An Attempt Towards Learning Semantics: Distributional Learning of IO Context-Free Tree Grammars

Ryo Yoshinaka
Kyoto University, Japan
ry@i.kyoto-u.ac.jp

Abstract

Solid techniques based on distributional learning have been developed targeting rich subclasses of CFGs and their extensions including linear context-free tree grammars. Along this line we propose a learning algorithm for some subclasses of IO context-free tree grammars.

1 Introduction

Several efficient algorithms have been proposed to learn different subclasses of context-free grammars (CFGs) based on *distributional learning* (e.g. (Clark and Eyraud, 2007; Yoshinaka, 2011b)). Distributional learning models and exploits the distribution of strings in contexts. Those techniques have soon been generalized to *mildly context-sensitive formalisms*: multiple CFGs (Yoshinaka, 2011a) and simple (non-deleting linear) context-free tree grammars (CFTGs) (Kasprzik and Yoshinaka, 2011). Those formalisms can be naturally encoded by abstract categorial grammars (ACGs) (de Groote and Pogodalla, 2004), which are based on the simply typed linear lambda calculus. By the flexible nature of lambda terms, ACGs can generate various types of data like strings, trees, meaning representations and their combinations. The distributional learning of ACGs is discussed in (Yoshinaka and Kanazawa, 2011).

It is quite recently shown that interesting subclasses of *parallel* MCFGs are also learnable by a distributional learning technique (Clark and Yoshinaka, 2012). PMCFGs are a non-linear extension of MCFGs, whose production rules may copy arguments. They are not considered to be an MCS formalism, but can model some (controversial) non-semilinear syntactic phenomena reported in linguistics.

Non-linear operations are more commonly required in generating semantic representations than in syntax. Consequently it is a natural direction of research to enhance the distributional learning techniques for ACGs to non-linear extensions of ACGs so that we can learn pairs of words and their meanings as lambda terms in the style of Montague semantics. However, treating general lambda terms involves technical difficulties. Instead, this paper discusses an easier case – the distributional learning of IO-CFTGs (Rounds, 1970; Engelfriet and Schmidt, 1977), which are also a non-linear formalism. Although trees are not satisfactory enough compared with lambda-terms, trees can be used as primitive models of meaning expressions. An example of an IO-CFTG that generates meaning representations of English sentences is found in Figure 1. For the sake of simplicity, this paper does not target the learning of word-meaning pairs, but it is possible to apply the technique presented in this paper to a natural formalism that generates pairs of a string and a tree through the same context-free derivation tree. We will concentrate on the distributional learning of IO-CFTGs.

Every grammar formalism for which distributional learning techniques have been proposed so far generate their languages through context-free derivation trees, whose nodes are labeled by production rules. The formalism and grammar rules determine how a context-free derivation tree t is mapped to a derived object $\phi(t) = T$. A context-free derivation tree t can be decomposed into a subtree s and a tree-context c with $t = c[s]$. The

Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11), pages 90–98, Paris, September 2012.

		CFG rules ;	IO-CFTG rules	
π_1	$\langle$	$I \to NP\ VP$;	$I \to VP(NP)$	$\rangle$
π_2	$\langle$	$VP \to V\ NP$;	$VP(x_1) \to V(x_1, NP)$	$\rangle$
π_3	$\langle$	$VP \to V$ himself ;	$VP(x_1) \to V(x_1, x_1)$	$\rangle$
π_4	$\langle$	$V \to$ loves ;	$V(x_1, x_2) \to \mathbf{Love}(x_1, x_2)$	$\rangle$
π_5	$\langle$	$V \to$ hates ;	$V(x_1, x_2) \to \mathbf{Hate}(x_1, x_2)$	$\rangle$
π_6	$\langle$	$V \to$ kills ;	$V(x_1, x_2) \to \mathbf{Kill}(x_1, x_2)$	$\rangle$
π_7	$\langle$	$V \to V$ and V ;	$V(x_1, x_2) \to \wedge(V(x_1, x_2), V(x_1, x_2))$	$\rangle$
π_8	$\langle$	$NP \to$ Adam ;	$NP \to \mathbf{Adam}$	$\rangle$
π_9	$\langle$	$NP \to$ Eve ;	$NP \to \mathbf{Eve}$	$\rangle$
π_{10}	$\langle$	$NP \to$ Steve ;	$NP \to \mathbf{Steve}$	$\rangle$

Figure 1: An IO-CFTG generating meanings expressions together with a CFG for sentences, where I is the initial symbol.

subtree determines a constituent $S = \phi(s)$ and the tree-context determines a contextual structure $C = \phi(c)$ in which the constituent is plugged to form the derived object $T = C \odot S$, where we represent the plugging operation by $\odot$. For example, in the CFG case, C is a string pair $\langle l, r \rangle$ and S is a string u and $\langle l, r \rangle \odot u = lur$, which may correspond to a derivation $I \overset{*}{\Rightarrow} lAr \overset{*}{\Rightarrow} lur$ where I is the initial symbol and A is a nonterminal symbol. A learner does not know how a positive example T is derived by the target grammar. A learner based on distributional learning simply tries all the possible decompositions of a positive example into arbitral two parts C' and S' such that $T = C' \odot S'$ where some grammar may derive T thorough a derivation tree $t' = c'[s']$ with $\phi(c') = C'$ and $\phi(s') = S'$. Based on observation on the relation between potential constituents S' and contextual structures C' collected from given examples, she constructs her hypothesis grammar.

The literature has proposed several distributional properties that give learnable subclasses of a concerned formalism. Those properties can be classified into two: One, which we call *primal*, assumes that the language generated by each nonterminal is characterized by a finite number of constituents, whereas they are characterized by contextual structures in the other type, which we call *dual*. Those approaches show a tidy symmetry in the learning of CFGs (Yoshinaka, 2011b).

An important property of a formalism that makes distributional learning approach tractable is the *linearity*. A constituent S corresponding to a subtree s of a derivation tree will not be duplicated through the derivation process — S "occurs" in $C \odot S$ just once. This property makes the decomposition of a positive example tractable. The PMCFGs are not a linear formalism in this sense. Some components of a constituent S of a PMCFG may be duplicated during the derivation process and appear more than once in $C \odot S$. However, still the linearity property holds on the other side. That is, no components from the contextual structure C will be duplicated in $C \odot S$ through the interaction with S. Based on the linearity on the constituent side, Clark and Yoshinaka (2012) have shown that PMCFGs are learnable by a straightforward modification of an existing dual approach of distributional learning, while they discussed difficulties in learning PMCFGs by a primal approach due to the non-linearity of the contextual structure side.

The formalism this paper targets is IO-CFTGs, where copying operations are embedded into both constituent and contextual sides, which contrasts the case of PMCFGs. Therefore, the difficulty pointed out by Clark and Yoshinaka confronts both primal and dual approaches. This paper discusses how we can overcome the difficulty at the expense of restriction on our learning target. Clark and Yoshinaka's result on the learning of PMCFGs is a strict generalization of the learnability results on (M)CFGs, but our learner for IO-CFTGs does not learn some languages which are learnable by Kasprzik and Yoshinaka's algorithm for simple CFTGs. In fact, some finite languages are not learnable by our technique despite the strong learning scheme. Our result is presented as a first step towards learning word-meaning pairs by distributional learning techniques.

2 Preliminaries

We denote the set of nonnegative integers by $\mathbb{N}$. For a *ranked* alphabet Σ, we denote by Σ_m the set of letters of rank $m \in \mathbb{N}$. The set $\mathbb{T}_\Sigma$ of *trees* on Σ is the smallest set s.t. $f(t_1, \dots, t_m) \in \mathbb{T}_\Sigma$ whenever $f \in \Sigma_m$ and $t_1, \dots, t_m \in \mathbb{T}_\Sigma$ where $m \geq 0$. For $t \in \mathbb{T}_\Sigma$ and $\Delta \subseteq \Sigma$, we denote the number of occurrences of symbols from Δ in t by $|t|_\Delta$. We drop the subscript Δ if $\Sigma = \Delta$. For a finite set D of trees, we define $\|D\| = \sum_{t \in D} |t|$. Let X be a countably infinite set of rank 0 *variables* $x_1, x_2, \dots$. An m*-stub* s is a tree in $\mathbb{T}_{\Sigma \cup \{x_1, \dots, x_m\}}$ in which every variable x_i ($1 \leq i \leq m$) occurs at least once. Thus 0-stub is a variable-free tree. An m-stub is said to be p*-copying* if every variable occurs at most p times. A 1-copying stub is also called a *linear* stub. The set of p-copying m-stubs is denoted by $\mathbb{S}_\Sigma^{m,p}$. We will use $*$ to denote unlimitedness: e.g. $\mathbb{S}_\Sigma^{m,*} = \bigcup_{p \in \mathbb{N}} \mathbb{S}_\Sigma^{m,p}$. A *leaf substitution* σ is a mapping from $\{x_1, \dots, x_m\}$ to $\mathbb{T}_\Sigma$ for some m, whose domain is extended to stubs in the standard way: $s\sigma$ is the tree obtained from $s \in \mathbb{S}_\Sigma^{m,p}$ by substituting $\sigma(x)$ for every occurrence of $x \in X$ in t. Let Y be another ranked alphabet of variables, whose elements are denoted by y with or without subscripts: $y, y_1, y_2, \dots$ etc. This paper flexibly assumes their ranks depending on the context.

An *infix substitution* θ is a partial map from Y to $\mathbb{S}_\Sigma^{*,*}$ such that $y \in Y_m$ implies $y\theta \in \mathbb{S}_\Sigma^{m,*}$ (if defined), which is extended so that $f(t_1, \dots, t_m)\theta = f(t_1\theta, \dots, t_m\theta)$ if $f \notin \mathrm{dom}(\theta)$ and $f(t_1, \dots, t_m)\theta = \theta(f)\sigma$ where $\sigma(x_i) = t_i\theta$ for each i if $f \in \mathrm{dom}(\theta)$. By assuming that the order of the variables of the domain of θ is understood, a substitution $\{ y_i \mapsto s_i \mid 1 \leq i \leq n \}$ is often specified as $[s_1, \dots, s_n]$. Moreover when $s_i = B_i(x_1, \dots, x_{m_i})$ for some $B_i \in \Sigma_{m_i}$, we write it by $[B_1, \dots, B_n]$. An m*-environment* e is a tree in $\mathbb{T}_{\Sigma \cup \{y\}}$ in which all subtrees rooted by y of rank m are identical: i.e., $e = e'[x_1 \mapsto y(t_1, \dots, t_m)]$ for some trees $t_i \in \mathbb{T}_\Sigma$ and a stub $e' \in \mathbb{S}_\Sigma^{1,*}$, where the rank of y is m. If y occurs at most p times in e, we call it p-copying. The set of p-copying m-environments is denoted by $\mathbb{E}_\Sigma^{m,p}$. For an m-environment $e \in \mathbb{E}_\Sigma^{m,*}$ and an m-stub $s \in \mathbb{S}_\Sigma^{m,*}$, $e \odot s$ denotes the tree $e[s]$. The operation $\odot$ is naturally extended to sets so that $E \odot S = \{ e \odot s \mid e \in E \text{ and } s \in S \}$ for $E \subseteq \mathbb{E}_\Sigma^{m,*}$ and $S \subseteq \mathbb{S}_\Sigma^{m,*}$. When the alphabet Σ is understood, we write $\mathbb{S}^{m,p}$ for $\mathbb{S}_\Sigma^{m,p}$ and so on.

For a tree set $D \subseteq \mathbb{T}_\Sigma$, we define

$$\mathrm{Sub}^{m,p}(D) = \{ s \in \mathbb{S}_\Sigma^{m,p} \mid \exists e \in \mathbb{E}_\Sigma^{m,*}, e \odot s \in D \},$$
$$\mathrm{Env}^{m,p}(D) = \{ e \in \mathbb{E}_\Sigma^{m,p} \mid \exists s \in \mathbb{S}_\Sigma^{m,*}, e \odot s \in D \}.$$

We note that $\mathrm{Sub}^{0,*}(D)$ is the set of *subtrees* of elements of D in the usual sense.

Lemma 1.

$$\mathrm{Sub}^{m,p}(D) = \{ s \in \mathbb{S}_\Sigma^{m,p} \mid \exists e \in \mathbb{E}_\Sigma^{m,1}, e \odot s \in D \},$$
$$\mathrm{Env}^{m,p}(D) = \{ e \in \mathbb{E}_\Sigma^{m,p} \mid \exists s \in \mathbb{S}_\Sigma^{m,1}, e \odot s \in D \}.$$

Proof. We prove the first claim. The second one can be shown in a similar manner. Suppose that $e \odot s \in D$ with $s \in \mathbb{S}^{m,p}$ and $e \in \mathbb{E}^{m,n}$ for some $n > 1$, where y occurs just n times in e, i.e.,

$e = f[\overbrace{y(\vec{t}), \dots, y(\vec{t})}^{n\text{-times}}] \in \mathbb{E}^{m,n}$ for some $f \in \mathbb{S}^{1,n}$, where $\vec{t}$ denotes a sequence of trees of length m. For $e' = f[y(\vec{t}), s[\vec{t}], \dots, s[\vec{t}]] \in \mathbb{E}^{m,1}$, which is obtained from e by substituting s for all but one occurrence of y, we have $e' \odot s \in D$ and thus $s \in \mathrm{Sub}^{m,p}(D)$. $\square$

Lemma 2. *For fixed* m *and* p*, one can enumerate all elements of* $\mathrm{Sub}^{m,p}(D)$ *and* $\mathrm{Env}^{m,p}(D)$ *in polynomial time.*

Proof. We prove the first claim only. Suppose that $t = e \odot s \in D$ for some $e = e'[y(t_1, \dots, t_m)] \in \mathbb{E}_\Sigma^{m,1}$ and $s \in \mathbb{S}^{m,p}$. Let $n_i \leq p$ be the number of occurrences of x_i in s. Then s can be written as $s'[x_1^{n_1}, \dots, x_m^{n_m}]$ with a linear stub s' where $x_i^{n_i}$ denotes the sequence of x_i of length n_i. We have $t = e[s'[t_1^{n_1}, \dots, t_m^{n_m}]]$. Hence such a pair $\langle e, s \rangle$ can be uniquely specified by the positions where s' and t_i occur in t, which are at most $1 + n_1 + \cdots + n_m \leq 1 + mp$ positions in total. Therefore, there are at most $\|t\|^{1+mp}$ elements in $\mathrm{Sub}^{m,p}(\{t\})$ by Lemma 1 and one can enumerate all in polynomial time. $\square$

An IO-CFTG[1] is a tuple $G = \langle \Sigma, N, P, I \rangle$ where N and Σ are disjoint ranked alphabets of *nonterminals* and *terminals*, respectively, P is the set of rules and $I \subseteq N_0$ is the set of initial symbols. Each rule in P has the form $A(x_1, \dots, x_m) \to s$ with $A \in N_m$ and $s \in \mathbb{S}_{\Sigma \cup N}^{m,*}$ for some m, which will be abbreviated as $A \to s$. We stipulate that hereafter when we denote a rule

[1] We consider only non-deleting CFTGs.

as $A \to s'[A_1, \ldots, A_n]$ with $A_i \in N$ for each i, it means that $|s'[A_1, \ldots, A_n]|_N = n$. For example a rule $A \to B(B(c))$ may be denoted as $A \to s_1[B, B]$ with $s_1 = y_1(y_2(c))$ but is never denoted as $A \to s_2[B]$ with $s_2 = y_1(y_1(c))$ or $s_2 = B(y_1(c))$.

We define the derivation of an IO-CFTG in a non-standard way. Derivation trees of G are defined as follows.

- for every rule $\pi = (A \to s)$ with $s \in \mathbb{S}_\Sigma^{*,*}$, π is an A-derivation tree and its *yield* is $\phi(\pi) = s$;
- for a rule $\pi = (A \to s[B_1, \ldots, B_n])$ and B_i-derivation trees τ_i for $i = 1, \ldots, n$, the tree $\pi(\tau_1, \ldots, \tau_n)$ is an A-derivation tree and its yield is $\phi(\pi(\tau_1, \ldots, \tau_n)) = s[\phi(\tau_1), \ldots, \phi(\tau_n)]$;
- nothing else is an A-derivation tree.

An A-derivation tree is simply called a *derivation tree* if $A \in I$. The *language* $\mathcal{L}(G, A) \subseteq \mathbb{S}^{m,*}$ generated by $A \in N_m$ is defined to be

$$\mathcal{L}(G, A) = \{\, \phi(\tau) \mid \tau \text{ is an } A\text{-derivation tree} \,\}\,.$$

The *language* of G is defined to be $\bigcup_{A \in I} \mathcal{L}(G, A)$. A *derivation tree-context* is a tree χ with exactly one occurrence of a rank 0 variable z such that $\chi[z \mapsto \tau]$ is a derivation tree for some A-derivation tree τ for some $A \in N$. The mapping ϕ is naturally applied to a derivation tree-context by $\phi(z) = y$ so that $\phi(\chi) \odot \phi(\tau) = \phi(\chi[\tau])$, where $\phi(\chi) \in \mathbb{E}^{m,*}$ with m the rank of both A and y.

Example 3. Figure 1 includes a CFG on the left column and an IO-CFTG on the right with common rule labels. $\pi_1(\pi_8, \pi_2(\pi_4, \pi_9))$ is a derivation tree, which yields a string Adam loves Eve by the CFG and a tree $\mathbf{Love}(\mathbf{Adam}, \mathbf{Eve})$ by the IO-CFTG.

Another derivation tree $\pi_1(\pi_8, \pi_3(\pi_7(\pi_4, \pi_5)))$ yields Adam loves and hates himself and $\wedge(\mathbf{Love}(\mathbf{Adam}, \mathbf{Adam}), \mathbf{Hate}(\mathbf{Adam}, \mathbf{Adam}))$.

Corollary 4. *Let τ be a derivation tree that has an A-derivation tree τ' as a subtree. There is $e \in \mathbb{E}^{m,1}$ such that $\phi(\tau) = e[\phi(\tau')]$ where m is the rank of A.*

Proof. By Lemma 1. □

Lemma 5. *Suppose that G_* generates L_*, and $t \in L_*$ is derived using a rule $A_0 \to s[A_1, \ldots, A_n]$ with $A_0 \in N_m$. Then there are $s_i \in \mathbb{S}_\Sigma^{m_i,1}$ such that $s[s_1, \ldots, s_n] \in \mathrm{Sub}^{m,p}(t)$.*

Proof. The idea of the proof is common to the one for Lemma 1 except that copies of arguments by other arguments require a little care. We prove the lemma only for a special case for understandability, which will easily be generalized. Suppose that we have a rule π of the form

$$A_0 \to s[A_1, A_2]$$

where $s[y_1, y_2] = y_1(a(y_2(b, x_1)), c)$. Let τ_1 and τ_2 be A_1- and A_2-derivation trees such that $\phi(\tau_1) = u_1 = u_1'[x_1, x_1, x_2] \in \mathcal{L}(G_*, A_1)$ and $\phi(\tau_2) = u_2 = u_2'[x_1, x_2, x_2] \in \mathcal{L}(G_*, A_2)$, where u_1 contains just two occurrences of x_1 and one occurrence of x_2 and so on. We then have

$$\begin{aligned}\phi(\pi(\tau_1, \tau_2)) &= s[u_1, u_2] \\ &= u_1'[a(u_2'[b, x_1, x_1]), a(u_2'[b, x_1, x_1]), c] \\ &\qquad\qquad \in \mathcal{L}(G, A_0)\,.\end{aligned}$$

Now let us consider a derivation tree τ that has $\pi(\tau_1, \tau_2)$ as a subtree. Let $t = \phi(\tau)$ and $u = \phi(\pi(\tau_1, \tau_2))$. By Lemma 4, $t = e[u]$ for some $e = t'[y(t'')] \in \mathbb{E}^{m,1}$. We then have

$$\begin{aligned}t &= e[u_1'[a(u_2'[b, t'', t'']), a(u_2'[b, t'', t'']), c]] \\ &= e[s[s_1, s_2]]\end{aligned}$$

for

$$\begin{aligned}s_1 &= u_1'[x_1, a(u_2'[b, t'', t'']), x_2] \in \mathbb{S}^{2,1}, \\ s_2 &= u_2'[x_1, x_2, t''] \in \mathbb{S}^{2,1}.\end{aligned}$$ □

By $\mathbb{G}(p, q, r)$ we denote the class of IO-CFTGs G such that $N = \bigcup_{m=1}^{r} N_m$ and for every rule $A \to s$, we have $s \in \mathbb{S}^{*,p}$ and $|s|_N \le q$. We will consider only grammars in $\mathbb{G}(p, q, r)$ with fixed and small numbers p, q, r.

Proposition 6. *The uniform membership problem for $\mathbb{G}(p, q, r)$ for fixed p, q, r can be solved in polynomial time.*

Proof. This proposition is a corollary to known results. Particularly Kanazawa's technique that reduces membership problems to datalog queries will give an elegant parsing algorithm (Kanazawa, 2007; Beeri and Ramakrishnan, 1991). Yet to make this paper self-contained, we give a brief sketch of a CKY-style algorithm for the problem. Let $\langle G, t\rangle$ with $G \in \mathbb{G}(p,q,r)$ and $t \in \mathbb{T}_\Sigma$ be an instance of the problem. For each nonterminal symbol A of rank m, we compute a set $Q_A \subseteq (\mathrm{Sub}^{0,*}(\{t\}))^{1+m}$ of $(1+m)$-tuples of subtrees of t so that $\langle t_0, t_1, \dots, t_m\rangle \in Q_A$ iff $s[t_1, \dots, t_m] = t_0$ for some $s \in \mathcal{L}(G, A)$. We initialize those sets Q_A to be empty and then monotonically and recursively expand the sets by referring to the rules of G. We have $t \in \mathcal{L}(G)$ if $t \in Q_A$ for some initial symbol $A \in I$. When all sets converge without satisfying this condition, we conclude $t \notin \mathcal{L}(G)$. Note that the bounds q and r play a crucial role for the polynomial-time computability. $\square$

3 Learning IO-CFTGs

Our learning scheme is *identification in the limit from positive data and membership queries* following Clark (2010). The learner is given an infinite sequence consisting of all and only trees from a learning target L_* and each time the learner gets a tree, it outputs an IO-CFTG as its conjecture. The learner has access to a membership oracle, which answers whether an arbitrary tree belongs to L_*. The sequence of the conjectures must eventually converge to an IO-CFTG representing L_*.

Hereafter we fix a target language L_*. Distributional learning observes the distribution of stubs in environments with respect to L_*. Let us define dual polar maps as follows. For $S \subseteq \mathbb{S}^{m,*}$ and $E \subseteq \mathbb{E}^{m,*}$,

$$S^{\rhd} = \{\, e \in \mathbb{E}^{m,*} \mid e \odot S \subseteq L_* \,\}\,,$$
$$E^{\lhd} = \{\, s \in \mathbb{S}^{m,*} \mid E \odot s \subseteq L_* \,\}\,.$$

We write $S^{\rhd\lhd}$ for $(S^{\rhd})^{\lhd}$ and so on. One can easily see that $S^{\rhd\lhd\rhd} = S^{\rhd}$ and $E^{\lhd\rhd\lhd} = E^{\lhd}$. A learner extracts stubs and environments from given positive examples and ask the oracle which combination of those give grammatical trees in L_*. When a positive example t is derived by a derivation tree τ that has a A-derivation subtree τ' for some A, in general there is no bound p such that $\mathcal{L}(G, A) \subseteq \mathbb{S}^{m,p}$ where m is the rank of A. That is, there exist exponentially many potential constituents $s' \in \mathrm{Sub}^{m,*}(\{t\})$ and extracting all such stubs is not tractable. The same holds for the environment side. In stead we consider only p-copying stubs and environments. We define restricted polar maps as follows:

$$S^{(E)} = S^{\rhd} \cap E \text{ and } E^{(S)} = E^{\lhd} \cap S\,.$$

Among the two types of approaches in the distributional learning, we first discuss the so-called primal one.

3.1 Primal property

The following definition is an easy translation of the k-FKP for CFGs (Yoshinaka, 2011b) to IO-CFTGs.

Definition 7. We say that an IO-CFTG G has *the* (k,p)*-kernel property* ((k,p)-KP) if every nonterminal A of rank m admits a set $S_A \subseteq \mathbb{S}^{m,p}$ s.t. $|S_A| \le k$ and $S_A^{\rhd\lhd} = \mathcal{L}(G, A)^{\rhd\lhd}$. We call such a set S_A a *characterizing (stub) set* of A.

In the CFG case, Clark et al. (2009) showed that CFGs with the 1-FKP generate all regular languages and other simple CFLs including the Dyck language. However the $(1,p)$-KP is still too strong to describe non-linear languages.

Example 8. Let an IO-CFTG consist of the following rules, where $\Sigma_0 = \{a\}$, $\Sigma_1 = \{b\}$, $\Sigma_2 = \{c\}$, $N_0 = \{I, A\}$, $N_1 = \{C\}$:

$$I \to C(A),\ C \to C(c(x_1, x_1)),\ C \to x_1,$$
$$A \to b(A),\ A \to a,$$

Let $b^0(a) = a$ and $b^{i+1}(a) = b(b^i(a))$. The defined language consists of trees of the form $s[y \mapsto b^k(a)]$ where s is a balanced binary tree whose internal nodes and leaves are labeled by c and y, respectively. The nonterminal A does not have a singleton characterizing set. Any element $b^k(a)$ of $\mathcal{L}(G, A)$ admits its unique environment $e = c(y, b^k(a))$, in the sense that $\{e\}^{\lhd} \cap \mathcal{L}(G, A) = b^k(a)$. Thus the $(1,p)$-KP does not hold. On the other hand, one can easily see that $\{a, b(a)\}$, $\{x_1, c(x_1, x_1)\}$ and $\{a, c(a, a)\}$ characterize A, C and I, respectively (2-KP).

The primal approaches use environments to check the correctness of constructed rules. In the linear case, we can extract all required environments, which are linear, from given examples in polynomial time, but in our general setting, we have to collect non-linear environments as well, which is computationally intractable. We further require the following condition.

Definition 9. We say that a language L_* has *the (p, r, k)-fiducial environment property* ((p, r, k)-FEP) if for any $S \subseteq \mathbb{S}^{m,p}$ with $m \le r$ and $|S| \le k$, we have $(S^{(\mathbb{E}^{m,p})})^{\triangleleft} = S^{\triangleright\triangleleft}$.

The (p, r, k)-FEP means that to validate whether a stub s may occur in every environment that accepts S, it is enough to confirm that it is the case for every p-copying environment. Actually the (p, r, k)-FEP is rather strong for $k > 1$. The finite language $\{a(d), b(d), c(d), a^{p+1}(d), b^{p+1}(d)\}$ does not have the $(p, 1, 2)$-FEP ($c(x_1) \in S^{(\mathbb{E}^{1,p})\triangleleft} \setminus S^{\triangleright\triangleleft}$ for $S = \{a(x_1), b(x_1)\}$). On the other hand, one can show that the $(p, r, 1)$-FEP is satisfied by every language. Our first learning target is the following class:

$$\mathbb{P}(p, q, r, k) = \{\, G \in \mathbb{G}(p, q, r) \text{ with the } (p, k)\text{-KP and } (p, r, k)\text{-FEP} \,\}$$

The above discussion on the property shows that the defined language class does not cover the class of simple CFTGs with k-FKP. Checking the property FEP is cumbersome even for the very simple IO-CFTG in Figure 1. Yet the author found no evidence suggesting that the grammar does not belong to $\mathbb{P}(2, 2, 2, 2)$.

3.2 Primal learner

Our learner (Algorithm 1) computes its conjecture $\mathcal{G}(T, F)$ from $T = \bigcup_{0 \le m \le r} T_m$ with $T_m \subseteq \mathrm{Sub}^{m,p}(D)$ and $F = \bigcup_{0 \le m \le r} F_m$ with $F_m = \mathrm{Env}^{m,p}(D)$ for a finite tree set $D \subseteq L_*$. The nonterminal set is $N^T = \bigcup_{0 \le m \le r} N^T_m$ where $N^T_m = \{\, [\![S]\!] \mid S \subseteq T_m \wedge |S| \le k \,\}$. $[\![S]\!] \in I$ if $S \subseteq L_*$. We have a rule of the form

$$[\![S_0]\!] \to s[[\![S_1]\!], \dots, [\![S_n]\!]]$$

iff for some $m_0, \dots, m_n \le r$, $[\![S_i]\!] \in N^T_{m_i}$ for $i = 0, \dots, n$, $s \in \mathbb{S}^{m_0,p}_{\Sigma \cup \{y_1,\dots,y_n\}}$ in which each of $y_1, \dots, y_n$ occurs just once in s with $n \le q$,

1. there are $s_i \in \mathbb{S}^{m_i,1}_{\Sigma}$ for $i = 1, \dots, n$ such that $s[s_1, \dots, s_n] \in T_{m_0}$,
2. $S_0^{(F_m)} \odot s[S_1, \dots, S_n] \subseteq L_*$.

Lemma 10. *One can construct $\mathcal{G}(T, F)$ in polynomial time in $\|D\|$ with the aid of a membership oracle.*

Algorithm 1 $\mathcal{A}(p, q, r, k)$

Data: trees $t_1, t_2, \dots \in L_*$
Result: IO-CFTGs $G_1, G_2, \dots$
let $D := T := F := \varnothing$; $\hat{G} := \mathcal{G}(T, F)$;
for $n = 1, 2, \dots$ **do**
  let $D := D \cup \{t_n\}$; $F := \bigcup_{0 \le m \le r} \mathrm{Env}^m(D)$;
  if $D \not\subseteq \mathcal{L}(\hat{G})$ **then**
    let $T := \bigcup_{0 \le m \le r} \mathrm{Sub}^{m,p}(D)$;
  end if
  output $\hat{G} = \mathcal{G}(T, F)$ as G_n;
end for

Proof. By Lemma 2, one can compute T and F in polynomial time in $\|D\|$.

We discuss the first condition of the rule construction. For each $t \in T_m$, there are at most $|t|^{1+m_1}$ pairs of $s_1 \in \mathbb{S}^{m_1,1}$ and t_1 with just one occurrence of y_1 such that $t = t_1[y_1 \mapsto s_1]$, since such a pair corresponds to at most $1 + m_1$ positions on a path from the root to a leaf of t. Recursively one can determine $s_2, s_3, \dots, s_n \in \mathbb{S}^{m_1,1}$ such that $t_i = t_{i+1}[y_{i+1} \mapsto s_{i+1}]$ and $y_1, \dots, y_i$ occur in t_{i+1}. t_n will be what we would like. There can be at most $|t|^{n+m_1+\cdots+m_n}$ possible choices and the enumeration can be done in polynomial time.

It is easy to see that checking the second condition can be done in polynomial time with the aid of a membership oracle. □

A rule of the form $[\![S_0]\!] \to s[[\![S_1]\!], \dots, [\![S_n]\!]]$ is said to be *incorrect* if $S_0^{\triangleright} \odot s[S_1, \dots, S_n] \not\subseteq L_*$. We say that $F \subseteq \mathbb{E}^{*,*}$ is *fiducial* on T (with respect to L_*) if $\mathcal{G}(T, F)$ has no incorrect rules. Clearly, if F is fiducial on T then so is every superset of F.

Lemma 11. *Every T admits a fiducial set $F \subseteq \mathbb{E}^{*,p}$ such that*

- *the cardinality $|F|$ is polynomially bounded by the description size of T,*
- *for each $e \in F_m$ there is $s \in \mathbb{S}^{m,p}$ such that $e \odot s \in L_*$.*

Proof. Suppose a rule $[\![S_0]\!] \to s[[\![S_1]\!], \dots, [\![S_n]\!]]$ is incorrect, which by the (p, r, k)-FEP means $S_0^{(\mathbb{E}^{m,p})} \odot s[S_1, \dots, S_n] \not\subseteq L_*$. There is $e \in \mathbb{E}^{m,p}$ such that $e \odot S_0 \subseteq L_*$ and $e \odot s[S_1, \dots, S_n] \not\subseteq L_*$. If e is in F, such a rule is excluded from $\mathcal{G}(T, F)$. In this case, we have $e \odot s \in L_*$ for any $s \in S_0 \subseteq \mathbb{S}^{m,p}$. □

Lemma 12. *Let $\hat{G} = \mathcal{G}(T, F)$ with F fiducial on T. For any $[\![S_0]\!] \in N^T_m$ and $e \in \mathbb{E}^{m,*}$,*

$$e \odot S_0 \subseteq L_* \implies e \odot \mathcal{L}(\hat{G}, [\![S_0]\!]) \subseteq L_* .$$

Proof. Suppose that $e \odot S_0 \subseteq L_*$. We prove by induction on the derivation of $s_0 \in \mathcal{L}(\hat{G}, [\![S_0]\!])$ that $e \odot s_0 \in L_*$. Let $s_0 \in \mathcal{L}(\hat{G}, [\![S_0]\!])$ be derived by a rule $[\![S_0]\!] \to s[[\![S_1]\!], \dots, [\![S_n]\!]]$ with $s_i \in \mathcal{L}(\hat{G}, [\![S_i]\!])$ and $s_0 = s[s_1, \dots, s_n]$. Since the rule is correct by the assumption, we have

$$e \odot s[S_1, \dots, S_n] \subseteq L_* ,$$

which implies

$$\begin{aligned}(e \odot s[y_1, S_2, \dots, S_n]) \odot S_1 \subseteq L_* \\ \implies (e \odot s[y_1, S_2, \dots, S_n]) \odot s_1 \\ = e \odot s[s_1, S_2 \dots, S_n] \subseteq L_*\end{aligned}$$

by the induction hypothesis on $[\![S_1]\!]$. By repeatedly applying the same argument, we finally obtain

$$e \odot s[s_1, \dots, s_n] = e \odot s_0 \in L_* . \qquad \square$$

Lemma 13. *If F is fiducial on T, then $\mathcal{L}(\mathcal{G}(T, F)) \subseteq L_*$.*

Proof. Apply Lemma 12 to an initial symbol $[\![S]\!] \in I$ with $e = y$. $\square$

Suppose that $G_* \in \mathbb{P}(p, q, r, k)$ generates L_*, We say that $T \subseteq \mathrm{Sub}^{*,*}(L_*)$ is *adequate* if (i) T_m includes a characterizing stub set S_A for every nonterminal $A \in N_m$ of G_* and (ii) for every rule $A \to s[A_1, \dots, A_n]$ of G_*, there are $s_i \in \mathbb{S}^{m_i,1}_\Sigma$ such that $s[s_1, \dots, s_n] \in T_m$. Clearly if T is adequate, every superset of T is adequate.

Lemma 14. *There is a finite set $D \subseteq L_*$ such that $\bigcup_{0 \le m \le r} \mathrm{Sub}^{m,p}(D)$ is adequate and $|D|$ is polynomially bounded by the description size of G_*.*

Proof. By Lemmas 4 and 5. $\square$

Lemma 15. *If T is adequate, $\mathcal{L}(G_*) \subseteq \mathcal{L}(\mathcal{G}(T, F))$ for any F.*

Proof. We show that for every rule $A_0 \to s[A_1, \dots, A_n]$ of G_*, $\hat{G}$ has a rule $[\![S_0]\!] \to s[[\![S_1]\!], \dots, [\![S_n]\!]]$ where S_i are characterizing sets of A_i for $i = 0, \dots, n$. Suppose that $e \odot S_0 \subseteq L_*$ for $e \in F_m$. Since S_0 is a characterizing set for A_0, we have $e \odot \mathcal{L}(G_*, A_0) \subseteq L_*$. The fact $S_i \subseteq \mathcal{L}(G_*, A_i)$ for $i = 1, \dots, n$ implies $s[S_1, \dots, S_n] \subseteq \mathcal{L}(G_*, A_0)$ and thus $e \odot s[S_1, \dots, S_n] \subseteq L_*$. Hence $\mathcal{G}(T, F)$ has the rule $[\![S_0]\!] \to s[[\![S_1]\!], \dots, [\![S_n]\!]]$. $\square$

Corollary 16. *If T is adequate and F is fiducial on T, then $\mathcal{L}(\mathcal{G}(T, F)) = L_*$.*

Proof. By Lemmas 12 and 15. $\square$

Proposition 17. *Algorithm 1 for fixed p, q, r, k identifies $\mathbb{P}(p, q, r, k)$ in the limit from positive data and membership queries.*

Proof. We first show that the conjecture never converges to a wrong grammar. If the current conjecture $\hat{G} = \mathcal{G}(T, F)$ is such that $\mathcal{L}(G_*) \not\subseteq \mathcal{L}(\hat{G})$, at some point some $t \in \mathcal{L}(G_*) \setminus \mathcal{L}(\hat{G})$ will be given, which expands the conjecture. If $\mathcal{L}(\hat{G}) \not\subseteq \mathcal{L}(G_*)$, Lemma 13 implies that F is not fiducial on T and $\hat{G}$ has an incorrect rule $[\![S_0]\!] \to s[[\![S_1]\!], \dots, [\![S_n]\!]]$. Lemma 11 implies that there is $e \in \mathrm{Env}^{m,p}(L_*)$ such that $e \odot S_0 \subseteq L_*$ and $e \odot s[S_1, \dots, S_n] \not\subseteq L_*$, which rejects the rule. The conjecture cannot be changed infinitely many times. At some point T will be adequate by Lemma 14, which fixes the nonterminal set of $\hat{G}$. Once T is fixed, expansion of F causes deletion of rules, which can happen at most finitely many times. Therefore, the conjecture converges to a grammar representing the target. $\square$

All in all, Algorithm 1 learns $\mathbb{P}(p, q, r, k)$ efficiently for small p, q, r, k.

3.3 Dual property

The dual properties of the (k, p)-KP and (p, r, k)-FEP are given as follows.

Definition 18. We say that an IO-CFTG G has *the (k, p)-environment property* (((k, p)-EP) if every nonterminal A of rank m admits a set $E_A \subseteq \mathbb{E}^{m,p}$ s.t. $|E_A| \le k$ and $E_A^{\lhd\rhd} = \mathcal{L}(G, A)^{\rhd}$. We call such a set E_A a *characterizing (environment) set* of A.

A language L_* has *the (p, r, k)-fiducial stub property* ((p, r, k)-FSP) if for any $E \subseteq \mathbb{E}^{m,p}$ with $m \le r$ and $|E| \le k$, we have $(E^{(\mathbb{S}^{m,p})})^{\rhd} = E^{\lhd\rhd}$.

The grammar of Example 8 satisfies the 1-EP, whereas 1-KP does not hold. The environment sets $\{b(y)\}$, $\{y(c(a, a))\}$ and $\{y\}$ characterize A, C and I, respectively. Hence, one might think the dual approach is somewhat better. However, while every tree language satisfies the $(p, r, 1)$-FEP, it is not the case for $(p, r, 1)$-FSP. For the language $\{a(c(d,d)), b(c(d,e)), b(c(e,d))\}$, one sees that $b(y(e)) \in \{a(y(d))\}^{\mathbb{S}^{1,1}\rhd} \setminus \{a(y(d))\}^{\lhd\rhd}$.

We target the following class of tree languages by a dual approach.

$$\mathbb{D}(p,q,r,k) = \{\, G \in \mathbb{G}(p,q,r) \text{ with the } (p,k)\text{-EP and } (p,r,k)\text{-FSP} \,\}\,.$$

3.4 Dual learner

Algorithm 2 is our learner for $\mathbb{D}(p,q,r,k)$, which is quite symmetric to Algorithm 1. It computes its conjecture $\mathcal{G}(H,F,T)$ from $H \subseteq D$, $F = \bigcup_{0\le m\le r} F_m$, and $T = \bigcup_{0\le m\le r} T_m$ with $F_m \subseteq \mathrm{Env}^{m,p}(D)$ and $T_m = \mathrm{Sub}^{m,p}(D)$ for a finite tree set $D \subseteq L_*$. The nonterminal set is $N^F = \bigcup_{0\le m\le r} N^F_m$ where $N^F_m = \{\, [\![E]\!] \mid E \subseteq F_m \wedge |E| \le k \,\}$ and the initial symbol set is the singleton $I = \{[\![\{y\}]\!]\}$. We have a rule of the form

$$[\![E_0]\!] \to s[[\![E_1]\!], \dots, [\![E_n]\!]]$$

iff for some $m_0, \dots, m_n \le r$, $[\![E_i]\!] \in N^F_{m_i}$ for $i = 0, \dots, n$, $s \in \mathbb{S}^{m_0,p}_{\Sigma\cup\{y_1,\dots,y_n\}}$ in which each of $y_1, \dots, y_n$ occurs just once with $n \le q$,

1. there are $s_i \in \mathbb{S}^{m_i,1}_{\Sigma}$ for $i = 1, \dots, n$ such that $s[s_1, \dots, s_n] \in \mathrm{Sub}^{m_0,p}(H)$,
2. $E_0 \odot s[E_1^{(T_{m_1})}, \dots, E_n^{(T_{m_n})}] \subseteq L_*$.

Algorithm 2 $\mathcal{B}(p,q,r,k)$

Data: trees $t_1, t_2, \dots \in L_*$
Result: IO-CFTGs $G_1, G_2, \dots$
let $D := H := F := T := \varnothing$; $\hat{G} := \mathcal{G}(H,F,T)$;
for $n = 1, 2, \dots$ **do**
  let $D := D \cup \{t_n\}$; $T := \bigcup_{0\le m\le r} \mathrm{Sub}^m(D)$;
  if $D \not\subseteq \mathcal{L}(\hat{G})$ **then**
    let $H := D$ and $F := \bigcup_{0\le m\le r} \mathrm{Env}^{m,p}(D)$;
  end if
  output $\hat{G} = \mathcal{G}(H,F,T)$ as G_n;
end for

The following lemmas, corollary and proposition are exactly in parallel with those in the primal approach, namely, Lemmas 10 to 15, Corollary 16 and Proposition 17.

Lemma 19. *One can construct $\mathcal{G}(H,F,T)$ in polynomial time in $\|D\|$ with the aid of a membership oracle.*

A rule of the form $[\![E_0]\!] \to s[[\![E_1]\!], \dots, [\![E_n]\!]]$ is said to be *incorrect* if $E_0 \odot s[E_1^{\lhd}, \dots, E_n^{\lhd}] \not\subseteq L_*$. We say that $T \subseteq \mathbb{S}^{*,*}$ is *fiducial* on $\langle H, F\rangle$ (with respect to L_*) if $\mathcal{G}(H,F,T)$ has no incorrect rules. Clearly, if T is fiducial on $\langle H, F\rangle$ then so is every superset of T.

Lemma 20. *Every $\langle H, F\rangle$ admits a fiducial set $T \subseteq \mathbb{E}^{*,p}$ such that*

- *the cardinality $|T|$ is polynomially bounded by the description size of $\langle H, F\rangle$,*
- *for each $s \in T_m$ there is $e \in \mathbb{E}^{m,p}$ such that $e \odot s \in L_*$.*

Lemma 21. *Suppose that T is fiducial on $\langle H, F\rangle$. For every $[\![E]\!] \in F^T_m$ we have*

$$E \odot \mathcal{L}(\mathcal{G}(H,F,T), [\![E]\!]) \subseteq L_*\,.$$

Lemma 22. *If T fiducial on $\langle H, F\rangle$, then $\mathcal{L}(\mathcal{G}(H,F,T)) \subseteq L_*$.*

Suppose that $G_* \in \mathbb{D}(p,q,r,k)$ generates L_*, We say that a pair of $F \subseteq \mathrm{Env}^{*,*}(L_*)$ and $H \subseteq L_*$ is *adequate* if F_m includes a characterizing environment set E_A for every nonterminal $A \in N_m$ of G_* and for every rule $A \to s[A_1, \dots, A_n]$ of G_*, there are $e \in \mathbb{E}^{n,1}$ and $s \in \mathbb{S}^{m_0,p}_{\Sigma\cup\{y_1,\dots,y_n\}}$ such that $e \odot s \in H$ and each of $y_1, \dots, y_n$ occurs just once in s with $n \le q$. Clearly if $\langle H, F\rangle$ is adequate, every pair $\langle H', F'\rangle$ with $H' \supseteq H$ and $F' \supseteq F$ is adequate.

Lemma 23. *There is a finite set $D \subseteq L_*$ such that $\langle D, \bigcup_{0\le m\le r} \mathrm{Env}^{m,p}(D)\rangle$ is adequate and $|D|$ is polynomially bounded by the description size of G_*.*

Lemma 24. *If $\langle H, F\rangle$ is adequate, $\mathcal{L}(G_*) \subseteq \mathcal{L}(\mathcal{G}(H,F,T))$ for any T.*

Corollary 25. *If $\langle H, F\rangle$ is adequate and T is fiducial on $\langle H, F\rangle$, then $\mathcal{L}(\mathcal{G}(H,F,T)) = L_*$.*

Proposition 26. *Algorithm 2 for fixed p, q, r, k identifies $\mathbb{D}(p,q,r,k)$ in the limit from positive data and membership queries.*

All in all, Algorithm 2 learns $\mathbb{D}(p,q,r,k)$ efficiently.

4 Discussion

Motivated for investigating the learning of meanings with words, we in this paper have discussed how distributional learning techniques can be applicable to IO-CFTGs. Copying operations seem very important for generating natural meaning representations in spite of technical difficulties in learning non-linear structures. The approaches presented in this paper are rather naive applications of existing techniques with additional conditions, the fiducial environment property and fiducial stub property, which are convenient assumptions for making our learners run in polynomial-time. It is not clear whether the introduced conditions are too much restrictive for meaning representations. The author hopes this paper to become a basis for other distributional properties more reasonable for expressivity and learnability.

Generalizing the learning of IO-CFTGs to *almost linear* ACGs must be very important future work. A simply typed lambda term is said to be almost linear if it has no vacuous λ-abstraction and only variables assigned atomic types may occur more than once. It is shown that almost linear lambda terms inherit nice properties of linear lambda terms (Kanazawa, 2007; Kanazawa, 2011). Targeting almost linear ACGs seems quite promising.

Clark (Clark, 2011) has proposed an algorithm that learns an interesting subclass of synchronous CFGs from positive data only, where languages in the class satisfy functionality. Though the relation between words and meanings in natural languages are not a function, the relation is very sparse. Therefore combination of Clark's and our approaches is an interesting direction of further research for a basic model of natural language acquisition taking syntax-semantics interface into account.

References

Catriel Beeri and Raghu Ramakrishnan. 1991. On the power of magic. *Journal of Logic Programming*, 10(3&4):255–299.

Alexander Clark and Rémi Eyraud. 2007. Polynomial identification in the limit of substitutable context-free languages. *Journal of Machine Learning Research*, 8:1725–1745.

Alexander Clark and Ryo Yoshinaka. 2012. Beyond semilinearity: Distributional learning of parallel multiple context-free grammars. In Jeffrey Heinz, Colin de la Higuera, and Tim Oates, editors, *Proceedings of the Eleventh International Conference on Grammatical Inference*, volume 21 of *JMLR Workshop and Conference Proceedings*, pages 84–96.

Alexander Clark, Rémi Eyraud, and Amaury Habrard. 2009. A note on contextual binary feature grammars. In *EACL 2009 workshop on Computational Linguistic Aspects of Grammatical Inference*, pages 33–40.

Alexander Clark. 2010. Learning context free grammars with the syntactic concept lattice. In José M. Sempere and Pedro García, editors, *ICGI*, volume 6339 of *Lecture Notes in Computer Science*, pages 38–51. Springer.

Alexander Clark. 2011. Inference of inversion transduction grammars. In Lise Getoor and Tobias Scheffer, editors, *ICML*, pages 201–208. Omnipress.

Philippe de Groote and Sylvain Pogodalla. 2004. On the expressive power of abstract categorial grammars: Representing context-free formalisms. *Journal of Logic, Language and Information*, 13(4):421–438.

Joost Engelfriet and Erik Meineche Schmidt. 1977. IO and OI. I. *J. Comput. Syst. Sci.*, 15(3):328–353.

Makoto Kanazawa. 2007. Parsing and generation as datalog queries. In John A. Carroll, Antal van den Bosch, and Annie Zaenen, editors, *ACL*. The Association for Computational Linguistics.

Makoto Kanazawa. 2011. Parsing and generation as datalog query evaluation.

Anna Kasprzik and Ryo Yoshinaka. 2011. Distributional learning of simple context-free tree grammars. In Jyrki Kivinen, Csaba Szepesvári, Esko Ukkonen, and Thomas Zeugmann, editors, *Algorithmic Learning Theory*, volume 6925 of *Lecture Notes in Computer Science*, pages 398–412. Springer.

William C. Rounds. 1970. Mappings and grammars on trees. *Mathematical Systems Theory*, 4(3):257–287.

Ryo Yoshinaka and Makoto Kanazawa. 2011. Distributional learning of abstract categorial grammars. In Sylvain Pogodalla and Jean-Philippe Prost, editors, *LACL*, volume 6736 of *Lecture Notes in Computer Science*, pages 251–266. Springer.

Ryo Yoshinaka. 2011a. Efficient learning of multiple context-free languages with multidimensional substitutability from positive data. *Theoretical Computer Science*, 412(19):1821–1831.

Ryo Yoshinaka. 2011b. Towards dual approaches for learning context-free grammars based on syntactic concept lattices. In Giancarlo Mauri and Alberto Leporati, editors, *Developments in Language Theory*, volume 6795 of *Lecture Notes in Computer Science*, pages 429–440. Springer.