
A Constructionist Approach to Grammar Inference

Oskar Kohonen, Sami Virpioja, Krista Lagus
Department of Information and Computer Science
Helsinki University of Technology
FIN-02015 Espoo, Finland
firstname.lastname@tkk.fi

Abstract

Models based on Chomskyan grammar are not as pervasive in certain NLP applications as might be expected from their status in linguistics, comparing to, e.g., statistical n-gram models or vector space models. In contrast, here grammar inference is approached from the viewpoint of constructionist theories of language, which avoid the separation of morphology, syntax, semantics and pragmatics. We consider the utility of constructionist theories in NLP applications, present a computational framework for learning constructions, and discuss related methodological work.

1 Introduction

The ability to recognise and produce new, meaningful sentences can currently be considered a largely unsolved problem both in natural language processing (NLP) and in the study of human cognition. Various aspects of grammar, sentence processing and lexical representations are under current research in many disciplines, including linguistics, psycholinguistics, cognitive science and neuropsychology. The objective in language engineering is to describe grammatical knowledge in a manner that practical NLP applications can utilize. A grammar can then be viewed as a model of sentence-level phenomena in language.

In linguistics and language engineering, the dominant tradition of grammar is the Chomskyan generative theory of grammar. Grammar inference most commonly refers to the problem of learning a grammar from a certain class in the Chomsky hierarchy. The classes are typically Context-Free Grammars [1, 2, 3, 4] or Regular Grammars [5]. Grammar inference may also refer to learning other kinds of representations, such as Dependency Grammar [3, 4, 6]. In NLP applications, models based on generative grammar are not as pervasive as might be expected from their status in linguistics. In many applications, such as automatic speech recognition (ASR), statistical machine translation (SMT) and information retrieval (IR), much simpler models are commonly used. For example, ASR and SMT systems apply n-gram models that just predict word sequences, and IR systems may discard everything related to grammar and apply the bag-of-words model for encoding the meaning of words and documents.

Currently, most grammar inference approaches have two central problems from the perspective of applications, namely problems with learnability of the models, and the underlying assumption of the separation of grammar and meaning. Some recent approaches seem promising regarding solving the problems of learnability [7]. A basis for solving the latter problem can be found in certain recent linguistic theories, namely the constructionist theories of language [8]. While these theories are supported by recent evidence from psycholinguistic and brain imaging studies, there seems to be a lack of a clear computational formulation of the grammar inference problem from a constructionist standpoint.

In this article, we attempt to provide such a problem formulation, and discuss some recent work by us and others regarding how to solve this problem. The article is structured as follows: Section 2

introduces the constructionist approaches to language and motivates why they should be useful in NLP. Section 3 presents a computational framework for learning constructions. Section 4 describes some methods that can be viewed as instances of applying the framework. Section 5 discusses some implications of the central ideas and Section 6 concludes the article.

2 Constructionist theories of grammar

Constructionist theories of grammar share with Chomskyan generative grammar the objectives to describe how to combine known grammatical structures to produce novel utterances. However, the theories disagree strongly on many key points. Here we give a short treatment of constructionist theories of grammar, covering the parts relevant to the arguments in this article. For a thorough overview, see [8]. Constructionist theories mostly share the following points:

1. There exist no different levels of language, such as morphology, syntax, semantics or pragmatics. Rather, all phenomena are described using form-meaning pairs, and these are called *constructions*. The form component can be, for example, a morpheme (*anti-*, *-ing*), a word, an idiom (“*kick the bucket*”) an idiom with an abstract category “*pull X’s leg*”, or an abstract sentence construction (*SUBJ V OBJ*). The meaning component includes both semantic and discourse function. The theories state that every regularity in language is expressed using a single framework, namely constructions. Consequently, there is no special emphasis on syntax. Moreover, because syntax is described using form-meaning pairs, syntax is not represented independently of semantics, in contrast to what is postulated in the Chomskyan generative grammar.
2. No special cognitive modules for grammar are posited, but constructions are considered to be learned using general cognitive mechanisms.
3. All knowledge a speaker possesses of a language is encoded in her *construction lexicon*. The construction lexicon is a network of constructions, describing both the form and meaning poles of each construction, and the relationships between the constructions.

What then is a construction, and what is not? Goldberg writes in [8]:

Any linguistic pattern is recognized as a construction as long as some aspect of its form or function is not strictly predictable from its component parts or from other constructions recognized to exist.

This definition defines a minimal set of constructions a speaker must know in order to be able to understand language. In this article, we refer to this set as *minimal constructions*. Goldberg continues on predictability and frequency:

In addition, many constructionist approaches argue that patterns are stored even if they are fully predictable as long as they occur with sufficient frequency.

It may seem peculiar to include constructions that are not strictly speaking needed. However, there is evidence, that humans do that in some cases. In psycholinguistics, an active topic of research concerns the question of what is stored in the “mental lexicon”. A particular form of this question is, whether frequent, inflected word forms are stored as complete forms or in terms of their constituent segments. According to [9] the answer seems to be, both.

2.1 Motivation for a constructionist approach to NLP

Due to associating directly to meaning, constructions can be considered from a cognitive linguistic point of view as basic elements of cognitive representation. Using these elements should directly benefit the subsequent processing of the linguistic objects in an NLP application whose goal is to process meanings. For example, these could be utilised as features in vector-space modeling of meaning based on word contexts [10, 11, 12, 13].

3 Computational framework for learning constructions

We attempt here to provide an operationalised account of constructions and of learning them from corpora. First, we define some central terms that are needed in order to describe the construction learning problem in a precise way. These definitions are inspired by linguistic theories of constructions.

- *Construction lexicon* is the set of all constructions stored by a human or a computational model.
- *Construction* is a pattern of language that consist of form and meaning, and that is stored in the construction lexicon.
 - The *form* of a construction describes how the construction appears in language data, that is, in speech or text.
 - The *meaning* of a construction describes the semantic and discourse functions of the construction.
- *Compound construction* is a construction whose *form* can be expressed using the forms of two or more constructions. These constructions are the *component parts* of the compound construction.

Examples: The word “cars” is a compound construction of two morphemes, “car” and “s”. The term “white house” is a compound construction of two words, “white” and “house”, regardless of whether its meaning is “The White House” or “a white house”.

- *Minimal construction* is a construction that is either a construction whose form cannot be expressed using other constructions (not a compound construction), or a compound construction whose meaning cannot be predicted from its component parts.

Examples: All morphemes of a language (“walk”, “car”, plural ending “s”, prefix “un”) are minimal constructions that are not compound constructions. The collocation “The White House” and the idiom “kick the bucket” are minimal constructions that are also compound constructions.

- *Redundant construction* is a compound construction that is not a minimal construction.

Examples: Words “cars” and “walked”, as well as phrases “my house” and “red and blue” are redundant constructions (if they exist in the construction lexicon).

3.1 Problem definition

Next, we consider learning of a construction lexicon given language data as input. We define two types of construction lexicons for a data set:

- *Minimal construction lexicon* is the set of minimal constructions that can produce the data set.
- *Sufficient construction lexicon* contains the minimal construction lexicon but can also include redundant constructions.

The learning problem that we are concentrating on is to *find a sufficient construction lexicon* for the given data set. There are three reasons for selecting this task instead of finding of the minimal lexicon: First, determining whether a construction is redundant or minimal is a hard problem and requires information on the meanings of the constructions. Second, redundant constructions may be useful for computationally efficient modeling of language data, even though they are not required for representing the meanings in the language data. Third, there is also evidence from psycholinguistic that redundant constructions, such as inflected word forms, exist in the mental lexicon [9].

Next, we consider two different types of the construction learning problem: The *full learning problem* is the search for both the forms and the meanings of constructions, whereas the *form learning problem* is the identification of only the forms of the constructions.

It is generally thought that humans learn constructions using a combination of contextual information such as embodied meanings active during speech and statistical regularities within the speech

signal. There is some work on learning constructions where both form and meaning are encoded as markup in the input data and both are modeled together [14]. Learning from untagged text, that is, without any coding of meaning present in the data can be considered as solving only the form learning problem.

3.2 Evaluation approaches

There are at least three possible ways of evaluating the learned construction lexicons: (1) Comparison with the studies of meaning representations in humans. For example, one could evaluate whether the *forms* of the constructions are the same that humans store. (2) Direct comparison to human-tagged meanings in corpora. That is, one may assess whether the learned constructions correspond to our intuitions regarding those meanings. (3) Indirect evaluation in NLP tasks, such as speech recognition, machine translation and information retrieval.

3.3 Methods for learning the forms of constructions

3.3.1 Controlling the complexity of the lexicon

If the purpose is to represent the construction lexicon that models the whole data set, and not just to identify some most interesting and surprising patterns in it, there are two aspects to be taken into account. The first aspect is the *representation ability*, the model should be able to represent all of the data. The second is *compactness*.

A sufficient construction lexicon, which was identified as the goal, can represent all the data. In practice, when one applies, e.g., a probabilistic model, the representation ability can be measured by the likelihood of the data.

Regarding compactness one might say that given two sufficient construction lexicons the more compact one is the better. The complexity of the lexicon can be selected beforehand or optimised during the learning process. For controlling the compactness of the model there are several options. Heuristic ones include fixed model sizes and stopping criteria. Explicit ones include, e.g., different models selection methods and non-parametric Bayesian methods. We discuss one well-motivated model selection method in more detail: The Minimum Description Length (MDL) principle, that comes from the field of information theory [15]. The basic idea in MDL resembles that of Occam's razor, which states that when you have two equally accurate theories (models), you should select the theory (model) that is less complex. When modeling data, controlling the model complexity is essential in order to avoid overlearning, i.e., a situation where the properties of the input data are learned so precisely that the model does not generalise well to new data.

There are different flavors of MDL. The earliest is called the two-part coding scheme. The more recent versions of the MDL principle, such as Refined MDL [16], address the theoretical weaknesses of the two-part coding (see, e.g., [17]). However, their application to practical tasks such as language learning is not as straightforward. The intuitive idea behind the two-part MDL is as follows: Modeling can be viewed as a problem of how to encode a data set efficiently in order to transmit it to a listener with a minimal number of bits. In order to transmit a data set, one first transmits the model, then the data set by referring to the model. Respectively, the cost function to minimise consists of (1) the cost of encoding the model, and (2) the cost of representing the observed data in terms of the model. The first part penalises models that are overly complex, whereas the second part penalises models that are not accurate descriptions of the data. Thus, the two-part code expresses an optimal balance between the specificity and the generalization ability of the model.

Note that if the costs of the representations are described as probabilities, the two-part coding is equivalent to using a certain prior in a Bayesian *Maximum a Posteriori* estimation: The cost of the model comes from the negative logarithm of the prior probability $P(\mathcal{M})$, and the cost of the data comes from the negative logarithm of the likelihood $P(\text{corpus}|\mathcal{M})$.

3.3.2 Identifying the forms of constructions

Irrespective of the particular method one applies to modeling the construction lexicon and the data, there are two opposite situations for the learning process: On one hand, the constructions can be too specific, such as all the sentences in the data set. On the other hand, they can be too general, such

as all the letters in the data. In the former case, the forms of the minimal constructions are smaller patterns of the data, and in the latter case, they are larger patterns. Both of the extremes might be useful initializations for the learning task—e.g., when using a *top-down* or a *bottom-up* algorithm—but it is necessary to be able to recognise too specific or too general forms of constructions.

First, we consider the problem of identifying the forms of minimal, non-compound constructions. In principle, this can be very easy: If we can say that an item has a meaning of its own, and that it cannot be a combination of some other constructions, then it has to be a construction. For example, if something occurs as a word, it is very likely to have a meaning. And if there is no way to produce it as a combination of known morphemes it is probably a morpheme itself and thus a minimal construction.

It would be very hard to recognise any constructions larger than morphemes, if the only test would be whether a form candidate is a combination of already found constructions or not. Even for morphemes, this would lead to many false negatives: e.g. “fiber” could be analysed as a concatenation of morphemes “fib” and “er”. Moreover, we would not be able to get any idioms, if their words were already in the lexicon.

As we assume that the meanings contribute to the statistics of a text, it is a reasonable hypothesis that constructions are more frequent than other observed combinations of forms. That is, if a form is frequent compared to the frequency of its component parts, this indicates a construction. (However, it is not necessarily a minimal construction.) Naturally, this simple observation has been applied for many related tasks of NLP, such as finding collocations. Methods that can be used directly with the frequencies are, e.g., hypothesis testing and measuring mutual information [18]. Due to the Zipfian distributions in language data, the number of high-frequency forms is always restricted, and the frequency information at least provides an easy way to exclude most of the theoretical form candidates.

However, frequency information can be misleading as well. Especially bottom-up learning may produce a situation where a high frequency of a compound item is due to some construction of a higher abstraction level that produces the observed form. For example, if the frequency of the form “work he” in an English corpus were larger than what could be predicted from the frequencies of its parts, the phenomenon might have been produced from a certain noun phrase construction “NP (that) VP” (e.g., “the work he did”).

4 Related methodological work

Since there is very little work on discovering constructions by that name, we will here take a wider outlook. Grammar inference of PCFGs or dependency grammars can be viewed as learning certain kinds constructions. but is covered well by others in the literature. Thus we focus here on other cases of construction learning. Finding collocations [18, Ch. 5] may be considered as a very simple form of construction discovery where the whole data is not modeled—instead, only some interesting patterns are identified. In contrast, a task where a text corpus is modeled as a whole, is unsupervised word segmentation from text [19]. Word-internal constructions and sometimes also their connections are discovered in unsupervised learning of morphology, which has recently been a target of a lot of active research [20, 21, 22]. Algorithms for inferring morphology have very practical uses in the preprocessing of morphologically rich languages for certain NLP applications [23, 24]. Also the modeling of sentence-level patterns is a relevant problem for many NLP applications: For example, the current statistical machine translation systems are often based on translation of phrase pairs extracted from a sentence-aligned corpus [25]. The obtained phrase table can be viewed as a construction lexicon that includes the forms in two languages.

Next we will describe in some detail the Morfessor algorithms [21] for morphology learning. In addition, we briefly outline two algorithms for the discovery of complex sentence-level patterns [26, 27].

4.1 Discovery of morphemes

Finding morphemes of a language can be regarded as a subproblem of learning the construction lexicon, especially in inflective languages. As morphemes are the minimal constructions that are

needed to build up any lexical constructions (such as words and anything that consists of words), it is an important starting point.

Unsupervised learning of morphology has been an active research topic at least for a decade, and the results are encouraging for several languages. Morphology induction algorithms have been evaluated, e.g., in the Morpho Challenge competitions that have been organised from 2005 to 2009 [24]. A successful algorithm is Morfessor [21], which applies the MDL principle for finding an optimal morpheme lexicon. The first Morfessor algorithm, Morfessor Baseline, is conceptually very simple. In Morfessor Baseline, the model is a lexicon of word segments called morphs, and the data is encoded as a sequence of pointers to the morph lexicon. The lexicon is encoded as a set of strings which form the morphs. The two-part coding scheme gives the cost function that is to be optimised by the method. A simple greedy algorithm is applied: Initially, the lexicon contains all the words in the corpus. Words are picked in a random order, and for each word, a segmentation to two parts is attempted. Furthermore, this step includes a recursive re-splitting attempt for each of the parts. As a result, each word can consist of segments which in turn may consist of other segments.

Surprisingly, the output of the Morfessor Baseline algorithm has been found to perform as well as or better than linguistic morphemes or words as tokens for language models utilised in speech recognition [23]. In contrast to linguistic analyses, the algorithm tends to oversegment words that were seen very rarely, whereas it undersegments very common words. While this is an undesired phenomenon with respect to a purely linguistic morphology analysis, it seems to be in agreement with the “construction forming view” of language, that is, that frequent linguistic phenomena become gradually lexicalised (i.e., becoming codes of their own), although initially they may have been productive (i.e., they have been merely assembled from their parts). Later Morfessor versions, such as Morfessor Categories-MAP [21] produces analyses that are more similar to a gold standard segmentation. Despite this, they do not outperform Morfessor Baseline in practical applications, such as speech recognition, information retrieval and machine translation [24]. This underlines that discovering only the minimal constructions, such as morphemes, may not be so important for the applications.

An interesting feature in Morfessor Categories-MAP, as well as in de Marcken’s model for word segmentation [19], is that they apply hierarchical lexicons, where the component parts of a form are used to store the form itself. This is in contrast to Morfessor Baseline, where the decision to join two morphs means that the original morphs are removed and no longer used to encode the data, and agrees well with the idea of learning a construction lexicon.

4.2 Discovery of sentence patterns

Recently, we applied an MDL-based approach, similar to Morfessor Baseline, to the problem of finding a compact description of sentence-level utterances [26]. The basic form of the task is very similar: instead of a lexicon of morphs that is used to encode words, we have a lexicon of constructions that is used to encode utterances. The applied model structure allows only a very restricted set of possible constructions, namely exact phrases (*white house*) and partially filled constructions that have exactly one abstract category that can be filled by one word (example $[X] \text{ house}$, $X \Rightarrow \{\text{white, red, green, brown, black, blue}\}$). The results demonstrate that the same principles that work on the morphological level can also work to some extent on the sentence-level. However, currently the model has only been evaluated using manual inspection after training it on a small corpus of children’s stories. The method suffers from issues with computational complexity since, as with learning CFGs the space of possible models is very large.

A more mature algorithm that can be viewed as searching for sentence-level constructions is ADIOS [27]. It does not learn explicit grammar rules, but rather generalizations in a specific context using mutual information based measures. The model is flexible enough to represent complex and realistic patterns. ADIOS outperforms standard n-gram models in a language prediction task for small spontaneous speech corpora (ATIS, CHILDES), but has problems with grammatically complex texts [28].

5 Discussion

From psycholinguistic perspective, MDL and related “compression approaches” to modeling language are sometimes criticised using the argument that this is not what humans do: humans have

language representations that are not compact but rather show redundancy and multiplicity. This criticism is in fact in accordance with our suggestion that the goal is not to find a minimal construction lexicon, but a sufficient one.

When would it be a reasonable goal for a communicative agent to represent just the minimal construction lexicon? The answer is straightforward: only when the data is stationary, i.e. the language properties do not change over time. However, it is well-known that languages in themselves are not at all stationary: new words and terms appear all the time, the probabilities of the existing ones change, and their contextual properties and complex constructions change even more.

Moreover, even if the language as a whole were in fact stationary, an individual agent is never in a stationary situation with respect to the language data. Any change in the social group, a change regarding the domains where the agent interacts, and so on, will lead to changes in the data encountered by the agent, and therefore a different solution to the question of the minimal lexicon. In other words, being in a constant 'learning state' regarding one's language representations can be considered useful for a language agent throughout its existence.

In a learning state in the distributional learning paradigm the agent must accrue evidence about possible new constructions. A good hypothesis for a construction is provided by frequency: if an item is frequent, it may, over time, accrue its own meaning, *even if it currently can be deduced from the meanings of its parts*. This explains both why it is useful to represent redundant constructions, and why frequent items are generally good candidates for storing new constructions in the lexicon.

A major open problem in construction learning is related to computational complexity. If the models used are flexible enough to express realistic constructions, then the space of possible models is correspondingly very large. Thus, model-based approaches become infeasible, and one has to resort to heuristic methods.

6 Conclusions

In this article we have presented a starting-point for grammar inference that stems from constructionist theories of language, and that provides an alternative to the currently pervasive Chomskyan tradition. Moreover, we have argued how such a constructionist approach for inferring grammatical knowledge might be both applicable to various NLP problems and justified from a psycholinguistic modeling point of view. Moreover, we have discussed some work in both morphology discovery and the discovery of a construction inventory that can be considered as examples of this approach. While the outlined problem itself appears to be fruitful, the development of efficient learning strategies and evaluation methods on the sentence level is at this point only in its infancy.

Acknowledgments

We thank Tiina Lindh-Knuutila for comments on the article. The financial support from Academy of Finland and Graduate School of Language Technology in Finland is gratefully acknowledged.

References

- [1] E. Mark Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.
- [2] Menno van Zaanen. Bootstrapping syntax and recursion using alignment-based learning. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1063–1070. Stanford University, Morgan Kaufmann Publishers, July 2000.
- [3] Dan Klein and Christopher D. Manning. Natural language grammar induction with a constituent-context model. *Pattern Recognition*, 38(9):1407–1419, 2005.
- [4] Dan Klein. *The Unsupervised Learning of Natural Language Structure*. PhD thesis, Stanford University, 2005.
- [5] Pieter W. Adriaans and Carl Jacobs. Using MDL for grammar induction. In *In Proceedings of the 8th International Colloquium on Grammatical Inference (ICGI). Lecture Notes in Artificial Intelligence vol. 4201*, 2006.
- [6] Noah A. Smith. *Novel Estimation Methods for Unsupervised Discovery of Latent Structure in Natural Language Text*. PhD thesis, Johns Hopkins University Department of Computer Science, Baltimore, MD, 2006.

- [7] Alexander Clark, Rémi Eyraud, and Amaury Habrard. A polynomial algorithm for the inference of context free languages. In *Proceedings of International Colloquium on Grammatical Inference*, pages 29–42. Springer, September 2008.
- [8] Adele E. Goldberg. Constructions: a new theoretical approach to language. *Trends in Cognitive Sciences*, 7(5):219–224, 2003.
- [9] Joseph P. Stemberger and Brian MacWhinney. Frequency and the lexical storage of regularly inflected forms. *Memory and Cognition*, 14:17–26, 1986.
- [10] Helge Ritter and Teuvo Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61:241–254, 1989.
- [11] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
- [12] Timo Honkela. *Self-Organizing Maps In Natural Language Processing*. PhD thesis, Helsinki University of Technology, Neural Networks Research Centre, 1997.
- [13] Magnus Sahlgren. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. PhD thesis, Department of Linguistics, Stockholm University, 2006.
- [14] Nancy Chang and Olya Gurevich. Context-driven construction learning. In *Proceedings of the 26th Annual Meeting of the Cognitive Science Society*, Chicago, IL., August 2004.
- [15] Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publ. Co., New Jersey, 1989.
- [16] Peter Grünwald. A tutorial introduction to the Minimum Description Length principle. In *Advances in Minimum Description Length: Theory and Applications*. MIT Press, 2005.
- [17] Pieter W. Adriaans and Paul M. B. Vitányi. The power and perils of MDL. In *2007 IEEE International Symposium on Information Theory (ISIT)*, pages 2216–2220, June 2007.
- [18] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Language Processing*. The MIT Press, 1999.
- [19] Carl. G. de Marcken. *Unsupervised Language Acquisition*. PhD thesis, MIT, 1996.
- [20] John Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–189, 2001.
- [21] Mathias Creutz and Krista Lagus. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1), 2007.
- [22] Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54, 2009.
- [23] Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pykkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Transactions on Speech and Language Processing*, 5(1), 2007.
- [24] Mikko Kurimo, Sami Virpioja, Ville T. Turunen, Graeme W. Blackwood, and William Byrne. Overview and results of Morpho Challenge 2009. In *Working Notes for the CLEF 2009 Workshop*, Corfu, Greece, September 2009.
- [25] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, 2003.
- [26] Krista Lagus, Oskar Kohonen, and Sami Virpioja. Towards unsupervised learning of constructions from text. In M. Sahlgren and O. Knutsson, editors, *Proceedings of the Workshop on Extracting and Using Constructions in NLP of 17th Nordic Conference on Computational Linguistics, NODALIDA*, May 2009. SICS Technical Report T2009:10.
- [27] Zach Solan, David Horn, Eytan Ruppín, and Shimon Edelman. Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences*, 102(33):11639–11634, 2005.
- [28] Jonathan Berant, Yaron Gross, Matan Mussel, Ben Sandbank, Eytan Ruppín, and Shimon Edelman. Boosting unsupervised grammar induction by splitting complex sentences on function words. In *Proceedings of the 31st annual Boston University Conference on Language Development*, 2007.