Category Translation: Learning to understand information on the Internet

Mike Perkowitz* **Oren Etzioni**
Department of Computer Science and Engineering, FR-35
University of Washington, Seattle, WA 98195
{map, etzioni}@cs.washington.edu
(206) 616-1845 Fax: (206) 543-2969

Abstract

This paper investigates the problem of automatically learning declarative models of information sources available on the Internet. We report on *ILA*, a domain-independent program that learns the *meaning* of external information by explaining it in terms of internal categories. In our experiments, *ILA* starts with knowledge of local faculty members, and is able to learn models of the Internet service `whois` and of the personnel directories available at Berkeley, Brown, Caltech, Cornell, Rice, Rutgers, and UCI, averaging fewer than 40 queries per information source. *ILA*'s hypothesis language is compositions of first-order predicates, and its bias is compactly encoded as a determination. We analyze *ILA*'s sample complexity both within the Valiant model, and using a probabilistic model specifically tailored to *ILA*.

1 Introduction and Motivation

The number and diversity of information sources on the Internet is increasing rapidly. A number of tools such as Gopher, WAIS, and Web Crawlers are available to help people search for the information they need. However, these tools are unable to interpret the results of their searches and are unable to use multiple information sources in concert. A number of more sophisticated AI systems have emerged, including SIMS [Knoblock *et al.*, 1994], the Information Manifold at AT&T [Kirk *et al.*, 1995], and the Internet softbot [Etzioni and Weld, 1994]. However, each of these AI systems requires sophisticated models of the different information sources it is able to access. As a result, there are two barriers that prevent AI approaches from keeping up with the explosion of information sources on the Internet. First, effort has to be devoted to hand coding a model of each source. Second, sources unknown to the programmers associated with each AI system cannot be modeled. To enable the AI approaches to scale with the growth of the Internet, we explore the problem of automatically learning models of information sources. This learning problem raises four fundamental questions:

- **Discovery:** how does the learner find new and unknown information sources? (*e.g.*, a web page representing the Brown phone directory has recently come on-line.)
- **Protocol:** what are the mechanics of accessing an information source and parsing the response into tokens? (the Brown directory is searched by sending a string such as "`kaelbling`" to the Brown server, and receiving back a string.)
- **Semantics:** how does the learner come to understand the information available at the source? (the tokens describe Kaelbling's e-mail address, phone number, department, etc.)
- **Quality:** What is the accuracy, reliability, and scope of the information source? (the directory contains accurate information about people at Brown, not elsewhere.)

Satisfactory answers to all of these questions would enable us to construct an autonomous Internet learning agent able to discover and use information resources effectively. As a first step, this paper investigates the question of learning semantics.

Our learning method is based on the following idea, due to St. Augustine [Wittgenstein, 1958]. Consider how you might learn the Latin term *uxor* by example. Suppose I tell you "George Washington's *uxor* was Martha." You might reason that, because "Martha" was the name of Washington's wife, perhaps "uxor" means "wife". If, however, you knew that Washington also had a sister named "Martha", you might wait until you saw another example, perhaps asking "Who was Jefferson's uxor?" This method of learning relies on three key assumptions. First, you are familiar with George Washington. Second, you have a concept corresponding to *uxor*, e.g. *wife*. Third, you are willing to establish a general *correspondence* between your concept *wife* and the concept *uxor* based on the example given. As we show below, this leap of faith can be viewed as an inductive *bias* and formalized as a determination. We refer to this determination

*We thank Dayne Freitag, Craig Knoblock, and Tom Mitchell for inspiring discussions that contributed to our problem formulation. We thank Dymitr Mozdyniewicz for his assistance with experiments, and Anchana Kullavanijaya for her transcription help. This research was funded in part by Office of Naval Research grant 92-J-1946 and by National Science Foundation grant IRI-9357772. Mike Perkowitz is supported, in part, by an NSF graduate fellowship.

as the *correspondence heuristic*.

This paper is organized as follows. In the next section, we define the category translation problem. We then present the Internet Learning Agent (*ILA*) and explain its learning method. Following that, we describe experiments in a simple Internet domain and present results. We then describe a probabilistic model of *ILA* that bounds its sample complexity. We conclude with a critique and directions for future work.

2 The Category Translation Problem

Below, we present both a concrete example and a general formulation of the category translation problem. Suppose, for example, the agent queries the University of Washington staff directory with the token **Etzioni** and gets the response **Oren Etzioni 685-3035 FR-35**. Based on its knowledge about Etzioni, we'd like the agent to come up with the general model of the directory shown at the bottom of Table 1 under "Response".

To solve the category translation problem, an agent has to *generalize* from the observed queries and responses to a logical expression made up of model attributes. To generalize effectively, the agent has to employ some inductive bias. *ILA* has to assume that the information source ($\mathcal{IS}$) is (roughly) invariant when responding to queries about different individuals. It would be difficult to learn a model of an $\mathcal{IS}$ that responded with random facts about each individual queried — the phone number for one person, the birth date of a second, and the social security number for a third.

As with St. Augustine's method, *ILA* requires an overlap between its world model and the information returned by the source. First, *ILA* and the $\mathcal{IS}$ must share some individuals. If the agent is only familiar with UW faculty, and the $\mathcal{IS}$ contains information about current Brown undergraduates, learning will prove problematic. Second, *ILA* and the $\mathcal{IS}$ must share some categories. If *ILA* is familiar with people and the $\mathcal{IS}$ is a catalog of stellar constellations, there is no basis for learning. Of course, *ILA* is not limited to people and their phone numbers. The same method could be used to learn about movie databases, product catalogs, etc.

We formalize the learning problem as follows. Let I be an $\mathcal{IS}$ that contains k fields. We represent I with the functions $I_1(o) \ldots I_k(o)$, where each $I_j(o)$ returns the jth field in the output when I is queried with object o. For the example in Table 1, we would say that $\texttt{staffdir}_4(P_1)$ = FR-35. In standard concept learning terms, *ILA* is trying to learn or approximate the functions $I_1..I_k$, based on examples of the form $I_1(o) = T_1, I_2(o) = T_2, \ldots, I_k(o) = T_k$.

The correspondence heuristic has two components. First, a match of tokens between the agent's model and the $\mathcal{IS}$ indicates a match of categories. For example, when *ILA* queries with Etzioni and sees **685-3035** in the third field, it explains this response as Etzioni's `phone-number`. Second, a category match on one individual suggests a match on all individuals — the meaning of the $\mathcal{IS}$ field is consistent across individuals. Continuing the previous example, *ILA* would now hypothesize that the third field was `phone-number`, for every individual in the $\mathcal{IS}$. In order to find a hypothesis for which this generalization holds, we must test hypotheses on multiple observations and choose the best hypothesis.

More formally, let $S(I, o)$ be true if o is an object in I. If two objects o_1 and o_2 are both in an $\mathcal{IS}$, then we believe that, for some attribute M^*, if we query the $\mathcal{IS}$ with o_1 and the jth field is $M^*(o_1)$ then, when we query with o_2, the jth field will be $M^*(o_2)$, where M^* is an arbitrary composition of model attributes.

We can state the correspondence heuristic as the following determination:

$$\exists(M^*)\forall(o_1, o_2)[S(I, o_1) \wedge S(I, o_2) \wedge (I_j(o_1) = M^*(o_1)) \rightarrow (I_j(o_2) = M^*(o_2))]$$

The equation $I_j(o_1) = M^*(o_1)$ encodes the assumption that the jth element of I's response can be described as a logical expression M^*, which is composed of model attributes. The implication encodes the assumption that, for some M^*, the equality observed for one individual holds for all others for which S is true. The existential quantifier suggests that we need to search for the appropriate M^*. The next section describes *ILA*'s search strategy, and its use of multiple queries to track down M^*.

3 Algorithm

In essence, *ILA* queries the $\mathcal{IS}$ with known objects and searches for token correspondences between its model and information returned by the $\mathcal{IS}$. *ILA* generates hypotheses based on these correspondences and ranks them with respect to how often they accord with observations. To flesh out this algorithm sketch we have to answer several questions:

1. Which object in its internal model should *ILA* query the $\mathcal{IS}$ with?
2. What is the appropriate mapping from that internal object to a query string? (in the case of a person, the $\mathcal{IS}$ might be queried with the person's last name, full name, social security number, etc.)
3. What are possible explanations (denoted by M^* in our determination) for each token in the response?
4. How should *ILA* evaluate competing explanations?

We consider each question in turn.

Initially, *ILA* may use any object in its internal model to query the $\mathcal{IS}$: a person, a tech report, a movie, etc. To constrain the set of possible queries, *ILA* utilizes any information it has about the $\mathcal{IS}$. This knowledge could be expressed as a constraint on the type of object that can be in the $\mathcal{IS}$ or as an attribute that is only true of objects likely to be found in the $\mathcal{IS}$. For example, if it knows that the $\mathcal{IS}$ is a personnel directory, *ILA* will not query the $\mathcal{IS}$ with movie titles.

In addition, *ILA* employs several heuristics to reduce the number of queries necessary to converge to a satisfactory model of the $\mathcal{IS}$. Most important, *ILA* attempts to *discriminate* between two competing hypotheses by choosing an object for which the hypotheses make different predictions (cf. [Rajamoney, 1993]). For example, if *ILA* has seen the record **Oren Etzioni 685-3035 FR-35**, it will consider both `lastname` and `userid` as

Given:

1. Incomplete internal world model:
   - Objects (*e.g.*, persons: P_1, P_2, ..., P_n).
   - Attributes of the objects (*e.g.*, `lastname`(P_1)=Etzioni, `department`(P_1)=CS, `userid`(P_1)=Etzioni, `mail-stop`(CS)=FR-35,...).
2. An external information source ($\mathcal{IS}$) that responds to queries.
   e.g. `staffdir`, the University of Washington personnel directory:
   Query: **Etzioni**
   Response: **Oren Etzioni 685-3035 FR-35**

Determine: A set of logical expressions composed of model attributes which explains the observed query/response pairs. *e.g.*
Query: `lastname`(person)
Response: first field = `firstname`(person)
second field = `lastname`(person)
third field = `phone-number`(person)
fourth field = `mail-stop`(`department`(person))

Note that the explanation may involve compositions of model attributes, as in the case of `mail-stop`, and that we seek to minimize the number of queries made.

Table 1: The Category Translation Problem.

hypotheses for the second field because Etzioni's userid is his last name. To discriminate between the two hypotheses, *ILA* will attempt to query with someone whose userid is different from her last name. If no discriminating query is possible, *ILA* will attempt to find an object that has the potential to disconfirm the leading hypothesis. In the above example, if *ILA* hypothesizes that the third field is `phone-number`, it will choose a person whose phone number is known over a person whose phone number is not. Finally, if neither a discriminating nor a disconfirming query is possible, *ILA* will query with an object about which it has much information, in order to increase the likelihood of recognizing some token in the response. Discriminating queries typically accelerate *ILA*'s ability to converge on a satisfactory hypothesis; in the case of `staffdir`, for example, when *ILA* does not make use of discriminating queries, it requires 50% more queries to converge on the same hypotheses.

Once a particular object is chosen, *ILA* has to decide which query string to actually send to the $\mathcal{IS}$. Initially, *ILA* will try all known facts about the object as possible query strings, attempting to learn the appropriate query string for the $\mathcal{IS}$. The learning mechanism used is, in essence, the same as the one described below for learning to interpret the $\mathcal{IS}$'s output.

Once *ILA* obtains a response from the external $\mathcal{IS}$, it attempts to *explain* each token in the response. An explanation is a chain of one or more model attributes composed into a relation between the object and the token seen. For example, in *ILA*'s model, people are associated with departments and departments associated with mail-stops. The relation between a person and her mail-stop, then, is a composition of `department` and `mail-stop` — the mail-stop of P is `mail-stop(department(`P`))`.

We employ a variant of *relational pathfinding* [Richards and Mooney, 1992] to discover a relation between the query object and each response token. Richards and Mooney's pathfinding technique performs a bidirectional breadth-first search in which constants are nodes in the graph and attributes on constants are edges between nodes. We use a *fuzzy matcher* to compare tokens from the $\mathcal{IS}$ to constants in *ILA*'s model. Our current matching function ignores punctuation and spacing and can allow substring matches (*e.g.*, the learner can recognize "(206) 616-1845" and "616.1845" as being the same token). Consequently, our pathfinding is unidirectional, proceeding from the query object to fuzzily-matched tokens.[1]

Suppose the agent starts with the model shown in Table 1. It queries the $\mathcal{IS}$ with the last name of object P_1 and gets the response **Oren Etzioni 685-3035 FR-35**. It will now try to explain each response token in turn. For example, in order to explain "FR-35", *ILA* will start with P_1 and spread out one step through the model, *e.g.*, to `CS` and `Etzioni`. Since neither matches the target token "FR-35", *ILA* will continue spreading out from the current frontier, retaining the path to each current node (*e.g.*, the attribute path from P_1 to `CS` is `department`(x)). From `CS`, *ILA* will get to `FR-35`. Thus, the path to `FR-35` is `mail-stop(department`(x)`)`. Since `FR-35` matches the target, this path will be returned as an explanation.

Next, *ILA* evaluates the hypothesized explanation. With respect to a particular query, a hypothesis may be *explanatory* (it predicted the output actually seen), *inconsistent* (it predicted something else), or *consistent* (it made no prediction). Thus, a hypothesis h partitions the set of responses to queries into Explanatory, Inconsistent, and Consistent subsets. We denote the number of elements in each subset by the ordered triple $(E(h), I(h), C(h))$. We refer to the triple as the *score* of the hypothesis h. Since a hypothesis is only generated when it successfully explains some response, we know

[1] To perform bidirectional pathfinding, we would have to find the set of matching tokens in *ILA*'s model, an expensive computation due to the size of the model.

that, for any h, $E(h) \geq 1$.

The predictions of a new hypothesis are compared against old responses to compute the hypothesis's score. Overall, *ILA* compares each hypothesis against each response exactly once, so learning time is linear in both the number of responses and the number of hypotheses. To determine whether one hypothesis is better than another, *ILA* compares the number of inconsistent predictions by the two hypotheses. If the number of inconsistent predictions is equal, *ILA* compares the number of explanatory predictions. More formally, we say that the hypothesis h is better than the hypothesis h' if and only if:

$$\text{Better}(h, h') \equiv [I(h) < I(h')] \vee [I(h) = I(h') \wedge E(h) > E(h')]$$

That is, *ILA* chooses the hypothesis with the lowest I score and uses E scores to break ties. This is a good policy when incomplete information is more common than incorrect information because the I score (how often the hypothesis was inconsistent) is a better indicator of the accuracy of the hypothesis. An inconsistency arises either when the hypothesis is inaccurate or when the information is incorrect. Because incorrect information is rare in our domain, a bad (high) I score indicates an inaccurate hypothesis. A hypothesis may fail to explain an observation due to incomplete information, because if we lack the relevant fact, the hypothesis makes no prediction. Since incomplete information is relatively common, a bad (low) E score does not necessarily indicate low accuracy of the hypothesis. Therefore, $I(h)$ is a better indicator of the quality of h than $E(h)$. Suppose *ILA* knows everybody's last name but only a few people's userid. When trying to learn the `userid` field, the `userid` hypothesis will explain only a few observations (because it will make very few predictions) but will never be inconsistent. In contrast, `lastname` will explain many observations but will be inconsistent on others. Because *ILA* prefers low I scores, it makes the right choice.

ILA terminates the learning process when one of two conditions occur. One, it has run out of objects with which to query the $\mathcal{IS}$. Two, its leading hypothesis is "significantly" better than its other hypotheses. The difference in I scores that is deemed significant is controlled by a parameter to *ILA*. Although *ILA*'s running time is exponential in the depth of the relational pathfinding search for an explanatory hypothesis, the maximal search depth is typically set to a small constant, keeping *ILA* fast. As mentioned earlier, the running time is linear in the number of queries made and the number of explanatory hypotheses generated. In fact, as the experiments in Table 2 show, *ILA*'s running time is dominated by Internet transmission time.

4 Experimental Results

In this section, we report on preliminary experiments designed to test whether our approach is viable in a real-world domain. We find that *ILA* is able to learn models of simple information sources on the Internet.

To factor out the issues of protocol (which we do not address in this paper), *ILA* is provided with an interface that standardizes the interaction with the information sources used. Each interface takes query strings as input and outputs a list of tokens which *ILA* attempts to understand. In our first experiment, *ILA* is provided with complete and correct models of faculty in the University of Washington's (UW) Computer Science Department, and is asked to learn a model of `staffdir`, the UW personnel directory. The first line of Table 2 shows the results of this experiment. We see that in 16 queries *ILA* was able to learn a correct model of `staffdir`. *ILA* spent 19 seconds interacting with `staffdir` and 24 CPU seconds searching for, and evaluating, hypotheses.

Below, we show the final scores of the leading hypotheses for interpreting the second field of `staffdir`'s output:

$\texttt{staffdir}_2(x) = \texttt{lastname}(x)$	Expl: 11	Incons: 0
$\texttt{staffdir}_2(x) = \texttt{userid}(x)$	Expl: 8	Incons: 3

We see that for eight people, both the `lastname` and `userid` hypotheses correctly explained the second field in the output of `staffdir`. However, for three people, the `userid` hypothesis failed, leading *ILA* to consider `lastname` to be the correct hypothesis.

A general problem that arises in relying on token correspondence to infer type correspondence is the occurrence of *puns*. A pun occurs when matching tokens are not actually instances of the same concept. A hypothesis arising from a pun amounts to finding an incorrect composition of model attributes — one that is not true for all x and y. A pun is an instance of the general problem of an incorrect hypothesis resulting in a correct classification of a training example. One type of pun is entirely coincidental; a person's area code turns out to be the same as his office number. A spurious hypothesis resulting from a coincidental pun is easy to reject — it is unlikely to prove explanatory for more than a single example. However, we also encounter *semi-regular* puns — where there is a correlation between the two concepts which gives rise to the pun. As pointed out above, many people's userids are also their last names. Semi-regular puns may require many more queries to converge on the correct hypothesis, because both the correct and spurious hypotheses will make accurate predictions in many cases. Discriminating queries aim to address this problem by finding examples where the correct and spurious hypotheses make different predictions.

No matter how regular a pun, there must eventually be a difference between the correct hypothesis and the competitor.[2] How hard it is to choose the best hypothesis is a function of the learner's knowledge and the regularity of the pun. The system faces a tradeoff: it must balance time spent learning against confidence in the result. If *ILA* collects more examples, it can be more confident in the correctness of its conclusions. The learner can never be fully certain it is not the victim of a particularly regular pun, but it will have some estimate of the likelihood that it has the right solution. We provide a quantitative analysis of this intuition in the next section.

One possible criticism of *ILA* is that it relies on an

[2] If there is no difference in extension between the two hypotheses, then they are equally good solutions to the learning problem.

	Fields							Queries			Time	
	1	2	3	4	5	6	7	queries	hits	discr	Internet	CPU
staffdir	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	16	16	5	0:19	0:24
whois	$\surd$	$\surd$			$\surd$	$\surd$	$\surd$	50	22	19	26:09	3:04
Berkeley	$\surd$	$\surd$			-	-	$\surd$	24	5	7	6:07	0:54
Brown	$\surd$	$\surd$			$\surd$	-	-	69	6	0	11:06	2:58
Caltech	$\surd$	$\surd$				$\surd$	$\surd$	22	11	4	4:02	0:17
Cornell	$\surd$	$\surd$			$\surd$	$\surd$	$\surd$	41	13	0	2:17:57	12:23
Rice	$\surd$	$\surd$			-	$\surd$	x	36	2	0	6:53	1:15
Rutgers	$\surd$	$\surd$						36	8	2	5:29	0:57
UCI	$\surd$	$\surd$			-	-	$\surd$	34	13	2	12:02	6:40

Table 2: Learning to understand information sources by bootstrapping from `staffdir` and `whois`. We report number of queries, number of responses, number of discriminating queries. Time spent querying (on the Internet) is in real seconds. Local processing time is in CPU seconds. Fields: 1=firstname, 2=lastname, 3=title, 4=dept, 5=phone, 6=email, 7=userid. Fields tagged $\surd$ were learned; "-" could have been learned but weren't, and those left blank could not have been learned, because the field was not reported by the $\mathcal{IS}$. The field marked x was mistaken for field 1 due to the paucity of hits at Rice.

overlap between the individuals in its model and individuals in the $\mathcal{IS}$ it is trying to learn. However, *ILA* benefits from the presence of *spanning* information sources on the Internet. A *spanning* information source is one that contains objects from a wide variety of information sources. For example, the Internet service called `whois` reports information on individuals from a wide range of sites on the Internet and will, for example, return people from a particular school when queried with that school's name. *ILA* relies on its knowledge of local individuals to learn a model of `whois`, and then leverages its model of `whois` to learn models of a wide variety of remote sites on the Internet. Instead of relying on individuals from its model, *ILA* will query `whois` for new individuals at the target site. For example, when trying to learn the Brown directory, *ILA* will query `whois` with "Brown" to get information about people at Brown and use its learned model of `whois` to interpret the output. Our second experiment demonstrates this process (Table 2). The second line of the table shows the results of learning `whois` from knowledge of local people. Given the learned model of `whois`, we report on *ILA*'s performance in learning models of the personnel directories available at Berkeley, Brown, Cal-Tech, Cornell, Rice, Rutgers, and UCI. As the results in Table 2 demonstrate, *ILA* is able to learn fairly accurate models of these information sources averaging fewer than 40 queries per source, most taking less than 15 minutes each, where the bulk of that time is spent in Internet communication. The processing time for *ILA* is less than three CPU minutes in most cases. Slow network connections contributed to the unusually large Internet times for `whois` and Cornell. The size of the Cornell directory and the generality of its matching contributed to the large processing time for that directory.

5 Theoretical Analysis

We would like to understand how the accuracy of (and confidence in) *ILA*'s hypotheses scale with the number of queries it makes, the size of its hypothesis space, the correctness of its information, and so on. We consider both Valiant's PAC model and an alternative probabilistic model of *ILA*. *ILA* is not a standard concept-learning program — *ILA* is learning a *function* from the query it issues to the response of the $\mathcal{IS}$. Furthermore, the oracle used by *ILA* is a generalized membership oracle. However, learnability results in function learning theory are specific to classes of functions learned, such as polynomial, or real-valued [Auer *et al.*, 1995]. Similarly, although specific concept classes have been shown to be learnable under the membership oracle [Angluin, 1988], we are not aware of any sample complexity results that apply directly to *ILA*. For this reason, we chose to make a number of strong simplifying assumptions and analyze *ILA* under the PAC model.

The PAC model provides a convenient framework for analyzing *ILA*'s behavior, under the simplifying assumption that queries to the $\mathcal{IS}$ are random. $|H|$ is the size of the hypothesis space explored by *ILA*. We posit a probability distribution P over queries to the information source I. The error E of a hypothesis h is the probabilistic weight of the queries on which the hypothesis h disagrees with the actual behavior of I:

$$E(h, I) = \sum_{o \in I \quad s.t.\ h(o) \neq I(o)} P(o)$$

For simplicity, we consider an $\mathcal{IS}$ with a single output field, where $I(o)$ is the token returned by the $\mathcal{IS}$, and $h(o)$ is the value predicted by h. Haussler [Haussler, 1988] derives a lower bound on the number of examples necessary for PAC learning. If h is any hypothesis that agrees with at least n queries from I, where $n \geq \frac{1}{\varepsilon} \ln(\frac{|H|}{\delta})$, then we have the following: $P(E(h, I) \geq \varepsilon) \leq 1 - \delta$.

To apply this bound to *ILA*, we have to assume that the information in I and in *ILA*'s model is error free, that a perfect model of I can be found in *ILA*'s hypothesis space, and that token matching is working perfectly. We can model the violation of these assumptions as random classification noise and use the bound due to [Angluin and Laird, 1988]: $n \geq \frac{2}{\varepsilon^2(1-2\eta_b)^2} \ln(\frac{2|H|}{\delta})$, where η_b is an upper bound on the frequency of noisy classifications, and the learner chooses the hypothesis that is correct

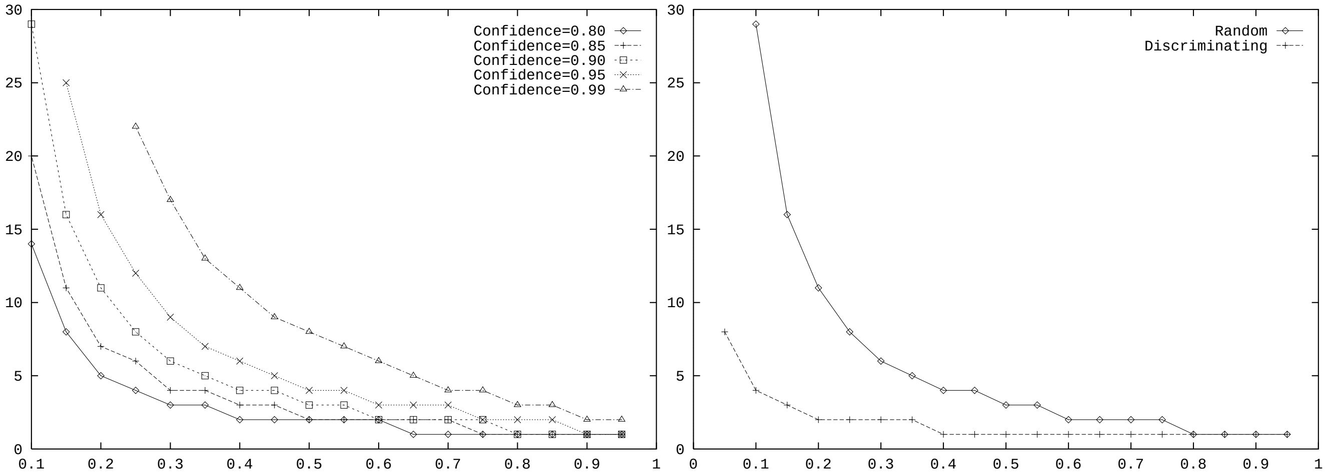

Figure 1: (a) The number of queries required to be confident of choosing the better hypothesis as a function of γ ($\gamma = A_g - A_b$), at varying confidence levels. (b) Number of queries required to be 90% sure of choosing the better hypothesis, as a function of γ. The lower line is using discriminating queries; the upper line is without. In both graphs A_g is fixed at 0.95, and we set an upper bound of 30 on the number of queries. Note that in both figures the curves meet but do not cross.

most often. Unfortunately, the number of queries suggested by the bound is very large. For example, if we set $\delta = 0.1, \eta_b = 0.05$, and $|H| = 2$, we see that the number of queries necessary to guarantee $\varepsilon \leq 0.05$ exceeds 3,000. How does *ILA* get away with relying on far fewer queries in our experiments?

The PAC bounds are overly conservative for two reasons. First, the bounds presuppose randomly distributed queries, whereas *ILA* makes discriminating queries which enhance its ability to converge quickly. Second, the PAC bounds are based on a worst case analysis where there is at least one hypothesis whose accuracy is just less than $1-\varepsilon$, where the *accuracy* of a hypothesis h with respect to an $\mathcal{IS}$ I is $1 - E(h, I)$. The learning algorithm has to observe enough examples to rule it out and find a hypothesis whose accuracy is at least $1 - \varepsilon$ with probability $1 - \delta$. Typically, puns are not as pernicious as this worst-case analysis would suggest. In the case of the lastname-userid pun in `staffdir`, for example, the best hypothesis (`lastname`) has accuracy of 1.0 and the next best hypothesis (`userid`) has accuracy 0.37. Only 2 queries are required to have 90% confidence that *ILA* will prefer the better hypothesis. Below, we derive a general bound that yields this observation.

Instead of asking how likely a learner is to find a hypothesis whose accuracy is at least $1 - \varepsilon$, we ask how likely the learner is to pick the best hypothesis in its hypothesis space based on its observations. We can no longer guarantee that the hypothesis chosen has accuracy $1 - \varepsilon$. However, we can guarantee that the learner is likely to perform as well as possible, given its hypothesis space. Furthermore, when the best hypothesis is *much* better than the other hypotheses in the space, relatively few queries are required to converge on the best hypothesis with high confidence. Below, we formalize this intuition using elementary probability theory.

If a hypothesis has probability p of making a correct prediction, then the probability that the hypothesis will yield x correct predictions on n independent queries is the binomial distribution: $p^x\,(1-p)^{n-x} \binom{n}{x}$. For simplicity and brevity, we analyze the case where the space contains exactly two hypotheses, and the learner chooses the hypothesis that makes fewest incorrect predictions.[3] Thus, if we consider the hypothesis g, with accuracy A_g, and the hypothesis b, with accuracy A_b, then the probability that after n queries *ILA* will prefer the hypothesis g is at least:

$$\sum_{x=1}^{n} A_g^x\,(1-A_g)^{n-x} \binom{n}{x} \left(\sum_{y=0}^{x-1} A_b^y\,(1-A_b)^{n-y} \binom{n}{y}\right) \quad (1)$$

This formula sums the probability, under the binomial distribution, of the different states where g results in more correct predictions than b.[4] We can use the above formula to determine the number of queries n necessary to guarantee, with confidence at least c, that *ILA* will choose the hypothesis with accuracy at least A_g over the hypothesis with accuracy at most A_b. To do so, we choose the desired confidence level c, set the above formula equal to c, and solve for n.

Figure 1(a) shows the number of queries required to choose hypothesis g over hypothesis b with a given confidence, as a function of γ, the difference in accuracy between g and b. The accuracy of g is fixed at 0.95. Curves are shown at various confidence levels. For example, when $\gamma = 0.50$, *ILA* must perform 3 queries in order to have 90% confidence of choosing g.

[3] In our model, if the two hypotheses have equal scores, they have equal probability of being chosen.

[4] We assume independence between the errors of g and b. Also, under the worst-case assumption that $n-1$ hypotheses all have accuracy A_b, a similar formula can be derived for a hypothesis space of size n [Etzioni and Perkowitz, 1995].

As γ shrinks, *ILA* requires more queries to achieve high confidence. Note though that when γ is very small, discriminating between g and b is less important. Furthermore, discriminating queries enable *ILA* to converge on the high-accuracy hypothesis quickly even for small γ. To demonstrate the advantage of discriminating queries over random queries, Figure 1(b) shows the number of random and discriminating queries necessary to achieve at least 90% confidence that the learner will prefer hypothesis g over hypothesis b, again as a function of γ. The accuracy of g is fixed at 0.95. The random-queries curve is derived from Equation 1. Due to lack of space, we omit the derivation and assumptions underlying the discriminating-queries curve, but see [Etzioni and Perkowitz, 1995]. When $A_g = 0.95$ and $\gamma = 0.50$, *ILA* requires only a single discriminating query to have 90% confidence that it has found the better hypothesis. In essence, it is so unlikely for g to be wrong *and* for b to be right on the same query that one query is sufficient for *ILA* to choose a hypothesis with high confidence.

In short, if there is a large gap between the best hypothesis and its closest competitor, or we are able to perform discriminating queries, our probabilistic model shows that relatively few queries are necessary to have high confidence in choosing the best hypothesis. The model helps to explain how, using a small number of queries, *ILA* was able to learn accurate models of information sources in the experiments summarized in Table 2.

6 Critique and Future Work

Our contributions include: formulating the Category Translation problem, developing *ILA*'s algorithm, and formalizing its bias as a determination. We have tested *ILA* experimentally on a simple Internet domain, and analyzed its sample complexity within the PAC framework and using a specialized probabilistic model.

We have identified several problems that *ILA* does not yet address. *Category mismatch* occurs when *ILA* fails to find categories corresponding to those of the external information source [Wiederhold, 1992]. For example, the $\mathcal{IS}$ records fax numbers, of which *ILA* is ignorant. *Token mismatch* occurs when, despite having appropriate categories, *ILA* fails to find matching tokens due to a difference in representation. For example, *ILA* may record prices in dollars, but a Japanese information source may store prices in yen. Finally, *ILA*'s conjunctive bias can prevent it from learning a category that corresponds to a disjunction of *ILA*'s categories. In future work we expect to test *ILA* on substantially more complex Internet domains, explore solutions to the above problems, and investigate the discovery and quality problems mentioned in the introduction.

References

[Angluin and Laird, 1988] D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

[Angluin, 1988] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, April 1988.

[Auer *et al.*, 1995] Peter Auer, Philip M. Long, Wolfgang Maass, and Gerhard J. Woeginger. On the complexity of function learning. *Machine Learning*, 18(2/3):187–230, February/March 1995.

[Etzioni and Perkowitz, 1995] O. Etzioni and M. Perkowitz. A probabilistic model of sample complexity. 1995. in preparation.

[Etzioni and Weld, 1994] O. Etzioni and D. Weld. A softbot-based interface to the internet. *CACM*, 37(7):72–76, July 1994.

[Haussler, 1988] D. Haussler. Quantifying inductive bias: AI learning algorithms and valiant's learning framework. *Artificial Intelligence*, 36(2):177–221, 1988.

[Kirk *et al.*, 1995] Thomas Kirk, Alon Y. Levy, Yehoshua Sagiv, and Divesh Srivastava. The information manifold. In *Working Notes of the AAAI Spring Symposium: Information Gathering from Heterogeneous, Distributed Environments*, pages 85–91, Stanford University, 1995. AAAI Press.

[Knoblock *et al.*, 1994] Craig Knoblock, Yigal Arens, and Chun-Nan Hsu. Cooperating agents for information retrieval. In *Proceedings of the Second International Conference on Cooperative Information Systems*, Toronto, Canada, 1994.

[Rajamoney, 1993] S. Rajamoney. The design of discrimination experiments. *Machine Learning*, 12(1/2/3), August 1993.

[Richards and Mooney, 1992] B. L. Richards and R. J. Mooney. Learning relations by pathfinding. In *Proc. 10th Nat. Conf. on A.I.*, pages 50–55, 1992.

[Wiederhold, 1992] G Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, pages 38–49, March 1992.

[Wittgenstein, 1958] Ludwig Wittgenstein. *Philosophical Investigations*. Macmillan Publishing Co., Inc., 1958. Translated by G.E.M. Anscombe.