SEVENTH FRAMEWORK PROGRAMME

Project Acronym:	Xperience
Project Type:	IP
Project Title:	Robots Bootstrapped through Learning from Experience
Contract Number:	215821
Starting Date:	01-01-2011
Ending Date:	31-12-2015

XPERIENCE.ORG

Deliverable Number:	D4.2.4
Deliverable Title :	Structural Bootstrapping for Language and Communication
Type (Internal, Restricted, Public):	PU
Authors:	Mark Steedman, Peter Kaiser, Mike Lewis, Ronald Petrick, Nathaniel Smith, Tamim Asfour
Contributing Partners:	UEDIN, KIT

Contractual Date of Delivery to the EC: 31-01-2014
Actual Date of Delivery to the EC: 28-02-2014

Contents

1 Executive Summary

This deliverable deals with lexicalized grounded conceptual representations, consisting of reusable, predictive, and applicable domain knowledge, structurally bootstrapped from raw text, that serve as the basis for deliberative planning, and a mechanism for their application in grounded communication between computational and human agents.

Significant developments that we report in this deliverable are the following:

- We show that spatial relations between domain objects can be inferred from unlabeled text in two complementary ways, and that such knowledge can be practically deployed to aid a domain planner.
- We show that is is possible to combine pragmatic learning of the "use" of language, based on reasoning recursively about dialog acts as resulting from planning by goal directed agents, with Bayesian models of acquisition the "form" of language, of the kind investigated by Kwiatkowski et al. (2012) and reported under earlier deliverables in WP4.

To address these problems in Structural Bootstrapping for Language and Communication, we first present work on the semiautomatic extension of ontological domain knowledge for deliberative planners (including the PKS planner reported under deliverable D3.2.3 for WP3 and the Elexir planner described under D4.2.3), using large volumes of unlabeled (raw) text. We then discuss pragmatic learning for dialog planning.

1.1 Text-mining Planning Domain Knowledge

The construction of planning domain knowledge representations for even simple action domains such as the kitchen demos used in the Xperience project, is tedious, error-prone, typically incomplete, and nontransferable to other domains. The Xperience project seeks to automate this process on several fronts, including work reported elsewhere under WP3, D3.2.3 on induction of PDDL planning operators themselves from observation of change, and on collaborative planning. However, such rule induction depends on knowledge of a prior ontological representation of the entity types in the domain and the action types that they afford. The present work addresses the problem of semiautomatic acquisition of such knowledge from free text available in large quantities on the web.

The first of two attached papers (Kaiser et al. 2014) approaches this problem in two phases. In the first phase, the English words denoting general types of objects (cups etc.), actions (mixing, etc.), and relations (in etc.) that are specific to the planning domain are identified, either by hand and observation, or by extracting them automatically from specialized text (such as cookery manuals). This first level ontology is typically very incomplete. It is fleshed out in a second more computationally intensive phase where much larger amounts of general text in the form of the Google n-gram corpus, which contains 500B words as 5-tuples including dependency relations, in several languages, harvested from Google Books, part of which is examined using templates constructed from the entity types and relation words discovered in the first phase. From this (noisy, partially observed) data, domain knowledge such as that milk is often in the fridge, and that pans are often in cupboards, can be identified. The usefulness of this knowledge extraction can be evaluated via the planner, showing that when supplied with this knowledge in a standard (PDDL) form, it can achieve plans that it was unable to find unaided.

A number of tasks in this area of knowledge acquisition and knowledge mining remain open at the time of this report and are the subject of ongoing and future work under the Xperience project:

- A known problem with data-mining knowledge from human-generated text in this way is that some of the knowledge that is needed for planning by machine is so blindingly obvious to humans that it is never mentioned, no matter how much text is examined. (For example, the fact that cups are often in cupboards turns out to be of this banal kind.) Our current and future work seeks to generalize the mined relations to unseen but plausible relations. (For example, the fact that cups and pans are rigid containers that can be clustered automatically on the basis of their similar bag-of-words collocations in text (Lin and Pantel, 2001b; 2001a) in the first phase of knowledge elicitation should provide a basis for the generalization to their being found in similar locations, such as cupboards.

- As in the case of domain knowledge arising from observation and action in the world, the relations that we bootstrap from text about that world are probabilistic. We continue to investigate the use of probabilistic models in connection with knowledge mined from text, as discussed in D3.2.3.

1.2 Pragmatic Learning

Language users must solve two contradictory problems: to learn, they must work out what words mean in general (or *literally*); to speak and understand, they need to figure out what words mean in context (or *pragmatically*). For example, the sentence “he ate some of the cookies” is literally true even if he ate all of the cookies, but would generally be used and interpreted pragmatically in context as implying that not all of the cookies were eaten. Pragmatic effects can be captured elegantly by Bayesian models which reason recursively about dialog acts as resulting from planning by goal directed agents – in essence, the models reason: “If all of the cookies were gone, she could have truthfully said either ‘some’ or ‘all’ were eaten; but she would have preferred to say ‘all’, because it would be more informative; but, she did not say ‘all’, therefore they must not all have been eaten” (Frank and Goodman 2012; Goodman and Stuhlm???ler 2013). Furthermore, Bayesian learning models can capture a wide variety of language learning phenomena ((Kwiatkowski et al. 2012; Frank et al. 2009). However, there are no models which combine the two, and this turns out to be fundamentally difficult: learning models assume there is an underlying hidden lexicon in use, which is not true when other agents are using planning rather than lexicon lookup to select utterances. Combining learning and pragmatics in the same model seems to force us to either reason about unbounded higher-order beliefs, which are wildly intractable, or else accept misspecification.

Smith et al. (2013) (attached) propose a particular misspecified model in which each agent assumes, incorrectly, that there is an underlying literal lexicon which the other agents know (the “social anxiety” assumption), and then tries to learn what this is. The resulting model is the first to capture a wide variety of both learning and pragmatic phenomena within a single framework. It is also, we believe, the first learning model that can learn both by listening to what more knowledgeable speakers say, and also by speaking to more knowledgeable listeners and observing how they react, and generalize from one to the other. And, it is the first model we know of that can infer underlying literal meanings while observing pragmatically strengthed uses in context, even when these differ.

In addition, we find that when two of our agents interact, even if they both start out ignorant about the lexicon (because there simply is no language in prior use), they quickly converge on a shared lexicon. What is more, they turn out to have a systematic bias towards converging on those lexicons which are most useful for the current task (for example, ones which use short/cheap messages for referring to common entities). This work thus demonstrates a possible route by which local, task-adapted multi-agent norms can be bootstrapped using pure structural learning mechanisms.

A number of tasks in this area of pragmatic and semantic language acquisition remain open at the time of this report and are the subject of ongoing and future work under the Xperience project:

- This work complements work using reward adaptive planning by Valtazanos (2014), reported under D3.2.3, and we seek in future work to combine these approaches with a view to scaling both in combination with deliberative dialog planning.

References

Frank, M. and Goodman, N. (2012). Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998.

Frank, S., Goldwater, S., and Keller, F. (2009). Evaluating models of syntactic category acquisition without using a gold standard. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, pages 2576–2581.

Goodman, N. and Stuhlm???ler, A. (2013). Knowledge and implicature: Modeling language understanding as social cognition. *Topics in Cognitive Science*, 5:173–184.

Kaiser, P., Lewis, M., Petrick, R., Asfour, T., and Steedman, M. (2014). Extracting common sense knowledge from text for robot planning. In *International Conference on Robotics and Automation (ICRA)*, IEEE. To appear.

Kwiatkowski, T., Goldwater, S., Zettlemoyer, L., and Steedman, M. (2012). A probabilistic model of syntactic and semantic acquisition from child-directed utterances and their meanings. In *Proceedings of the 13th Conference of the European Chapter of the ACL (EACL 2012)*, pages 234–244, Avignon. ACL.

Lin, D. and Pantel, P. (2001a). Dirt—discovery of inference rules from text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data-Mining (KDD-01)*, pages 323 – 328, San Francisco.

Lin, D. and Pantel, P. (2001b). Induction of semantic classes from natural language text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data-Mining (KDD-01)*, pages 317–322, San Francisco.

Smith, N., Goodman, N., and Frank, M. (2013). Learning and using language via recursive pragmatic reasoning about other agents. In *Advances in Neural Information Processing Systems*, pages 3039–3047.

Valtazanos, A. (2014). Reward-adaptive planning with unknown teammates from limited observation. In *Proceedings of the 31st Workshop of the UK Planning and Scheduling Special Interest Group*. PlanSIG.

2 Attached Papers

Kaiser et al. (2014) Extracting Common Sense Knowledge from Text for Robot Planning, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (to appear)

Autonomous robots often require domain knowledge to act intelligently in their environment. This is particularly true for robots that use automated planning techniques, which require symbolic representations of the operating environment and the robots capabilities. However, the task of specifying domain knowledge by hand is tedious and prone to error. As a result, we aim to automate the process of acquiring general common sense knowledge of objects, relations, and actions, by extracting such information from large amounts of natural language text, written by humans for human readers. We present two methods for knowledge acquisition, requiring only limited human input, which focus on the inference of spatial relations from text. Although our approach is applicable to a range of domains and information, we only consider one type of knowledge here, namely object locations in a kitchen environment. As a proof of concept, we test our approach using an automated planner and show how the addition of common sense knowledge can improve the quality of the generated plans.

Smith et al. (2013) Learning and using language via recursive pragmatic reasoning about other agents, *Advances in Neural Information Processing Systems*, 3039-3047.

Language users are remarkably good at making inferences about speakers intentions in context, and children learning their native language also display substantial skill in acquiring the meanings of unknown words. These two cases are deeply related: Language users invent new terms in conversation, and language learners learn the literal meanings of words based on their pragmatic inferences about how those words are used. While pragmatic inference and word learning have both been independently characterized in probabilistic terms, no current work unifies these two. We describe a model in which language learners assume that they jointly approximate a shared, external lexicon and reason recursively about the goals of others in using this lexicon. This model captures phenomena in word learning and pragmatic inference; it additionally leads to insights about the emergence of communicative systems in conversation and the mechanisms by which pragmatic inferences become incorporated into word meanings.

Extracting Common Sense Knowledge from Text for Robot Planning

Peter Kaiser[1] Mike Lewis[2] Ronald P. A. Petrick[2] Tamim Asfour[1] Mark Steedman[2]

Abstract— **Autonomous robots often require domain knowledge to act intelligently in their environment. This is particularly true for robots that use automated planning techniques, which require symbolic representations of the operating environment and the robot's capabilities. However, the task of specifying domain knowledge by hand is tedious and prone to error. As a result, we aim to automate the process of acquiring general common sense knowledge of objects, relations, and actions, by extracting such information from large amounts of natural language text, written by humans for human readers.**

We present two methods for knowledge acquisition, requiring only limited human input, which focus on the inference of spatial relations from text. Although our approach is applicable to a range of domains and information, we only consider one type of knowledge here, namely object locations in a kitchen environment. As a proof of concept, we test our approach using an automated planner and show how the addition of common sense knowledge can improve the quality of the generated plans.

I. Introduction and Related Work

Autonomous robots that use automated planning to make decisions about how to act in the world require symbolic representations of the robot's environment and the actions the robot is able to perform. Such models can be aided by the presence of *common sense knowledge*, which may help guide the planner to build higher quality plans, compared with the absence of such information. In particular, knowledge about default locations of objects (*the juice is in the refrigerator*) or the most suitable tool for an action (*knives are used for cutting*) could help the planner make decisions on which actions are more appropriate in a given context.

For example, if a robot needs a certain object for a task, it can typically employ one of two strategies in the absence of prior domain knowledge: the robot can ask a human for the location of the object, or the robot can search the domain in an attempt to locate the object itself. Both techniques are potentially time consuming and prevent the immediate deployment of autonomous robots in unknown environments. By contrast, the techniques proposed in this paper allow the robot to consider likely locations for an object, informed by common sense knowledge. This potentially improves plan quality, by avoiding exhaustive search, and does not require the aid of a human either to inform the robot directly or to encode the necessary domain knowledge a priori.

While it is not possible to automatically generate all the domain knowledge that could possibly be required, we propose two methods for learning useful elements of domain knowledge based on information gathered from natural language texts. These methods will provide the set of object and action types for the domain, as well as certain relations between entities of these types, of the kind that are commonly used in planning. As an evaluation, we build a domain for a robot working in a kitchen environment (see Fig. 1) and infer spatial relations between objects in this domain. We then show how the induced knowledge can be used by an automated planning system. (The generated symbols will not be grounded in the robot's internal model; however, approaches to establish these links given names of objects or actions are available (e.g., [1], [2], [3] and [4]).)

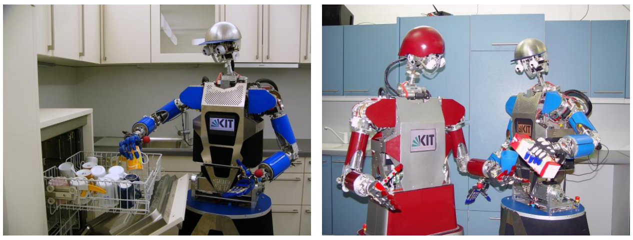

Fig. 1: The humanoid robots ARMAR-IIIa (left) and ARMAR-IIIb working in a kitchen environment ([5], [6]).

The extraction of spatial relations from natural language has been studied in the context of understanding commands and directions given to robots in natural language (e.g., [7], [8], [9]). In contrast to approaches based on annotated corpora of command executions or route instructions, or the use of knowledge bases like *Open Mind Common Sense* [10] explicitly created for artificial intelligence applications, we extract relevant relations from large amounts of text written by humans for humans. The text mining techniques used in [11], [12], [13] to extract action-tool relations to disambiguate visual interpretations of kitchen actions are related. In [14], spatial relations are inferred based on search engine queries and common sense databases.

In the following, we describe a process for learning domain ontologies (Section II) and for extracting relations (Section III). The last two sections evaluate both methods (Section IV) and describe how the resulting knowledge can be used in an automated planning system (Section V).

II. Automatic Domain Ontology Learning

In this section, we propose a method for automatically learning a *domain ontology* $\mathcal{D}$—a set of symbols that refer to a robot's environment or capabilities—with very little human input. The method can be configured to learn a domain

[1]Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany {peter.kaiser, asfour}@kit.edu
[2]School of Informatics, University of Edinburgh, Edinburgh, Scotland, United Kingdom {mike.lewis, rpetrick, steedman}@inf.ed.ac.uk

of objects or actions. With robotic planning in mind, it is crucial that in either of these cases, the contained symbols are not too abstract. In terms of a kitchen environment, interesting objects might be *saucepan*, *refrigerator* or *apple*, while abstract terms like *minute* or *temperature* that do not directly refer to objects are avoided. Similarly, we focus on actions that are directly applied to objects like *knead*, *open* or *screw*, and ignore more abstract actions like *have* or *think*.

Automatic domain ontology learning is based on a *domain-defining corpus* $\mathcal{C}_D$, which contains texts concerning the environment that the domain should model. For example, a compilation of recipes is a good domain-defining corpus for a kitchen environment. Note that these texts have been written by humans for human readers, and no efforts are taken to make them more suitable for $\mathcal{C}_D$. However, $\mathcal{C}_D$ needs to be part-of-speech (POS) tagged and possibly parsed if compound nouns are to be appropriately recognized.[1]

The domain-defining corpus $\mathcal{C}_D$ is used to retrieve an initial vocabulary $\mathcal{V}$ which is then filtered for abstract symbols. Depending on the type of symbol that this vocabulary is meant to model, only nouns or verbs are included in $\mathcal{V}$. In the first step, $\mathcal{C}_D$ is analyzed for word frequency and the k most frequent words are extracted (see Alg. 1). Only words with a part-of-speech-tag (*POS-tag*) equal to $p \in \{\text{noun}, \text{verb}\}$ are considered. The resulting vocabulary $\mathcal{V}$ is then filtered according to the score $\Theta(w, p)$ which expresses the *concreteness* of a word w.

Algorithm 1: *learnDomainOntology*$(\mathcal{C}_D, p, k, \Theta_{\text{min}})$

$\mathcal{V} \leftarrow \textit{mostFrequentWords}(\mathcal{C}_D, p, k)$
$\mathcal{D} \leftarrow \{w \in \mathcal{D} : \Theta(w, p) \geq \Theta_{\text{min}}\}$
return $\mathcal{D}$

Fig. 2 gives an overview of the domain ontology learning process. Additionally, it shows details about the relation extraction procedure that will be discussed below, and the interoperability between the two methods. In the following section, we discuss the concreteness score Θ in detail.

A. The Concreteness Θ

Having a measure of concreteness is necessary for filtering symbols that are too abstract to play a role in our target domain. In particular, the score $\Theta(w, p)$ resembles the concreteness of a word w with POS-tag p using the lexical database *WordNet* ([16], [17]). For nouns, WordNet features an ontology that differs between physical and abstract entities. However, a word can have different meanings, some of which could be abstract and others not. WordNet solves this issue by working on word-senses rather than on words. For a word w with a sense[2] s from the set $S(w)$ of possible senses of w, we can compute a Boolean indicator $c_{w,s}$ that

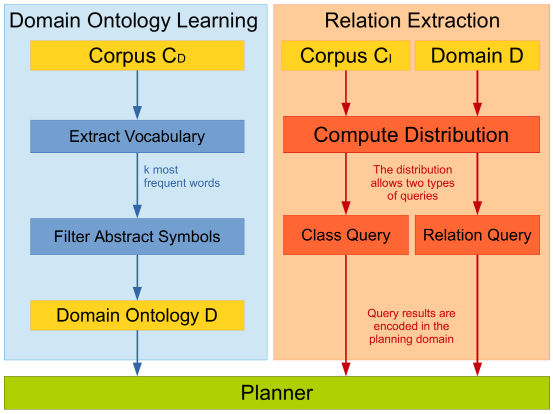

Fig. 2: The process of domain ontology learning (left) and relation extraction (right). The ontology resulting from the first method can be used as input to relation extraction.

tells us if s is a physical meaning of w:

$$c_{w,s} = \begin{cases} 1, & \text{if } s \text{ is a physical meaning of } w \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

WordNet also features a frequency measure $f_{w,s}$ that indicates how often a word w was encountered in the sense of s, based on a reference corpus. As we are not doing semantic parsing on $\mathcal{C}_D$, we do not know which of the possible senses of w is true. However, we can compute a weighted average of the concreteness of the different meanings of w, weighing each word-sense with its likeliness:

$$\Theta(w) = \frac{\sum_{s \in S(w)} f_{w,s} \cdot c_{w,s}}{\sum_{s \in S(w)} f_{w,s}}. \tag{2}$$

As a byproduct, Θ can only have nonzero values for words that are contained in WordNet, which filters out misspelled words or parsing errors.

As there is no suitable differentiation in WordNet's ontology for verbs, we can not apply the exact same approach here. However, WordNet features a rough clustering of verbs that we use to define the filter. We set $c_{w,s}$ to 1, if the verb w with sense s is in one of the following categories: *verb.change*, *verb.contact*, *verb.creation* or *verb.motion*.

III. Relation Extraction

The second technique we propose for information acquisition deals with relations between symbols, defined using syntactic patterns. Such patterns capture the syntactic contexts that describe the relevant relations as well as the relations' arguments. For example, the pattern

$$\big((\textit{\#object}, \textit{noun}), (\textit{in}, \textit{prep}), (\textit{\#location}, \textit{noun})\big) \tag{3}$$

describes a prepositional relation between two nouns using the preposition *in*. The pattern also defines two classes,[3] *#object* and *#location*, which stand for the two arguments of the relation. Given the above syntactic pattern, two types of questions are relevant in this work:

[1] We use the *Stanford Parser* [15] to do this.
[2] WordNet numbers the different word-senses, so $S(w) \subset \mathbb{N}$.
[3] In examples we use a hash to indicate a classname.

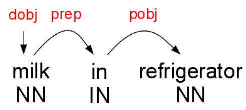

Fig. 3: A dependency path for the fragment *milk in refrigerator* contains the words, their respective POS-tag and the syntactic relation between them.[4]

- *Class inference*: Is a symbol w more likely an object or a location?
- *Relation inference*: What is the most likely location for a symbol w?

The acquisition of relational information is interesting for endowing a robot with initial knowledge of its environment. Our main application for this method is the extraction of spatial relations in a kitchen setting, such as the location of common objects. However, the proposed method is not constrained to objects and locations, and we will show different use cases in the evaluation in Section IV-C.

The relation extraction process works in two phases:

- In the *crawling phase*, the text sources are searched for predefined syntactic patterns. Words falling into classes defined in the patterns are counted. The counts are compiled to a set of distributions.
- In the *query phase*, information can be queried from the distributions computed in the crawling phase. Different kinds of queries are possible.

The foundation for relation extraction is the *domain-independent corpus* $\mathcal{C}_I$. In contrast to the domain-defining corpus $\mathcal{C}_D$, $\mathcal{C}_I$ contains unrestricted text. Because it is rare for common sense information to be explicitly expressed, the size of $\mathcal{C}_I$ is crucial. We assume that the domain-independent corpus is dependency parsed, i.e., consists of syntactic dependency paths of the kind shown in Fig. 3. A further discussion of $\mathcal{C}_I$ is given in Section IV-C.

In the following sections, we give a formal definition of a syntactic pattern and explain the two phases in further detail. Fig. 2 gives an overview of relation extraction.

A. Syntactic Patterns

The goal of the crawling phase is to search large amounts of texts for syntactic patterns predefined by the user. These patterns are designed to specify a relation between classes of words. For example, pattern (3) describes a spatial relation between the two classes *#object* and *#location*. The fragment *milk in refrigerator* would match the pattern and would result in the assignments *#object=milk* and *#location=refrigerator*.

Formally, a syntactic pattern is defined as a sequence of tuples containing a symbol s_i and a POS-tag p_i:

$$\Pi = \big((s_1, p_1), \cdots, (s_k, p_k)\big). \tag{4}$$

When matching the pattern to a sequence of words, the tuples will match exactly one word of the sequence. The condition for a match depends on the symbol s_i:

- If s_i is a word, the i-th tuple matches to this exact word with POS-tag p_i.
- If s_i is a classname, the i-th tuple matches to all words from $\mathcal{D}$ having the POS-tag p_i.

We will use the predicates $\mathit{isclass}(s_i)$ and $\mathit{isword}(s_i)$ to differ between the two possible meanings of the symbol s_i.

The search for matches happens on word-sequences. Such sequences can represent sentences or, as is the case for our corpus $\mathcal{C}_I$, dependency paths. A word-sequence contains words w_i together with their respective POS-tag t_i:

$$\Sigma = \big((w_1, t_1), \cdots, (w_n, t_n)\big), \tag{5}$$

Alg. 2 decides if an element (s, p) from a syntactic pattern matches an element (w, t) from a word-sequence. If the symbol s is a class, it only checks if the word w is part of the domain ontology $\mathcal{D}_p$ that contains the valid words with POS-tag p. If s is a word, it must equal w. In both cases, the POS-tags p and t have to match.

Algorithm 2: $\mathit{match}((s,p),(w,t),\mathcal{D})$

if $\mathit{isclass}(s)$ **then**
  return $p = t \wedge w \in \mathcal{D}_p$
end
else if $\mathit{isword}(s)$ **then**
  return $p = t \wedge s = w$
end

Alg. 3 describes the matching process for a complete syntactic pattern (using Alg. 2). If a match is found, the class configuration is returned as a set of class assignments, i.e., class-word pairs. For example, using pattern (3), the fragment *milk in refrigerator* results in the class configuration:

$$\mathcal{K} = \big\{(\mathit{\#object}, \mathit{milk}), (\mathit{\#location}, \mathit{refrigerator})\big\}. \tag{6}$$

Algorithm 3: $\mathit{configuration}(\Sigma, \Pi, \mathcal{D})$

for $i = 1, \cdots, |\Sigma| - |\Pi| + 1$ **do**
  $\mathcal{I} \leftarrow \{0, \cdots, |\Pi| - 1\}$
  if $\mathit{match}(\Pi_{j+1}, \Sigma_{i+j}, \mathcal{D})\ \forall j \in \mathcal{I}$ **then**
    return $\{(s_{j+1}, w_{i+j}) : j \in \mathcal{I},\ \mathit{isclass}(s_{j+1})\}$
  end
end
return $\emptyset$

B. The Crawling Phase

In the crawling phase, $\mathcal{C}_I$ is searched for pattern matches using Alg. 2 and Alg. 3. Two different distributions are then computed based on the resulting class configurations:

- The *Relation Distribution* D_R counts the occurrences of class configurations (e.g., (6)). D_R is suitable for answering the question: How likely is a class configuration for the relation induced by pattern Π?

[4] *NN* - noun, *IN* - preposition
dobj - direct object, *prep* - preposition, *pobj* - prepositional object

- The *Class Distribution* D_C counts the occurrences of individual class assignments. It is suitable for answering the question: How likely is a class for a given word?

Alg. 4 shows how D_R and D_C are computed given a set of dependency paths $\mathcal{S}$ and a syntactic pattern Π.

Algorithm 4: *computeDistribution*$(\mathcal{S}, \Pi, \mathcal{D})$

$D_R \leftarrow$ Empty Relation Distribution
$D_C \leftarrow$ Empty Class Distribution
foreach $\Sigma = ((w_1, t_1), \cdots, (w_n, t_n)) \in \mathcal{S}$ **do**
  $\mathcal{K} \leftarrow$ *configuration*$(\Sigma, \Pi, \mathcal{D})$
  if $\mathcal{K} \neq \emptyset$ **then**
    $D_R[\mathcal{K}] \leftarrow D_R[\mathcal{K}] + 1$
    foreach $(c, w) \in \mathcal{K}$ **do**
      $D_C[(c, w)] \leftarrow D_C[(c, w)] + 1$
    end
  end
end
return (D_R, D_C)

C. The Query Phase

The query phase uses the distributions D_R and D_C to compute pseudo-probabilities for class assignments.

A *class query* $\gamma(c, w)$ approximates the probability of a word w falling into a class c. If $\Gamma = \{c_1, \cdots, c_l\}$ is the set of defined classes, the class query can be formulated as:

$$\gamma(c, w) = \frac{D_C[(c, w)]}{\sum_{x \in \Gamma} D_C[(x, w)]}. \tag{7}$$

A *relation query* $\rho(Q, c_*)$ approximates the probability of a relation with class-assignments $Q = \{(c_1, w_1), \cdots, (c_l, w_l)\}$, normalizing over the possible values of the class c_*. With $Q_* = \{(c, w) \in Q : c \neq c_*\}$, the relation query can be formulated as:

$$\rho(Q, c_*) = \frac{D_R[Q]}{\sum_{v \in \mathcal{D}} D_R[Q_* \cup \{(c_*, v)\}]}. \tag{8}$$

In the evaluation, we will consider both types of queries.

IV. Evaluation

To evaluate the proposed methods of domain learning and relation extraction, we first show that it is possible to use a specialized corpus to generate a domain ontology of entity types that matches people's expectations for the kitchen environment. We then use another, more general text corpus, to infer spatial relations and action-tool relations for those entities. These components are independent: we show in Section V that hand specification of the domain entities by a human expert can aid the automated extraction processes.

A. Prerequisites

To learn a domain ontology using the proposed method, the two text corpora $\mathcal{C}_D$ and $\mathcal{C}_I$ must first be defined.

1) The domain-defining Corpus $\mathcal{C}_D$: This corpus is used to generate an initial vocabulary by analysing word frequencies. $\mathcal{C}_D$ should therefore be reasonably large but, more importantly, should contain descriptions of common objects and actions from the desired domain. For a kitchen environment, we chose to build $\mathcal{C}_D$ from a set of about 11,000 recipes,[5] with a total size of 19.5 MB.

2) The domain-independent Corpus $\mathcal{C}_I$: The domain-independent corpus is used to sort entities into different classes according to the results of syntactic pattern matches. $\mathcal{C}_I$ does not need to be a different corpus than $\mathcal{C}_D$, but it is difficult to extract reliable information on rare relations from small corpora. This is especially true for common sense knowledge that isn't often explicitly expressed. Hence, $\mathcal{C}_I$ should be extensive. As it is often difficult to gather large amounts of text about a specific topic, it is useful to separate $\mathcal{C}_I$ from $\mathcal{C}_D$, and use a large standard corpus for $\mathcal{C}_I$.

We use the *Google Books Ngrams Corpus* [18], in the following referred to as the *Google Corpus*, which contains a representation of 3.5 million English books containing about 345 billion words in total. The corpus is already parsed, tagged and frequency counted. The Google Corpus does not work on sentences, but on *syntactic ngrams* (Fig. 3), which are n content-word long subpaths of the dependency paths. We use the corpus in its *arcs* form which contains syntactic ngrams with two content-words ($n = 2$) plus possible non-content-words like prepositions or conjunctions. However, the proposed methods can also be used in combination with corpora containing longer syntactic ngrams.

B. Domain Ontology Learning

Using the corpora mentioned above, we can run the method for automatic domain ontology learning. Generating a domain ontology for nouns using parameter values of $k = 300, \Theta_{\text{min}} = 0.35$ results in an ontology of 198 words of which the 80 most frequent ones are listed in Table I. The 20 most frequent nouns that were part of the initial vocabulary, but did not pass the concreteness filter are listed in Table II.

Analogously, the 80 most frequent actions from a domain ontology learnt from verbs using the parameters $k = 300, \Theta_{\text{min}} = 0.2$ are depicted in Table III. The full domain ontology contains 206 verbs. The 20 most frequent verbs that did not pass the concreteness filter are listed in Table IV.

Results show that for objects, as well as actions, the generated domain ontologies are reasonable, but contain obvious mistakes. For example, the concrete noun *cream* was rejected while abstract nouns like *top* and *bottom* were included. The reason for this is the diversity of possible word senses present in WordNet that can mislead the filter Θ.

To evaluate the strength of the domain learning method, we asked four people[6] to manually extract kitchen-related objects and actions from the sets of the 300 most frequent nouns and verbs from $\mathcal{C}_D$. Fig. 4 shows the F_1-scores of the automatically learnt domain ontologies for objects and

[5] From `http://www.ehow.com`.

[6] Native speakers of English, not involved in the research.

TABLE I: Automatic domain ontology learning (objects)

$k = 300, \ \Theta_{\text{min}} = 0.35$

1	wine	21	milk	41	container	61	salad
2	water	22	bottle	42	home	62	tea
3	meat	23	fruit	43	bag	63	grill
4	bowl	24	pot	44	garlic	64	center
5	sugar	25	dough	45	skillet	65	soup
6	mixture	26	glass	46	hand	66	alcohol
7	pan	27	side	47	lid	67	coffee
8	oil	28	pepper	48	onion	68	beer
9	top	29	meal	49	skin	69	sheet
10	oven	30	flour	50	saucepan	70	world
11	salt	31	fish	51	egg	71	diet
12	dish	32	refrigerator	52	beef	72	freezer
13	cheese	33	drink	53	layer	73	blender
14	cup	34	chocolate	74	piece	89	batter
15	butter	35	turkey	55	liquid	75	pasta
16	chicken	36	bottom	56	spoon	76	pork
17	juice	37	cake	57	surface	77	addition
18	bread	38	place	58	restaurant	78	dinner
19	rice	39	ice	59	fat	79	vodka
20	sauce	40	knife	60	plate	80	powder

TABLE II: Not part of the object domain ontology

$k = 300, \ \Theta_{\text{min}} = 0.35$

1	time	6	taste	11	temperature	16	type
2	flavor	7	way	12	day	17	tbsp
3	heat	8	variety	13	process	18	color
4	recipe	9	cream	14	boil	19	hour
5	food	10	amount	15	cooking	20	half

TABLE III: Automatic domain ontology learning (actions)

$k = 300, \ \Theta_{\text{min}} = 0.2$

1	add	21	cool	41	come	61	stick
2	make	22	fill	42	press	62	beat
3	place	23	leave	43	freeze	63	clean
4	remove	24	go	44	garnish	64	begin
5	cook	25	bring	45	pick	65	burn
6	pour	26	hold	46	open	66	spread
7	stir	27	reduce	47	slice	67	replace
8	do	28	follow	48	become	68	whisk
9	put	29	heat	49	refrigerate	69	boil
10	take	30	pan	50	soak	70	produce
11	get	31	sprinkle	51	dip	71	preheat
12	turn	32	dry	52	form	72	squeeze
13	set	33	start	53	shake	73	chill
14	cut	34	melt	74	cause	89	top
15	cover	35	sit	55	pull	75	peel
16	mix	36	chop	56	break	76	fit
17	combine	37	drain	57	wash	77	move
18	create	38	rinse	58	simmer	78	coat
19	prepare	39	blend	59	lay	79	increase
20	bake	40	roll	60	transfer	80	seal

TABLE IV: Not part of the action domain ontology

$k = 300, \ \Theta_{\text{min}} = 0.2$

1	be	6	keep	11	eat	16	check
2	use	7	let	12	choose	17	enjoy
3	have	8	allow	13	need	18	give
4	serve	9	try	14	help	19	see
5	show	10	find	15	buy	20	want

actions, using different values for Θ_{min}, compared to the domains created by the human participants. The results show that enabling the concreteness filter ($\Theta_{\text{min}} > 0$) significantly increases the quality of the resulting domain for nouns as well as for verbs. The results also show that values of roughly $\Theta_{\text{min}} > 0.5$ result in a too restrictive filter. While in the case of nouns, the restrictive filter still produces a better domain than if no filter is applied, this is not true for verbs: the quality of a verb-domain drops dramatically the more restrictive the filter gets. The reason for the difference between the two plots is that verbs often have a variety of possible meanings. By contrast, nouns usually have a predominant interpretation, at least in terms of the differentiation between physical and abstract meanings. This is also reflected in the fact that the participants found it significantly harder to create a domain of actions than to create a domain of nouns.

C. Inference

We now evaluate the relation and class inference mechanisms described in Section III. To illustrate the capabilities of these methods, we generated the results using the manually created domain ontology as a gold standard. We additionally show how false positives can affect the process by using the automatically learnt domain ontology. The parameter Θ_{min} can be determined in practice by generating and evaluating an ontology for a subset of the initial vocabulary using plots similar to Fig. 4. Different syntactic patterns can be used to conduct different kinds of inference. The following sections show examples of possible queries.

1) Location Inference: A good use of knowledge acquisition is the exploration of spatial relations between objects and locations using prepositional contexts. For instance, pattern (3) matches fragments where two nouns, *#object* and *#location*, are linked by the preposition *in*. This pattern can be used in combination with the above object ontology (Table I) to infer spatial relations in a kitchen environment. Table V shows the most likely locations for the ten most frequent objects from the automatically learnt domain ontology.[7] Note that for generating the results we used pattern (3) combined with three similar patterns using the prepositions *on*, *at* and *from*. Table V presents two sets of locations for each object: the upper, highlighted rows refer to the manually created domain ontology and the lower, non-highlighted rows refer to the automatically learnt domain ontology in Table I.

Results from the automatically learnt domain ontologies are more noisy, and distractive terms like *side* or *bottom* haven't been filtered out. (We can also tune domain generation to work in a more restrictive way, e.g., by using the $F_{0.5}$ measure instead of F_1 to emphasize precision over recall.)

The results demonstrate that the system is able to infer typical locations for objects. However, two problems constrain its performance. First, the automatically learnt domain ontology does not contain typical locations like *cupboard* or *drawer*, because these words do not frequently appear in the initial vocabulary. Second, the system is not able to differ between container objects like *pot* or *pan*, and actual locations like *refrigerator* or *oven* (i.e., objects that have a fixed position in the kitchen). Improving the domain entity specification by using more diverse but relevant domain specific corpora is the subject of ongoing research. In Section V

[7] We consider *top* and *oven* not to be objects.

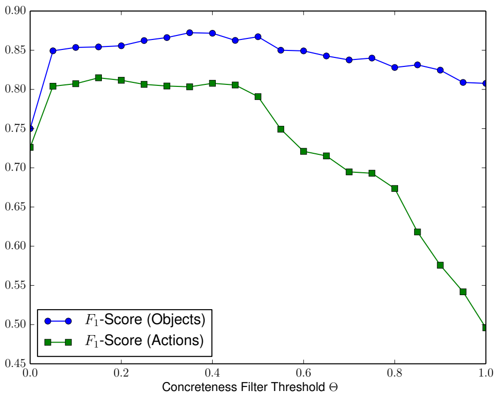

Fig. 4: Automatically learnt domain ontologies are evaluated using different values of $\Theta_{\min}$, by comparing them to domain ontologies created manually by human participants.

TABLE V: Results for location inference

Highlighted rows: Manually created domain ontology. Non-highlighted rows: Automatically learnt domain ontology.
refr. - refrigerator, scp. - saucepan, swc. - sandwich, kit. - kitchen

object	first	second	third	fourth
wine	glass / 0.20	bottle / 0.20	table / 0.15	cup / 0.14
	glass / 0.13	bottle / 0.13	table / 0.10	cup / 0.09
water	surface / 0.09	bottle / 0.07	water / 0.07	glass / 0.07
	bottom / 0.06	side / 0.06	surface / 0.05	bottle / 0.04
meat	table / 0.08	pan / 0.06	swc. / 0.05	pot / 0.05
	diet / 0.11	table / 0.05	pan / 0.04	swc. / 0.04
bowl	table / 0.51	refr. / 0.06	stem / 0.05	kit. / 0.05
	table / 0.27	hand / 0.20	top / 0.06	side / 0.05
sugar	water / 0.15	bowl / 0.15	scp. / 0.10	milk / 0.08
	water / 0.13	bowl / 0.13	scp. / 0.08	milk / 0.06
mixture	pan / 0.10	bowl / 0.08	water / 0.08	dish / 0.08
	top / 0.10	pan / 0.07	bowl / 0.05	water / 0.05
pan	oven / 0.40	stove / 0.19	rack / 0.18	pan / 0.02
	oven / 0.29	stove / 0.14	rack / 0.13	hand / 0.07
oil	skillet / 0.22	pan / 0.19	scp. / 0.08	board / 0.07
	skillet / 0.19	pan / 0.16	scp. / 0.07	board / 0.06
salt	water / 0.40	bowl / 0.17	scp. / 0.05	food / 0.04
	water / 0.33	bowl / 0.14	diet / 0.06	scp. / 0.04
dish	table / 0.30	oven / 0.24	menu / 0.11	pan / 0.03
	table / 0.20	oven / 0.16	hand / 0.12	top / 0.04

we show the effect of more helpful entity specification.

2) Tool Inference: A similar approach that also includes the action domain ontology uses the preposition *with* to infer relations between actions and tools. The following syntactic pattern matches a verb *#action* and a noun *#tool* from the respective domain ontology, linked together by *with*:

$$\big((\#action, verb), (with, prep), (\#tool, noun)\big). \qquad (9)$$

Table VI shows the three most probable tools for different actions from the kitchen domain. The results are shown for both the manually created domain ontology (upper rows, highlighted) and the automatically learnt one (lower rows).

3) Class Inference: Another possible result the system can compute is the probability that a word falls into a certain class of syntactic pattern. For example, given the above pattern (3), the system can approximate the probability that a word names an object or a location (Table VII). These results

TABLE VI: Results for tool inference

Highlighted rows: Manually created domain ontology. Non-highlighted rows: Automatically learnt domain ontology.

object	first	second	third
cut	knife / 0.80	fork / 0.01	machine / 0.01
	knife / 0.68	hand / 0.04	world / 0.03
flip	spatula / 0.89	spoon / 0.06	fork / 0.03
	spatula / 0.65	hand / 0.24	spoon / 0.05
mash	fork / 0.58	spoon / 0.16	butter / 0.09
	fork / 0.59	spoon / 0.16	butter / 0.09
stir	spoon / 0.50	fork / 0.20	spatula / 0.08
	spoon / 0.48	fork / 0.19	spatula / 0.08

can be used to improve the location inference results, e.g., by dropping words that seem unlikely to name a location.

TABLE VII: Results for class inference

Manually created domain ontology (left), automatically learnt domain ontology (right)

symbol	object	location	object	location
wine	0.80	0.20	0.85	0.15
water	0.48	0.52	0.56	0.44
meat	0.77	0.23	0.81	0.19
bowl	0.11	0.89	0.16	0.84
sugar	0.93	0.07	0.95	0.05
mixture	0.72	0.28	0.77	0.23
pan	0.17	0.83	0.18	0.82
oil	0.82	0.18	0.83	0.17
oven	0.12	0.88	0.11	0.89
salt	0.95	0.05	0.96	0.04

4) Computation Time: In this work, the domain ontology learning and distribution computation steps are considered to be run offline. However we note that the computation time for these steps depends heavily on the sizes and representations of $\mathcal{C}_D$ and $\mathcal{C}_I$. Processing the Google Corpus[8] requires especially high computational power. On the other hand, the inference step consists of simple lookups in precomputed distributions, and can therefore be done online.

V. Planning with Common Sense Knowledge

In this section we show how the domain knowledge induced by the processes described above can be used with an automated planning system to improve the quality of generated plans. We have chosen to use the PKS (*Planning with Knowledge and Sensing* [19], [20]) planner for this task, since PKS has previously been deployed in robot environments like the one in Fig. 1 [21]. However, one of the strengths of the above approach is that it is not planner (or domain) dependent, and the method we outline for PKS can be adapted to a range of different planners and domains.

As an example scenario, we will focus on the use of spatial relations in a small kitchen domain. The domain contains the entities *cereal*, *counter*, *cup*, *cupboard*, *juice*, *plate*, *refrigerator* and *stove*. Table VIII shows the results of the location inference method. We will first postprocess this data for planning by considering the entity *juice*.

A. Postprocessing

Given the initial domain of objects, and using pattern (3), we can approximate the probability of an object o being

[8] The Google Arcs Corpus contains 38G of compressed text.

TABLE VIII: Location inference for a small domain

Omitted values are zero.

object	counter	cup	cupboard	dishwasher	juice	plate	refrigerator	stove
cereal			1.00					
cup	0.25	0.33	0.13	0.01		0.18	0.02	0.08
juice	0.02	0.43			0.21	0.08	0.26	
plate	0.14	0.10	0.07			0.58	0.06	0.06

spatially related to a location l by issuing a relation query:

$$P(loc = l|obj = o) = \rho\Big(\big\{(obj, o), (loc, l)\big\}, loc\Big). \quad (10)$$

To put these results into a suitable form for planning, we introduce the predicate *at* and output the computed likelihoods for pairs of objects. The resulting relations that are extracted, and their likelihoods, are shown in the top half of Table IX.

The postprocessor must now refine the results, possibly making use of additional information about the structure of the planning domain and the types of objects that are available. Refinement can be done in three possible ways:

1) *Symbol Mapping*: A word that describes an object in natural language may not necessarily match the symbol name for that object in the planning domain. This is currently corrected by an appropriate mapping process that uses a dictionary of likely synonyms. E.g., the word *refrigerator* may be mapped to *fridge*.
2) *Type Filtering*: Many planners have the concept of object *types*, which enables us to filter relations that have entity arguments of the incorrect type. Assuming the planning domain provides us with a type *location* that is required for the second argument of *at*, the postprocessor can then remove the extracted relations *at(juice, cup)*, *at(juice, juice)*, and *at(juice, plate)*, since the entities *cup*, *juice*, and *plate* are not locations.
3) *Instantiation*: The symbols extracted by our processes will often refer to *classes* of objects, rather than the specific object identifiers used by the planner. Making use of type information in the planning domain, the postprocessor can instantiate objects of the appropriate types from the extracted relational information. For instance, the class *juice* might be instantiated into two objects, *applejuice* and *orangejuice*. These objects can subsequently be substituted in any relations that contains the appropriate class type.

The final set of postprocessed relations from our example is shown in the bottom half of Table IX. We note that the necessity and possibility of applying these postprocessing steps depends on the nature of the planning domain. Furthermore, the information that is needed to perform the postprocessing, i.e., the symbol mapping table or the type information, needs to be manually encoded in the planning domain.

Given the postprocessed set of relations, the final step is to decide how this information will be included in the planning domain. For planners that work with probabilistic representations, the relation/likelihood information could be directly encoded. For planners like PKS that do not deal with probabilities, there are two main possibilities:

1) The most probable location for each object could be encoded as a single fact in the planner's knowledge, i.e., *at(applejuice, fridge)* and *at(orangejuice, fridge)*.
2) Some or all of the most probable locations could be encoded as a disjunction of possible alternatives, i.e., *at(applejuice, fridge)* | *at(applejuice, counter)*, and *at(orangejuice, fridge)* | *at(orangejuice, counter)*.

Depending on the domain, either form may be appropriate.

TABLE IX: Extracted and postprocessed relations

Extracted	
Relations	**Likelihood**
at(juice, cup)	0.43
at(juice, refrigerator)	0.27
at(juice, juice)	0.21
at(juice, plate)	0.08
at(juice, counter)	0.02
Postprocessed	
Relations	**Likelihood**
at(applejuice,fridge)	0.27
at(applejuice,counter)	0.02
at(orangejuice,fridge)	0.27
at(orangejuice,counter)	0.02

B. Plan Generation

Consider the task of finding the apple juice container in the kitchen. In the absence of precise information as to the object's location, but knowing there are various places in the kitchen where objects could be located (e.g., *counter*, *cupboard*, *fridge*, *stove*), a planner could potentially build a plan for a robot to exhaustively check all locations: *move-robot-to-counter*, *check-for-apple-juice*, *if not present move-robot-to-cupboard*, *check-for-apple-juice*, *if not present move-robot-to-fridge*, etc., until all locations have been checked. If the robot does not have information-gathering capabilities to check for the apple juice in a particular location, the planner may not be able to generate such a plan at all.

With the availability of more certain information about the location of the apple juice, the planner can potentially eliminate some parts of the plan (e.g., by ignoring certain locations), or at least prioritise certain likely locations over others, resulting in higher quality plans. For instance, in the case that the planner had the knowledge *at(applejuice, fridge)*, resulting from the above relation extraction process, then the planner could build the simple plan *move-robot-to-fridge*, under the assumption that the extracted information was true.

Similarly, if the planner had the disjunctive information *at(applejuice, fridge)* | *at(applejuice, counter)* then the planner could build the plan: *move-robot-to-fridge*, *check-for-apple-juice*, *if not present move-robot-to-counter*. Again, this plan improves on the exhaustive search plan by only considering the most likely locations for the apple juice, resulting from the extracted relational information.

One inherent danger when dealing with common sense knowledge is that the plans that are built from such information alone may ultimately fail to achieve their goals in the real world. For instance, even though relation extraction provides us with likely locations for the apple juice, there is no guarantee that this is the way the robot's world is actually configured. (E.g., another robot may have left the apple juice on the stove.) However, such information does give us a starting point for building plans, in the absence of more certain information, and can also aid plan execution monitoring to guide replanning activities in the case of plan failure. (E.g., if a plan built using common sense knowledge fails to locate the apple juice, fall back to the exhaustive search plan for the locations that haven't been checked.)

Finally, we note that the use of common sense knowledge may improve the efficiency of plan generation, since in general more specific information helps constrain the plan generation process. However, plan generation time is both domain and planner dependent, and it is difficult to quantify any improvements without experimentation. (E.g., planning time went from 0.003s to 0.001s in our small examples.)

VI. Conclusion and Future Work

We have presented two techniques for reducing the amount of prior, hardcoded knowledge that is necessary for building a robotic planning domain. Using the methods described here, a domain ontology of object and action types can be defined automatically, over which user-defined relations can be inferred automatically from sources of natural language text. The resulting representation of common sense domain knowledge has been tested using an automated planning system, improving the quality of the generated plans.

As future work, we are exploring a number of improvements to our techniques. First, more specialized corpora $\mathcal{C}_I$, longer syntactic patterns, or databases of common sense knowledge might help in overcoming the sparsity of common sense information in text sources. Second, the location inference does not perform any checks for plausibility. While the class inference will help in filtering results that are not locations at all, additional methods are needed to differentiate between locations for temporary storage and locations for long-term storage. Another interesting improvement would be the generalization of inferred relations to still missing knowledge. For example one could conclude by analyzing text sources that *bowl* and *dish* are conceptually similar and therefore apply relations inferred for *bowls* also to *dishes*. Finally, we are investigating the application of our methods to robot domains other than the kitchen environment.

Acknowledgment

The research leading to these results received funding from the European Union's 7th Framework Programme FP7/2007-2013, under grant agreement №270273 (Xperience).

References

[1] M. Ternoth, U. Klank, D. Pangercic, and M. Beetz, "Web-enabled robots," *Robotics & Automation Magazine*, vol. 18, no. 2, pp. 58–68, 2011.

[2] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Galvez-Lopez, K. Haussermann, R. Janssen, J. Montiel, A. Perzylo, B. Schiessle, M. Tenorth, O. Zweigle, and R. van de Molengraft, "Roboearth," *Robotics Automation Magazine, IEEE*, vol. 18, no. 2, pp. 69–82, 2011.

[3] K. Welke, P. Kaiser, A. Kozlov, N. Adermann, T. Asfour, M. Lewis, and M. Steedman, "Grounded spatial symbols for task planning based on experience," in *13th International Conference on Humanoid Robots (Humanoids)*. IEEE/RAS, 2013.

[4] A. Kasper, R. Becher, P. Steinhaus, and R. Dillmann, "Developing and analyzing intuitive modes for interactive object modeling," in *ICMI '07: Proceedings of the 9th international conference on Multimodal interfaces*. New York, NY, USA: ACM, 2007, pp. 74–81.

[5] T. Asfour, K. Regenstein, P. Azad, J. Schröder, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An integrated humanoid platform for sensory-motor control," in *IEEE International Conference on Humanoid Robots (Humanoids)*, 2006, pp. 169–175.

[6] T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schröder, and R. Dillmann, "Toward humanoid manipulation in human-centred environments," *Robotics and Autonomous Systems*, vol. 56, no. 1, pp. 54–65, 2008.

[7] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proceedings of the 25th National Conference on Artificial Intelligence*. AAAI, 2011, pp. 1507–1514.

[8] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward understanding natural language directions," in *Proceedings of the 5th International Conference on Human-Robot Interaction (HRI)*. IEEE, 2010, pp. 259–266.

[9] D. Chen and R. Mooney, "Learning to interpret natural language navigation instructions from observations," in *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-2011)*, 2011, pp. 859–865.

[10] P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, and W. L. Zhu, "Open mind common sense: Knowledge acquisition from the general public," in *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*. Springer, 2002, pp. 1223–1237.

[11] C. Teo, Y. Yang, H. Daumé III, C. Fermüller, and Y. Aloimonos, "A corpus-guided framework for robotic visual perception," in *Workshop on Language-Action Tools for Cognitive Artificial Agents, held at the 25th National Conference on Artificial Intelligence*. San Francisco: AAAI, 2011, pp. 36–42.

[12] ——, "Toward a Watson that sees: Language-guided action recognition for robots," in *IEEE International Conference on Robotics and Automation*. St. Paul, MN: IEEE, 2012, pp. 374–381.

[13] M. Tamosiunaite, I. Markelic, T. Kulvicius, and F. Wörgötter, "Generalizing objects by analyzing language," in *11th International Conference on Humanoid Robots (Humanoids)*. IEEE/RAS, 2011, pp. 557–563.

[14] K. Zhou, M. Zillich, H. Zender, and M. Vincze, "Web mining driven object locality knowledge acquisition for efficient robot behavior," in *2012 International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, 2012, pp. 3962–3969.

[15] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics ACL 03*, vol. 1, pp. 423–430, 2003.

[16] G. A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, pp. 39–41, 1995.

[17] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.

[18] Y. Goldberg and J. Orwant, "A dataset of syntactic-ngrams over time from a very large corpus of english books," in *Second Joint Conference on Lexical and Computational Semantics*, 2013, pp. 241–247.

[19] R. P. A. Petrick and F. Bacchus, "A knowledge-based approach to planning with incomplete information and sensing," in *International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2002)*, 2002, pp. 212–221.

[20] ——, "Extending the knowledge-based approach to planning with incomplete information and sensing," in *International Conference on Automated Planning and Scheduling (ICAPS 2004)*, 2004, pp. 2–11.

[21] R. Petrick, N. Adermann, T. Asfour, M. Steedman, and R. Dillmann, "Connecting knowledge-level planning and task execution on a humanoid robot using Object-Action Complexes," in *Proceedings of the International Conference on Cognitive Systems (CogSys 2010)*, 2010.

Learning and using language via recursive pragmatic reasoning about other agents

Nathaniel J. Smith* University of Edinburgh
Noah D. Goodman Stanford University
Michael C. Frank Stanford University

Abstract

Language users are remarkably good at making inferences about speakers' intentions in context, and children learning their native language also display substantial skill in acquiring the meanings of unknown words. These two cases are deeply related: Language users invent new terms in conversation, and language learners learn the literal meanings of words based on their pragmatic inferences about how those words are used. While pragmatic inference and word learning have both been independently characterized in probabilistic terms, no current work unifies these two. We describe a model in which language learners assume that they jointly approximate a shared, external lexicon and reason recursively about the goals of others in using this lexicon. This model captures phenomena in word learning and pragmatic inference; it additionally leads to insights about the emergence of communicative systems in conversation and the mechanisms by which pragmatic inferences become incorporated into word meanings.

1 Introduction

Two puzzles present themselves to language users: What do words mean in general, and what do they mean in context? Consider the utterances "it's raining," "I ate some of the cookies," or "can you close the window?" In each, a listener must go beyond the literal meaning of the words to fill in contextual details ("it's raining here and now"), infer that a stronger alternative is not true ("I ate some but not all of the cookies"), or more generally infer the speaker's communicative goal ("I want you to close the window right now because I'm cold"), a process known as *pragmatic reasoning*. Theories of pragmatics frame the process of language comprehension as inference about the generating goal of an utterance given a rational speaker [14, 8, 9]. For example, a listener might reason, "if she had wanted me to think 'all' of the cookies, she would have said 'all'—but she didn't. Hence 'all' must not be true and she must have eaten some *but not all* of the cookies." This kind of reasoning is core to language use.

But pragmatic reasoning about meaning-in-context relies on stable literal meanings that must themselves be learned. In both adults and children, uncertainty about word meanings is common, and often considering speakers' pragmatic goals can help to resolve this uncertainty. For example, if a novel word is used in a context containing both a novel and a familiar object, young children can make the inference that the novel word refers to the novel object [22].[1] For adults who are proficient language users, there are also a variety of intriguing cases in which listeners seem to create situation- and task-specific ways of referring to particular objects. For example, when asked to refer to idiosyncratic geometric shapes, over the course of an experimental session, participants create conventionalized descriptions that allow them to perform accurately even though they do not begin with shared labels [19, 7]. In both of these examples, reasoning about another person's goals informs

*nathaniel.smith@ed.ac.uk

[1] Very young children make inferences that are often labeled as "pragmatic" in that they involve reasoning about context [6, 1], though in some cases they are systematically 'too literal' (e.g. failing to strengthen SOME to SOME-BUT-NOT-ALL [23]). Here we remain agnostic about the age at which children are able to make such inferences robustly, as it may vary depending on the linguistic materials being used in the inference [2].

language learners' estimates of what words are likely to mean.

Despite this intersection, there is relatively little work that takes pragmatic reasoning into account when considering language learning in context. Recent work on grounded language learning has attempted to learn large sets of (sometimes relatively complex) word meanings from noisy and ambiguous input (e.g. [10, 17, 20]). And a number of models have begun to formalize the consequences of pragmatic reasoning in situations where limited learning takes place [12, 9, 3, 13]. But as yet these two strands of research have not been brought together so that the implications of pragmatics for learning can be investigated directly.

The goal of our current work is to investigate the possibilities for integrating models of recursive pragmatic reasoning with models of language learning, with the hope of capturing phenomena in both domains. We begin by describing a proposal for bringing the two together, noting several issues in previous approaches based on recursive reasoning under uncertainty. We next simulate findings on pragmatic inference in one-shot games (replicating previous work). We then build on these results to simulate the results of pragmatic learning in the language acquisition setting where one communicator is uncertain about the lexicon and in iterated communication games where both communicators are uncertain about the lexicon.

2 Model

We model a standard communication game [19, 7]: two participants each, separately, view identical arrays of objects. On the *Speaker's* screen, one object is highlighted; their goal is to get the *Listener* to click on this item. To do this, they have available a fixed, finite set of words; they must pick one. The Listener then receives this word, and attempts to guess which object the Speaker meant by it. In the psychology literature, as in real-world interactions, games are typically iterated; one view of our contribution here is as a generalization of one-shot models [9, 3] to the iterated context.

2.1 Paradoxes in optimal models of pragmatic learning. Multi-agent interactions are difficult to model in a normative or optimal framework without falling prey to paradox. Consider a simple model of the agents in the above game. First we define a *literal listener* L_0. This agent has a *lexicon* of associations between words and meanings; specifically, it assigns each word w a vector of numbers in $(0, 1)$ describing the extent to which this word provides evidence for each possible object[2].To interpret a word, the literal listener simply re-weights their prior expectation about what is referred to using their lexicon's entry for this word:

$$P_{L_0}(\text{object}|\text{word}, \text{lexicon}) \propto \text{lexicon}(\text{word}, \text{object}) \times P_{\text{prior}}(\text{object}). \tag{1}$$

Because of the normalization in this equation, there is a systematic but unimportant symmetry among lexicons; we remove this by assuming the lexicon sums to 1 over objects for each word. Confronted with such a listener, a speaker who chooses approximately optimal actions should attempt to choose a word which soft-maximizes the probability that the listener will assign to the target object—modulated by the effort or cost associated with producing this word:

$$P_{S_1}(\text{word}|\text{object}, \text{lexicon}) \propto \exp\Big(\lambda\big(\log P_{L_0}(\text{object}|\text{word}, \text{lexicon}) - \text{cost}(\text{word})\big)\Big). \tag{2}$$

But given this speaker, then the naive L_0 strategy is not optimal. Instead, listeners should use Bayes rule to invert the speaker's decision procedure [9]:

$$P_{L_2}(\text{object}|\text{word}, \text{lexicon}) \propto P_{S_1}(\text{word}|\text{object}, \text{lexicon}) \times P_{\text{prior}}(\text{object}). \tag{3}$$

Now a difficulty becomes apparent. Given such a listener, it is no longer optimal for speakers to implement strategy S_1; instead, they should implement strategy S_3 which soft-maximizes P_{L_2} instead of P_{L_0}. And then listeners ought to implement L_4, and so on.

One option is to continue iterating such strategies until reaching a fixed point equilibrium. While this strategy guarantees that each agent will behave normatively given the other agent's strategy, there is no guarantee that such strategies will be near the system's global optimum. More importantly,

[2]We assume words refer directly to objects, rather than to abstract semantic features. Our simplification is without loss of generalization, however, because we can interpret our model as marginalizing over such a representation, with our literal $P_{\text{lexicon}}(\text{object}|\text{word}) = \sum_{\text{features}} P(\text{object}|\text{features})P_{\text{lexicon}}(\text{features}|\text{word})$.

there is a great deal of evidence that humans do not use such equilibrium strategies; their behavior in language games (and in other games [5]) can be well-modeled as implementing S_k or L_k for some small k [9]. Following this work, we recurse a finite (small) number of times, n. The consequence is that one agent, implementing S_n, is fully optimal with respect to the game, while the other, implementing L_{n-1}, is only nearly optimal—off by a single recursion.

This resolves one problem, but as soon as we attempt to add uncertainty about the meanings of words to such a model, a new paradox arises. Suppose the listener is a young child who is uncertain about the lexicon their partner is using. The obvious solution is for them to place a prior on the lexicon; they then update their posterior based on whatever utterances and contextual cues they observe, and in the mean time interpret each utterance by making their best guess, marginalizing out this uncertainty. This basic structure is captured in previous models of Bayesian word learning [10]. But when combined with the recursive pragmatic model, a new question arises: Given such a listener, what model should the speaker use? A rational speaker attempts to maximize the listener's likelihood of understanding, so if an uncertain listener interpets by marginalizing over some posterior, then a fully knowledgeable speaker should disregard their own lexical knowledge, and instead model and marginalize over the listener's uncertainty. But if they do this, then their utterances will provide no data about their lexicon, and there is nothing for the rational listener to learn from observing them.[3]

One final problem is that under this model, when agents switch roles between listener and speaker, there is nothing constraining them to continue using the same language. Optimizing task performance requires my lexicon as a speaker to match your lexicon as a listener and vice-versa, but there is nothing that relates my lexicon as a speaker to my lexicon as a listener, because these never interact. This clearly represents a dramatic mismatch to typical human communication, which almost never proceeds with distinct languages spoken by each participant.

2.2 A conventionality-based model of pragmatic word learning. We resolve the problems described above by assuming that speakers and listeners deviate from normative behavior by assuming a conventional lexicon. Specifically, our final convention-based agents assume: (a) There is some single, specific literal lexicon which everyone should be using, (b) and everyone else knows this lexicon, and believes that I know it as well, (c) but in fact I don't. These assumptions instantiate a kind of "social anxiety" in which agents are all trying to learn the correct lexicon that they assume everyone else knows.

Assumption (a) corresponds to the lexicographer's illusion: Naive language users will argue vociferously that words have specific meanings, even though these meanings are unobservable to everyone who purportedly uses them. It also explains why learners speak the language they hear (rather than some private language that they assume listeners will eventually learn): Under assumption (a), observing other speakers' behavior provides data about not just that speaker's idiosyncratic lexicon, but the consensus lexicon. Assumption (b) avoids the explosion of hyper^n-distributions described above: If agent n knows the lexicon, they assume that all lower agents do as well, reducing to the original tractable model without uncertainty. And assumption (c) introduces a limited form of uncertainty at the top level, and thus the potential for learning. To the extent that a child's interlocutors do use a stable lexicon and do not fully adapt their speech to accomodate the child's limitations, these assumptions make a reasonable approximation for the child language learning case. In general, though, in arbitrary multi-turn interactions in which both agents have non-trivial uncertainty, these assumptions are incorrect, and thus induce complex and non-normative learning dynamics.

Formally, let an unadorned L and S denote the listener and speaker who follow the above assumptions. If the lexicon were known then the listener would draw inferences as in L_{n-1} above; but by assumption (c), they have uncertainty, which they marginalize out:

$$P_L(\text{object}|\text{word}, L\text{'s data}) = \sum_{\text{lexicon}} P_{L_{n-1}}(\text{object}|\text{word}, \text{lexicon}) P(\text{lexicon}|L\text{'s data}) \quad (4)$$

[3] Of course, in reality both parties will generally have some uncertainty, making the situation even worse. If we start from an uncertain listener with a prior over lexicons, then a first-level uncertain speaker needs a prior over priors on lexicons, a second-level uncertain listener needs a prior over priors over priors, etc. The original $L_0 \to S_1 \to \dots$ recursion was bad enough, but at least each step had a constant cost. This new recursion produces hyper^n-distributions for which inference almost immediately becomes intractable even in principle, since the dimensionality of the learning problem increases with each step. Yet, without this addition of new uncertainty at each level, the model would dissolve back into certainty as in the previous paragraph, making learning impossible.

Phenomenon	Ref.	WL	PI	PI+U	PI+WL	Section
Interpreting scalar implicature	[14]		x	x	x	3.1
Interpreting Horn implicature	[15]			x	x	3.2
Learning literal meanings despite scalar implicature	[21]				x	4.1
Disambiguating new words using old words	[22]	x		x	x	4.2
Learning new words using old words	[22]	x			x	4.2
Disambiguation without learning	[16]			x	x	4.2
Emergence of novel & efficient lexicons	[11]				x	5.1
Lexicalization of Horn implicature	[15]				x	5.2

Table 1: Empirical results and references. WL refers to the word learning model of [10]; PI refers to the recursive pragmatic inference model of [9]; PI+U refers to the pragmatic inference model of [3] which includes lexical uncertainty, marginalizes it out, and then recurses. Our current model is referred to here as PI+WL, and combines pragmatic inference with word learning.

Here L's data consists of her previous experience with language. In particular in the iterated games explored here it consists of S's previous utterances together with whatever other information L may have about their intended referents (e.g. from contextual clues). By assumption (b), L treats these utterances as samples from the knowledgeable speaker S_{n-2}, not S, and thus as being informative about the lexicon. For instance, when the data is a set of fully observed word-referent pairs $\{w_i, o_i\}$:

$$P(\text{lexicon}|L\text{'s data}) \propto P(\text{lexicon}) \prod_i P_{S_{n-2}}(w_i|o_i, \text{lexicon}) \tag{5}$$

The top-level speaker S attempts to select the word which soft-maximizes their utility, with utility now being defined in terms of the informativity of the expectation (over lexicons) that the listener will have for the right referent[4]:

$$P_S(\text{word}|\text{object}, S\text{'s data}) \propto \\ \exp\Big(\lambda\big(\log \sum_{\text{lexicon}} P_{L_{n-1}}(\text{object}|\text{word}, \text{lexicon}) P(\text{lexicon}|S\text{'s data}) - \text{cost}(\text{word})\big)\Big) \tag{6}$$

Here $P(\text{lexicon}|S\text{'s data})$ is defined similarly, when S observes L's interpretations of various utterances, and treats them as samples from L_{n-1}, not L. However, notice that if S and L have the same subjective distributions over lexicons, then S is approximately optimal with respect to L in the same sense that S_k is approximately optimal with respect to L_{k-1}. In one-shot games, this model is conceptually equivalent to that of [3] restricted to $n = 3$; our key innovations are that we allow learning by replacing their $P(\text{lexicon})$ with $P(\text{lexicon}|\text{data})$, and provide a theoretical justification for how this learning can occur.

In the remainder of the paper, we apply the model described above to a set of one-shot pragmatic inference games that have been well-studied in linguistics [14, 15] and are addressed by previous one-shot models of pragmatic inference [9, 3]. These situations set the stage for simulations investigating how learning proceeds in iterated versions of such games, described in the following section. Results captured by our model and previous models are summarized in Table 1. In our simulations throughout, we somewhat arbitrarily set the recursion depth $n = 3$ (the minimal value that produces all the qualitative phenomena), $\lambda = 3$, and assume that all agents have shared priors on the lexicon and full knowledge of the cost function. Inference is via importance sampling from a Dirichlet prior over lexicons.

3 Pragmatic inference in one-shot games

3.1 Scalar implicature. Many sets of words in natural language form scales in which each term makes a successively stronger claim. "Some" and "all" form a scale of this type. While "I ate some

[4] An alternative model would have the speaker take the expectation over informativity, instead of the informativity of the expectation, which would correspond to slightly different utility functions. We adopt the current formulation for consistency with [3].

of the cookies" is compatible with the followup "in fact, I ate *all* of the cookies," the reverse is not true. "Might" and "must" are another example, as are "OK," "good," and "excellent." All of these scales allow for *scalar implicatures* [14]: the use of a less specific term pragmatically implies that the more specific term does not apply. So although "I ate some of the cookies" could in principle be compatible with eating ALL of them, the listener is lead to believe that SOME-BUT-NOT-ALL is the likely state of affairs. The recursive pragmatic reasoning portions of our model capture findings on scalar implicature in the same manner as previous models [3, 13].

3.2 Horn implicature. Consider a world which contains two words and two types of objects. One word is expensive to use, and one is cheap (call them "expensive" and "cheap" for short). One object type is common and one is rare; denote these COMMON and RARE. Intuitively, there are two possible communicative systems here: a good system where "cheap" referes to COMMON and "expensive" refers to RARE, and a bad system where the opposite holds. Obviously we would prefer to use the good system, but it has historically proven very difficult to derive this conclusion in a game theoretic setting, because both systems are stable equilibria: if our partner uses the bad system, then we would rather follow and communicate at some cost than switch to the good system and fail entirely [3].

Humans, however, unlike traditional game theoretic models, do make the inference that given two otherwise equivalent utterances, the costly utterance should have a rare or unusual meaning. We call this pattern *Horn implicature*, after [15]. For instance, "Lee got the car to stop" implies that Lee used an unusual method (e.g. not the brakes) because, had he used the brakes, the speaker would have chosen the simpler and shorter (less costly) expression, "Lee stopped the car" [15]. Surprisingly, Bergen et al. [3] show that the key to achieving this favorable result is ignorance. If a listener assigns equal probability to her partner using the good system or the bad system, then their best bet is to estimate $P_S(\text{word}|\text{object})$ as the average of $P_S(\text{word}|\text{object}, \text{good system})$ and $P_S(\text{word}|\text{object}, \text{bad system})$. These might seem to cancel out, but in fact they do not. In the good system, the utilities of the speaker's actions are relatively strongly separated compared to the bad system; therefore, a soft-max agent in the bad system has noiser behavior than in the good system, and the behavior in the good system dominates the average. Similar reasoning applies to an uncertain speaker. For example, in our model with a uniform prior over lexicons and $P_{\text{prior}}(\text{COMMON}) = 0.8$, cost("cheap") $= 0.5$, cost("expensive") $= 1.0$, the symmetry breaks in the appropriate way: Despite total ignorance about the conventional system, our modeled speakers prefer to use simple words for common referents ($P_S(\text{"cheap"}|\text{COMMON}) = 0.88, P_S(\text{"cheap"}|\text{RARE}) = 0.46$), and listeners show a similar bias ($P_L(\text{COMMON}|\text{"cheap"}) = 0.77, P_L(\text{COMMON}|\text{"expensive"}) = 0.65$).

This preference is weak; the critical point is that it exists at all, given the unbiased priors. We return to this in §5.2. [3] report a much stronger preference, which they accomplish by applying further layers of pragmatic recursion on top of these marginal distributions. On the one hand, this allows them to better fit their empirical data; on the other, it removes the possibility of learning the literal lexicon that underlies pragmatic inference – further recursion above the uncertainty means that it is only hypothetical agents who are ignorant, while the actual speaker and listener have no uncertainty about each other's generative process.

4 Pragmatics in learning from a knowledgable speaker

4.1 Learning literal meanings despite scalar implicatures. The acquisition of quantifiers like "some" provides a puzzle for most models of word learning: given that in many contexts, the word "some" is used to mean SOME-BUT-NOT-ALL, how do children learn that SOME-BUT-NOT-ALL is not in fact its literal meaning? Our model is able to take scalar implicatures into account when learning, and thus provide a potential solution, congruent with the observation that no known language in fact lexicalizes SOME-BUT-NOT-ALL [21].

Following the details of §3.1, we created a simulation in which the model's prior fixed the meaning of "all" to be a particular set ALL, but was ambiguous about whether "some" literally meant SOME-BUT-NOT-ALL (incorrect) or SOME-BUT-NOT-ALL OR ALL (correct). The model was then exposed to training situations in which "some" was used to refer to SOME-BUT-NOT-ALL. Despite this training, the model maintained substantial posterior probability on the correct hypothesis about the meaning of "some." Essentially, the model reasoned that although it had unambiguous evidence for "some" being used to refer to SOME-BUT-NOT-ALL, this was nonetheless consistent with a literal meaning of SOME-BUT-NOT-ALL OR ALL which had then been pragmatically strengthened.

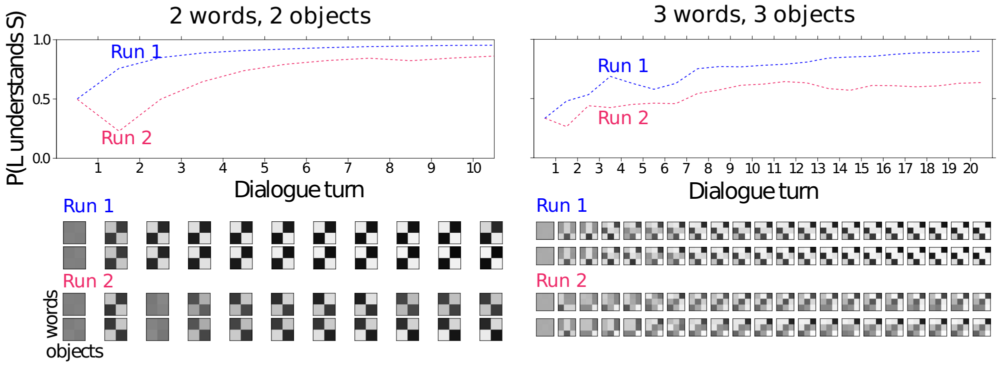

Figure 1: Simulations of two pragmatic agents playing a naming game. Each panel shows two representative simulation runs, with run 1 chosen to show strong convergence and run 2 chosen to show relatively weaker convergence. At each stage, S and L have different, possibly contradictory posteriors over the conventional, consensus lexicon. From these posteriors we derive the probability $P(L \text{ understands } S)$ (marginalizing over target objects and word choices), and also depict graphically S's model of the listener (top row), and L's actual model (bottom row).

Thus, a pragmatically-informed learner might be able to maintain the true meaning of SOME despite seemingly conflicting evidence.

4.2 Disambiguation using known words. Children, when presented with both a novel and a familiar object (e.g. an eggbeater and a ball), will treat a novel label (e.g. "dax") as referring to the novel object, for example by supplying the eggbeater when asked to "give me the dax" [22]. This phenomenon is sometimes referred to as "mutual exclusivity." Simple probabilistic word learning models can produce a similar pattern of findings [10], but all such models assume that learners retain the mapping between novel word and novel object demonstrated in the experimental situation. This observation is contradicted, however, by evidence that children often do not retain the mappings that are demonstrated by their inferences in the moment [16].

Our model provides an intriguing possible explanation of this finding: when simulating a single disambiguation situation, the model gives a substantial probability (e.g. 75%) that the speaker is referring to the novel object. Nevertheless, this inference is not accompanied by an increased belief that the novel word literally refers to this object. The learner's interpretation arises not from lexical mapping but instead from a variant of scalar implicature: the listener knows that the familiar word *does not* refer to the novel object—hence the novel word will be the best way to refer to the novel object, even if it literally could refer to either. Nevertheless, on repeated exposure to the same novel word, novel object situation, the learner does learn the mapping as part of the lexicon (congruent with other data on repeated training on disambiguation situations [4]).

5 Pragmatic reasoning in the absence of conventional meanings

5.1 Emergence of efficient communicative conventions. Experimental results suggest that communicators who start without a usable communication system are able to establish novel, consensus-based systems. For example, adults playing a communication game using only novel symbols with no conventional meaning will typically converge on a set of new conventions which allow them to accomplish their task [11]. Or in a less extreme example, communicators asked to refer to novel objects invent conventional names for them over the course of repeated interactions (e.g., "the ice skater" for an abstract figure vaguely resembling an ice skater, [7]). From a pure learning perspective this behavior is anomalous, however: Since both agents know perfectly well that there is no existing convention to discover, there is nothing to learn from the other's behavior. Furthermore, even if only one partner is producing the novel expressions, their behavior in these studies still becomes more regular (conventional) over time, which would seem to rule out a role for learning—even if there is some pattern in the expressions the speaker chooses to use, there is certainly nothing for the *speaker* to learn by observing these patterns, and thus their behavior should not change over time.

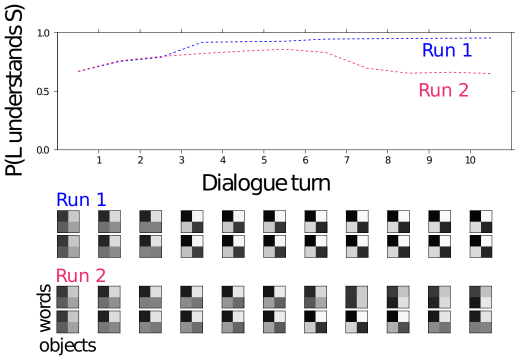

Figure 2: Example simulations showing the lexicalization of Horn implicatures. Plotting conventions are as above. In the first run, speaker and listener converge on a sparse and efficient communicative equilibrium, in which "cheap" means COMMON and "expensive" means RARE, while in the second they reach a sub-optimal equilibrium. As shown in Fig. 3, the former is more typical.

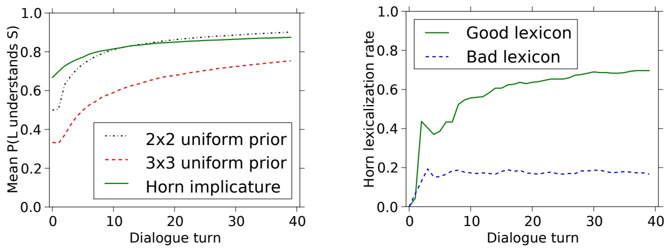

Figure 3: Averaged behavior over 300 dialogues as in Figs. 1 and 2. Left: Communicative success by game type and dialogue turn. Right: Proportion of dyads in the Horn implicature game (§5.2) who have converged on the 'good' or 'bad' lexicons and believe that these are literal meanings.

To model such phenomena, we imagine two agents playing the simple referential game introduced in § 2. On each turn the speaker is assigned a target object, utters some word referring to this object, the listener makes a guess at the object, and then, critically, the speaker observes the listener's guess and the listener receives feedback indicating the correct answer (i.e., the speaker's intended referent). Both agents then update their posterior over lexicons before proceeding to the next trial. As in [19, 7], the speaker and listener remain fixed in the same role throughout.

Fig. 1 shows the result of simulating several such games when both parties begin with a uniform prior over lexicons. Notice that: (a) agents' performance begins at chance, but quickly rises – a communicative system emerges where none previously existed; (b) they tend towards structured, sparse lexicons with a one-to-one correspondence between objects and words – these communicative systems are biased towards being useful and efficient; and (c) as the speaker and listener have entirely different data (the listener's interpretations and the speaker's intended referent, respectively), unlucky early guesses can lead them to believe in entirely contradictory lexicons—but they generally recover and converge. Each agent effectively uses their partner's behavior as a basis for forming weak beliefs about the underlying lexicon that they assume must exist. Since they then each act on these beliefs, and their partner uses the resulting actions to form new beliefs, they soon converge on using similar lexicons, and what started as a "superstition" becomes normatively correct. And unlike some previous models of emergence across multiple generations of agents [18, 25], this occurs within individual agents in a single dialogue.

5.2 Lexicalization and loss of Horn implicatures. A stronger example of how pragmatics can create biases in emerging lexicons can be observed by considering a version of this game played in the "cheap"/"expensive"/COMMON/RARE domain introduced in our discussion of Horn implicature (§3.2). Here, a uniform prior over lexicons, combined with pragmatic reasoning, causes each agent to start out weakly biased towards the associations "cheap" $\leftrightarrow$ COMMON, "expensive" $\leftrightarrow$ RARE. A fully rational listener who observed an uncertain speaker using words in this manner would therefore discount it as arising from this bias, and conclude that the speaker was, in fact, highly uncertain. Our convention-based listener, however, believes that speakers do know which convention is in use, and therefore tends to misinterpret this biased behavior as positive evidence that the 'good' system is in use. Similarly, convention-based speakers will wager that since on average they will succeed more often if listeners are using the 'good' system, they might as well try it. When they succeed, they take their success as evidence that the listener was in fact using the good system all along. As a result, dyads in this game end up converging onto a stable system at a rate far above chance, and preferentially onto the 'good' system (Figs. 2 and 3).

In the process, though, something interesting happens. In this model, Horn implicatures depend on uncertainty about literal meaning. As the agents gather more data, their uncertainty is reduced, and thus through the course of a dialogue, the implicature is replaced by a belief that "cheap" *literally* means COMMON (and did all along). To demonstrate this phenomenon, we queried each agent in each simulated dyad about how they would refer to or interpret each object and word, *if* the two objects were equally common, which cancels the Horn implicature. As shown in Fig. 3 (right), after 30 turns, in nearly 70% of dyads both S and L used the 'good' mapping even in this implicature-free case, while less than 20% used the 'bad' mapping (with the rest being inconsistent).

This points to a fundamental difference in how learning interacts with Horn versus scalar implicatures. Depending on the details of the input, it is possible for our convention-based agents to observe pragmatically strengthened uses of scalar terms (e.g., "some" used to refer to SOME-BUT-NOT-ALL), without becoming confused into thinking that "some" *literally* means SOME-BUT-NOT-ALL (§4.1). This occurs because scalar implicature depends only on recursive pragmatic reasoning (§2.1), which our convention-based agents' learning rules are able to model and correct for. But, while our agents are able to use Horn implicatures in their own behaviour (§ 3.2), this happens implicitly as a result of their uncertainty, and our agents do not model the uncertainty of other agents; thus, when they observe other agents using Horn implicatures, they cannot interpret this behavior as arising from an implicature. Instead, they take it as reflecting the actual literal meaning. And this result isn't just a technical limitation of our implementation, but is intrinsic to our convention-based approach to combining pragmatics and learning: in our system, the only thing that makes word learning possible at all is each agent's assumption that other agents are better informed; otherwise, other agents' behavior would not provide any useful data for learning. Our model therefore makes the interesting prediction that all else being equal, uncertainty-based implicatures should over time be more prone to lexicalizing and becoming part of literal meaning than recursion-based implicatures are.

6 Conclusion

Language learners and language users must consider word meanings both within and across contexts. A critical part of this process is reasoning pragmatically about agents' goals in individual situations. In the current work we treat agents communicating with one another as assuming that there is a shared conventional lexicon which they both rely on, but with differing degrees of knowledge. They then reason recursively about how this lexicon should be used to convey particular meanings in context. These assumptions allow us to create a model that unifies two previously separate strands of modeling work on language usage and acquisition and account for a variety of new phenomena. In particular, we consider new explanations of disambiguation in early word learning and the acquisition of quantifiers, and demonstrate that our model is capable of developing novel and efficient communicative systems through iterated learning within the context of a single simulated conversation.

Our assumptions produce a tractable model, but because they deviate from pure rationality, they must introduce biases, of which we identify two: a tendency for pragmatic speakers and listeners to accentuate useful, sparse patterns in their communicative systems (§5.1), and for short, 'low cost' expressions to be assigned to common objects (§5.2). Strikingly, both of these biases systematically drive the overall communicative system towards greater global efficiency. In the long term, these processes should leave their mark on the structure of the language itself, which may contribute to explaining how languages become optimized for effective communication [26, 24].

More generally, understanding the interaction between pragmatics and learning is a precondition to developing a unified understanding of human language. Our work here takes a first step towards joining disparate strands of research that have treated language acquisition and language use as distinct.

Acknowledgments

This work was supported in part by the European Commission through the EU Cognitive Systems Project Xperience (FP7-ICT-270273), the John S. McDonnell Foundation, and ONR grant N000141310287.

References

[1] D.A. Baldwin. Early referential understanding: Infants' ability to recognize referential acts for what they are. *Developmental Psychology*, 29(5):832–843, 1993.

[2] D. Barner, N. Brooks, and A. Bale. Accessing the unsaid: The role of scalar alternatives in childrens pragmatic inference. *Cognition*, 118(1):84, 2011.

[3] L. Bergen, N. D. Goodman, and R. Levy. That's what she (could have) said: How alternative utterances affect language use. In *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, 2012.

[4] R.A.H. Bion, A. Borovsky, and A. Fernald. Fast mapping, slow learning: Disambiguation of novel word–object mappings in relation to vocabulary learning at 18, 24, and 30months. *Cognition*, 2012.

[5] C. F. Camerer, T.-H. Ho, and J.-K. Chong. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, 119(3):861–898, 2004.

[6] E.V. Clark. On the logic of contrast. *Journal of Child Language*, 15:317–335, 1988.

[7] Herbert H Clark and Deanna Wilkes-Gibbs. Referring as a collaborative process. *Cognition*, 22(1):1–39, 1986.

[8] R. Dale and E. Reiter. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263, 1995.

[9] M. C. Frank and N. D. Goodman. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998, 2012.

[10] M. C. Frank, N. D. Goodman, and J. B. Tenenbaum. Using speakers' referential intentions to model early cross-situational word learning. *Psychological Science*, 20:578–585, 2009.

[11] B. Galantucci. An experimental study of the emergence of human communication systems. *Cognitive science*, 29(5):737–767, 2005.

[12] D. Golland, P. Liang, and D. Klein. A game-theoretic approach to generating spatial descriptions. In *Proceedings of EMNLP 2010*, pages 410–419. Association for Computational Linguistics, 2010.

[13] Noah D. Goodman and Andreas Stuhlmüller. Knowledge and implicature: Modeling language understanding as social cognition. *Topics in Cognitive Science*, 5:173–184, 2013.

[14] H.P. Grice. Logic and conversation. *Syntax and Semantics*, 3:41–58, 1975.

[15] L. Horn. Toward a new taxonomy for pragmatic inference: Q-based and r-based implicature. In *Meaning, form, and use in context*, volume 42. Washington: Georgetown University Press, 1984.

[16] J. S. Horst and L. K. Samuelson. Fast mapping but poor retention by 24-month-old infants. *Infancy*, 13(2):128–157, 2008.

[17] G. Kachergis, C. Yu, and R. M. Shiffrin. An associative model of adaptive inference for learning word–referent mappings. *Psychonomic Bulletin & Review*, 19(2):317–324, April 2012.

[18] S. Kirby, H. Cornish, and K. Smith. Cumulative cultural evolution in the laboratory: An experimental approach to the origins of structure in human language. *Proceedings of the National Academy of Sciences*, 105(31):10681–10686, 2008.

[19] R. M. Krauss and S. Weinheimer. Changes in reference phrases as a function of frequency of usage in social interaction: A preliminary study. *Psychonomic Science*, 1964.

[20] T. Kwiatkowski, S. Goldwater, L. Zettlemoyer, and M. Steedman. A probabilistic model of syntactic and semantic acquisition from child-directed utterances and their meanings. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 234–244, 2012.

[21] S.C. Levinson. *Presumptive meanings: The theory of generalized conversational implicature*. MIT Press, 2000.

[22] E. M. Markman and G. F. Wachtel. Children's use of mutual exclusivity to constrain the meanings of words. *Cognitive Psychology*, 20:121–157, 1988.

[23] A. Papafragou and J. Musolino. Scalar implicatures: Experiments at the semantics-pragmatics interface. *Cognition*, 86(3):253–282, 2003.

[24] S. T. Piantadosi, H. Tily, and E. Gibson. Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences*, 108(9):3526 –3529, 2011.

[25] R. van Rooy. Evolution of conventional meaning and conversational principles. *Synthese*, 139(2):331–366, 2004.

[26] G. Zipf. *The Psychobiology of Language*. Routledge, London, 1936.