

Language Identification in the Limit  
An Overview of Gold's paper

Marek Doniec      Viksit Gaur

September 13th 2005

# 1 Identification in the Limit

In this paper, we dwell upon the concept of learnability, with relation to languages. One of the fundamental issues in Artificial Intelligence is the ability of a machine to learn a language, in such a way as not to have been programmed explicitly. A language like English for example, can't be written down explicitly in terms of rules and can't therefore be taught to a machine (or another similar entity) - the latter will have to learn its rules from implicit information which we need to provide it with. To achieve this end, E. Mark Gold proposed a precise model for the notion of "being able to speak a language" in order to theoretically investigate how it would be done artificially.

Before we start with the above process, a definition of Learnability needs to be developed in order to build upon later. We assume that time is quantized, and has a finite starting time. At each time, the learner receives a unit of information, and makes a guess as to the identity of the unknown language on the basis of the information received so far. This process continues forever, and the class of languages will be considered learnable with respect to the specified method of information presentation if there is an algorithm having the following property:

Given any language of the class, there is some finite time after which the guesses will all be the same and they will be correct. We consider a finite set of alphabet,  $A$ , and let  $\Sigma A$  be the set of all finite strings from  $A$ 's elements.  $L$  is a language which is a subset of  $A$ . We can now define a learnability model.

Time is considered quantized, and is a finite quantity -  $t = 1, 2, \dots$ . At each time  $t$ , a unit of information related to the unknown language is presented to the learner as a set of training sequences  $i_1, i_2, \dots$ . At each time  $t$ , the learner makes a guess  $g_t$  about the language  $L$  based on information obtained through time  $t$ .  $g_t = G(i_1, i_2, \dots, i_t)$ .  $L$  is said to be identified in the limit if all guesses after some finite time are the same.

## 2 Information Presentation through text

For the learner to be able to actually *learn* a language, it needs to be presented with information about which it can make its guesses. There are a number of ways of doing this, but we will consider doing so using text. A sequence of strings  $x_1, x_2, \dots$  for the language  $L$  such that every string occurs at least once in it, is called a text for  $L$ . At time  $t$ ,  $x_t$  is presented to the learner. There might be 3 ways of presenting text:

1. (1) Arbitrary text -  $x_t$  is any function of  $t$
2. (2) Recursive text -  $x_t$  is an recursive function of  $t$
3. (3) Primitive Recursive text -  $x_t$  may be any primitive recursive function of  $t$

Two types of functions for naming relations will be dealt with here - testers and generators. A Tester for  $L$  is a turing machine, which defines functions from strings to natural numbers, value 1 for strings in  $L$ , and 0 for those not. On the other hand, a Generator is a turing machine which defines functions from natural numbers to strings, such that range of this function is exactly  $L$ .

An example information function for the english language could look like this:

1.  $I(0) = (\text{"You are an algorithm."})$
2.  $I(1) = (\text{"Ein Apfel ist grun."})$
3.  $I(2) = (\text{"This is an apple"})$
4. ...

### 3 Algorithm for learning

Now that we have defined the ways of presenting information, let us develop an algorithm which would allow the learner to learn from the information thus obtained. We first define an input function  $I(t)$ , and at each time step  $t$ , it produces an output  $O(t)$ . The latter is the guess which is made by the learner about the language for which the information is being presented to it. The input may be presented in the form of ("xy"), which may be represented as  $a$ . The learner then checks if  $\forall \varepsilon > 0$  a certain  $n_0$  exists  $\forall n > n_0, f(n) - a < \varepsilon$ . If the condition is satisfied, then  $a$  is the correct answer, and the next piece of information is processed.

1. Define input function  $I(t)$  and an output function  $O(t)$ .
2. Read input  $S(t) = I(t)$
3.  $\lim O(t) = a$  is the correct answer
4. ...

## 4 Abstract model of identification

We may term two language learnability models as equivalent if, with respect to either model, exactly the same classes of language are identifiable in the limit. On the other hand, two naming relations are termed equivalent if the two language learnability models obtained for every method of information presentation are equivalent. In the same vein, two methods of information presentation are said to be equivalent if every naming relation yields two equivalent language learnability models.

We consider a class  $\Omega$  of objects, out of which the learner will be presented. There are three terms we need to define when presenting such a model.

- 1 Information presentation At each time  $t$ , learner receives unit of information  $i_t$  belongs to set  $I$ . The method consists of specifying for each  $\omega$  belongs to  $\Omega$  which sequences of information,  $i_1, i_2, \dots$  are allowable. Set of allowable sequences is labelled  $I^\infty(\omega)$ .
- 2 Naming relation - consists of a set  $N$  of names and a function  $f$  which assigns an object to each name,  $f : N \rightarrow \Omega$ .
- 3 The identification problem is now to determine whether there is indeed a rule which the learner can use to accomplish : For any object  $\omega$  belongs to  $\Omega$ , and for any information sequence from  $I^\infty(\omega)$ , on the basis of that information sequence the rule will yield a name  $n$  of  $\omega$ , that is,  $f(n) = \omega$ .

## 5 Information presentation by informant

In contrast to presentation by text where the learning algorithm is only fed with positive examples presentation by informant supplies as well positive as negative learning examples. In presentation by informant the learning algorithm receives a 2-tupel  $(S, b)$  at each time step as input.  $S$  represents an arbitrary string over  $\Sigma^*$  whereas  $b$  is 0 when  $S$  is not part of the language presented and 1 when  $S$  is in the language presented. Thus the input can be seen as a function over time:

$$I : N_0 \rightarrow (\Sigma^*, \{0, 1\}).$$

As an example an informant function for the English language could look like this:

1.  $I(0) = (\text{"You are an algorithm."}, 1)$

2.  $I(1) = (\text{"An apple is green."}, 1)$
3.  $I(2) = (\text{"Das ist ein Apfel."}, 0)$
4. ...

As we will see later a learning algorithm can learn a context free language in the limit when provided with an informant whereas it can not even learn a language from a class of regular languages when only provided with text.

## 6 Identification by enumeration

In identification by enumeration the learner first need a function

$$L : N \rightarrow \mathcal{L}$$

that assigns a language from the selected class of languages  $\mathcal{L}$  to each natural number. The range of  $L$  has to be exactly  $\mathcal{L}$ .  $L$  is indeed an enumeration of the languages from the class  $\mathcal{L}$ . The learner is again presented with an input  $I(t)$  at each time step  $t$  and produces an output  $O(t)$ . The output is the current guess as to which language is being represented by the input received so far. In learning by enumeration the learner starts out by comparing the first input with the first value of  $L$ . In the case of an informant the input might be  $(\text{"xy"}, 1)$ . In this case the learner will check if "xy" is contained in  $L(0)$ . If so it outputs  $L(0)$ , if not it will check the input with the second Language, i.e.  $L(1)$ . Once a language has been discarded it is never considered again (except for if the enumeration is ambiguous) for the counter for the languages is never reset. Instead when the last output was  $L(n)$  then the next input is only going to be compared to  $L(n)$  and eventually successive languages. Formally the algorithm looks as follows:

1. choose  $L : N_0 \rightarrow \mathcal{L}$ , set  $n = 0$
2. read input  $(S(t), b(t)) = I(t)$
3. while  $\neg((S(t) \in L(n) \wedge b(t)) \vee (S(t) \notin L(n) \wedge \neg b(t)))$  do  $n++$
4. output  $L(n)$
5. go to step 2. (infinite loop)

Indeed this algorithm will always identify the correct language in the limit if the searched class of languages is context-free and the input is presented by an informant.

*Proof:* Let us assume the language to be learned is  $L(m)$ ,  $m$  being a natural number, and that  $L(m) \neq L(k)$  for all  $k < m$  (i.e.  $m$  being the minimal number in the enumeration for the language chosen). Suppose now that the learning algorithm will stop in the limit at language  $L(p)$ ,  $p < m$ . Since the informant will provide information on every string of  $\Sigma^*$  this means that  $L(p)$  accepts all strings contained in  $L(m)$  and rejects all those that are not contained in  $L(m)$ . Thus  $L(p) = L(m)$  which is contrary to our minimal choice of  $m$ . Thus the algorithm will not stop before reaching  $L(m)$ . Once it has reached  $L(m)$  it is evident that it will continue to provide  $L(m)$  as an output since all its input will coincide with  $L(m)$  by definition of the problem.

## 7 Information Presentation by text and Regular Languages

We will prove here that the class of regular languages  $\mathcal{R}$  cannot be learned in the limit when given Information presentation by text. To show this we assume we have a learner  $M$  and give a counterexample. The language to be learned is  $A = a^n | n \in \mathbb{N}$  and we will also consider the languages  $A(k) = \sum_{i=1}^k a^i \subset A$ . These languages are clearly regular.

At the beginning we present only the one symbol  $a \in A(1)$  to the learner. We assume that the learner is able to identify languages in the limit. This means that the output  $O(t)$  of the learner has a limit  $\lim O(t) = A(1)$  and thus there is a moment  $t_1$  from which on the learner will give only  $A(1)$  as answer. This is a reasonable assumption given the case that we only presented the symbol  $a$  as input. But when we switch to giving strings from  $A(2)$  after time  $t_1$  the learner will have to change his output. Again assuming that he is able to learn in the limit he will at some point  $t_2$  continue to only give  $A(2)$  as output. We proceed in this way, i.e. every time the learner is presented only with symbols from  $A(k)$  he will eventually start outputting  $A(k)$  at time  $t_k$ . Each time this happens we enrich our inputs to be taken from the language  $A(k+1)$ . As we can easily see there is no end to this procedure since  $A$  is infinite. Still the strings presented to the learner are always from finite languages and he will make wrong guesses at least at the time  $t_k$  when he guesses  $A(k)$ . Thus our learner will never learn  $A$  in the limit and our assumption that such a learner exists is false.