theoretical and practical studies developed in GI with a linguistic motivation.

2.1 Theoretical studies of classes linguistically relevant

GI researchers have been mainly focused on studying regular (REG) and context-free (CF) grammars, which constitute the first two levels of the Chomsky hierarchy (de la Higuera, 2010). There are not too many studies focused on classes of languages more powerful than CF, since it is already very hard to get positive learnability results with these two classes. E.M. Gold (Gold, 1967) gave one of the main results in GI: superfinite classes of languages (i.e., classes of languages that contain all finite languages and at least one infinite language) are not learnable from only positive data. This implies that any of the classes defined by Chomsky are learnable from only positive data. The main problem is that if we produce a grammar whose language is larger than the target language (i.e., we overgeneralize), by only having access to positive data, we will not be able to refute our hypothesis. In order to avoid this problem, several solutions have been adopted in GI. For example:

- To study subclasses of the languages to be learned, such as reversible languages (Angluin, 1982), k-testable languages (García and Vidal, 1990), etc.
- To provide structural information to the learner, i.e., information about the structure of the input (Sakakibara, 1992)
- To make available to the learner negative data, i.e., strings that do not belong to the target language (Oncina and García, 1992)

For more detailed information, the reader is referred to (de la Higuera, 2010). It is worth noting that the main goal of most of the works developed in GI have been to prove that it is possible to learn grammars in an efficient way, without any concern about whether the grammars were relevant or not to study the natural language. However, if we really want to come back to the original linguistic motivations of GI, the following questions become important: are REG or CF (or subclasses of them) enough to study natural language? where are natural languages located in the Chomsky hierarchy?

CF were one of the first kind of grammars used to define the syntax of natural language. A grammar is CF if every production rule is of the form:

- $A \to w$, with $A \in N$, $w \in (N \cup T)^*$

Hence, the expansion of a symbol in a CF grammar does not depend on its context (i.e., the symbols that surround the nonterminal symbol); the single nonterminal symbol on the left hand side of the rule can always be replaced by the symbols on the right hand side of the rule.

Table 1 shows an example of a CF grammar for a small fragment of Spanish. As we can see:

- nonterminal symbols represent grammatical categories, such as: sentence (S), noun phrase (NP), verb phrase (VP), noun (N), verb (V) and preposition (PREP).
- terminal symbols represent words in the language.
- from the start symbol we can obtain a sentence.
- production rules express how to construct different grammatical categories.

$S \rightarrow NP \quad VP$
$NP \rightarrow N$
$VP \rightarrow V \quad PP$
$PP \rightarrow PREP \quad NP$
$N \rightarrow Juan$ $N \rightarrow María$
$V \rightarrow ama$
$PREP \rightarrow a$

Table 1: Example of a CF grammar for a small fragment of Spanish

Example of sentences generated by this grammar are: *Juan ama a María* (John loves Mary), María ama a Juan (Mary loves John).

Although CF grammars are enough to model some of the main linguistic phenomena of natural language, they are not able to generate all the constructions that are present in natural languages. For instance, in the 80's, it was discovered some examples of natural language constructions that cannot be described by using a CF grammar; in particular, examples of multiple agreement, crossed agreement and duplications structures in Dutch (Bresnan et al., 1987), Bambara (Culy, 1987) or Swiss-German (Shieber, 1987). Here is an example of a crossed agreement structure in Swiss-German:

Jan säit das mer (d'chind)^m (em Hans)^n es huus haend wele (laa)^m (hälfe)^n aastriiche
(English translation: Jan said that we wanted to let the children help Hans paint the house)

This sentence has the structure $xwa^mb^nyc^md^nz$, where a, b stand for accusative, dative noun phrases, respectively, and c, d for the corresponding accusative, dative verb phrases, respectively. This kind of constructions (and some others found in several natural languages) cannot be described by using a CF grammar, which shows that natural languages are not CF. Hence, more generative capacity than CF is necessary to describe natural languages. But, how much power is it necessary?

The following level in the Chomsky hierarchy is context-sensitive (CS). A grammar is CS if every production rule is of the form:

- $u_1Au_2 \rightarrow u_1wu_2$, with $u_1, u_2, w \in (N \cup T)^*$, $A \in N$ and $w \neq \lambda$

Hence, the expansion of a symbol in a CS grammar depends on its context. Although CS grammars have enough expressive power to describe natural languages, they have a problem: they are computationally very complex.

Taking into account the limitations of CF and CS to study natural language, A. K. Joshi (Joshi, 1985) proposed the idea of having a mechanism able to generate CF and non-CF constructions, but keeping under control the generative power. According to him, this mechanism should have the following three properties:

- limited crossed-serial dependencies
- constant growth

- polynomial parsing

A.K. Joshi called classes of grammars with these properties *mildly context-sensitive* (MCS), because they are slightly more powerful than CF, but they preserve many of the interesting features of the CF. He claimed that grammars that are adequate for natural languages structures are found in the class of MCS grammars.

Several formalisms have been proposed to fabricate MCS families of languages, like *Tree Adjoining Grammars* (Joshi and Schabes, 1997), *Head Grammars* (Roach, 1987), *Combinatory Categorial Grammars* (Steedman, 1985), *Linear Indexed Grammars* (Gazdar and Pullum, 1985), etc. A.K. Joshi proved that these 4 formalisms are equivalent, i.e., they describe the same class of languages.

All these works assume that MCS contain all CF languages and are included in CS, that is, they occupy a concentric position in the Chomsky hierarchy. However, some researchers have suggested that natural languages could occupy an *orthogonal position* in this hierarchy, that is, they contain some REG and some CF languages, but are included in CS (Kudlek et al., 2002; Manaster-Ramer, 1999). Several arguments support this idea: we can find some examples of natural languages constructions that are neither REG or CF, and also, there exist some REG and CF constructions that do not appear naturally in sentences.

Therefore, taking into account all these ideas, it would be desirable to study mechanisms having the following properties:

i) They generate MCS languages (in the sense that, they can generate multiple agreements, crossed agreements and duplication structures, and they are computationally feasible).

ii) They occupy an orthogonal position in the Chomsky hierarchy.

Researchers in GI have studied classes of languages with such desirable properties from a linguistic point of view. An example of this class is the *Simple p-dimensional External Contextual* grammars (SEC). A SEC grammar is $G = (\Sigma, B, C)$, where:

- Σ is the alphabet of G.
- B is a singleton of p-words over Σ. B is called the base. A p-word is a p-dimensional vector whose components are words over Σ (i.e., $x = (x_1, x_2, \ldots, x_p)$, where $x \in \Sigma^*$, $1 \geq i \geq p$).
- C is a finite set of p-contexts over Σ. C is called the set of contexts of G. A p-context is a p-dimensional vector whose components are contexts over Σ (i.e., $c = (c_1, c_2, \ldots, c_p)$, where $c_i = (u_i, v_i)$, $u_i, v_i \in \Sigma^*$, $1 \geq i \geq p$).

As we can see in the definition, a SEC grammar does not involve nonterminals and it does not have production rules. A SEC grammar produces a language starting from the base (i.e., a word), and iteratively adding contexts (i.e., pair of words) at the ends of the currently generated word. With this kind of grammars we can easily generate non-CF languages. For example, we can generate $\{a^n b^n c^n | n \leq 0\}$ with the following SEC grammar:

- $\Sigma = a, b, c$
- $B = (\lambda, \lambda)$
- $C = [(a, b), (c, \lambda)]$

If we apply just once the context to the base we obtain: $(a\lambda b, c\lambda\lambda) = (ab, c) = abc$. If we apply again the context we obtain: $(aabb, cc) = aabbcc$. And so on.

In (Becerra-Bonache, 2006), it was proved that SEC can generate MCS languages and occupy an orthogonal position in the Chomsky hierarchy. Positive learnability results have also been obtained with this class in (Becerra-Bonache and Yokomori, 2004; Oates et al., 2006; Becerra-Bonache, 2006; Becerra-Bonache et al., 2010). Hence, we can find in the field of GI works that study the learnability of classes that are relevant to model some aspects of natural language syntax. Some other results on the learnability of other classes that also generate MCS languages or are linguistically relevant can be found in (Yoshinaka, 2009; Clark and Yoshinaka, 2014).

2.2 Practical Studies with Natural Language Data

Although most part of the work in GI are theoretical, we can also find some practical approaches in GI based on natural language data. Next we review some of the main approaches.

Depending on how the information is provided to the learner, we can distinguish three different GI approaches to natural language (D'Ulizia et al., 2011):

- *Unsupervised approach*: the teacher provides *unstructured* sentences to the learner (i.e., the learner does not receive explicit information about the structure of the sentences in the target language).
- *Supervised approach*: the teacher provides (unstructured) sentences to the learner and can also validate the hypothesis made by the learner about the target language (i.e., the learner can use the teacher to verify if a sentence is correct in the target language).
- *Semi-supervised approach*: in addition to the *structured* data, the learner receives *unstructured* data.

Moreover, we can also take into account the kind of information given to the learner. In this case, we will distinguish two different methods:

- *Learning from positive data*: only examples of correct sentences are given to the learner. This is also known as *text* presentation.
- *Learning from positive and negative data*: examples of correct and incorrect sentences are given to the learner. This is also known as *informant* presentation.

Most part of GI methods for natural language have also been focused on CF grammars. Most of them are based on an *unsupervised approach*, and only use *positive data* during the learning process. According to (D'Ulizia et al., 2011), one of the main raisons of using unsupervised learning is that methods based on it are able to learn larger and more complex models than the ones based on supervised learning. Moreover, unsupervised approaches are less time-consuming and costly.

There exist different techniques to evaluate the GI algorithms. We can mainly grouped them in three (van Zaanen et al., 2004; D'Ulizia et al., 2011):

- *Looks-Good-to-Me*: the GI algorithm is applied to unstructured data. The output produced by the algorithm is checked manually by the evaluator (usually, the developer of the system), who selects the grammatical structures that look good to him/her. The advantage of this approach is that, since only unstructured data is necessary as input, it can be applied to different languages. However, since the evaluation is not done by an independent expert, it is difficult to get objective evaluations.

- *Rebuilding Known Grammars*: a (simple) grammar is selected beforehand. The input to the algorithm is a set of example sentences generated with this grammar. The grammar learned (i.e., output of the algorithm) is compared to the original grammar. If the result is equal or similar to the original grammar, then the learning system is considered good. In contrast to the "Looks-Good-to-Me" technique, the evaluation can be done automatically, leading to more objective evaluations. However, one of the disadvantages of this technique is that this kind of evaluation strongly depends on the grammar chosen beforehand.

- *Compare against Treebank*: a set of plain natural language sentences is extracted from a treebank (i.e., a linguistic corpus in which sentences are annotated with their syntactic structure, often represented in the form of a tree), which is selected as the "gold standard". This plain corpus of sentences are given to the algorithm as input. Then, the GI algorithm generates structured sentences and these sentences are compared against the original structured sentences from the treebank. Different metrics can be used to do this comparison, but the most used are *precision* (which shows how many learned structures are correct, describing in that way the correctness of the learned grammar) and *recall* (which shows how many of the correct structures have been learned, giving in that way the completeness of the learned grammar). Another metric is the f_1 *score*, which can be considered as a weighted average of the precision and recall metrics. The main disadvantage of this technique is that structured corpora are needed. This could be a problem when we try to evaluate natural languages for which the underlying grammar is not known (it would be necessary to build a treebank for that language).

The most common method to evaluate GI systems is the *Compare against Treebank* technique. There exist different treebanks, but the most used in GI are:

- *Air Traffic Information System (ATIS) treebank* (Marcus et al., 1993): English corpus that contains 577 sentences on air traffic.

- *OpenbaarVervoer Informatie Systeem (OVIS) treebank* (Bonnema et al., 1997): Dutch corpus that contains 10.000 sentences on travel information (train schedules in the Netherlands).

- *Wall Street Journal (WSJ) treebank* (Marcus et al., 1993): English corpus that contains sentences extracted from newspaper articles.

Examples of GI systems for natural language learning are: EMILE, ABL and ADIOS. They mainly use the ATIS and/or OVIS corpus for the evaluation. We review next some of the main ideas of these systems. For detailed information about these algorithms, other GI methods and comparisons between them, see (D'Ulizia et al., 2011).

EMILE was proposed by P. Adriaans in the 90's (Adriaans, 1992) (several improved versions of the algorithm have been developed later). This GI method uses a *supervised approach* and only *positive data*; there is a teacher that generates sentences that are grammatically correct and the learner can ask to the teacher if a certain sentence is correct. Following (D'Ulizia et al., 2011), we can distinguish five main steps in EMILE algorithm:

- *first order explosion*: the input to the learner (or learning algorithm) is a set of sentences grammatically correct. The learner tries to discover the different subexpressions of each sentence. For instance, given the sentences “Bob calls Alice” and “Alice works”, a possible set of subexpressions is the following: “Bob”, “Bob calls”, “Bob calls Alice”, “calls Alice”, “Alice”, “Alice works”, “works”.

- *verification*: the learner tries to verify substitutions of all expressions in each context, by asking to the teacher whether or not each subexpression is a correct sentence.

- *clustering*: this step consists in clustering context rules into types. Expressions that can be substituted in the same contexts receive the same type label. For example, the context rules *S/calls Alice* → *Bob* and *S/calls Alice* → *Alice* imply that Bob and Alice can be substituted in the same context and consequently they belong to the same type A.

- *rule induction*: the rules (associated to specific types during the clustering step) are generalized toward more general types and, in that way, new grammar rules are introduced. For example, the grammar rules *S/calls Alice* → *A* and *A* → *Bob* | *Alice* are introduced.

- *rule rewriting*: the rules are rewritten with the outcome of producing CF grammar rules.

ABL (Alignment-Based Learning) was proposed by M. van Zaanen in (van Zaanen, 2001). This GI method uses an *unsupervised approach* and only *positive data*; the ABL algorithm learns syntactic structures from positive sentences without any a priori knowledge of the language. It is based on Harri’s substitutability idea, which states that if two constituents are of the same type, then they can be substituted by each other. Following (D’Ulizia et al., 2011), we can distinguish three main phases in ABL algorithm:

- *alignment learning*: the first phase finds possible constituents by aligning all sentences in the input corpus and finding the group of words that are different in all pairs of sentences. A reverse version of Harri’s implication is used to search for constituents: if parts of sentences can be substituted by each other then they are constituents of the same type. For example, given the sentences “Book Delta 128 from Dallas to Boston” and “Give me all flights from Dallas to Boston”, the group of words that are different are considered constituents, that is, “Book Delta 128” and “Give me all flights”.

- *selection learning*: the second phase eliminates overlapping constituents found in the previous phase and select the best constituents by using probabilistic methods. The output of this step is a structured tree-bank.

- *grammar extraction*: the last phase extracts a stochastic grammar from this tree-bank.

ADIOS (Automatic DIstillation Of Structure) was proposed by Z. Solan et al. in (Solan et al., 2005). This GI uses an *unsupervised approach* and only *positive data*. It takes a set of correct sentences as input, and outputs a grammar for that language by using a statistical method. Following (D’Ulizia et al., 2011), we can distinguish three main steps in ADIOS algorithm:

- *initialization*: the first step is to load the corpus onto a directed pseudograph (i.e. a non-simple graph in which both loops and multiple edges are permitted). Every unique word in the corpus is represented in the graph as a node. Every sentence is represented by a path over the graph, starting at a special node called the *begin node*, following the graph edges and traversing the nodes corresponding to the words in the order in which appear in the sentence, and ending at a second special node called *end node*. Every path generates exactly one sentence.

- *pattern distillation*: the second step consists in the extraction of significant patterns (i.e. sequences of nodes) from the pseudograph by finding sub-paths of high probability, considering the number of outgoing and incoming edges of a sub-path. A set of candidate patterns are generated.
- *generalization*: the third step looks for finding the most significant pattern, then generalizes over the graph by creating equivalence classes from all of the variable node elements in the pattern.

In GI there has also been some efforts for taking into account more natural aspects during the language learning process, like for instance the correspondence between sentences and the context in which these sentences are produced (children acquire language not just from sentences, but from sentences in a particular context). An example of it is (Angluin and Becerra-Bonache, 2008; Angluin and Becerra-Bonache, 2010; Angluin and Becerra-Bonache, 2011). The purpose of these works was not to learn a CF grammar, but to investigate the effect of semantics and corrections in the process of learning to understand and speak a natural language. In fact, in the early stages of children's linguistic development, semantic information seem to play an important role (Chouinard and Clark, 2003). Taking into account that most works in GI have only been focused on syntax learning and results in GI show that language learning is hard, the following question was posed: can semantic information simplify the learning problem? In order to answer this question, a simple computational model was developed which takes into account semantics for language learning. The model was tested with ten different natural languages by using a simplified version of the Miniature Language Acquisition task (this task involves sentences that describe visual scenes), and the results show that:

- Access to semantic information facilitates language learning.
- The presence of meaning-preserving corrections has an effect of language learning, even if the learner does not treat them specially.

Therefore, the results obtained with these works were not only of interest for GI, but also for the linguistic community.

Researchers in GI have also developed methods for other tasks, such as machine translation. For example, we can find several works focused on learning *stochastic finite-state transducers* (SFST) for machine translation (Casacuberta and Vidal, 2007). A SFST involves two different alphabets (source and target alphabet), and associates probabilities to the transitions and final states. The main advantages of working with these models are the following:

- There exist efficient search algorithms for translation.
- These techniques are less computational expensive (in general) than most pure statistical approaches.
- They allow an easy integration with other information sources, such as acoustics models, making easier the applications of SFST to more difficult tasks, such as speech translation.

It is worth noting that these methods have been successfully applied to different non-trivial tasks, such as Miniature Language Acquisition task, Traveler task, etc. For more information, the reader is refered to (Casacuberta and Vidal, 2007).

3. Agents and Formal Languages

In this section, we consider the linguistic applications of models based on the interrelation between *multi-agent systems* and *formal language theory* (Rozenberg and Salomaa, 1997).

In order to understand the role of formal language theory in linguistics, we have to go back to the origins of this field. The area of formal languages was born in the middle of the 20th century. The field emerged as an interdisciplinary area, relating two disciplines that usually are viewed as totally opposed: *mathematics* and *linguistics*. In the area of mathematics, the key names are A. Thue, E. Post and A. Turing. Thue (Thue, 1906; Thue, 1912) and Post (Post, 1936) introduced the formal notion of a rewriting system, while Turing (Turing, 1936) introduced the general idea of finding models of computing where the power of a model could be described by the complexity of the language it generates/accepts. From linguistics the name is Chomsky (Chomsky, 1956). Building on the work by Thue, Turing and Post, Noam Chomsky started in the 1950s the study of grammars and the grammatical structure of a language. Chomsky's goal was to give a precise characterization of the structure of natural languages. He wanted to define the syntax of languages using simple and precise mathematical rules. In order to reach this goal, he introduced his grammar hierarchy as a tool for modelling natural languages. In fact, formal language theory may be considered as one of the best examples of an active area of mathematical research whose origins lie in linguistics. The first generation of formal grammars, based on rewriting, formalized classical computing models. The view of natural languages as formal languages played a significant role in the development of linguistics in the second half of the 20th century. The idea of languages as sets of strings underlay the early development of generative grammar (Newmeyer, 1997). It was a period dominated by the interest on syntax, and the formalism introduced by Chomsky was considered a good mathematical approach to solve the problem of approaching natural language from a formal point of view.

From the 60s to the 80s, linguistics was the central application of formal language theory and linguists were very much interested in applying formal language models to the formalization of natural language. On the contrary, from the 90s, the interest of linguistics in formal languages seems to have disappeared and formal language theorists have found innumerable applications of their theory different from linguistics. The reasons that explain this separation of linguistics from formal language theory could be very different; among them we stress the following ones:

1. Problems that linguists found when trying to describe natural language by the classical theory of formal languages. For example, the difficulty of locating natural language in the Chomsky hierarchy and, therefore, the necessity of defining different new formalisms.

2. The growing interest on less theoretical/formal areas of linguistics –as *cognitive linguistics* (Geeraerts and Cuyckens, 2010)– and, therefore, the need for looking for more natural computational systems to give account of natural language processing. Rewriting methods used in natural language approaches based on formal languages seemed to be not very adequate, from a cognitive perspective, to account for the processing of language.

3. The importance gained in the field of linguistics of less abstract subdisciplines such as *semantics* (Lappin, 1997), *pragmatics* (Horn and Ward, 2005) or *sociolinguistics* (Bayley et al., 2013). Formal language theory was very useful in linguistics when the main interest of this discipline was to describe

the syntactical competence of speakers, this is, the knowledge of the syntax of a natural language. When linguists tried to approach dynamical parts of natural language that depends on the context of use, classical models of formal languages became too rigid.

Those problems related to the first generation of formal languages based on rewriting systems were the reason for the divorce between formal languages and linguistics. However, models proposed from the 90s in the area of formal language theory may solve those classic problems. Among those new models we find the *agent-based models* of formal languages that constitute an important subfield of the theory. While in classic formal language theory, grammars and automata modelled classic computing devices where computation was accomplished by one central agent, new models in formal languages take into account the idea of modern computer science where distributed computation plays a major role.

The idea of several devices collaborating for achieving a common goal was formalized in the theory of *colonies* (Kelemen and Kelemenová, 1992), in *grammar systems* (Csuhaj-Varjú et al., 1994) and in *eco-grammar systems* (Csuhaj-Varjú et al., 1996). Those models are examples of a new generation of formal languages that defines grammatical models of agent systems. These multi-agent formal languages have been applied to natural language description and processing. In general, it can be shown that those non-standard models in formal languages can solve the classical problems related to the first generation of formal languages and can cover the whole range of linguistic disciplines, from phonology to pragmatics. In the next section, we show some examples of those possible linguistic applications.

3.1 Linguistic Applications of Agent-based Formal Language Theory

3.1.1 Colonies

Colonies are the first agent-based model we consider. Colonies as well-formalized language generating devices have been proposed in (Kelemen and Kelemenová, 1992), and developed during the nineties in several directions. Colonies can be thought of as grammatical models of multi-agent systems motivated by Brooks' subsumption architectures (Brooks, 1990). They describe language classes in terms of behavior of collections of very simple, purely reactive, situated agents with emergent behavior. Colonies, as proposed originally, capture some formal aspects of systems of finite number of autonomous components capable to perform very simple reactive computing tasks. The behaviour of the colony really emerges from interactions of its components with their symbolic environment and can considerably surpass the individual behaviours of its components. The main advantage of colonies is their generative power, the class of languages describable by colonies that make use of strictly regular components is beyond the set describable in terms of individual regular grammars.

In (Bel-Enguix et al., 2009), colonies have been proposed as a tool to generate natural language by the interaction of a finite number of finite-state devices that generate finite languages. This application takes into account the idea of describing natural languages as a number of modules that interact in a nonsimple way. Colonies offer a modular theory where the various dimensions of linguistic representation may be arranged in a distributed framework and where the language of the system is the result of the interaction of those independent cooperative modules. Therefore, colonies allow us to generate natural language by the interaction of a finite number of finite-state devices that generate finite languages.

In computational linguistics, finite-state devices have been used for phonology, morphology, lexicon and syntax of natural languages. The advantages of approaching such parts of natural languages with finite-state devices have been demonstrated. So, from a practical point of view, finite-state descriptions of natural languages have revealed as adequate. The problems of using such devices arise when we consider natural languages as a whole, pointing our attention on the complexity of such an object and on its infinite nature. With colonies we can define finite-state devices for morphology, phonology, syntax, lexicon and any other required module in language. We, then, place those finite-state devices in the colony, and they interact in order to generate natural language. The result of the interaction is, not a regular language, but a language that is context-free or more. This formal framework increases the power of regular grammars thanks to interaction. What is important here is the fact that although the generative power of colonies goes beyond the regular family of languages, the derivation process is done in a regular (finite-state) manner. Therefore, colonies may reveal as a device able of conjoining the simplicity of finite-state machines with a stronger generative power able to account for the infiniteness (context-free or more) of natural languages.

3.1.2 Grammar Systems

Another important agent based-model in formal languages are the so-called *grammar systems* (Csuhaj-Varjú et al., 1994). Grammar system theory is a consolidated and active branch in the field of formal languages that provides syntactic models for describing multi-agent systems at the symbolic level, using tools from formal grammars and languages. A grammar system is a set of grammars working together, according to a specified protocol, to generate a language. Note that while in classical formal language theory *one* grammar (or automaton) works individually to generate (or recognize) *one* language, here we have *several* grammars working together in order to produce *one* language. The theory was launched in 1988 and has developed into several directions, motivated by several scientific areas. Easy generation of non-context-free structures using context-free rules, modularity, parallelism, interaction, distribution, and cooperation are just some of the advantages that grammar systems have over classical models

Grammar systems may offer useful tools to account for arrangement and interaction of the various dimensions of natural language grammar. In order to define a grammar system approach for grammar architecture, a set of postulates of linguistic theory must be followed (Sadock, 1991):

1. we take a grammar to be a set of subgrammars called modules;
2. each of these modules is a grammar of an independent level of linguistic representation;
3. these modules are not hierarchically related to one another;
4. a module need not wait for the output of another to do its work, but has the power to generate (analyze) an infinite set of representations quite independently of what is going on in any of the other components;
5. each component is a self-contained system, with its own independent set of rules, principles and basic vocabulary;
6. the lexicon plays a special, transmodular role in the theory.

In order to capture every feature of the above list, a new variant of grammar systems has been introduced: *Linguistic Grammar Systems* (LGS) (Jiménez-López, 2006). Since the basic definition of grammar systems would be too simple to fully account for the intricate arrangement of components in grammar, LGS were defined as an alternative model which, by combining several properties of different variants of grammar systems, may provide an adequate description of grammar architecture. LGS is defined as a model of parallel grammatical representations where different modules work independently and collaborate in order to generate a well-formed linguistic structure. LGS offers a highly modular general device in which properties such as modularity, parallelism, interaction, coordination, and distribution can be captured.

LGS features fit with the idea of explaining the functioning of natural language as a number of modules that interact in a nonsimple way. Natural language understanding and generation needs cooperative phonological, morphological, lexical, syntactic, pragmatic, semantic... modules. If we adopt a *modular* strategy to write our grammar model we will obtain advantages such as the following:

1. a complex task is divided into convenient and manageable subtasks;
2. we deal with one structure, or set of structures, at one time;
3. each module can be written and tested in isolation;
4. minor changes and/or additions can be introduced in one module if necessary, without having to modify the whole grammar.

On the other hand, in general, formal and computational approaches to natural language demand non-hierarchical, parallel, distributed models in order to explain the complexity of linguistic structures as the result of the interaction of a number of independent but cooperative modules.

Linguistic grammar systems offer a modular theory where the various dimensions of linguistic representation are arranged in a parallel distributed framework and where the language of the system is the result of the interaction of those independent cooperative modules that form the LGS. Therefore, *linguistic grammar systems* provide a suitable framework according to the requirements of most of the current formal/computational approaches to natural language. Moreover, LGS offer a highly formalized model that has several advantages with respect to classical formal languages and that can be easily implemented, two important features for the computational treatment of natural language. For formal definitions of LGS the reader can see (Jiménez-López, 2006).

3.1.3 Eco-Grammar Systems

Very relevant in the area of multi-agent models of formal languages are the so-called *eco-grammar systems* (Csuhaj-Varjú et al., 1996). Eco-grammar systems provide a syntactical framework for eco-systems, this is, for communities of evolving agents and their interrelated environment. An eco-grammar system is defined as a multi-agent system where different components, apart from interacting among themselves, interact with a special component called 'environment'. Within an eco-grammar system we can distinguish two types of components *environment* and *agents*. Both are represented at any moment by a string of symbols that identifies the current state of the component. These strings change according to sets of evolution rules. Interaction among agents and environment is carried out through agents' actions performed on the environmental state by the application of some productions from the set of action rules of agents.

Among the multiple linguistic applications of this formal model, we highlight two:

1. eco-grammar systems application to *dialogue modelling*;
2. eco-grammar systems application to the *modelling of language evolution*.

According to the idea that dialogue can be understood as the sustained production of mutually-dependent acts, constructed by two or more agents each monitoring and building on the actions of the other, eco-grammar systems can describe dialogue as a sequence of *acts* performed by two or more agents in a common environment. An example of this application is presented in (Bel-Enguix and Jiménez-López, 2008). In this paper, a formal model of dialogue based on eco-grammar systems is introduced: *Conversational Grammar Systems* (CGS). Cooperation, coordination, emergence, dynamism, flexibility, and capacity to capture the main elements and mechanisms in a dialogue make of CGS a conversational-like model that emulates human linguistic behaviour in some specific situations such as natural language interfaces.

In a CGS, the essential structure in dialogue can be well and easily formalized. The most important elements and mechanisms at work in dialogue are modelled by means of formal tools. The dialogue setting has been defined as one in which there are two elements: *context* and *agents*. Both are described at any moment of time by a string of symbols over a specific alphabet. The functioning of the system –dialogue development– is understood in terms of state changes. Both context and agents change their state in the course of dialogue. A change in agents' strings is due to the updating of the agents' state according to what is happening in the dialogue. On the other hand, modifications of the context string are due both to agents' actions and to its own evolution. In order to allow agents to change the state of the context, they are endowed with sets of action rules. The action to be performed at any moment of time is selected according to both the state of the context and the state of the agent itself. In this way, CGS guarantee that agents act appropriately, in accordance with what is required by the state of dialogue at a given moment. Dialogue proceeds by alternating between action and evolution steps. At the end, CGS provide a sequence of states, that is, a series of states that have been reached, from the initial state, during the dialogue interchange. Two formal tools, derivation modes and stop/start conditions, capture –in a very simple way– the turn-taking found in dialogue. On the other hand, some formal counterparts of adjacency pairs and similar notions are defined in order to constrain the set of allowable actions according to the current state of the context. Moreover, different ways of closing a dialogue are defined. Coherence, dynamism and emergence in dialogue are also preserved in the model. The lack of any external control make of CGS a dynamic, flexible and emergent framework that accounts for the unplanned and opportunistic nature of conversation.

Flexibility and interdisciplinarity are two of the most important features of CGS. The formal apparatus of CGS guarantees a high degree of flexibility what means that CGS accept new concepts and modify rules, protocols and settings during the computation. Interdisciplinarity is on the basis of CGS: to define this formal model we combine ideas from linguistics, specifically from the area of conversational analysis, with theories coming from theoretical computer science and formal language theory in order to define a dialogue system that may contribute to the building of better human-computer dialogues through a simulation of human language use.

Considering the adequacy of agent systems for the simulation of language evolution, in (Jiménez-López, 2012) we introduce a formal-language-theoretic multi-agent model based on eco-grammar systems that may account for language change: *Cultural Grammar Systems*.

Taking into account the different evolutionary processes considered in the origins and evolution of language, in a Cultural Grammar System we distinguish five elements:

1. a *physical environment* (to account for socio-ecological evolution);
2. a *genetic system* (to account for biological evolution);
3. a set of *cultures* (to account for cultural evolution);
4. a set of *subcultures*;
5. and a set of *agents*.

Those elements are related as follows:

- The set of *agents* belong to a specific *cultural environment* and are constrained by their *genetic* properties as well as by the *physical environment* where they are living. An intermediate level between cultures and agents is introduced: *subcultures*. This further division can represent several things: socioeconomic classes, religions, castes, dialects, etc. The criteria by which subgroups are identified as strata usually depend on variables such as the trait under study.
- The genetic system and culture are not two independent inheritance systems, but they are interdependent. Those two systems are able to influence or modify each other. A change (usually very rapid) in culture can cause a (slower) change in gene frequency. Therefore, culture can alter somehow the direction and/or rate of genetic evolution. On the other hand, the genetic constitution of the individuals composing a society influences the nature of that society, so the genetic system is capable of modifying or conditioning changes in culture.
- Agents may be influenced by the physical environment where they live.

In order to reflect all those relationships, cultural grammar systems: 1) relate genetic system to agents, in order to capture the influence that biological environment has in the evolution of agents; 2) relate subculture to agents, in order to indicate cultural constraints on agents' actions. 3) and finally, relate the physical environment to the whole system, in order to put in evidence that physical environment constrains evolution of genetic system as well as evolution of agents. And, of course, by influencing evolution of agents, it constrains indirectly changes on cultural environments, because agents and only agents are the responsible for cultural evolution.

The dynamics of the system reflects the ideas recurrently found in the literature of cultural change. There exists a difference in the rates of evolution between cultural and biological systems. Cultural systems evolve faster than genetic ones. Culture is man-made. Man is the responsible for creation and changes in cultural systems. According to this, in cultural grammar systems, culture evolves faster than the genetic system, and it evolves thanks to direct action of agents. In contrast, genetic system evolves in a slower fashion and admits little –or none– direct modification on behalf of man.

Cultural Grammar Systems accounts for the different adaptive systems identified in the literature –individual learning, cultural transmission and biological evolution (Christiansen and Kirbi, 2003)–, since it introduces: *agents* with evolution rules that describe the individual learning and show how they evolve according to the dynamics of the whole system; a *genetic system* that accounts for biological evolution; and *cultures* that describe the cultural transmission and evolve according to agents needs.

The improvement of computational techniques have led many researchers to use computational modeling to understand the origin and evolution of language. Computer simulations have become widespread in the last years (Bartlett and Kazakov, 2004; Cangelosi, 2007; Cangelosi and Parisi, 2001; Reitter and Lebiere, 2010; Steels, 2006). Many of those simulations have been carried out within multi-agent settings. Cultural Grammar Systems have been defined by taking into account the adequacy of multi-agent systems for the simulation of language evolution and following the research line opened by *artificial life* (Langton, 1989) where organisms have been simulated using symbolic models.

4. Conclusions

We have considered three different areas: *agent systems, machine learning* and *formal languages*. Each of those fields has intrinsically good features that enable them to cope with many real-world problems. The formal apparatus of agent technology provides a powerful and useful set of structures and processes for designing and building complex applications. Machine learning is one of the core fields of Artificial Intelligence since the ability to learn is one of the most fundamental attributes of intelligent behavior. And finally, formal language theory provides the flexibility and the abstraction necessary in order to be applied to fields such as linguistics, economic modeling, developmental biology, cryptography, sociology, etc. Therefore, multi-agent systems, machine learning and formal language theory provide flexible and useful tools that can be used in different research areas due to their versatility. In this paper, we have tried to show that the individual power of those systems may be increased if they collaborate among them.

In this work, we have focused on two possible intersections: formal languages and learning; and, agents and formal languages. Our main objective here has been to show the contributions of those intersections to the study of natural language. The interaction between researchers in those three topics can provide good techniques and methods for improving our knowledge about how languages are processed.

Language is one of the most challenging issues that remain to be explained. As a complex system, the explanation, formal modelling and simulation of language present important difficulties. If we deal with language, we need to connect and integrate several academic disciplines in order to find a solution. In this interdisciplinary environment, formal languages, machine learning and agents systems can collaborate –as shown in this paper– in the description, explanation and processing of language.

In this paper, we have focused our attention in the role that the interrelation among agents technologies, machine learning and formal languages can play in the field of linguistics. However, the interdisciplinarity we defend is bi-directional, this is, we think that models in formal language theory, in agent technology and in machine learning can benefit from linguistic theories as well. Formal language theory would find in linguistics a ‘natural’ inspiration for defining new models that could become the theoretical basis for future computational systems that will, for sure, improve our interaction with computers. Agent technology could benefit, for example, from the coordination protocols followed in natural language conversation and explained in the areas of pragmatics and conversational analysis. And machine learning could improve its algorithms by artificially simulating the process of learning a natural language. Therefore, we would like to call the attention of researchers working in those fields about the possibilities that the human processing of language -and its study in the field of linguistics– offers as motivation/inspiration for new frameworks in their research areas.

5. References

Adriaans, P., 1992. *Language learning from a categorial perspective*. Ph.D. thesis, University of Amsterdam.

Angluin, D., 1982. Inference of reversible languages. *Journal of the Association for Computing Machinery*, 29(3):741–765.

Angluin, D. and Becerra-Bonache, L., 2008. Learning Meaning Before Syntax. In *ICGI: International Colloquium on Grammatical Inference*, pages 1–14. Berlin.

Angluin, D. and Becerra-Bonache, L., 2010. *A model of semantics and corrections in language learning*. YALEU/DCS/TR-1425.

Angluin, D. and Becerra-Bonache, L., 2011. Effects of Meaning-Preserving Corrections on Language Learning. In *CoNLL: International Conference on Computational Natural Language Learning*, pages 97–105.

Bartlett, M. and Kazakov, D., 2004. The role of environment structure in multi-agent simulations of language evolution. In *Proceedings of the Fourth Symposium on Adaptive Agents and Multi-Agent Systems*. Springer.

Bayley, R., Cameron, R., and Lucas, C., 2013. *The Oxford Handbook of Sociolinguistics*. Oxford University Press, Oxford.

Becerra-Bonache, L., 2006. *On the Learnability of Mildly Context-Sensitive Languages using Positive Data and Correction Queries*. Ph.D. thesis, Rovira i Virgili University.

Becerra-Bonache, L., Case, J., Jain, S., and Stephan, F., 2010. Iterative learning of simple external contextual languages. *Theoretical Computer Science*, 411:2741–2756.

Becerra-Bonache, L. and Yokomori, T., 2004. Learning Mild Context-Sensitiveness: Toward Understanding Children's Language Learning. In *ICGI: International Colloquium on Grammatical Inference*, pages 53–64.

Bel-Enguix, G. and Jiménez-López, M., 2008. Modelling dialogue as inter-action. *International Journal of Speech Technology*, 11(3/4):209–221.

Bel-Enguix, G., Jiménez-López, M., and Martín-Vide, C., 2009. Using finite-state methods for getting infinite languages: A preview. *Romanian Journal of Information, Science and Technology*, 12(2):125–137.

Bonnema, R., Bod, R., and Scha, R., 1997. A DOP Model for Semantic Interpretation. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 159–167.

Bresnan, J., Kaplan, R., Peters, S., and Zaenen, A., 1987. Cross-Serial Dependencies in Dutch. In Savitch, W., Bach, E., Marsh, W., and Safran-Naveh, G., editors, *The Formal Complexity of Natural Language*, pages 286–319. D. Reidel, Dordrecht.

Brooks, R., 1990. Elephants don't play chess. *Robotics and Autonomous Systems*, 6:3–15.

Cangelosi, A., 2007. Adaptive agent modeling of distributed language: investigations on the effects of cultural variation and internal action representations. *Language Sciences*, 29(5):633–549.

Cangelosi, A. and Parisi, D., 2001. Computer simulation: A new scientific approach to the study of language evolution. In Cangelosi, A. and Parisi, D., editors, *Simulating the Evolution of Language*, pages 3–28. Springer.

Casacuberta, F. and Vidal, E., 2007. Learning finite-state models for machine translation. *Machine Learning*, 66(1):69–91.

Chomsky, N., 1956. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124.

Chouinard, M. and Clark, E., 2003. Adult Reformulations of Child Errors as Negative Evidence. *Journal of Child Language*, 30:637–669.

Christiansen, M. and Kirbi, S., 2003. Language evolution: Consensus and controversy. *Trends in Cognitive Sciences*, 7(7):300–307.

Clark, A. and Yoshinaka, R., 2014. Distributional learning of parallel multiple context-free grammars. *Machine Learning*, 96(1-2):5–31.

Csuhaj-Varjú, E., Dassow, J., Kelemen, J., and Păun, G., 1994. *Grammar systems: A grammatical approach to distribution and cooperation*. Gordon and Breach, London.

Csuhaj-Varjú, E., Kelemen, J., Kelemenová, A., and Păun, G., 1996. Eco-grammar systems: A grammatical framework for life-like interactions. *Artificial Life*, 3(1):1–28.

Culy, C., 1987. The Complexity of the Vocabulary of Bambara. In Savitch, W., Bach, E., Marsh, W., and Safran-Naveh, G., editors, *The Formal Complexity of Natural Language*, pages 349–357.

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

García, P. and Vidal, E., 1990. Inference of k-Testable Languages in the Strict Sense and Application to Syntactic Pattern Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(9):920–925.

Gazdar, G. and Pullum, G. K., 1985. Computationally Relevant Properties of Natural Languages and Their Grammars. *New Generation Computing*, 3:273–306.

Geeraerts, D. and Cuyckens, H., 2010. *The Oxford Handbook of Cognitive Linguistics*. Oxford University Press, Oxford.

Gold, E., 1967. Language identification in the limit. *Information and Control*, 10:447–474.

de la Higuera, C., 2010. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press.

Horn, L. and Ward, G., 2005. *The Handbook of Pragmatics*. Blackwell, Oxford.

Jiménez-López, M., 2006. A grammar systems approach to natural language grammar. *Linguistics and Philosophy*, 29:419–454.

Jiménez-López, M., 2012. A grammar-based multi-agent system for language evolution. In Bajo Pérez, J. e. a., editor, *Highlights on Practical Applications of Agents and Multiagent Systems*, pages 45–52. Springer.

Joshi, A. K., 1985. How Much Context-Sensitivity is Required to Provide Reasonable Structural Descriptions: Tree Adjoining Grammars. In Dowty, D., Karttunen, L., and Zwicky, A., editors, *Natural Language Parsing: Psychological, Computational and Theoretical Perspectives*, pages 206–250. Cambridge University Press, New York, NY.

Joshi, A. K. and Schabes, Y., 1997. Tree-Adjoining Grammars. In Rozenberg, G. and Salomaa, A., editors, *Handbook of Formal Languages*, volume 3, pages 69–123. Springer-Verlag, Berlin.

Kelemen, J. and Kelemenová, A., 1992. A grammar-theoretic treatment of multiagent systems. *Cybernetics and Systems*, 23:621–633.

Kudlek, M., Martín-Vide, C., Mateescu, A., and Mitrana, V., 2002. Contexts and the concept of mild context-sensitivity. *Linguistics and Philosophy*, 26(6):703–725.

Langton, C., 1989. Artificial Life. In Langton, C., editor, *Artificial Life*, pages 1–47. Addison-Wesley.

Lappin, S., 1997. *The Handbook of Contemporary Semantic Theory*. Blackwell, Oxford.

Manaster-Ramer, A., 1999. Some Uses and Abuses of Mathematics in Linguistics. In Martín-Vide, C., editor, *Issues in Mathematical Linguistics*, pages 73–130. John Benjamins, Amsterdam.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A., 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Newmeyer, F., 1997. *Generative Linguistics: An Historical Perspective*. Routledge, London.

Oates, T., Armstrong, T., Becerra-Bonache, L., and Atamas, M., 2006. Inferring grammars for mildly context sensitive languages in polynomial-time. In *ICGI: International Colloquium on Grammatical Inference*, pages 137–147.

Oncina, J. and García, P., 1992. Identifying Regular Languages In Polynomial Time. In *Advances in Structural and Syntactic Pattern Recognition*, pages 99–108. World Scientific.

Post, E., 1936. Finite combinatory processes-formulation. *Journal of Symbolic Logic*, 1:103–105.

Reitter, D. and Lebiere, C., 2010. Did social networks shape language evolution? A Multi-Agent Cognitive Simulation. In *Proceedings of ACL 2010*, pages 9–17. ACL.

Roach, K., 1987. Formal Porperties of Head Grammars. In Manaster-Ramer, A., editor, *Mathematics of Language*, pages 293–348. John Benjamins, Amsterdam.

Rozenberg, G. and Salomaa, A., 1997. *Handbook of Dormal Languages*. Springer, Berlin.

Sadock, J., 1991. *Autolexical syntax. A theory of parallel grammatical representations*. University of Chicago Press, Chicago.

Sakakibara, Y., 1992. Efficient learning of context-free grammars from positive structural examples. *Information Processing Letters*, 97:23–60.

Shieber, S., 1987. Evidence Against the Context-Freeness of Natural Languages. In Savitch, W., Bach, E., Marsh, W., and Safran-Naveh, G., editors, *The Formal Complexity of Natural Language*, pages 320–334. D. Reidel, Dordrecht.

Solan, Z., Horn, D., Ruppin, E., and Edelman, S., 2005. Unsupervised learning of natural languages. In *Proceedings of the National Academy of Sciences of the USA*, volume 102(33), pages 11629–11634.

Steedman, M., 1985. Dependency and Coordination in the grammar of Dutch and English. *Language*, 61:523–568.

Steels, L., 2006. How to do experiments in artificial language evolution and why. In *Proceedings of the 6th International Conference on the Evolution of Language*, pages 323–332.

Thue, A., 1906. Über unendliche Zeichenreihen. *Norske Vid. Selsk. Skr., I Mat. Nat. Kl., Kristiania*, 7:1–22.

Thue, A., 1912. Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen. *Norske Vid. Selsk. Skr., I Mat. Nat. Kl., Kristiania*, 1:1–67.

Turing, A., 1936. On computable numbers with an application to the Entscheidungsproblem. In *Proceedings London Mathematical Society*, pages 230–265.

Yoshinaka, R., 2009. Learning Mildly Context-Sensitive Languages with Multidimensional Substitutability from Positive Data. In *ALT: International Conference on Algorithmic Learning Theory*, pages 278–292.

van Zaanen, M., 2001. ABL: Alignment-Based Learning. *CoRR*, cs.LG/0104006.

van Zaanen, M., Roberts, A., and Atwell, E., 2004. A multilingual parallel parsed corpus as gold standard for grammatical inference evaluation. In *LREC: Workshop on The Amazing Utility of Parallel and Comparable Corpora*, pages 58–61.