On the Learnability of Recursively Enumerable Languages from Good Examples

Sanjay Jain [a], Steffen Lange [b], Jochen Nessel [c]

[a] *Department of Information Systems and Computer Science, National University of Singapore, Singapore 119260, Email: sanjay@iscs.nus.edu.sg*

[b] *Institut für Informatik, Universität Leipzig, PF 920, D-04009 Leipzig, Germany, Email: slange@informatik.uni-leipzig.de*

[c] *FB Informatik, Universität Kaiserslautern, PF 3049, D-67653 Kaiserslautern, Germany, Email: nessel@informatik.uni-kl.de*

Abstract

The present paper investigates identification of indexed families $\mathcal{L}$ of recursively enumerable languages from *good examples*. We distinguish *class preserving* learning from good examples (the good examples have to be generated with respect to a hypothesis space having the same range as $\mathcal{L}$) and *class comprising* learning from good examples (the good examples have to be selected with respect to a hypothesis space comprising the range of $\mathcal{L}$). A learner is required to learn a target language on every finite superset of the good examples for it. If the learner's first and only conjecture is correct then the underlying learning model is referred to as *finite identification from good examples* and if the learner makes a finite number of incorrect conjectures before always outputting a correct one, the model is referred to as *limit identification from good examples*.

In the context of class preserving learning, it is shown that the learning power of finite and limit identification from good text examples coincide. When class comprising learning from good text examples is concerned, limit identification is strictly more powerful than finite learning. Furthermore, if learning from good informant examples is considered, limit identification is superior to finite identification in the class preserving as well as in the class comprising case.

Finally, we relate the models of learning from good examples to one another as well as to the standard learning models in the context of Gold-style language learning.

Keywords: Inductive Inference, Computational Learning Theory, Good Examples.

1 Introduction

Consider the identification of formal languages from positive data. A machine is fed all the strings and no non-strings of a language L, in any order, one string at a time. The machine, as it receives strings of L, outputs a sequence of grammars. The machine is said to identify L just in case the sequence of grammars converges to a grammar for L. This is the paradigm of identification in the limit introduced by Gold [12].

But there are other situations in life, where we speak of learning without hesitation, and which are not covered by Gold's model. As an example, consider an ordinary school. If a pupil has to learn a language, this is not achieved by just presenting all correct sentences of that language in random order. Instead, a teacher, who knows the language the pupil wants to learn, carefully selects the sentences the pupil will see. Furthermore, at some point the teacher will stop teaching, because he feels the examples should suffice for the learning task.

This observation led Freivalds, Kinber and Wiehagen [10] to study models in which learners are provided with only finitely many examples (of a possibly infinite language), though these examples may include important ones. Freivalds, Kinber and Wiehagen referred to these important examples as *good examples*. The revised learning model then requires the learner to come up with a grammar for the language when it is provided a set of examples containing all good examples. If the learner's first and only conjecture is correct then the model is referred to as *finite identification from good examples* and if the learner makes a finite number of incorrect conjectures before converging to a correct one, the model is referred to as *identification in the limit from good examples*.[1]

It should be noted that in the model just described, the learner may receive some superset of the good examples and not necessarily just the good examples. This avoids some trivial cases where learnability can be achieved by a suitable encoding of a correct grammar into the good examples (see for example [3]). The model places as an additional requirement that it has to be possible to effectively generate the good examples for a language (from any of its grammars in the hypothesis space). This allows a helpful teacher to provide the good examples needed for learning. We refer the reader to [10] and [15] for additional motivation and discussion on these models.

[1] It should be noted that the learning power of the criteria of inference from good examples considered in this paper is not effected if we consider set-driven learning in the sense that the output of an admissible learning machine depends exclusively on the content of the input, thereby neglecting the length and order of the data sequence.

As a concrete example consider pattern languages [1]. For a pattern p, we can take $\{w \in \Sigma^* \mid |w| = |p|, w \in L(p)\}$ as a set of good examples for $L(p)$ (for class preserving finite identification from good text examples). [2]

Learning from good examples was first considered by Freivalds, Kinber and Wiehagen in the context of function learning [10]. Lange, Nessel and Wiehagen [15] extended this study to include indexed families of recursive languages. For this latter case, they showed that the power of finite identification and identification in the limit is the same as long as class preserving learning is considered. They left open the issue of whether a similar result holds in the context of learning language classes that are not necessarily indexed families of recursive languages. In this paper, we provide a solution to this question.

Other authors have attacked the problem of learning with help from selected examples as well. In [19], Shinohara and Miyano study the learnability from finite sets of examples. However, these finite sets need not be effectively computable from the grammars in the hypothesis space and the learning machines are not required to learn when a proper superset of these good examples is given as input (in contrast to our requirements, as in Definition 5 below). Motoki in [16] and Baliga, Case and Jain in [3] consider language learning from all positive data and a finite amount of negative data. Again, they did not require negative examples to be computable. However, [3] did consider the case of allowing supersets of negative examples. A model similar to the one studied here is also investigated by Goldman and Mathias in [13]. Their main interest lies in decribing concept classes that can be learned in polynomial time with polynomial size (in the description of the target concept) example sets. They also address the problem of coding the target concept into the examples.

In the case of learning indexed families of recursive languages, the hypothesis space chosen is also an indexed family of recursive languages. The effective generation of the good examples is with respect to the grammars from the hypothesis space. Regarding the choice of hypothesis spaces, usually two situations are considered: *class preserving* (when the hypothesis space contains exactly the languages in the class being learned) and *class comprising* (when the hypothesis space may contain descriptions for languages in addition to the language class being learned). In this paper we consider learning from good examples for indexed families of recursively enumerable languages. [3] We take

[2] Note that we do not need such a large set of good examples for learning pattern languages (for example see [1] and [21]). Furthermore, if we are interested in class comprising identification in the limit from good examples, for $k \in N$, for unions of k pattern languages, we can obtain good examples based on Theorem 4(a). Such a set of good examples can also be explicitly obtained using a trick used in [14,6]. We do not know at present whether we can do this for class preserving finite identification from good examples.

[3] We refer the reader to [9] for some nice characterizations for learnability of indexed

the hypothesis space also to be an indexed family of recursively enumerable languages. Some of our results can also be extended to the case of learning arbitrary classes of r.e. languages, i.e., classes which may not possess an enumeration that contains all and only the languages in the class.

In the present paper we consider combinations of learning from good text or good informant examples, finite or limit learning, class preserving or class comprising learning. In addition we compare the resulting inference types with the standard inference types. Some of the highlights are briefly discussed next.

We first consider learning from good text examples. We show the following: (i) for class preserving learning, the power of finite learning and limit learning from good text examples coincide (Theorem 1 (a)); (ii) for class comprising learning, the power of finite and limit learning from good text examples differ (Theorem 1 (b)). As noted, the above two results resolve an open question in [15]. Theorem 3 shows that *TxtEx*, the class of all families of r.e. languages identifiable in the limit from text, is incomparable to the class of all families of r.e. languages that can be finitely learned from good text examples. In contrast, for class comprising learning, every class in *TxtEx* is learnable in the limit from good text examples (Theorem 4 (a)).

For learning from informant, we show that, for both class preserving and class comprising learning, the power of finite and limit learning from good informant examples differ (Theorem 7). This also addresses an open question in [15].

We now proceed formally.

2 Preliminaries

Any unexplained recursion theoretic notation is from [17]. N denotes the set of natural numbers, $\{0, 1, 2, \ldots\}$. Let $L \subseteq N$. We set $\overline{L} = N \setminus L$, i.e., $\overline{L}$ is the complement of L. We will use χ_L to indicate the characteristic function for language L, i.e., $\chi_L(x) = 1$ if $x \in L$, and 0 otherwise. The cardinality of a set S is written $\mathrm{card}(S)$. The maximum and minimum of a set are denoted by $\max(\cdot)$ and $\min(\cdot)$, respectively, where $\max(\emptyset) = 0$ and $\min(\emptyset) = \infty$. Let $D_0, D_1, \ldots$ be some canonical recursive indexing of finite sets [17]. *FIN* will stand for the family of all finite sets of natural numbers and *INIT* for the family $\{X_i \mid i \in N\}$, where $X_i = \{x \mid x < i\}$.

We let $\langle \cdot, \cdot \rangle$ stand for an arbitrary, computable, bijective mapping from $N \times N$ onto N [17]. Similarly, for each $m \in N$, one can define an encoding $\langle \cdot, \ldots, \cdot \rangle$

---

families of recursively enumerable languages, along the lines of characterizations for learnability of indexed families of recursive languages by Angluin [2].

for all m-tuples of natural numbers onto N.

For a partial function α, domain(α) and range(α) are the domain and range of α, respectively. We write $\alpha(x)\downarrow$ to denote the fact that $\alpha(x)$ is defined on x and $\alpha(x)\uparrow$ to denote that $\alpha(x)$ is undefined on x.

A programming system ψ is a partial recursive function of two arguments. For a programming system ψ, let ψ_i be the partial recursive function $\lambda x.\psi(i,x)$. i is also called a ψ-program for the (possibly partial) function ψ_i. We use W_i^ψ to denote the domain of ψ_i. Thus we can consider W_i^ψ as the language accepted by ψ-grammar i. Let $\mathcal{L}_\psi = \{W_i^\psi \mid i \in N\}$. Intuitively, $\mathcal{L}_\psi$ is the class of languages accepted by programs in programming system ψ. Suppose Ψ is some fixed Blum complexity measure for the programming system ψ, cf. [4]. $W_{i,s}^\psi$ will stand for the set $\{x < s \mid \Psi_i(x) < s\}$. Intuitively, $W_{i,s}^\psi$ will be the part of W_i^ψ enumerated within s steps by some fixed effective procedure for enumeration of all W_i^ψ.

A language class $\mathcal{L}$ is said to be an indexable class of r.e. languages provided there is a programming system ψ such that $\mathcal{L}_\psi = \mathcal{L}$. A language class $\mathcal{L}$ is said to be an indexable class of recursive languages provided there is a programming system ψ such that $\mathcal{L}_\psi = \mathcal{L}$ and a recursive function d such that $d(\langle i,x\rangle) = 1$ iff $\psi_i(x)\downarrow$. In other words, for an indexable class of recursive languages it is uniformly decidable whether or not $x \in W_i^\psi$.

We let $\beta_0, \beta_1, \ldots$ denote some effective enumeration of all the computable programming systems, cf. [17].

A programming system η is called an acceptable programming system [17], iff for all programming systems ψ, there exists a recursive function h, such that $(\forall i)[\eta_{h(i)} = \psi_i]$. Throughout the following, φ will be a fixed *acceptable* programming system. Let Φ be an arbitrary fixed Blum complexity measure [4] for the φ-system. For ease of notation, we sometimes use W_i instead of W_i^φ.

In order to specify the effective generation of good examples, we need to consider the notion of computable functions from N to FIN. We say that a (possibly partial) mapping F from N to FIN, is (partially) computable, iff there exists a partial recursive function α such that $\alpha(x)\downarrow$ iff $F(x)$ is defined, and for $x \in$ domain(F), $F(x) = D_{\alpha(x)}$. In other words, some Turing machine on input x, enumerates $F(x)$ and then signals that it has completed the enumeration. We let $F_0, F_1, \ldots$ denote an effective enumeration of all the partial computable mappings from N to FIN. For example, such an enumeration can easily be obtained from any acceptable numbering.

Quantifiers $\forall, \exists, \overset{\infty}{\forall}, \overset{\infty}{\exists}$, respectively, denote, for all, there exists, for all but finitely many and there exist infinitely many.

We first define the notion of text for languages. A text T is a mapping from N into $N \cup \{\#\}$. content(T) denotes the set of natural numbers in the range of T. Thus, the content of a text never includes $\#$. A text T is for L iff content$(T) = L$. Intuitively, a text T for a language L is a presentation of elements of L (repetition allowed) and no non-elements of L; $\#$'s in the presentation may be thought of as modeling pauses in data. The initial sequence of text T of length n is denoted $T[n]$. The set of all finite sequences of natural numbers and $\#$'s is denoted by SEQ. It is easy to see that there exists a computable bijection between SEQ and N. Members of SEQ are inputs to machines that learn grammars (acceptors) for r.e. languages. We let σ range over SEQ. Λ denotes the empty sequence and content(σ) the set of natural numbers in the range of σ. For two sequences σ and σ', we write their concatenation as $\sigma \cdot \sigma'$.

A language learning machine (from text) is an algorithmic mapping (possibly partial) from SEQ into N. M denotes a typical variable for a language learning machine from texts. We say that M converges on text T to i (written: $M(T)$ converges to i; $M(T)\!\downarrow = i$) just in case, for all but finitely many n, $M(T[n]) = i$. Let $M_0, M_1, \ldots$ be an enumeration of all learning machines.

We interpret the output of a machine as programs in some programming system (which need not to be acceptable). This programming system is called the hypothesis space for the machine.

The following definition introduces standard criteria for successful identification of languages.

Definition 1 ([12]). Suppose ψ is a hypothesis space.

(a) M *$TxtEx_\psi$-identifies* a text T, if $(\exists i \mid W_i^\psi = \text{content}(T))[M(T)\!\downarrow = i]$.
(b) M *$TxtEx_\psi$-identifies* an r.e. language L (written: $L \in TxtEx_\psi(M)$) just in case M $TxtEx_\psi$-identifies each text T for L.
(c) M *$TxtEx_\psi$-identifies* $\mathcal{L}$ iff M $TxtEx_\psi$-identifies each $L \in \mathcal{L}$.

Let $TxtEx_\psi$ be the collection of all indexable classes $\mathcal{L}$ of r.e. languages such that some M $TxtEx_\psi$-identifies $\mathcal{L}$. By $TxtEx$ we denote $\bigcup_\psi TxtEx_\psi$.

Definition 2 ([18,11]). M is *rearrangement independent* iff
$(\forall \sigma, \sigma' \mid \text{content}(\sigma) = \text{content}(\sigma') \ \wedge \ |\sigma| = |\sigma'|)[M(\sigma) = M(\sigma')]$

Lemma 1 ([18,11]). *If $\mathcal{L} \in TxtEx$, then there exists a hypothesis space ψ and a rearrangement independent M which $TxtEx_\psi$-identifies each language in $\mathcal{L}$.*

Definition 3 ([5]). Suppose ψ is a hypothesis space.
σ is a *locking sequence* for M on L iff

(a) $\text{content}(\sigma) \subseteq L$,
(b) $(\forall \sigma' \supseteq \sigma \mid \text{content}(\sigma') \subseteq L)[M(\sigma') = M(\sigma)]$, and
(c) $W^{\psi}_{M(\sigma)} = L$

Lemma 2 ([5]). *Suppose ψ is a hypothesis space. If M TxtEx_{ψ}-identifies L, then there exists a locking sequence for M on L.*

Definition 4 ([7]). Suppose ψ is a hypothesis space.

(a) M *TxtBc_{ψ}-identifies* a text T just in case, $(\stackrel{\infty}{\forall} n)[W^{\psi}_{M(T[n])} = \text{content}(T)]$.
(b) M *TxtBc_{ψ}-identifies* an r.e. language L (written: $L \in \mathit{TxtBc}_{\psi}(M)$) just in case M TxtBc_{ψ}-identifies each text T for L.
(c) M *TxtBc_{ψ}-identifies* $\mathcal{L}$ iff M TxtBc_{ψ}-identifies each $L \in \mathcal{L}$.

Let TxtBc_{ψ} be the collection of all indexable classes $\mathcal{L}$ of r.e. languages such that some M TxtBc_{ψ}-identifies $\mathcal{L}$. By TxtBc we denote $\bigcup_{\psi} \mathit{TxtBc}_{\psi}$.

It is easy to verify that $\mathit{TxtEx}_{\varphi} = \mathit{TxtEx}$, and $\mathit{TxtBc}_{\varphi} = \mathit{TxtBc}$.

2.2 Language identification from informant

We next introduce the notion of informant for languages. An informant I is a mapping from N into $(N \times \{0,1\})$ such that $\{x \mid (x,1) \in \text{range}(I)\}$ and $\{x \mid (x,0) \in \text{range}(I)\}$ partition the set of natural numbers. $\text{Pos}(I)$ denotes the set $\{x \mid (x,1) \in \text{range}(I)\}$ and $\text{Neg}(I)$ will stand for the set $\{x \mid (x,0) \in \text{range}(I)\}$. An informant I is for L, iff $\text{Pos}(I) = L$ (and thus $\text{Neg}(I) = \overline{L}$). The initial sequence of informant I of length n is written $I[n]$. SEG means the set of initial sequences of informants, i.e., $\text{SEG} = \{I[n] \mid n \in N \ \wedge \ I$ is an informant $\}$. The canonical informant for a language L is the informant I, such that $I(x) = (x, \chi_L(x))$.

Intuitively, an informant I for a language L is a presentation of the characteristic function of L. It is easy to see that there exists a computable bijection between SEG and N. Members of SEG are inputs to machines that learn grammars (acceptors) for r.e. languages from informant. We let τ range over SEG. $\text{Pos}(\tau)$ denotes the set $\{x \mid (x,1) \in \text{range}(\tau)\}$ and $\text{Neg}(\tau)$ denotes the set $\{x \mid (x,0) \in \text{range}(\tau)\}$.

A language learning machine (from informant) is an algorithmic mapping (possibly partial) from SEG into N. We use M to also denote a language learning machine from informant. Context will determine whether a learning machine from text or a learning machine from informant is meant. We say that M converges on informant I to i (written: $M(I)$ converges to i; $M(I)\!\downarrow = i$) just in case for all but finitely many n, $M(I[n]) = i$.

The notion of *InfEx*-identifiability and *InfBc*-identifiability, respectively, as well as the corresponding learning types are defined analogously as their text counterparts by replacing everywhere text by informant (cf. Definitions 1 and 4). It is easy to verify that $\mathit{InfEx}_{\varphi} = \mathit{InfEx}$, and $\mathit{InfBc}_{\varphi} = \mathit{InfBc}$.

3 Language Learning from good examples

Intuitively, for learning a language L from good examples, a learner is given a set of examples from L, which contain all the *good examples*. Here we assume that the set of good examples is finite. The learner is then expected to come up with a grammar for L either recursively (for finite identification) or in the limit depending on the criteria. Note that the learner is given not just the good examples, but examples which contain all the good examples. The reason is to disallow some coding of the language in the good examples (for example see [3]). We further require that the set of good examples should be effectively generable from any grammar for L in the hypothesis space. Here by effective generation we mean, there exists a recursive function which maps grammars for L in the hypothesis space to finite sets (using the canonical indexing of finite sets).

3.1 Learning from good text examples

We first consider the class comprising version.

Definition 5. Let $\mathcal{L}$ be an indexable class of r.e. languages.
M CTxtGFin-identifies $\mathcal{L}$ iff there exists a hypothesis space ψ and a recursive function G_p from N into FIN such that

(a) $\mathcal{L} \subseteq \mathcal{L}_{\psi}$,
(b) for each i, if $W_i^{\psi} \in \mathcal{L}$ then $G_p(i) \subseteq W_i^{\psi}$,
(c) for each i, if $W_i^{\psi} \in \mathcal{L}$, then $(\forall \sigma \in \mathrm{SEQ} \mid G_p(i) \subseteq \mathrm{content}(\sigma) \subseteq W_i^{\psi})(\exists n \mid W_n^{\psi} = W_i^{\psi})[M(\sigma) = n]$.

Intuitively, $G_p(i)$ above gives the set of good positive examples (text examples) for $W_i^{\psi} \in \mathcal{L}$. Part (b) in the above definition says that good examples for $W_i^{\psi} \in \mathcal{L}$ can be effectively given by $G_p(i)$. Part (c) says that, for any language $W_i^{\psi} \in \mathcal{L}$, if the input sequence σ contains the good examples $G_p(i)$, then M on σ outputs a ψ-grammar for W_i^{ψ}.

By *CTxtGFin* we denote the collection of all indexable classes $\mathcal{L}$ of r.e. languages for which there is a learning machine M which *CTxtGFin*-identifies $\mathcal{L}$.

We now define the class preserving version of learning from good examples.

Definition 6. Let $\mathcal{L}$ be an indexable class of r.e. languages.
M *PTxtGFin-identifies* $\mathcal{L}$ iff there exists a hypothesis space ψ and a recursive function G_p from N into *FIN* such that

(a) $\mathcal{L} = \mathcal{L}_\psi$,
(b) for each i, $G_p(i) \subseteq W_i^\psi$,
(c) for each i, $(\forall \sigma \in \mathrm{SEQ} \mid G_p(i) \subseteq \mathrm{content}(\sigma) \subseteq W_i^\psi)(\exists n \mid W_n^\psi = W_i^\psi)[M(\sigma) = n]$.

By *PTxtGFin* we denote the collection of all indexable classes $\mathcal{L}$ of r.e. languages for which there is a learning machine M which *PTxtGFin*-identifies $\mathcal{L}$.

We now consider limit identification from good examples. For this we need the learning machines to be a limiting recursive function from SEQ to N. Thus, for such identification criteria, we take machines to be a mapping from SEQ $\times N$ to N. We use M as a typical variable for these kinds of machines too. It will be clear from context, which type of machine is meant.

Definition 7. Let $\mathcal{L}$ be an indexable class of r.e. languages.
M *CTxtGEx-identifies* $\mathcal{L}$ iff there exists a hypothesis space ψ and a recursive function G_p from N into *FIN* such that

(a) $\mathcal{L} \subseteq \mathcal{L}_\psi$,
(b) for each i, if $W_i^\psi \in \mathcal{L}$ then $G_p(i) \subseteq W_i^\psi$,
(c) for each i, if $W_i^\psi \in \mathcal{L}$, then $(\forall \sigma \in \mathrm{SEQ} \mid G_p(i) \subseteq \mathrm{content}(\sigma) \subseteq W_i^\psi)(\exists n \mid W_n^\psi = W_i^\psi)(\overset{\infty}{\forall} m)[M(\sigma, m) = n]$.

The notion of *PTxtGEx*-identifiability can be defined in a similar manner as above. *CTxtGEx* and *PTxtGEx* will denote the collections of all indexable classes $\mathcal{L}$ of r.e. languages for which there is a learning machine M which *CTxtGEx*-identifies (*PTxtGEx*-identifies) $\mathcal{L}$.

3.2 Learning from good informant examples

Definition 8. Let $\mathcal{L}$ be an indexable class of r.e. languages.
M *CInfGFin-identifies* $\mathcal{L}$ iff there exists a hypothesis space ψ and recursive functions G_p and G_n from N into *FIN* such that

(a) $\mathcal{L} \subseteq \mathcal{L}_\psi$,
(b) for each i, if $W_i^\psi \in \mathcal{L}$ then $G_p(i) \subseteq W_i^\psi$ and $G_n(i) \subseteq \overline{W_i^\psi}$,
(c) for each i, if $W_i^\psi \in \mathcal{L}$, then $(\forall \tau \in \mathrm{SEG} \mid G_p(i) \subseteq \mathrm{Pos}(\tau) \subseteq W_i^\psi \wedge G_n(i) \subseteq \mathrm{Neg}(\tau) \subseteq \overline{W_i^\psi})(\exists n \mid W_n^\psi = W_i^\psi)[M(\tau) = n]$.

Intuitively, $G_p(i)$ above gives the set of positive good examples for $W_i^\psi \in \mathcal{L}$ and $G_n(i)$ gives the set of negative good examples for $W_i^\psi \in \mathcal{L}$. Subsequently, we use the term 'good informant examples' to refer to both sets $G_P(i)$ and $G_n(i)$. Part (b) in the above definition says that good positive and negative examples for $W_i^\psi \in \mathcal{L}$ can be effectively generated (as given by $G_p(i)$ and $G_n(i)$). Part (c) says that, for any language $W_i^\psi \in \mathcal{L}$, if the input information sequence τ contains the (appropriately labeled) good positive examples $G_p(i)$ and good negative examples $G_n(i)$, then M on τ outputs a ψ-grammar for W_i^ψ.

As above, *CInfGFin* denotes the collection of all indexable classes $\mathcal{L}$ of r.e. languages for which there is a learning machine M which *CInfGFin*-identifies $\mathcal{L}$.

One can similarly define *PInfGFin*, *CInfGEx*, and *PInfGEx*.

The following proposition follows immediately from the corresponding definitions.

Proposition 1. *Let $\lambda \in \{P, C\}$. Then,*
(a) λ*TxtGFin* $\subseteq \lambda$*TxtGEx*.
(b) λ*InfGFin* $\subseteq \lambda$*InfGEx*.
(c) λ*TxtGFin* $\subseteq \lambda$*InfGFin*.
(d) λ*TxtGEx* $\subseteq \lambda$*InfGEx*.

As we shall see, all the stated inclusions are proper, except *PTxtGFin* $\subseteq$ *PTxtGEx*.

4 Results on learning from good text examples

The following theorem shows that, for class preserving learning from good examples, there is no increase in learning power when we consider limit learning instead of finite learning. On the other hand, for class comprising learning from good examples, there is an increase in learning power when we consider limit learning instead of finite learning. Interestingly, this phenomenon can be observed on the fairly concrete level of indexable classes of recursive languages.

Theorem 1. (a) *PTxtGFin* $=$ *PTxtGEx*.
(b) *CTxtGFin* $\subset$ *CTxtGEx*.

Proof. We begin with (a). Since, by Proposition 1 (a), we have *PTxtGFin* $\subseteq$ *PTxtGEx*, it remains to verify that *PTxtGEx* $\subseteq$ *PTxtGFin*. Suppose M, $\mathcal{L}$, and G_p are given such that M *PTxtGEx*-identifies $\mathcal{L}$ using hypothesis space ψ, where G_p gives the good examples.

Let M' be defined as follows. M' on input σ, searches for an i such that $G_p(i) \subseteq \mathrm{content}(\sigma) \subseteq W_i^\psi$; $M'(\sigma)$ then outputs the first i, if any, found in the search.

We now verify that M' *PTxtGFin*-identifies $\mathcal{L}$ using hypothesis space ψ, where G_p gives the set of good examples: let i, σ be arbitrary such that $G_p(i) \subseteq \mathrm{content}(\sigma) \subseteq W_i^\psi$. Therefore, $M'(\sigma)\downarrow = j$ for some j satisfying $G_p(j) \subseteq \mathrm{content}(\sigma) \subseteq W_j^\psi$. If $W_i^\psi \neq W_j^\psi$, then M would fail to identify one of the languages W_i^ψ and W_j^ψ from input σ. Thus, we must have $W_i^\psi = W_j^\psi$.

We continue with (b). Since, by Proposition 1 (a), $CTxtGFin \subseteq CTxtGEx$, it suffices to define a language class $\mathcal{L}$ separating $CTxtGEx$ and $CTxtGFin$.

For every $i \in N$, the infinite language $L_i = \{\langle i, x\rangle \mid x \in N\}$ belongs to $\mathcal{L}$. Let K be the diagonal halting set of the fixed acceptable programming system φ, i.e., $K = \{x \mid \varphi_x(x)\downarrow\}$. Recall that Φ denotes a fixed Blum complexity measure for the φ-system. For every $j \in K$ and every $y \leq \Phi_j(j)$, the finite language $L_{\langle j,y\rangle} = \{\langle j, x\rangle \mid x \leq y\}$ belongs to $\mathcal{L}$ as well. Since $\mathcal{L} \in TxtEx$ (cf. [20]), $\mathcal{L} \in CTxtGEx$ follows directly from Theorem 4 (a). The remaining part, i.e., $\mathcal{L} \notin CTxtGFin$, will be shown by reducing the halting problem for the φ-system to $\mathcal{L} \in CTxtGFin$. Thus, suppose M, and G_p are given such that M *CTxtGFin*-identifies $\mathcal{L}$ using hypothesis space ψ, where G_p gives the set of good examples.

Based on M and ψ, we define an algorithm A which solves the halting problem for the φ-system.

Algorithm A:

1. On input $k \in N$, determine the first $z \in N$ such that there is an $y > z$ with $\langle k, y\rangle \in W_j^\psi$, where $j = M(\sigma)$ and $\sigma = \langle k, 0\rangle, \dots, \langle k, z\rangle$.
2. Test whether or not $\Phi_k(k) \leq z$. In case it is, output '$\varphi_k(k)\downarrow$.' Otherwise, output '$\varphi_k(k)\uparrow$.'

Since M, in particular, *CTxtGFin*-identifies every infinite language L_k on the basis of finitely many good examples for it, one easily verifies that algorithm A terminates on every input k. Clearly, if A outputs '$\varphi_k(k)\downarrow$,' then $\varphi_k(k)$ is indeed defined. Thus, suppose that $k \in K$, but algorithm A terminates with '$\varphi_k(k)\uparrow$.' Let z be the corresponding index determined by A on input k, and let $L' = \{\langle k, x\rangle \mid x \leq z\}$. Since $k \in K$ and $\Phi_k(k) > z$, we have $L' \in \mathcal{L}$. Moreover, L' is finite, and, therefore, it must be the case that $W_j^\psi = L'$ for $j = M(\sigma)$ with $\sigma = \langle k, 0\rangle, \dots, \langle k, z\rangle$. However, $\langle k, y\rangle \in W_j^\psi$ for some y with $y > z$, and, thus, $W_j^\psi \neq L'$, a contradiction.

Hence, algorithm A solves the halting problem for the φ-system, a contradiction, and, thus, $\mathcal{L} \notin \mathit{CTxtGFin}$ follows. $\square$

The following theorem and its implications show the disadvantages of requiring class preserving hypothesis spaces.

Theorem 2. *There exists a recursively enumerable subset $\mathcal{L}_{\mathit{fin}}$ of FIN such that $\mathcal{L}_{\mathit{fin}} \notin \mathit{PInfGEx}$.*

Proof. For each $i = \langle u, v, w\rangle$, we will define two finite languages, L_i^1 and L_i^2 below. Let $\mathcal{L}_{\mathit{fin}} = \{L_i^1 \mid i \in N\} \cup \{L_i^2 \mid i \in N\}$. We will show that, for each $i = \langle u, v, w\rangle$, either $\mathcal{L}_{\beta_u} \neq \mathcal{L}_{\mathit{fin}}$, or no M can witness that $\mathcal{L}_{\mathit{fin}} \in \mathit{PInfGEx}$ using hypothesis space β_u, with positive good examples given by F_v and negative good examples given by F_w.

We now give the construction of L_i^1 and L_i^2.

Definition of L_i^1 and L_i^2

1. Suppose $i = \langle u, v, w\rangle$.
   Enumerate $\langle i, 0\rangle$ in L_i^1.
2. Search for j such that $\langle i, 0\rangle \in W_j^{\beta_u}$, $F_v(j)\!\downarrow$ and $F_w(j)\!\downarrow$.
3. If $F_v(j) \not\subseteq \{\langle i, 0\rangle\}$, then go to step 9.
4. Else, if $F_w(j) \cap \{\langle i, x\rangle \mid x \in N\} \neq \emptyset$, then enumerate an element of $F_w(j) \cap \{\langle i, x\rangle \mid x \in N\}$ in L_i^1 and go to step 9.
5. Otherwise enumerate $\langle i, 1\rangle$ in L_i^1 and wait until $W_j^{\beta_u}$ enumerates $\langle i, 1\rangle$.
6. Enumerate $\langle i, 0\rangle$ in L_i^2.
7. Search for k such that $\langle i, 0\rangle \in W_k^{\beta_u}$, $F_w(k)\!\downarrow$ and $F_w(k) \cap \{\langle i, x\rangle \mid x \in N\} \neq \emptyset$.
8. If and when such a k is found, enumerate an element of $F_w(k) \cap \{\langle i, x\rangle \mid x \in N\}$ in both L_i^1 and L_i^2.
9. Do not enumerate any more elements in L_i^1 and L_i^2 (i.e., definition ends).

End

It is easy to verify that L_i^1 and L_i^2 are both finite. Thus $\mathcal{L}_{\mathit{fin}} \subseteq \mathit{FIN}$. We now show that $\mathcal{L}_{\mathit{fin}} \notin \mathit{PInfGEx}$. Suppose by way of contradiction that some machine M *PInfGEx*-identifies $\mathcal{L}_{\mathit{fin}}$ using hypothesis space β_u, with positive good examples given by F_v and negative good examples given by F_w. Let $i = \langle u, v, w\rangle$.

We consider the following cases in the definition of L_i^1, L_i^2.

Case 1: Search in step 2 does not succeed.

In this case, either $\mathcal{L}_{\beta_u} \neq \mathcal{L}_{\mathit{fin}}$, or F_u, F_v do not give the good positive, good negative examples.

Case 2: If-clause in step 3 succeeds.

In this case, L_i^1 is the only language in $\mathcal{L}_{\mathit{fin}}$ which contains $\langle i, 0\rangle$. Thus $W_j^{\beta_u}$ must be equal to L_i^1 (otherwise $\mathcal{L}_{\beta_u} \neq \mathcal{L}_{\mathit{fin}}$). However, $F_v(j) \not\subseteq L_i^1$ violating the requirements of good positive examples.

Case 3: In step 4, If-condition succeeds.

In this case, L_i^1 is the only language in $\mathcal{L}_{\mathit{fin}}$ which contains $\langle i, 0\rangle$. Thus $W_j^{\beta_u}$ must be equal to L_i^1 (otherwise $\mathcal{L}_{\beta_u} \neq \mathcal{L}_{\mathit{fin}}$). However, $F_w(j) \not\subseteq \overline{L_i^1}$ violating the requirements of good negative examples.

Case 4: In step 5, procedure waits forever.

In this case, $L_i^1 = \{\langle i, 0\rangle, \langle i, 1\rangle\}$, is the only language in $\mathcal{L}_{\mathit{fin}}$ which contains $\langle i, 0\rangle$. However, $W_j^{\beta_u}$ contains $\langle i, 0\rangle$ but not $\langle i, 1\rangle$. Thus $\mathcal{L}_{\beta_u} \neq \mathcal{L}_{\mathit{fin}}$.

Case 5: Search in step 7 does not succeed.

In this case $L_i^1 = \{\langle i, 0\rangle, \langle i, 1\rangle\}$, $L_i^2 = \{\langle i, 0\rangle\}$. Suppose $\mathcal{L}_{\beta_u} = \mathcal{L}_{\mathit{fin}}$. Then $W_j^{\beta_u}$ must be equal to L_i^1. Let k be a β_u-grammar for L_i^2. Now, $F_v(j) \subseteq L_i^2 \subset L_i^1$, and $F_v(k) \subseteq L_i^2 \subset L_i^1$. Moreover, $F_w(j) \subseteq \overline{L_i^1} \subset \overline{L_i^2}$, and $F_w(k) \subseteq \overline{L_i^1} \subset \overline{L_i^2}$. Let $\tau \in \mathrm{SEG}$ be such that $\mathrm{Pos}(\tau) = L_i^2$ and $\mathrm{Neg}(\tau) = F_w(j) \cup F_w(k)$. Then M on τ must converge to a grammar for both L_i^1 and L_i^2, an impossible task.

Case 6: Search in step 7 succeeds.

In this case, either $W_k^{\beta_u}$ is not in $\mathcal{L}_{\mathit{fin}}$, or it must be one of L_i^1 and L_i^2. But $F_w(k)$ is not a subset of either $\overline{L_i^2}$ or $\overline{L_i^1}$, violating the condition for negative good examples.

From the above cases we have that $\mathcal{L}_{\mathit{fin}} \notin \mathit{PInfGEx}$ and we are done. $\square$

Applying the result above, we can show that, for learning from good examples, it is advantageous to use class comprising hypothesis spaces instead of class preserving ones. This nicely contrasts the fact that learning in the limit of indexable classes of r.e. languages is invariant with respect to the choice of the underlying hypothesis space, cf. [9].

Corollary 1. *PTxtGEx* $\subset$ *CTxtGFin.*

Proof. Since *PTxtGEx* $=$ *PTxtGFin* and *PTxtGFin* $\subseteq$ *CTxtGFin*, by definition, it suffices to separate *CTxtGFin* and *PTxtGFin*. Obviously, *FIN* $\in$

CTxtGFin and, thus, every subclass of *FIN* belongs to *CTxtGFin* as well. Consequently, the wanted separation follows immediately via Theorem 2 and Proposition 1 (d). $\square$

Our next result points out a difference to learning indexable classes of recursive languages from good examples. We show that there are indexable classes of r.e. languages which are class preservingly learnable from good examples, but which are not learnable in the limit from text. In contrast, class preserving learning of indexable classes of recursive languages from good examples is less powerful than learning in the limit from text, cf. [15].

Theorem 3. (a) $\mathit{TxtEx} \setminus \mathit{CTxtGFin} \neq \emptyset$.
(b) $\mathit{PTxtGFin} \setminus \mathit{InfEx} \neq \emptyset$.

Proof. The language class $\mathcal{L}$ used in the proof of Theorem 1 (b), separates *TxtEx* and *CTxtGFin*, and (a) follows.

Next we prove (b). We will define a numbering ψ. The diagonalizing class $\mathcal{L}$ will be formed by using the non-empty languages in $\mathcal{L}_\psi$. Let $M_0, M_1, \ldots$ be an enumeration of all *InfEx*-learning machines. For every i, $\mathcal{L}$ will contain a non-empty language $L \subseteq \{\langle i, x\rangle \mid x \in N\}$ which M_i fails to InfEx_φ-identify. (Recall that $\mathit{InfEx} = \mathit{InfEx}_\varphi$.)

Fix i. Below we give the description of $\psi_{\langle i,\cdot\rangle}$. Enumerate $\langle i, 0\rangle$ in $W^\psi_{\langle i,0\rangle}$. For a finite set S let I_S denote the canonical information sequence for S. Let $x_0 = \langle i, 0\rangle + 1$.

Stage s
1. Let $q_0, q_1 \in \{\langle i, x\rangle \mid x \in N\}$ be such that $x_s < q_0 < q_1$.
2. Let S denote the set of elements enumerated in $W^\psi_{\langle i,0\rangle}$ until now.
3. Enumerate $S \cup \{q_0\}$ into $W^\psi_{q_0}$.
   Enumerate $S \cup \{q_1\}$ into $W^\psi_{q_1}$.
4. Search for a $t > q_1$ such that, $M_i(I_{S\cup\{q_0\}}[t]) \neq M_i(I_{S\cup\{q_1\}}[t])$.
5. If and when such a t is found, let $i \in \{0, 1\}$ be such that $M_i(I_{S\cup\{q_i\}}[t]) \neq M_i(I_{S\cup\{q_i\}}[x_s])$.
   Enumerate q_i in $W^\psi_{\langle i,0\rangle}$.
   Let $W^\psi_{q_i}$ enumerate whatever $W^\psi_{\langle i,0\rangle}$ enumerates from now on. Thus $W^\psi_{q_i} = W^\psi_{\langle i,0\rangle}$.
   Let $x_{s+1} = t$.
   Go to stage $s + 1$.

End stage s

Let $\mathcal{L} = \{L \in \mathcal{L}_\psi\} - \{\emptyset\}$. First, we show that $\mathcal{L} \notin \mathit{InfEx}$. Suppose to the contrary that there is a machine M_i which InfEx_φ-identifies $\mathcal{L}$. Consider the construction of the languages $W^\psi_{\langle i,\cdot\rangle}$.

Case 1: Each stage s is entered and subsequently terminates.

Then, by construction, $W^\psi_{\langle i,0\rangle}$ is an infinite language. However, M_i on the canonical informant for $W^\psi_{\langle i,0\rangle}$, diverges.

Case 2: Some stage s is entered but never subsequently terminates.

Then, let S, q_0, q_1 be as defined in stage s. Then, $M_i(I_{S\cup\{q_0\}}) = M_i(I_{S\cup\{q_1\}})$. Thus, M_i fails to InfEx_φ-identify at least one of $W^\psi_{q_0}$ and $W^\psi_{q_1}$.

It follows from the above cases that M_i does not InfEx_φ-identify $\mathcal{L}$.

Finally, we show that $\mathcal{L}$ belongs to *PTxtGFin*. For this purpose, choose a total recursive function g with range$(g) = \{j \mid W^\psi_j \neq \emptyset\}$. Clearly, such a recursive function exists. Let $\psi'_j = \psi_{g(j)}$ for all $j \in N$. We define a machine M which *PTxtGFin*-identifies $\mathcal{L}$ with respect to the above numbering ψ' of $\mathcal{L}$, where the good examples are given by $G_p(j) = \{g(j)\}$, for every $j \in N$.

On input σ for an unknown language $L \in \mathcal{L}$, M behaves as follows: It determines the maximum x with $\langle i, x\rangle \in \mathrm{content}(\sigma)$, and outputs the least z with $g(z) = \langle i, x\rangle$.

Let y be the least index such that $L = W^\psi_{\langle i,y\rangle}$. Clearly, if $x = y$, then z is a correct guess for L. Otherwise, we know that $x > y$. Since $\langle i, x\rangle \in \mathrm{content}(\sigma) \subseteq W^\psi_{\langle i,y\rangle} = L$ and $x > y$, one easily verifies that both $W^\psi_{\langle i,x\rangle}$ and $W^\psi_{\langle i,y\rangle}$ must equal $W^\psi_{\langle i,0\rangle}$. Hence, z is a correct guess for L, and M behaves as required. $\square$

The next corollary summarizes the established relations between finite learning from good examples and learning in the limit from text.

Corollary 2. (a) *PTxtGFin # TxtEx.*
(b) *CTxtGFin # TxtEx.*

The next result shows the limitations of finite learning of indexable classes of r.e. languages from good examples. It illustrates a difference to learning of recursive functions from good examples, where finite learning from good examples turns out to be of the same learning power as *Bc*-inference, cf. [10].

Corollary 3. *CTxtGFin* $\subset$ *TxtBc.*

Proof. *CTxtGFin* $\subseteq$ *TxtBc* follows from the definition of *CTxtGFin*. Moreover, since, by definition, *TxtEx* $\subseteq$ *TxtBc* and *TxtEx* $\setminus$ *CTxtGFin* $\neq \emptyset$ (cf. Theorem 3 (a)), we are done. $\square$

Our final results in this section provide some more insight in the power of learning machines that are allowed to process the good examples in the limit.

Theorem 4. (a) *TxtEx* $\subset$ *CTxtGEx*.
(b) *CTxtGEx* $\setminus$ *TxtBc* $\neq \emptyset$.

Proof. In order to show (a) it suffices to verify that *TxtEx* $\subseteq$ *CTxtGEx*. Note that, by definition, *TxtEx* $\subseteq$ *TxtBc*, and, thus, *CTxtGEx* $\setminus$ *TxtEx* $\neq \emptyset$ follows directly from (b).

Next, we show *TxtEx* $\subseteq$ *CTxtGEx*. Suppose $\mathcal{L} \in$ *TxtEx* as witnessed by M (using hypothesis space φ). Assume, without loss of generality that M is rearrangement independent (cf. Definition 2 and Lemma 1). We also assume, without loss of generality, that $\emptyset \notin \mathcal{L}$ (we can easily modify the following proof to take care of $\emptyset$). We consider two cases.

Case 1: $N \notin \mathcal{L}$

Let ψ be defined as follows. Note that we assume an implicit coding of all members of SEQ onto N. When we use σ in a pairing function below, we assume such an encoding.

Definition of W_j^ψ,
Suppose $j = \langle \sigma, i \rangle$.
1. If content(σ) $\not\subseteq W_i^\varphi$ or $i \neq M(\sigma)$, then let $W_j^\psi = \emptyset$.
2. Dovetail steps 3 and 4 until, if ever, step 3 succeeds. If and when step 3 succeeds, go to step 5.
3. Search for a σ' extending σ such that content(σ') $\subseteq W_i^\varphi$ and $M(\sigma') \neq M(\sigma)$.
4. For $s = 0$ to ∞
    Enumerate $W_{i,s}^\varphi$ in W_j^ψ.
    EndFor
5. Enumerate N in W_j^ψ.
    (* Intuitively step 5, denotes spoiling of ψ-grammar j *)

End definition of W_j^ψ.

Let $G(j) = \text{content}(\sigma)$, where $j = \langle \sigma, i \rangle$.

Now, for each $L \in \mathcal{L}$, let σ be a locking sequence for M on L (cf. Definition 3 and Lemma 2). Then clearly, $W_{\langle \sigma, M(\sigma) \rangle}^\psi = L$. Thus, $\mathcal{L} \subseteq \mathcal{L}_\psi$.

Now consider the following M'. On input $\hat{\sigma}$, $M'(\hat{\sigma}, t)$ outputs the least $j' = \langle \sigma', i' \rangle$, if any, such that (i) $G(j') \subseteq \text{content}(\hat{\sigma}) \subseteq W^{\psi}_{j',t} \neq \emptyset$, and, (ii) in the construction above for $W^{\psi}_{j'}$, the procedure does not reach step 5 by time t. If no such j' exists then $M'(\hat{\sigma}, t)$ outputs 0.

Thus, M' on $\hat{\sigma}$, in the limit, converges to the least $j' = \langle \sigma', i' \rangle$, if any, such that (i) $G(j') \subseteq \text{content}(\hat{\sigma}) \subseteq W^{\psi}_{j'} \neq \emptyset$, and, (ii) in the construction above for $W^{\psi}_{j'}$, the procedure does not reach step 5.

We claim that M' *CTxtGEx*-identifies $\mathcal{L}$ with respect to hypothesis space ψ.

So, suppose $L \in \mathcal{L}$. Let $j' = \langle \sigma', i' \rangle$ be a ψ-grammar for L. Note that this implies σ' is a locking sequence for M on L, and $W^{\varphi}_{i'} = L$. Consider any $\hat{\sigma}$ such that $\text{content}(\sigma') \subseteq \text{content}(\hat{\sigma}) \subseteq L$. We claim that M' on $\hat{\sigma}$ converges to a ψ-grammar for L. First note that the sequence $(M'(\hat{\sigma}, t))_{t \in N}$ must converge since j' above satisfies $G(j') \subseteq \text{content}(\hat{\sigma}) \subseteq W^{\psi}_{j'} \neq \emptyset$, and procedure for $W^{\psi}_{j'}$ does not reach step 5. Now suppose M' on $\hat{\sigma}$ converges to $j'' = \langle \sigma'', i'' \rangle$. Then we have (A) $\text{content}(\sigma'') \subseteq \text{content}(\hat{\sigma}) \subseteq W^{\varphi}_{i''}$, $M(\sigma'') = i''$, σ'' is a locking sequence for M on $W^{\varphi}_{i''}$, and (B) $\text{content}(\sigma') \subseteq \text{content}(\hat{\sigma}) \subseteq W^{\varphi}_{i'}$, $M(\sigma') = i'$, σ' is a locking sequence for M on $W^{\varphi}_{i'}$. Now since M is rearrangement independent, we may conclude that, $i'' = M(\sigma'') = M(\sigma'' \cdot \sigma' \cdot \hat{\sigma}) = M(\sigma' \cdot \sigma'' \cdot \hat{\sigma}) = M(\sigma') = i'$. Hence, we have $W^{\psi}_{j''} = W^{\psi}_{j'}$.

Case 2: $N \in \mathcal{L}$.

In this case let z be such that, for all $w \geq z$, $\{x \mid x \leq w\} \notin \mathcal{L}$. Such z exists, cf. [12], Theorems I.8 and I.9.

Let ψ be defined as follows. Note that we assume an implicit coding of all members of SEQ onto N. When we use σ in a pairing function below, we assume such an encoding.

Definition of W^{ψ}_{j},

Suppose $j = \langle \sigma, i \rangle$.

1. If $\text{content}(\sigma) \not\subseteq W^{\varphi}_{i}$ or $i \neq M(\sigma)$, then let $W^{\psi}_{j} = \emptyset$.
2. Dovetail steps 3 and 4 until, if ever, step 3 succeeds. If and when step 3 succeeds, go to step 5.
3. Search for a σ' extending σ such that $\text{content}(\sigma') \subseteq W^{\varphi}_{i}$ and $M(\sigma') \neq M(\sigma)$.
4. For $s = 0$ to ∞

   Enumerate $W^{\varphi}_{i,s}$ in W^{ψ}_{j}.

   EndFor
5. Let w be the largest element that has been enumerated in W^{ψ}_{j} so far. Enumerate $\{x \mid x \leq \max(\{z, w, \max(\text{content}(\sigma))\})\}$ in W^{ψ}_{j}.

End definition of W_j^{ψ}.

The rest of the proof of (a) is now identical to Case 1.

Finally, we refer the reader to the demonstration of Theorem 8 which contains a language class witnessing $\mathit{CTxtGEx} \setminus \mathit{InfBc} \neq \emptyset$. Since, by definition, $\mathit{TxtBc} \subseteq \mathit{InfBc}$, we have $\mathit{CTxtGEx} \setminus \mathit{TxtBc} \neq \emptyset$ as well. $\square$

The next result contrasts Theorem 4. We show that *TxtEx* and *CTxtGEx* coincide if *exclusively* indexable classes of recursive languages have to be learned. Thereby, we exploit the common assumption that, as long as learning of indexable classes of recursive languages is concerned, *only* indexable classes of recursive languages are allowed to serve as hypothesis spaces, cf. [2] and [20].

Theorem 5. *Let $\mathcal{L}$ be an indexable class of recursive languages. Then, the following are equivalent:*

(i) *$\mathcal{L}$ is TxtEx-identifiable using some indexable class of recursive languages comprising $\mathcal{L}$ as hypothesis space.*
(ii) *$\mathcal{L}$ is CTxtGEx-identifiable using some indexable class of recursive languages comprising $\mathcal{L}$ as hypothesis space.*

Proof. We first prove (ii) implies (i). Let $\mathcal{L} \in \mathit{CTxtGEx}$ using hypothesis space ψ and good examples given by G_p, where it is effectively decidable (in i and x) whether $x \in W_i^{\psi}$. If $\mathcal{L}$ is finite, then trivially it belongs to *TxtEx*. So assume $\mathcal{L}$ is infinite. Let η be such that $\mathcal{L}_\eta = \mathcal{L}$, and without loss of generality assume that $W_i^{\eta} \neq W_j^{\eta}$ for all $i \neq j$. Let c be a binary total recursive function such that, for all i and all but finitely many n, $c(i,n) = \min(\{j \mid W_i^{\eta} = W_j^{\psi}\})$, i.e., $c(i,n)$ equals almost always the minimal ψ grammar for W_i^{η}. For this purpose one can let $c(i,n) = \min(\{n+1\} \cup \{k \mid \{w \leq n \mid w \in W_i^{\eta}\} = \{w \leq n \mid w \in W_k^{\psi}\}\})$.

We let $T_i = W_i^{\eta} \cap \bigcup_{s \in N} G_p(c(i,s))$. Clearly, the family $\{T_i\}_{i \in N}$ is uniformly recursively enumerable and $T_i \subseteq W_i^{\eta}$ holds for all i. Since $c(i,n)$ equals almost always the minimal ψ grammar for W_i^{η}, we also have that T_i is finite.

Claim 1. If $W_i^{\psi} \in \mathcal{L}$, $W_j^{\psi} \in \mathcal{L}$, and $W_j^{\psi} \subset W_i^{\psi}$, then $G_p(i) \not\subseteq W_j^{\psi}$.

Suppose the opposite, i.e., there are i, j such that $W_i^{\psi} \in \mathcal{L}$, $W_j^{\psi} \in \mathcal{L}$, $W_j^{\psi} \subset W_i^{\psi}$, and $G_p(i) \subseteq W_j^{\psi}$. Since $G_p(j) \subseteq W_j^{\psi}$, by definition of good examples, we immediately get $G_p(j) \subseteq W_i^{\psi}$. However, this contradicts the assumption that $\mathcal{L}$ is learnable from good examples with respect to hypothesis space ψ, since any inference machine would have to infer a hypothesis for both W_i^{ψ} and W_j^{ψ} from $G_p(i) \cup G_p(j)$, an impossible task.

Claim 2. For each i there exists an m such that $W_m^\psi = W_i^\eta$ and $G_p(m) \subseteq T_i$.

Let $m = \min(\{j \mid W_j^\psi = W_i^\eta\})$. Since, $c(i,n) = m$ for all but finitely many n, it follows that $G_p(m) \subseteq T_i$.

Putting Claims 1 and 2 together, one immediately obtains that, for all i and j, if $T_i \subseteq W_j^\eta$ then $W_j^\eta \not\subset W_i^\eta$. Hence, each T_i serves as finite "tell-tale" set for the corresponding language W_i^η. Hence, $\mathcal{L}$ is *TxtEx*-identifiable using hypothesis space $\mathcal{L}_\eta$ via Angluin's characterization of *TxtEx*, cf. [2].

The justification of the remaining part, i.e., (i) implies (ii), is similar to that of Theorem 4 (a). W_j^ψ can be defined as in the proof of Theorem 4 (a), Case 1, except: (1) we replace W_i^φ there by W_i^η, (2) in step 4, we describe a decision procedure for W_i^ψ as similar to that of W_i^η, (3) in step 5, we need to diagonalize against all languages in $\mathcal{L}_\eta$. For this we cannot enumerate N or an initial segment of N, as in the proof of Theorem 4 (a), since step 4 may have excluded some elements from W_j^ψ. However, this is not a problem since one can effectively diagonalize against all languages in $\mathcal{L}_\eta$ (despite having already decided the membership for finite number of elements in step 4), using the fact that $\mathcal{L}_\eta$ is indexed family of recursive languages. We omit the details. $\square$

5 Results on learning from good informant examples

In this section we investigate learning when the good examples may come from the target language as well as from its complement.

The following theorem shows the advantages of having good informant examples, compared to good text examples.

Theorem 6. *Let $\mathcal{L} = FIN \cup \{N\}$. Then $\mathcal{L} \in InfEx \cap PInfGFin$, but $\mathcal{L} \notin TxtBc \cup CTxtGEx$.*

Proof. Clearly, $\mathcal{L} \in InfEx$. Also, define $W_0^\psi = N$, $W_{i+1}^\psi = D_i$, $G_p(0) = G_n(0) = \emptyset$, $G_p(i+1) = D_i$, $G_n(i+1) = \{\min(N \setminus D_i)\}$. Let

$$M(\tau) = \begin{cases} 0, & \text{if } \mathrm{Neg}(\tau) = \emptyset; \\ i+1, & \text{if } \mathrm{Neg}(\tau) \neq \emptyset \ \wedge \ \mathrm{Pos}(\tau) = D_i. \end{cases}$$

It is easy to verify that M witnesses that $\mathcal{L}$ is in *PInfGFin* using hypothesis space ψ, where positive and negative good examples are given by G_p and G_n respectively.

Since $\mathcal{L}$ is superfinite, $\mathcal{L} \notin TxtBc$ (see [12]). Now suppose by way of contradiction that M using hypothesis space ψ, and positive good examples given by

G_p, shows that $\mathcal{L} \in CTxtGEx$. Let i be such that $W_i^\psi = N$. Let $X = G_p(i)$. Let j be such that $W_j^\psi = X$. Let σ be such that content$(\sigma) = X$. Now M on σ must converge to a ψ-grammar for both N and X, an impossible task. Thus, $\mathcal{L} \notin CTxtGEx$. $\square$

The next theorem shows that *Bc*-learning from informant is at least as powerful as finite learning from good informant examples in class comprising hypothesis spaces.

Proposition 2. *CInfGFin* $\subseteq$ *InfBc*.

Proof. Follows immediately from the corresponding definitions. $\square$

By means of Theorem 7 we will be able to separate the identification types *PInfGFin*, *PInfGEx*, *CInfGFin* and *CInfGEx* from one another. This is explicitly done in Corollary 4.

Theorem 7. *CInfGFin* # *PInfGEx*.

Proof. We first prove *PInfGEx* $\setminus$ *CInfGFin* $\neq \emptyset$. This is accomplished by proving a stronger result, namely that *PInfGEx* $\setminus$ *InfBc* $\neq \emptyset$. (This result will be used again in Corollary 5.) Together with Proposition 2, this yields *PInfGEx* $\setminus$ *CInfGFin* $\neq \emptyset$.

Claim 1. PInfGEx $\setminus$ *InfBc* $\neq \emptyset$.

We will define a numbering ψ, and partial computable functions G_p and G_n, from N to *FIN*. It will be the case that, if W_i^ψ is non-empty, then $G_p(i)$ and $G_n(i)$ are both defined. The diagonalizing class will be formed using the non-empty languages in $\mathcal{L}_\psi$. (We could have directly defined a numbering for the diagonalizing class. However, the current approach makes the presentation simpler).

Let g be a total recursive function such that range$(g) = \{i \mid W_i^\psi \neq \emptyset\}$. We let $\mathcal{L} = \{W_{g(0)}^\psi, W_{g(1)}^\psi, \ldots\}$. We will use $\mathcal{L}$ as the diagonalizing class. Intuitively, $\mathcal{L}$ is a suitable ordering of $\mathcal{L}_\psi - \{\emptyset\}$. The good examples for *PInfGEx*-identification, will be given by $G_p(g(\cdot))$ (positive) and $G_n(g(\cdot))$ (negative).

We now proceed with the definition of ψ, G_p and G_n. It should be noted that $G_p(x)$, $G_n(x)$ may not be defined if $W_x^\psi = \emptyset$. However, G_p and G_n will be defined on x such that $W_x^\psi \neq \emptyset$.

For each $i \in N$, we will give below the construction (effective in i) of $W_{\langle i,k\rangle}^\psi$, $k \in N$. In the construction, we will define a sequence of numbers $j_1^i < j_2^i < \ldots$.

This sequence may be finite or infinite. $W^\psi_{\langle i,0\rangle}$ will be nonempty. For $k > 0$, $W^\psi_{\langle i,k\rangle}$ will be nonempty iff j^i_k is defined. In addition, for each $i \in N$, we will have the following properties.

(A) For all $k \in N$, if $W^\psi_{\langle i,k\rangle}$ is nonempty, then $W^\psi_{\langle i,k\rangle} \cap \{\langle 0,x\rangle \mid x \in N\} = \{\langle 0,i\rangle\}$.
(B) $W^\psi_{\langle i,0\rangle} \cap \{\langle 1,x\rangle \mid x \in N\} = \emptyset$.
(C) For $k \geq 1$, if j^i_{k+1} is defined then $W^\psi_{\langle i,k\rangle} \cap \{\langle 1,x\rangle \mid x \in N\} = \{\langle 1,0\rangle\}$; otherwise $W^\psi_{\langle i,k\rangle} \cap \{\langle 1,x\rangle \mid x \in N\} = \emptyset$.
(D) For $k \geq 1$, if j^i_k is defined then $W^\psi_{\langle i,k\rangle} \cap \{\langle 2,x\rangle \mid x \in N\} = \{\langle 2,j^i_r\rangle \mid 1 \leq r \leq k\}$.
(E) $W^\psi_{\langle i,0\rangle} \cap \{\langle 2,x\rangle \mid x \in N\} = \{\langle 2,j^i_k\rangle \mid 1 \leq k \ \wedge\ j^i_{k+1} \text{ is defined }\}$.
(F) $(\{W^\psi_{\langle i,k\rangle} \mid k \in N\} \setminus \{\emptyset\}) \not\subseteq \mathit{InfBc}_\varphi(M_i)$.

We will have $G_p(\langle i,0\rangle) = \{\langle 0,i\rangle\}$, and $G_n(\langle i,0\rangle) = \{\langle 1,0\rangle\}$. In addition, for $k \geq 1$, if j^i_k is defined, then $G_p(\langle i,k\rangle) = \{\langle 0,i\rangle, \langle 2,j^i_k\rangle\}$, and $G_n(\langle i,k\rangle) = \emptyset$.

It is easy to verify, from properties (A) to (E) that, $\mathcal{L} \in \mathit{PInfGEx}$. (For this consider a machine which, on input τ, first finds an i such that $\langle 0,i\rangle \in \mathrm{Pos}(\tau)$, and the maximum k, if any, such that $\langle 2,j^i_k\rangle \in \mathrm{Pos}(\tau)$. If no such k exists then the input language must be $W^\psi_{\langle i,0\rangle}$, so assume that such a k exists. Note that this restricts the input language to be either $W^\psi_{\langle i,0\rangle}$ or $W^\psi_{\langle i,k\rangle}$. Now, the input language is $W^\psi_{\langle i,0\rangle}$ iff $\langle 1,0\rangle \in \mathrm{Neg}(\tau)$ and $W^\psi_{\langle i,k\rangle}$ enumerates $\langle 1,0\rangle$.)

In addition (F) will imply that $\mathcal{L} \notin \mathit{InfBc}$ (since $\mathit{InfBc}_\varphi = \mathit{InfBc}$).

We now give the construction of $W^\psi_{\langle i,k\rangle}$, for $k \in N$, and, for $W^\psi_{\langle i,k\rangle} \neq \emptyset$, the definition of $G_p(\langle i,k\rangle)$ and $G_n(\langle i,k\rangle)$. We will define $W^\psi_{\langle i,\cdot\rangle}$ (and corresponding G_p and G_n) in stages $s = 0, 1, \ldots$

Definition of $W^\psi_{\langle i,k\rangle}$, for $k \in N$,
Stage 0:
(* Intuitively, P_s denotes the set of elements we have decided to keep in $W^\psi_{\langle i,0\rangle}$ before stage s. N_s denotes the set of elements we have decided to keep out of $W^\psi_{\langle i,0\rangle}$ before stage s. x_s denotes $\max(P_s \cup N_s)$. *)
Let $P_1 = \{\langle 0,i\rangle\}$.
Let $x_1 = \max(\{\langle 1,0\rangle, \langle 0,i\rangle\})$.
Let $N_1 = \{x \mid x \leq x_1 \ \wedge\ x \neq \langle 0,i\rangle\}$.
Enumerate P_1 in $W^\psi_{\langle i,0\rangle}$.
Let $G_p(\langle i,0\rangle) = \{\langle 0,i\rangle\}$.
Let $G_n(\langle i,0\rangle) = \{\langle 1,0\rangle\}$.
Let j^i_1 be such that $\langle 2,j^i_1\rangle > x_1$.
Go to stage 1.

Stage $s \geq 1$:
1. Let $Z_{i,s} = P_s \cup \{\langle 2, j^i_s \rangle\} \cup \{\langle 3, x \rangle \mid \langle 3, x \rangle > x_s\}$.
2. Enumerate $Z_{i,s}$ in $W^{\psi}_{\langle i,s \rangle}$.
3. Let $G_p(\langle i, s \rangle) = \{\langle 0, i \rangle, \langle 2, j^i_s \rangle\}$.
   Let $G_n(\langle i, s \rangle) = \emptyset$.
4. Let I be the canonical informant for $Z_{i,s}$.
5. Search for $n > j^i_s$ and $y > n$, such that $y \in W^{\varphi}_{M_i(I[n])}$.
6. If and when such n, y are found,
   Enumerate $\langle 1, 0 \rangle$ in $W^{\psi}_{\langle i,s \rangle}$.
   Let $x_{s+1} = y$.
   Let j^i_{s+1} be such that $\langle 2, j^i_{s+1} \rangle > x_{s+1}$.
   Let $P_{s+1} = \{z \mid z < y \ \wedge \ z \in Z_{i,s}\}$.
   Let $N_{s+1} = \{y\} \cup \{z \mid z < y \ \wedge \ z \notin Z_{i,s}\}$.
   Enumerate P_{s+1} in $W^{\psi}_{\langle i,0 \rangle}$.
   Go to stage $s + 1$.

End stage s
End of definition of $W^{\psi}_{\langle i,k \rangle}$, for $k \in N$.

Fix i. Consider the construction for the definition of $W^{\psi}_{\langle i,k \rangle}$, $k \in N$. It is easy to verify that the construction satisfies properties (A) to (E). We now show that ψ satisfies property (F). We consider two cases.

Case 1: There are infinitely many stages.

In this case, let I be a canonical informant for $W^{\psi}_{\langle i,0 \rangle}$. Now, $M_i(I[n])$ is not a φ-grammar for $W^{\psi}_{\langle i,0 \rangle}$ for infinitely many n, (due to success of step 5 and diagonalization at step 6). Thus M_i does not InfBc_{φ} identify $W^{\psi}_{\langle i,0 \rangle}$.

Case 2: Stage s starts but does not finish.

In this case let I be the canonical informant for $W^{\psi}_{\langle i,s \rangle}$. Since step 5 did not succeed, we have that, for all but finitely many n, $M_i(I[n])$ is a φ-grammar for a finite language. Since $W^{\psi}_{\langle i,k \rangle}$ is infinite, we have that M_i does not InfBc_{φ} identify $W^{\psi}_{\langle i,k \rangle}$.

From the above two cases we have that (F) is satisfied.

This proves Claim 1.

Claim 2. $\mathit{CInfGFin} \setminus \mathit{PInfGEx} \neq \emptyset$.

Theorem 2 gave a class $\mathcal{L}_{\mathit{fin}} \subseteq \mathit{FIN}$ which does not belong to $\mathit{PInfGEx}$. Since every subclass of FIN belongs to $\mathit{CInfGFin}$, the claim follows. $\square$

Corollary 4. (a) *PInfGFin* $\subset$ *PInfGEx*.
(b) *CInfGFin* $\subset$ *CInfGEx*.
(c) *PInfGFin* $\subset$ *CInfGFin*.
(d) *PInfGEx* $\subset$ *CInfGEx*.

Proof. Immediate from the corresponding definitions and the previous theorem. $\square$

The next two results suggest that there are major differences in what is learnable from good examples – even only considering text examples – and what is *Bc*-learnable from informant. We believe that the main reason for this is the fact that the learning process is "divided" when learning with good examples: first, the good examples are computed from a description of the language in question and secondly, the strategy is required to learn only if it receives (a superset of) these examples. When learning *Bc*-style from informant we require that the strategy learns from *every* informant. So the whole learning problem has to be solved by the strategy, without help from selected examples. On the other hand, when learning from informant the strategy may get information on every word it desires, whereas, when learning from good examples, the strategy only has access to the finite set it receives.

Corollary 5. *PInfGEx* $\#$ *InfBc*.

Proof. For *PInfGEx* $\setminus$ *InfBc* $\neq \emptyset$ consider Claim 1 in the proof of Theorem 7.

InfBc $\setminus$ *PInfGEx* $\neq \emptyset$, is again witnessed by class $\mathcal{L}_{\mathit{fin}}$ (cf. Theorem 2). $\square$

Theorem 8. *CTxtGEx* $\#$ *InfBc*.

Proof. First we will prove *CTxtGEx* $\setminus$ *InfBc* $\neq \emptyset$. Let ψ, g, G_p be as defined in the proof of Claim 1 in Theorem 7.

Define η as follows.

$W^{\eta}_{2x} = W^{\psi}_{g(x)} \setminus \{\langle 1,0\rangle\}$. $G'_p(2x) = G_p(g(x))$.

$$W^{\eta}_{2x+1} = \begin{cases} W^{\psi}_{g(x)} \cup \{\langle 1,0\rangle\}, & \text{if } g(x) \notin \{\langle i,0\rangle \mid i \in N\}; \\ \emptyset, & \text{otherwise.} \end{cases}$$

$$G'_p(2x+1) = \begin{cases} G_p(g(x)) \cup \{\langle 1,0\rangle\}, & \text{if } g(x) \notin \{\langle i,0\rangle \mid i \in N\}; \\ \emptyset, & \text{otherwise.} \end{cases}$$

It is easy to verify that $\mathcal{L} = \{W^{\psi}_{g(0)}, W^{\psi}_{g(1)}, \ldots\}$ can be *CTxtGEx* identified using hypothesis space η, and good examples given by G'_p. To verify this, consider a machine which, on input σ, first finds an i such that $\langle 0,i\rangle \in \text{content}(\sigma)$, and the maximum k, if any, such that $\langle 2, j_k^i\rangle \in \text{content}(\sigma)$. If no such k exists then the input language must be $W^\psi_{\langle i,0\rangle}$, so assume that such a k exists. Note that this restricts the input language to be either $W^\psi_{\langle i,0\rangle}$ or $W^\psi_{\langle i,k\rangle}$. Now, if $\langle 1, 0\rangle \in \text{content}(\sigma)$ then the input language is $W^\psi_{\langle i,k\rangle}$ (η grammar for which is $2x+1$, where $g(x) = \langle i, k\rangle$). If $\langle 1, 0\rangle \notin \text{content}(\sigma)$ and $W^\psi_{\langle i,k\rangle}$ enumerates $\langle 1, 0\rangle$ then the input language is $W^\psi_{\langle i,0\rangle}$ (η grammar for which is $2x$, where $g(x) = \langle i, 0\rangle$). If $\langle 1, 0\rangle \notin \text{content}(\sigma)$ and $W^\psi_{\langle i,k\rangle}$ does not enumerate $\langle 1, 0\rangle$ then the input language is $W^\psi_{\langle i,k\rangle}$ (η grammar for which is $2x$, where $g(x) = \langle i, k\rangle$).

$\mathcal{L} \notin \mathit{InfBc}$ was shown in Theorem 7.

Finally, $\mathit{InfBc} \setminus \mathit{CTxtGEx} \neq \emptyset$ follows from Theorem 6. □

Corollary 6. *CTxtGEx # PInfGFin.*

Proof. For $\mathit{CTxtGEx} \setminus \mathit{PInfGFin} \neq \emptyset$ first note that $\mathit{PInfGFin} \subseteq \mathit{CInfGFin}$ holds by definition. The assertion now follows immediately, since $\mathit{CInfGFin} \subseteq \mathit{InfBc}$, by Theorem 2, and Theorem 8 gives a class of languages in $\mathit{CTxtGEx} \setminus \mathit{InfBc}$.

$\mathit{PInfGFin} \setminus \mathit{CTxtGEx} \neq \emptyset$ follows from Theorem 6. □

Theorem 9. $\mathit{CTxtGFin} \cap \mathit{TxtEx} \setminus \mathit{PInfGEx} \neq \emptyset$.

Proof. Note that every subset of *FIN* belongs to $\mathit{CTxtGFin} \cap \mathit{TxtEx}$. Theorem now follows from Theorem 2. □

Finally, we present some more insight into the strength of class comprising learning from good examples.

We start with finite identification from good informant examples.

Theorem 10. $\mathit{TxtEx} \subseteq \mathit{CInfGFin}$.

Proof. Suppose M TxtEx_φ-identifies $\mathcal{L}$. Without loss of generality assume that M is rearrangement independent. For $\tau \in \text{SEG}$, let $H(\tau)$ denote a $\sigma \in \text{SEQ}$ such that $\text{content}(\sigma) = \text{Pos}(\tau)$ and $|\sigma| = 2 * \text{card}(\text{Pos}(\tau) \cup \text{Neg}(\tau))$.

For $\tau \in \text{SEG}$, we say that witness(τ, m) iff the following four conditions hold: (a) $\text{Pos}(\tau) \subseteq W^\varphi_m$, (b) $\text{Neg}(\tau) \subseteq \overline{W^\varphi_m}$, (c) $M(H(\tau)) = m$, and (d) $H(\tau)$ is a locking sequence for M on W^φ_m.

Note that, if witness(τ, m), then for all $\tau' \supseteq \tau$ consistent with W^φ_m, witness(τ', m).

For $\tau \in \mathrm{SEG}$, let

Possible $= \{\langle \tau, m\rangle \mid \mathrm{Pos}(\tau) \subseteq W_m^{\varphi} \ \wedge \ M(H(\tau)) = m\}$, and

Spoiled $= \{\langle \tau, m\rangle \in \text{Possible} \mid \neg\text{witness}(\tau, m)\}$.

Note that Possible and Spoiled are r.e. sets.

Without loss of generality assume that Possible is an infinite set. We consider two cases based on whether $N \in \mathcal{L}$ or not.

Case 1: $N \notin \mathcal{L}$.

Let g be a 1–1 recursive function such that range(g) = Possible. Let ψ be defined as follows.

$$W_i^{\psi} = \begin{cases} W_m^{\varphi}, & \text{if } g(i) = \langle \tau, m\rangle \text{ and } \langle \tau, m\rangle \notin \text{Spoiled}; \\ \mathrm{N}, & \text{otherwise.} \end{cases}$$

It is easy to verify that ψ is a computable numbering. Moreover, for $g(i) = \langle \tau, m\rangle$, if witness$(\tau, m)$ is true, then $W_i^{\psi} = W_m^{\varphi}$; otherwise $W_i^{\psi} = N$ (and thus not in $\mathcal{L}$).

For i such that $g(i) = \langle \tau, m\rangle$, let $G_p(i) = \mathrm{Pos}(\tau)$, and $G_n(i) = \mathrm{Neg}(\tau)$.

Define M' as follows. On input $\hat{\tau}$, M' outputs i such that $g(i) = \langle \hat{\tau}, M(H(\hat{\tau}))\rangle$. Consider any $W_i^{\psi} \in \mathcal{L}$. Suppose $g(i) = \langle \tau', m'\rangle$. Note that witness$(\tau', m')$ holds. Suppose $\hat{\tau}$ is such that $\mathrm{Pos}(\tau') \subseteq \mathrm{Pos}(\hat{\tau}) \subseteq W_i^{\psi} = W_{m'}^{\varphi}$, and $\mathrm{Neg}(\tau') \subseteq \mathrm{Neg}(\hat{\tau}) \subseteq \overline{W_i^{\psi}} = \overline{W_{m'}^{\varphi}}$. Thus witness$(\hat{\tau}, m')$ holds. Hence $m' = M(H(\hat{\tau}))$, and i such that $g(i) = \langle \hat{\tau}, M(H(\hat{\tau}))\rangle$ is a ψ grammar for W_i^{ψ}. Thus M' *CInfGFin*-identifies $\mathcal{L}$, using hypothesis space ψ and good positive and negative examples given by G_p and G_n, respectively.

Case 2: $N \in \mathcal{L}$.

Recall that, for all $i \in N$, $X_i = \{x \mid x \leq i\}$ and $INIT = \{X_i \mid i \in N\}$. Applying Lemma 2, one easily verifies that $INIT \cap \mathcal{L}$ is finite, since $N \in \mathcal{L}$. Hence, there is a $n \in N$ such that, for all $i \geq n$, $X_i \notin \mathcal{L}$.

Let g be a 1–1 recursive function such that range(g) = Possible. Let ψ be defined as follows.

$$W_i^{\psi} = \begin{cases} W_m^{\varphi}, & \text{if } g(i) = \langle \tau, m\rangle \text{ and } \langle \tau, m\rangle \notin \text{Spoiled}; \\ X_j, & \text{otherwise, for some } j \geq n. \end{cases}$$

Note that, for the second clause above, W_i^ψ can just enumerate some initial segment of N, once it discovers that $\langle \tau, m\rangle \in$ Spoiled. Thus, ψ is a computable numbering. Moreover, for $g(i) = \langle \tau, m\rangle$, if witness$(\tau, m)$ is true, then $W_i^\psi = W_m^\varphi$; otherwise $W_i^\psi = X_j$, for some $j \geq n$ and thus not in $\mathcal{L}$.

For i such that $g(i) = \langle \tau, m\rangle$, let $G_p(i) = \mathrm{Pos}(\tau)$, and $G_n(i) = \mathrm{Neg}(\tau)$.

Define M' as follows. On input $\hat{\tau}$, M' outputs i such that $g(i) = \langle \hat{\tau}, M(H(\hat{\tau}))\rangle$. Now as in Case 1, one can verify that M' *CInfGFin*-identifies $\mathcal{L}$, using hypothesis space ψ and good positive and negative examples given by G_p and G_n, respectively. $\square$

Interestingly, even class comprising finite learning (from good text examples) can outperform *TxtEx* inference provided that the target class contains only infinite languages.

Theorem 11. *Suppose $\mathcal{L} \in TxtEx$ consists of only infinite languages. Then, $\mathcal{L} \in CTxtGFin$.*

Proof. Suppose M $TxtEx_\varphi$-identifies $\mathcal{L}$. Without loss of generality assume that M is rearrangement independent. For $\sigma \in$ SEQ, let $H(\sigma)$ denote a $\sigma' \in$ SEQ such that content(σ') = content(σ) and $|\sigma'| = 2 * \mathrm{card}(\mathrm{content}(\sigma))$.

For $\sigma \in$ SEQ, we say that witness(σ, m) iff the following three conditions hold: (a) content$(\sigma) \subseteq W_m^\varphi$, (b) $M(H(\sigma)) = m$, and (c) $H(\sigma)$ is a locking sequence for M on W_m^φ.

The rest of the proof can now be done essentially in the same manner as in the proof of Theorem 10. We omit the details. $\square$

Finally, we show the power of limit learning from good informant examples in class comprising hypothesis spaces.

Before proving the next theorem, we define the following predicates and point to some of their properties.

m is a *minimal grammar* iff $m = \min(\{j \mid W_j^\varphi = W_m^\varphi\})$.

We say consistent(τ, j), iff $\mathrm{Pos}(\tau) \subseteq W_j^\varphi$, and $\mathrm{Neg}(\tau) \subseteq \overline{W_j^\varphi}$.

We say that bndincons(τ, j), iff $\mathrm{Pos}(\tau) \not\subseteq W_j^\varphi$, or $\mathrm{Neg}(\tau) \not\subseteq \overline{W_{j,|\tau|}^\varphi}$.

Note that if bndincons(τ, j), then consistent(τ, j) is false; however, the converse is not always true. Intuitively, bndincons(τ, j) just puts some computability constraints on inconsistency.

The following proposition is easy to prove.

Proposition 3. (a) *Suppose m is a minimal grammar. Then, there exists a τ such that* $\mathrm{consistent}(\tau, m)$ *and* $(\forall j < m)[\mathrm{bndincons}(\tau, m)]$.

(b) *Suppose τ, m meeting* $\mathrm{consistent}(\tau, m)$ *and* $(\forall j < m)[\mathrm{bndincons}(\tau, m)]$. *Then,* $(\forall \tau' \supseteq \tau)[\mathrm{consistent}(\tau', m) \Rightarrow (\forall j < m)[\mathrm{bndincons}(\tau', j)]]$.

(c) *Suppose τ, m meeting* $\mathrm{consistent}(\tau, m)$ *and* $(\forall j < m)[\mathrm{bndincons}(\tau, m)]$. *Then,* $m = \min(\{j \mid \mathrm{consistent}(\tau, j)\})$.

(d) *Suppose m is not a minimal grammar. Then, there exists no τ such that* $\mathrm{consistent}(\tau, m)$ *and* $(\forall j < m)[\mathrm{bndincons}(\tau, m)]$.

Intuitively, part (a) says that if m is a minimal grammar for some language, then there exists a 'witness' to this fact. Part (b) says that if τ is a 'witness' to m being a minimal grammar, then all consistent extensions of τ are also a witness. Part (c) gives a mechanism to find the minimal grammar using a witness.

Let Possible $= \{\langle \tau, m\rangle \mid \mathrm{Pos}(\tau) \subseteq W_m^\varphi \ \wedge \ \mathrm{Neg}(\tau) \subseteq \overline{W_{m,|\tau|}^\varphi}\}$.

Intuitively, Possible consists of $\langle \tau, m\rangle$ such that it is possible for τ to be a witness for m to be a minimal grammar. Now we define Spoiled as follows.

Let Spoiled $= \{\langle \tau, m\rangle \mid \langle \tau, m\rangle \in$ Possible $\wedge \ \neg[\mathrm{consistent}(\tau, m) \ \wedge \ (\forall j < m)[\mathrm{bndincons}(\tau, j)]]\}$.

Intuitively, Spoiled consists of those (τ, m) in Possible, such that τ is not a witness to m being minimal grammar. Clearly, Possible is r.e. Moreover, after a bit of reflection one verifies that Spoiled is r.e. as well.

We are now ready to prove the final theorem.

Theorem 12. *$\mathit{TxtBc} \subseteq \mathit{CInfGEx}$.*

Proof. Note that, for any $\mathcal{L} \in \mathit{TxtBc}$, either $N \notin \mathcal{L}$, or $\mathit{INIT} \cap \mathcal{L}$ is finite. Theorem now follows using Lemmas 3 and 4. □

Lemma 3. *Let $\mathcal{L} = \mathcal{E} \setminus \{N\}$. Then $\mathcal{L} \in \mathit{CInfGEx}$.*

Proof. Note that Possible is an infinite r.e. set. Let g be a 1–1, total recursive function such that $\mathrm{range}(g) =$ Possible. ψ is defined as follows:

$$W_i^\psi = \begin{cases} W_m^\varphi, & \text{if } g(i) = \langle \tau, m\rangle \text{ and } \langle \tau, m\rangle \notin \text{Spoiled;} \\ \text{N}, & \text{otherwise.} \end{cases}$$

It is easy to verify that ψ is a computable numbering. Moreover, for $g(i) = \langle \tau, m\rangle$, if τ is a witness to m being a minimal grammar, then $W_i^\psi = W_m^\varphi$;

otherwise $W_i^\psi = N$ (and thus not in $\mathcal{L}$).

For i such that $g(i) = \langle \tau, m \rangle$, let $G_p(i) = \text{Pos}(\tau)$, and $G_n(i) = \text{Neg}(\tau)$.

Define M as follows. On input $\hat{\tau}$, M converges, in the limit, to i, such that $g(i) = \langle \hat{\tau}, m \rangle$, where $m = \min(\{j \mid \text{consistent}(\hat{\tau}, j)\})$. It is now easy to verify, using Proposition 3, that M *CInfGEx*-identifies $\mathcal{L}$, using hypothesis space ψ and good positive and negative examples given by G_p and G_n, respectively. $\square$

Lemma 4. *Suppose $n \in N$. Let $X_i = \{x \mid x < i\}$. Let $\mathcal{L} = \mathcal{E} \setminus \{X_i \mid i \geq n\}$. Then $\mathcal{L} \in$ CInfGEx.*

Proof. This proof is similar to that of Lemma 3. Let g be a 1–1, total recursive function such that range(g) = Possible. ψ is defined as follows:

$$W_i^\psi = \begin{cases} W_m^\varphi, & \text{if } g(i) = \langle \tau, m \rangle \text{ and } \langle \tau, m \rangle \notin \text{Spoiled;} \\ X_j, & \text{otherwise, for some } j \geq n. \end{cases}$$

Note that, for the second clause above, W_i^ψ can just enumerate some initial segment of N once it discovers that $\langle \tau, m \rangle \in$ Spoiled. Thus, ψ is a computable numbering. Moreover, for $g(i) = \langle \tau, m \rangle$, if τ is a witness to m being a minimal grammar, then W_i^ψ is W_m^φ; otherwise $W_i^\psi = X_j$, for some $j \geq n$ (and thus not in $\mathcal{L}$).

For i such that $g(i) = \langle \tau, m \rangle$, let $G_p(i) = \text{Pos}(\tau)$, and $G_n(i) = \text{Neg}(\tau)$.

Define M as follows. On input $\hat{\tau}$, M converges, in the limit, to i such that $g(i) = \langle \hat{\tau}, m \rangle$, where $m = \min(\{j \mid \text{consistent}(\hat{\tau}, j)\})$. It is now easy to verify, using Proposition 3, that M *CInfGEx*-identifies $\mathcal{L}$, using hypothesis space ψ and good positive and negative examples given by G_p and G_n, respectively. $\square$

6 Concluding Remarks

As experience shows, in learning from examples there are important examples and less important ones. In order to solve the learning problem it often suffices to see the important examples rather than as much examples as possible. The approach of learning from good examples formalizes this intuitive idea.

In this paper we studied learning from good examples for indexed families of recursively enumerable languages. We considered the relationship between different criteria based on (i) whether the good examples contain only elements of the target language (so-called text examples) or the good examples contain both elements and non-elements of the target language (so-called informant examples), on (ii) whether the good examples are computed with respect to some class preserving or some class comprising hypothesis space, and on (iii) whether the learner has, when fed any superset of the good examples, to learn finitely or in the limit. Moreover, we related the resulting models of learning from good examples to the standard learning models in the context of Gold-style language learning.

We showed that the learning power of finite and limit learning from good text examples coincides in the class preserving case. On the other hand, in the class comprising case, limit learning from good text examples is more powerful than finite inference from good text examples. When learning from good informant examples is considered, limit learning is more powerful than finite inference, both in the class preserving and in the class comprising case. These results provide an answer to an open question posed by Lange, Nessel and Wiehagen in a similar study about learning indexed families of reursive languages from good examples (cf. [15]).

It turned out that learning from good examples may sometimes outperform learning in the limit and even behaviourally correct inference from text and informant, respectively. This additional power mainly comes from the following sources: the knowledge of the language to be learnt when computing the good examples to it and, in a sense simultaneously, the careful choice of an appropriate hypothesis space.

Furthermore, the results obtained allows to clarify the relation between finite learning from good examples and the standard models of finite identification from text (TxtFin) and finite identification from informant (InfFin), cf. Gold [12]. It is easy to see that $\mathit{TxtFin} \subset \mathit{PTxtGFin}$. Furthermore, it can be shown that $\mathit{InfFin} \subseteq \mathit{CTxtGFin}$. Since $\mathit{InfFin} \subseteq \mathit{TxtEx}$, as a corollary to Theorem 3 (b) we get, $\mathit{PTxtGFin} \setminus \mathit{InfFin} \neq \emptyset$. On the other hand, an easy modification of the proof of Theorem 2 can be used to verify that $\mathit{InfFin} \setminus \mathit{PInfGEx} \neq \emptyset$.

Finally, let us point to the relevant open problems. The most important questions are whether or not $\mathit{TxtBc} \subseteq \mathit{CTxtGEx}$ and $\mathit{InfBc} \subseteq \mathit{CInfGEx}$, respectively, hold. Besides that, we don't know whether $\mathit{CInfGFin} \subset \mathit{InfBc}$ holds.

7 Acknowledgements

We thank Arun Sharma and Rolf Wiehagen for helpful discussions. We also thank the anonymous referees for ALT'97 and this journal, whose comments helped to improve the paper.

References

[1] D. Angluin, Finding patterns common to a set of strings, *Journal of Computer and System Sciences* **21** (1980) 46–62.

[2] D. Angluin, Inductive inference of formal languages from positive data, *Information and Control* **45** (1980) 117–135.

[3] G. Baliga, J. Case and S. Jain, Language learning with some negative information, *Journal of Computer and System Sciences* **51** (1995) 273–285.

[4] M. Blum, A machine independent theory of the complexity of recursive functions, *Journal of the ACM* **14** (1967) 322–336.

[5] L. Blum and M. Blum, Toward a mathematical theory of inductive inference, *Information and Control* **28** (1975) 125–155.

[6] J. Case, S. Jain, S. Lange and T. Zeugmann, Incremental concept learning for bounded data mining, DOI Technical Report, DOI-TR-136, Kyushu University, Japan, 1997.

[7] J. Case and C. Lynes, Machine inductive inference and language identification, in: M. Nielsen and E.M. Schmidt, eds., *Proc. 9th Intern. Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science, Vol. 140 (Springer-Verlag, Berlin, 1982) 107–115.

[8] J. Case and C. Smith, Comparison of identification criteria for machine inductive inference, *Theoretical Computer Science* **25** (1983) 193–220.

[9] D. de Jongh and M. Kanazawa, Angluin's theorem for indexed families of r.e. sets and applications, in: A. Blum and M. Kearns, eds., *Proc. 9th Annual Conference on Computational Learning Theory* (ACM Press, New York, 1996) 193–203.

[10] R. Freivalds, E.B. Kinber and R. Wiehagen, On the power of inductive inference from good examples, *Theoretical Computer Science* **110** (1993) 131–144.

[11] M.A. Fulk, Prudence and other conditions on formal language learning, *Information and Computation* **85** (1990) 1–11.

[12] E.M. Gold, Language identification in the limit, *Information and Control* **10** (1967) 447–474.

[13] S.A. Goldman and H.D. Mathias, Teaching a smarter learner, *Journal of Computer and System Sciences* **52** (1996) 255–267.

[14] S. Jain and A. Sharma, Elementary formal systems, intrinsic complexity and procrastination, *Information and Computation* **132** (1997) 65–84.

[15] S. Lange, J. Nessel and R. Wiehagen, Language learning from good examples, in: S. Arikawa and K.P. Jantke, eds., *Proc. 5th Intern. Workshop on Algorithmic Learning Theory*, Lecture Notes in Artificial Intelligence, Vol. 872 (Springer-Verlag, Berlin, 1994) 423–437.

[16] T. Motoki, Inductive inference from all positive and some negative data, *Information Processing Letters* **39** (1991) 177–182.

[17] H. Rogers, *Theory of Recursive Functions and Effective Computability* (McGraw-Hill, New York, 1967; reprinted by MIT Press, Cambridge, Massachusetts, 1987).

[18] G. Schäfer-Richter, Über Eingabeabhängigkeit und Komplexität von Inferenzstrategien, Dissertation, Rheinisch Westfälische Technische Hochschule Aachen, 1984.

[19] A. Shinohara and S. Miyano, Teachability in computational learning, *New Generation Computing* **8** (1991) 337–347.

[20] T. Zeugmann and S. Lange, A guided tour across the boundaries of learning recursive languages, in: K.P. Jantke and S. Lange, eds., *Algorithmic Learning for Knowledge-Based Systems*, Lecture Notes in Artificial Intelligence, Vol. 961 (Springer-Verlag, Berlin, 1995) 190–258.

[21] T. Zeugmann, Lange and Wiehagen's pattern language learning algorithm: An average-case analysis with respect to its total learning time, RIFIS Technical Report, RIFIS-TR-CS-111, RIFIS, Kyushu University, Japan, 1995.