

A Gold Standard to Measure Relative Linguistic Complexity with a Grounded Language Learning Model

Leonor Becerra-Bonache
Univ. Lyon, UJM-St-Etienne
Saint-Étienne, France

leonor.becerra@univ-st-etienne.fr

Henning Christiansen
Roskilde University
Roskilde, Denmark
henning@ruc.dk

M. Dolores Jiménez-López
Universitat Rovira i Virgili
Tarragona, Spain
mariadolores.jimenez@urv.cat

Abstract

This paper focuses on linguistic complexity from a relative perspective. It presents a grounded language learning system that can be used to study linguistic complexity from a developmental point of view and introduces a tool for generating a gold standard in order to evaluate the performance of the learning system. In general, researchers agree that it is more feasible to approach complexity from an objective or theory-oriented viewpoint than from a subjective or user-related point of view. Studies that have adopted a relative complexity approach have showed some preferences for L2 learners. In this paper, we try to show that computational models of the process of language acquisition may be an important tool to consider children and the process of first language acquisition as suitable candidates for evaluating the complexity of languages.

1 Introduction

In this paper, we propose to use a grounded language learning model for measuring the relative complexity of natural languages.

Complexity is a controversial concept in linguistics. Eventhough, natural language complexity has been extensively studied for almost two decades –starting with McWhorter (2001) paper published in *Linguistic Typology*–, it still not clear how complexity has to be defined and measured. The equi-complexity dogma, which stated that the total complexity of a natural language is fixed because sub-complexities in linguistic sub-systems trade off, has been almost completely debunked. We have seen what Joseph and Newmeyer (2012) name the “decline in popularity of the equal complexity principle”. This situation has led to the proposal of many models, tools and criteria to quantify the level of complexity of languages (Dahl, 2004; Kusters, 2003; Miestamo et al., 2008; Sampson et al., 2009; Newmeyer and Preston, 2014). However, currently, there is no clear solution to measure linguistic complexity and each of the proposed models has advantages and disadvantages.

Criteria and measures of complexity remain unsolved and this may be due to the fact that there is no agreement about how to define complexity. Instead, in the literature, we can find a variety of approaches that has led to a linguistic complexity taxonomy: absolute complexity vs. relative complexity; global complexity vs. local complexity; system complexity vs. structural complexity, etc. With this diversity of definitions, measures and criteria to calculate complexity vary and depend on the specific research interests and on the definition of complexity adopted.

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: <http://creativecommons.org/licenses/by/4.0/>

In this paper, we adopt a relative approach to complexity. Specifically, from the following three different meanings of complexity that Pallotti (2015) identifies in the linguistic literature, we focus on the third one:

1. *Structural complexity*, a formal property of texts and linguistic systems having to do with the number of their elements and their relational patterns.
2. *Cognitive complexity*, having to do with the processing costs associated with linguistic structures.
3. *Developmental complexity*, the order in which linguistic structures emerge and are mastered in second (and, possibly, first) language acquisition.

From the two possibilities offered by the developmental meaning of complexity, we work on the second one, this is, we intend to calculate linguistic complexity by considering a child learner in the process of first language acquisition.

In order to calculate the relative complexity of language by taking into account the process of acquiring a language, we propose to use a computational model for first language acquisition. Specifically, we have chosen a machine learning model, since this kind of models deal with idealized learning procedures for acquiring grammars on the basis of exposure to evidence about languages (D’Ulizia et al., 2011).

In section 2, we introduce a language learning system to calculate linguistic complexity. The adequacy of this model for measuring linguistic complexity from a developmental point of view is based on the fact that the computational models developed in the area of grounded language learning can be useful for studying first language acquisition. An important advantage of grounded language learning tools is that they allow us to reproduce the learning context of first language acquisition. In fact, in these models we provide data to a learner, and a learner (or learning algorithm) must identify the underlying language from this data. This process has some similarities with the process of language acquisition where children receive linguistic data and from them they learn their mother tongue.

The model calculates the number of interactions that are necessary to achieve a good level of performance in a given language by using a unique algorithm to learn any language. Therefore, it allows us to calculate the *cost* –in terms of the number of interactions– to reach a good level of performance in a given language and offer the possibility to measure the *difficulty* of acquiring different natural languages, since it may show that not all the languages need the same number of linguistic interactions to reach the same level of performance.

Therefore, the grounded learning system introduced in section 2 may be a potential adequate tool to measure the linguistic complexity in *relative* terms. In fact, the unique algorithm used in the model to learn any natural language could be seen as somehow equivalent to the innate capacity that allows humans to acquire a language. Moreover, the learner –this is, the machine– has no previously knowledge about the language. The machine represents, therefore, the child that has to acquire a language by just being exposed to this language. To count the needed number of interactions for the machine to achieve a good level of performance in a specific domain of the language may be equivalent to calculate the child’s cost/difficulty to acquire a language. Finally, to show that with the same algorithm not every language requires the same number of interactions may be interpreted (in terms of complexity) as an evidence to defend that the difficulty/cost to acquire different languages is not the same and, therefore, languages differ in relative complexity.

One of the main problems in order to use this language learning model to calculate linguistic complexity is the evaluation of the performance of the system. Two measures will be used to evaluate the learning system: correctness and completeness. The *correctness* is the amount of the learner’s sentences that are in the set of sentences that denote correctly one object. The *completeness* is the amount of sentences that denote correctly one object and appear in the set of learner sentences. The problem with the model we use is that it is not trivial to specify which is the set of correct denoting sentences, this is, there is not a gold standard to evaluate the model. In order to solve this problem, in section 3 we present a language model that integrates grammar rules and contextual semantic (CS) knowledge in order to generate the

gold standard that will be used to evaluate the performance of the language learning model introduced in section 2, allowing the utilization of that model to calculate the complexity of natural language.

The model described in this paper will be used to determine the level of complexity of a set of natural languages. This model will be able to provide quantifiable experimental results that may show that languages differ in their relative complexity.

2 A grounded language learning system to study linguistic complexity

Learning a language is a challenging task that children have to face during the first years of their life. The difficulty of this task is well described by the classic example given by Quine (1960). Imagine that a linguist visits a culture with a different language than his own, and a native speaker says “gavagai” while observing a scene with a rabbit scurrying by. To understand the meaning of this word, the linguist should figure out if “gavagai” means “rabbit” or something else, such as the action performed by the rabbit or perhaps is just an expression used by the native speaker to catch his attention. Similarly, children learning their native language need to map the words they hear to their corresponding meaning in the scene they observe (Fazly et al., 2010). Hence, like in the previous example, children have to face, among others, the problem of *referential uncertainty* (i.e., they may perceive many aspects of the scene that are not related to the utterance they hear) and *alignment ambiguity* (i.e., to discover which word in the utterance refers to which part of the scene).

Taking into account all these aspects, Becerra-Bonache et al. (2015; 2016a; 2016b) developed an artificial system that, without any language-specific prior knowledge, is able to learn language models from pairs consisting of a *sentence* and the *context* in which this sentence has been produced. This type of learning is often called *grounded language learning*. This system is inspired by some research work developed by Angluin and Becerra-Bonache (2010; 2011; 2016). Note that these previous works were used in Jiménez-López and Becerra-Bonache (2016) to study the linguistic complexity of ten different natural languages, in *relative* terms (i.e., difficult/cost of learning a language).

In this section, we focus on the computational system developed by Becerra-Bonache et al. (2016a; 2016b), which uses a challenging dataset called *Abstract Scenes Dataset* (Zitnick and Parikh, 2013). It contains clip-art pictures of children playing outdoors and sentences that describe these images. This dataset was created using Amazon’s Mechanical Turk (AMT). First, AMT workers were asked to create scenes from 80 pieces of clip art depicting a boy and a girl with different poses and facial expressions, and some other objects, such as toys, trees, animals, hats, etc. Then, a new set of workers were asked to describe the scenes using one or two sentences description; the descriptions should use basic words that would appear in a children’s book. In total, the dataset contains 10.020 images and 60.396 sentences.

One of the main advantages of using abstract scenes versus real images is that they allow to study the scene description problem in isolation, without the noise introduced by computer vision tools while detecting objects in real images. Hence, the Abstract Scenes dataset allows Becerra-Bonache et al. (2016a; 2016b) to consider a scenario with a perfect vision system and focus on the language learning problem. In Figure 1, we can see an example of a scene, how the dataset encodes the objects in the scene and some of the human-written descriptions for that scene. It is worth noting that even if we know which objects are present in the image and their position, the alignment between clip-art images and sentences is not given, that is, we do not know which actions are depicted in the image (e.g., playing, eating) and which words can be used to describe them (e.g., *s_3s.png* is called *sun*)

The system developed by Becerra-Bonache et al. (2016a; 2016b) learns from (S, I) pairs, where S is a sentence that (partially) describes an image I . A sentence is represented as a sequence of words (n-grams). For the images, a basic pre-processing step transforms the information provided by the dataset (information given to the right in Figure 1) into a context C , by using a first-order logic based representation. Thus contexts are made up of a set of ground atoms that describe properties and relationships between the objects in the image. The meaning of an n-gram is whatever is in common among all the contexts in which it can be used. It is worth noting that a context describes what the learner can perceive in the world and, in contrast to other approaches, the meaning is not explicitly represented, the learner has to discover it. Hence, the input to the system is a dataset consisting on pairs (S, C) where S is a sentence



s_3s.png	0	3	469	31	2	0
p_7s.png	1	7	178	89	2	1
hb0_10s.png	2	10	100	250	1	0
hb1_4s.png	3	4	391	248	1	1
a_4s.png	4	4	205	98	1	0
c_7s.png	5	7	87	181	1	0
t_4s.png	7	4	279	115	1	1

“Mike is wearing a hat with horns”, “Mike kicks a ball to Jenny”, “An owl sits in the tree”

Figure 1: Example of an image extracted from the Abstract Scenes Dataset, its corresponding information (to the right), and three sentences related to the image (bottom).

related to a concrete C . Using inductive logic programming techniques, the system learns a mapping between n -grams and a semantic representation of their associated meaning. Experiments showed that the system was able to learn such a mapping and use it for a variety of purposes, including identifying the elements in a context that a sentence refers to and generating sentences describing a given context. For more details about the system and the experimental results, see Becerra-Bonache et al. (2016a; 2016b).

In this paper, we propose to use the artificial system developed in Becerra-Bonache et al. (2016a; 2016b) to study the complexity of languages from a relative point of view. This system is not only linguistically well motivated (for instance, the input given to the system has similar properties to those of the input received by children from their learning environment, and the system has no previous knowledge about the language to be learnt), but also allows to perform cross-linguistic analysis (a unique algorithm is used to learn any language, which could be equivalent to the innate capacity that allows humans to acquire a language). The question is: how to calculate the difficult/cost of learning a language by using this approach?

By following previous works (Jiménez-López and Becerra-Bonache, 2016), we could calculate the linguistic complexity in relative terms by counting the number of examples needed for the system to achieve a good level of performance in a given language. To evaluate the performance of the system, two measures can be used: correctness and completeness. Given a set of correct denoting sentences for a given image, the correctness of the learner is the fraction of learner’s sentences that are in the correct denoting set, and the completeness of the learner is the fraction of the correct denoting sentences that appear in the set of learner’s sentences. The problem with this approach is to define the set of correct denoting sentences for a given image, since it does not exist a *gold standard* to evaluate the system. In the next section we present a solution to solve this problem.

3 A tool to evaluate the performance of the learning system

3.1 The language model

We use a language model that integrates grammar rules and contextual semantic (CS) knowledge. A *contextual semantic knowledge base (CSKB)* is a set of logical facts, giving a “flat” representation, cf. Hobbs (1985) and Christiansen and Dahl (2005b), which is well-suited for representing observable information about the objects, their properties and interrelationships in static scenes. Our model, that we call *Contextual Semantic Grammars* in the present paper, is symmetric with respect to deductive and abductive reasoning, implemented by standard logic programming technology. Grammar rules are given by the familiar Definite Clause Grammar notation (DCG, illustrated below) as they are available in Prolog, and the CSKBs may be represented and maintained by Constraint Handling Rules (CHR); see Christiansen and Dahl (2005a) for an introduction and Christiansen (2014) for a clarification of the theory behind this approach. We explain this by a small example; consider a Definite Clause Grammar consisting of the following, single rule, as part of a Prolog program.

```
greeting --> [roar], {present(bear)}.
```

The logical goal in the curly brackets is a condition that must hold for this rule to apply; thus analyzing the utterance `[roar]` with the `present/1` predicate given as a Prolog predicate will succeed when `present(bear)` is a fact in the Prolog program, and fail otherwise. Similarly, we can use the program to generate all possible greetings, which here would be either only `[roar]` or, if `present(bear)` is not true, the empty set. In this way, the program is used in a *deductive* way, with the semantic predicates interpreted *closed world*.

Declaring instead `present/1` as a CHR constraint predicates provides an open world interpretation, which overall leads an *abductive* analysis of given sentences. CHR is integrated with Prolog, executing in its normal top-down fashion, extended with a constraint store to which new constraints (such as `present(bear)`) are added when encountered by the Prolog interpreter; after execution, the resulting constraint store is printed out as answer; `phrase/2` is a Prolog built-in used to parse (or generate) a phrase according to the given DCG:

```
?- phrase(greeting, [roar]).
   present(bear)
```

In other words, this answer can be taken as an abductive answer – the reason – why `phrase(greeting, [roar])` can be observed.

CHR includes also rules that govern the constraints in the store. While CHR originally was intended as a language for writing constraint solvers for numerical calculations and such – in which case it would be relevant to write CHR rules that define an equation solver – we use such rules to express general *semantic world knowledge*. In the CS Grammar used for our present experiments, each objects appearing in a scene has a unique identifier and a type, so, e.g., a bear will be represented by the two constraints, `object(ID)`, `type(ID, bear)`.¹ The following two rules indicate that the type of any object is unique and (by assumption about the clip-art images) that an image includes at most on bear. Logical variables are indicated by capital initial letters, thus distinguished from constants, predicates, etc.

```
type(ID, T1) \ type(ID, T2) <=> T1=T2.
type(ID1, bear) \ type(ID2, bear) <=> ID1=ID2.
```

Each rule applies as soon as constraints meeting the left side appear in the store; constraints following the backslash are removed, and those on the right side are added (in these examples, unifications are executed, perhaps leading to failure if a proposed interpretation is judged impossible). CHR has a variety of different rules and facilities, but the understanding of the details are not important for the present paper.

We can put these relationships into a logical formula as follows, considering a specific image.

$$Grammar \wedge WorldKnowledge \wedge CSKB \models sentence(S)$$

For the present applications, *Grammar* and *WorldKnowledge* are fixed. Enhancing a given knowledge base by knowledge embedded in sentences means that *CSKB* is partially known, which we may write it as $CSKB_0 \wedge ?Extension$, where the last component is unknown and filled in by an abductive interpretation of given sentences *S*.

Sentences may be approved or generated by a deductive analysis, i.e., *CSKB* is now the enhanced knowledge base effectively locked (close world) by converting it into Prolog facts. Here *S* is either a given sentence (for approval) or a logical variable that will be instantiated to alternative sentences by the execution.

Additionally, we may use the model to build part of the *WorldKnowledge* by analyzing a large collection of sentences for different images, for example to identify roles for verbs, e.g., which (types of) objects can eat and which are edible. However, we did not apply this for the present experiments.

3.2 Crafting a CS grammar for the clip-art image sentences

About 7000 images are given by partial descriptions, in the form of a *CSKB* for each image defining most of their objects and some – but not all – of their interrelationships. Each image is accompanied

¹These identifiers may, as shown here, be new, unused variables, but when stored in a file to be loaded later, it is practical to replace such variables consistently by unique constant symbols.

by typically three sentences created by natural language users. We have – to some extent - manually corrected the corpus for spelling and grammatical errors and removed some sentences whose contents obviously went far beyond what is seen in the images.

As mentioned, our goal is, for each image, to extend the given CSKB with the knowledge embedded in the sentences about the image, such that we can generate additional sentences consistent with the image as well as checking whether sentences from other sources have this property.

A suitable grammar is developed in an iterative process combining general knowledge about the English language and the constructions and vocabulary used in the corpus. At each iteration the coverage (i.e., the percentage of all sentences that can be parsed) is checked and samples of the extended CSKBs are checked manually.

We can indicate the flavour of our Contextual Semantic Grammars, showing excerpts of our current version, involved in processing the sentence “*A red and yellow hot air balloon is floating over the park*”. Any constant symbols used internally to represent contextual semantics starts by the characters “c_”; for simplicity of writing the rules, we use generic predicates `rel` with one, two or three arguments for various relationships, say `rel(c_rain)` (“*It is raining*”), `rel(c_sleep, X)`, `rel(c_eat, X, Y)`.

First the grammar rules:

```
sentence --> subject(X,Number), vp(X,Number).
subject(X,Number) --> np(X,Number).
np(X,Number) --> det(Number,AnA), adjp(A,AnA), noun(X,Number,_), {rel(A,X)}.
det(singular,a) --> [a].
det(singular,an) --> [an].
det(singular,_) --> [the].
noun(X,singular,a) --> [hot,air,balloon], {object(X), type(c_hot_air_balloon,X)}.
adjp(A1+A2,AnA) --> simpleAdjp(A1,AnA), [and], simpleAdjp(A2,_).
simpleAdjp(Ad, AnA) --> adj(Ad,AnA).
adj(c_red,a) --> [red].
adj(c_yellow,a) --> [yellow].
adj(c_orange,an) --> [orange].
vp(X,Number) --> verb(V,intrans, Number), {rel(V,X)}, pp(X).
verb(V,Val,singular) --> [is], verb_ing_form(V,Val).
verb_ing_form(c_float, intrans) --> [floating].
pp(X) --> prep(P), np(Y,_), {rel(P,X,Y)}.
prep(c_over) --> [over].
noun(X,singular,a) --> [park], {object(X),type(c_park,X)}.
```

When the subject “*a ... hot air ballon*” has been recognized, the variable `X` in the first rule is instantiated to an identifier, which may be a new variable. This `X` is sent to the predicate and the `pp` subphrases, as they are expected to express further properties that are naturally associated with `X`.

The detailed analysis of the subject refers to the CS constraints `object(X)`, `type(c_hot_air_balloon,X)`. Operationally speaking, “refers to” here means that the constraints are created when in abductive mode, and checked when in deductive mode.

Notice that we allow only one or two adjectives in a row in a `adjp`, which fits with the given corpus and, when generating sentences, suppresses the creation of infinitely long sentences. The analysis of the `adjp` “*red and yellow*” additionally introduces, first `rel(c_yellow+c_red,X)` which in turn is reduced by a CHR rule shown below to `rel(c_yellow,X)`, `rel(c_red,X)`.

The correct use of “*a*” and “*an*” is controlled by the arguments named `AnA` in the rules for `np` and `adjp`, see, e.g., the difference in the rules for adjectives `red` and `orange`. The rule for `adjp` indicates the principle that the choice of *a/an* follows immediately following word (adjective or noun).

The predicate gives rise to the CS constraints `rel(c_float,X)`² and the `pp` yields `object(Y)`, `type(c_park,Y)`, `rel(c_over,X,Y)`. A CHR rule introduces, as a consequences of the last one, also `rel(c_over,Y,X)`, which allows, in a next step, the generation of, say, “*The park is under the hot air ballon*”.

The Contextual Semantic Grammar includes also a collection of CHR rules, some that take care of operational needs such as avoiding loops and suppressing creation of duplicate constraints, and others

²It may be seen as a rather coarse simplification that we always attach proposition to the subject rather than the verb, but when using this grammar for analysis and generation we obtain results that look reasonable in most cases.

that express interesting knowledge. The processing of our chosen sample sentence involves activating the following rules.

```
rel(R,X) \ rel(R,X) <=> true.  
rel(Rel1+Rel2,A) <=> rel(Rel1,A), rel(Rel2,A).  
rel(c_over,X,Y) ==> rel(c_under,Y,X).
```

The first rule removes a duplicate constraint before any other rule is tried, which means that these rules also works together with the additional one `rel(c_under,X,Y) ==> rel(c_over,Y,X)` without looping.

4 Conclusions

In this paper, we have proposed to use a grounded language learning system –defined in Becerra-Bonache et al. (2015; 2016a; 2016b) for a different purpose– to study linguistic complexity from a developmental point of view. We have also introduced a tool for generating a gold standard in order to calculate the complexity of a language through the evaluation of the performance of the learning system.

Regarding the grounded language learning system, we may conclude that it presents several advantages to measuring linguistic complexity: it does not require any prior language-specific knowledge; it uses realistic data and psychologically plausible algorithms that include features like gradual learning, robustness to noise in the data, and learning incrementally.

In what refers to the tool for generating the gold standard, some final considerations are in order. As it appears, our grammar rules include several simplifications, but as is well-known, Definite Clause Grammars are quite flexible and there is a comprehensive literature since the 1970s on how to model various grammatical refinement. The additional use of CHR for abductive reasoning facilitates the use of a flat representation for the CS knowledge representation which avoids the difficulties of using a traditional compositional approach, involving that each sentence needs a meaning representation which is one huge structure covering the entire sentence; furthermore the entire contextual semantic knowledge base needs to be passed explicitly trough all phrases and subphrases.

In comparison to other approaches to abductive reasoning in logic programming, the present approach is note for its direct and efficient use of existing technology without any interpretational overhead. For a recent overview of Prolog based grammars, including the present ones, including lots of background references, see Christiansen and Dahl (2018).

The Contextual Semantic Grammars used in the present paper includes semantic information in a much simpler way, and the symmetry between abductive and deductive reasoning supports an intuition that every sentence reflects some underlying reality – as indicates by a particular clip-art image – independently of whether or nor this reality is known in all details to the language processor (whether human or machine).

As shown elsewhere (Christiansen et al., 2007a; Christiansen et al., 2007b), it is possible to integrate pronoun resolution in these sort of grammars, but in the present simplistic setting, there are very few pronouns that in most cases are resolved deterministically. For example, the only possible people are Mike and Jenny, so there is very little doubt to whom “*she*” refers.

In this paper, we claim that learning models can be seen as an alternative to the methods that have been used so far in the area of linguistic complexity. They are models that focused on the learning process and therefore open the door to consider children first language acquisition as the language use-type to calculate linguistic complexity. In general, recent work on language complexity takes an *absolute* perspective of the concept while the *relative* complexity approach –even though considered as conceptually coherent– has hardly begun to be developed. Computational models of language acquisition may be a way to revert this situation.

The proposed model may provide quantifiable experimental results and permits to perform crosslinguistic analysis. In order to determine the degree of complexity, we are working on experiments with a set of languages and we will be able to quantify the complexity of each language. Since our computational simulation allows us to reproduce exactly the same state/environment/requirements for the acquisition of any language we will assure crosslinguistic analysis regarding complexity.

Acknowledgements

This research has been supported by the Ministerio de Economía y Competitividad and the Fondo Europeo de Desarrollo Regional under the project number FFI2015-69978-P (MINECO/FEDER, UE) of the Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento.

The work of Leonor Becerra-Bonache has been performed during her teaching leave granted by the CNRS (French National Center for Scientific Research) in the Computer Science Department of Aix-Marseille University.

References

- D. Angluin and L. Becerra-Bonache. 2010. A model of semantics and corrections in language learning. Technical report, Yale University.
- D. Angluin and L. Becerra-Bonache. 2011. Effects of meaning-preserving corrections on language learning. In *Proceedings of the 15th International Conference on Computational Natural Language Learning, CoNLL 2011*, pages 97–105. Portland.
- D. Angluin and L. Becerra-Bonache. 2016. A model of language learning with semantics and meaning preserving corrections. *Artificial Intelligence*, 242:23–51.
- L. Becerra-Bonache, H. Blockeel, M. Galván, and F. Jacquenet. 2015. A first-order-logic based model for grounded language learning. In *Advances in Intelligent Data Analysis XIV - 14th International Symposium, IDA 2015*, pages 49–60.
- L. Becerra-Bonache, H. Blockeel, M. Galván, and F. Jacquenet. 2016a. Learning language models from images with regll. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016*, pages 55–58.
- L. Becerra-Bonache, H. Blockeel, M. Galván, and F. Jacquenet. 2016b. Relational grounded language learning. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, pages 1764–1765.
- H. Christiansen and V. Dahl. 2005a. HYPROLOG: A new logic programming language with assumptions and abduction. In Maurizio Gabbrielli and Gopal Gupta, editors, *Logic Programming, 21st International Conference, ICLP 2005, Sitges, Spain, October 2-5, 2005, Proceedings*, volume 3668 of *Lecture Notes in Computer Science*, pages 159–173. Springer.
- H. Christiansen and V. Dahl. 2005b. Meaning in Context. In Anind Dey, Boicho Kokinov, David Leake, and Roy Turner, editors, *Proceedings of Fifth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-05)*, volume 3554 of *Lecture Notes in Artificial Intelligence*, pages 97–111.
- H. Christiansen and V. Dahl. 2018. Natural language processing with (tabled and constraint) logic programming. In Michael Kifer and Annie Liu, editors, *Festschrift for David S. Warren*. To appear.
- H. Christiansen, Ch. Theil Have, and K. Tveitane. 2007a. From use cases to UML class diagrams using logic grammars and constraints. In *Recent Advances in Natural Language Processing (RANLP-2007)*, pages 128–132. Shoumen, Bulgaria: INCOMA Ltd.
- H. Christiansen, Ch. Theil Have, and K. Tveitane. 2007b. Reasoning about use cases using logic grammars and constraints. In *Proceedings of the 4th International Workshop on Constraints and Language processing (CSLP 2007)*, number 113 in *Computer Science Research Reports*, pages 40–52. Roskilde University.
- H. Christiansen. 2014. Constraint programming for context comprehension. In Patrick Brézillon and Avelino J. Gonzalez, editors, *Context in Computing – A Cross-Disciplinary Approach for Modeling the Real World*, pages 401–418. Springer.
- O. Dahl. 2004. *The Growth and Maintenance of Linguistic Complexity*. John Benjamins, Amsterdam.
- A. D’Ulizia, F. Ferri, and P. Grifoni. 2011. A survey of grammatical inference methods for natural language learning. *Artificial Intelligence Review*, 36(1):1–27.
- A. Fazly, A. Alishahi, and S. Stevenson. 2010. A probabilistic computational model of cross-situational word learning. *Cognitive Science*, 34(6):1017–1064.

- J.R. Hobbs. 1985. Ontological promiscuity. In William C. Mann, editor, *23rd Annual Meeting of the Association for Computational Linguistics, 8-12 July 1985, University of Chicago, Chicago, Illinois, USA, Proceedings.*, pages 61–69. ACL.
- M.D. Jiménez-López and L. Becerra-Bonache. 2016. Could machine learning shed light on natural language complexity? In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity*, pages 1–11.
- J.E. Joseph and F.J. Newmeyer. 2012. All languages are equally complex: The rise and fall of a consensus. *Historiographia Linguistica*, 39:2/3:341–368.
- W. Kusters. 2003. *Linguistic Complexity: The Influence of Social Change on Verbal Inflection*. LOT, Utrecht.
- J. McWhorter. 2001. The world's simplest grammars are creole grammars. *Linguistic Typology*, 6:125–166.
- M. Miestamo, K. Sinnemäki, and F. Karlsson. 2008. *Language Complexity: Typology, Contact, Change*. John Benjamins, Amsterdam.
- F.J. Newmeyer and L.B. Preston. 2014. *Measuring Grammatical Complexity*. Oxford University Press, Oxford.
- G. Pallotti. 2015. A simple view of linguistic complexity. *Second Language Research*, 31:117–134.
- W. V. O. Quine. 1960. *Word and object*. Cambridge, MA: MIT Press.
- G. Sampson, D. Gil, and P. Trudgill. 2009. *Language Complexity as an Evolving Variable*. Oxford University Press, Oxford.
- C.L. Zitnick and D. Parikh. 2013. Bringing semantics into focus using visual abstraction. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 3009–3016. Portland.