Summer Internship Report

for

CIS 9282 Independent Study

Submitted to Dr. Pei Wang

By

Ozkan Kilic

TUID 915085380-1

Department of Computer Science

Temple University

Contents

INTRODUCTION

The internship at AGI Innovations Inc was mainly about researching AGI systems, natural language parsing and automated text summarization. My research results are given below.

AGI SYSTEMS

For more than 60 years, AI researchers have been working on creating intelligent machines or programs that can perform intellectual tasks that a human being can. These tasks can vary a lot from playing chess, counting animals in a picture, performing commands given in a natural language to making jokes, having a conversation with a human, producing novel inventions and art. Weak-AI aims to have human-like intelligence for some problems while the ultimate goal of strong-AI is to have systems that have their own minds.

Searle's (1980) misleading Chinese Room Argument[1] was one of early and famous reactions to strong-AI. Such reactions, and also the complexity of strong-AI, have caused many researchers to work on solving various sub-problems with weak-AI approaches, which have resulted in uncombinable systems. On the contrary, humans can learn different problem solving approaches or algorithms and even combine and alter them to come up with novel ones. Furthermore, resent studies have claimed that everybody may be born as synesthetic, which means humans' knowledge representations are initially so flexible that they can represent and combine different data modalities (shapes, colors, sounds, meanings) within the same ontology when they are infants. Though human knowledge representations may be specialized as we grow up, our analogy making and even creativity skills seem to have relations with our minds' ability to unify different data modalities.

Research on general purpose systems returned stronger in 2000s. This time, the proponents of strong-AI have been better equipped with knowledge from other disciplines, such as neurology, cognitive science, linguistics, psychology, philosophy, and such. I think that the progress of Strong-AI is very central to the mind-body problem stated by Descartes because it may prove that there is no dualism and mind is a product of substance. I am for strong-AI but not soon because of two reasons: First, strong-AI research heavily depends on relatively young fields like neurology and cognitive science. Therefore, it requires updates and knowledge transfers from these young fields. Second, there is a partially circular dependency:

[1] http://plato.stanford.edu/entries/chinese-room/

strong-AI researchers use their minds to understand mind and create an artificial mind; and then, understanding mind will help the researchers to create an artificial mind. These make strong-AI field improve incrementally and slowly, just like human mind's life-long learning ability.

Thrun (1997) used the term "lifelong machine learning" for a system that is able to acquire knowledge through learning, retain or consolidate such knowledge, and use it for inductive transfer when learning new tasks. An AGI system needs to be able to transfer its experience to other domains in order to be a lifelong learning system. Thus, knowledge acquisition, representation, reasoning and knowledge transfer are the key components of an intelligent system. Most referred and ongoing AI systems with NLP or related features are reviewed in the following sections:

1 SNePS

Stuart Shapiro's SNePS[2] currently has the most developed NLP feature. His research group released the version 2.8. They have been working on a new version, 3.0. The SNePS agent was written in LISP. Every proposition is parsed into a FOL-like representation and then they are represented as nodes in a network. Relations among propositions are represented by arcs in the network.
For example, "*Any robot that talks is intelligent"* is represented as in Figure 1.

[2] http://www.cse.buffalo.edu/sneps/

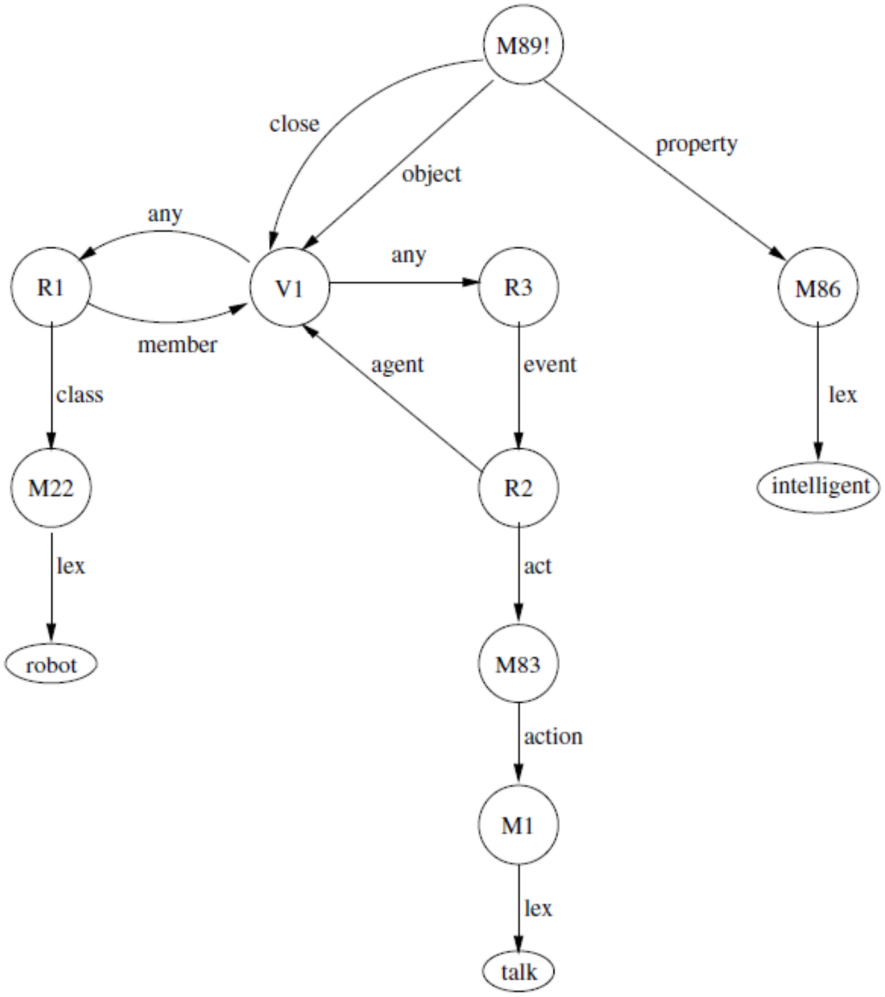


Figure 1 A propositional semantic network in SNePS

SNePS uses many redundant nodes (M89!, M83, M1) to represent semantics of agent, time, categories, action, act, event, and proposition-denoting terms. SNePS also has to make use of a lexicon to parse sentences. The lexicon is not automatically incremental. A user needs to update the lexicon. A sample lexicon is of the form:

```
("a" ((ctgy . art)(definite . nil)))
("the" ((ctgy . art)(definite . t)))
("Computer Science" ((ctgy . npr)))
("John" ((ctgy . npr)))
("Mary" ((ctgy . npr)))
("computer" ((ctgy . n)))
("Computer" ((ctgy . multi-start) (multi-rest . ("Science")))
((ctgy . n)(root . "computer")))
("dog" ((ctgy . n)))
```

```
("man" ((ctgy . n)(plur . "men")))
("men" ((ctgy . n)(root . "man")(num . plur)))
("woman" ((ctgy . n)(plur . "women")))
("women" ((ctgy . n)(root . "woman")(num . plur)))
("saw" ((ctgy . n))
((ctgy . v)(root . "see")(tense . past)))
("believe" ((ctgy . v)(stative . t)))
("bit" ((ctgy . v)(root . "bite")(tense . past)))
("bite" ((ctgy . v)(num . plur)(past . "bit")))
("like" ((ctgy . v)(num . plur)))
("see" ((ctgy . v)(past . "saw")))
("sleep" ((ctgy . v)(past . "slept")))
("slept" ((ctgy . v)(root . "sleep")(tense . past)))
("study" ((ctgy . v)))
("use" ((ctgy . v)))
("who" ((ctgy . wh)))
("what" ((ctgy . wh)))
```

For example, "*Mary believes that John likes the dog*" is parsed as

```
(S (MOOD DECL)
(NP (DEFINITE T) (NPR "Mary"))
(VP (V "believe")
(NP (DEFINITE T)
(S (MOOD DECL)
(NP (DEFINITE T) (NPR "John"))
(VP (V "like")
(NP (DEFINITE T) (N "dog")))))))
```

The queries are also parsed as in, "*Who likes a dog?*"

```
(S (MOOD QUESTION)
(NP ?)
(VP (V "like")
(NP (DEFINITE NIL) (N "dog"))))
```

And a reasoning mechanism uses the network to satisfy the question mark in the parsed query (or the empty node in the upcoming version 3.0). The network based representation provides flexibility: The SNePS agent may have different networks depending on the previous experience. Arcs can be used as flexible relations among propositions. However, everything depends on the lexicon, and the system cannot build up its own lexicon. Another problematic issue is the belief revision in SNePS. Assume that

the following propositions are input to the system, “All pirates are uneducated”, “John is a pirate”, “John is educated”. When the system comes across such a contradiction, it reports the contradiction, and then, either the system asks users to revise the contradicting propositions, or an automated belief revision method recently added to the system runs to determine *the least used proposition* and deletes it. However, this is not what a human being would probably do. Since the lexicon is handmade, it is easy to represent degrees quantification or levels or certainty. A better solution would be, revise the propositions to either one of these:

- John may be a pirate.
- John might be educated.
- Some pirates are uneducated.

Rapaport (2013) states that such semantic networks built online by SNePS agents have the ability to represent Meinongian Semantics (i.e., semantics of impossible objects in real life) because every SNePS term represents an intensional (mental) entity. For example, “unicorns have big horns”, and “A round square is round” are Meinongian type propositions. Semantic networks build by an agent corresponds to a human’s mental repository, and the tree-like structure makes reasoning faster. However, SNePS framework requires use of an ANSI Common LISP interpreter, which makes it less user-friendly environment.

2 LIDA

LIDA[3] is a Java based cognitive framework using Global Workspace Theory[4]. The architecture ties in well with both neuroscience and cognitive psychology, but it deals most thoroughly with “lower level” aspects of intelligence, handling more advanced aspects like language and reasoning only somewhat sketchily. The LIDA agent can run in a virtual world, sense the world (colors, numbers, obstacles), and achieves some tasks according to *cognitive cycles* as shown in Figure 2. The LIDA framework mimics three types of learning: Perceptual, episodic and procedural. Perceptual learning concerns learning of new objects, categories, relations, etc, represented as nodes in the perceptual memory. Episodic learning, on the other hand, involves learning to memorize specific events (i.e., the what, where, and when). Finally, procedural learning concerns learning of new actions and action sequences with which to accomplish new tasks (e.g., “turn left, walk, grab, walk, turn right” might be a successful solution for the command “fetch my cup”). This architecture may explain many features of mind; however, it remains to be far away from

[3] http://ccrg.cs.memphis.edu/

[4] GWT resembles the concept of Working Memory, and is proposed to correspond to a "momentarily active, subjectively experienced" event in working memory. It closely related to conscious experiences.

understanding language, vision, and such. The developers claim that LIDA is the best implementation of "consciousness" because its agent is aware of itself, its experience, time and space. Despite dealing with lower level abilities, LIDA is a good at implementation of different memory types, and dealing with self-awareness, grounded knowledge and reasoning under uncertainty.

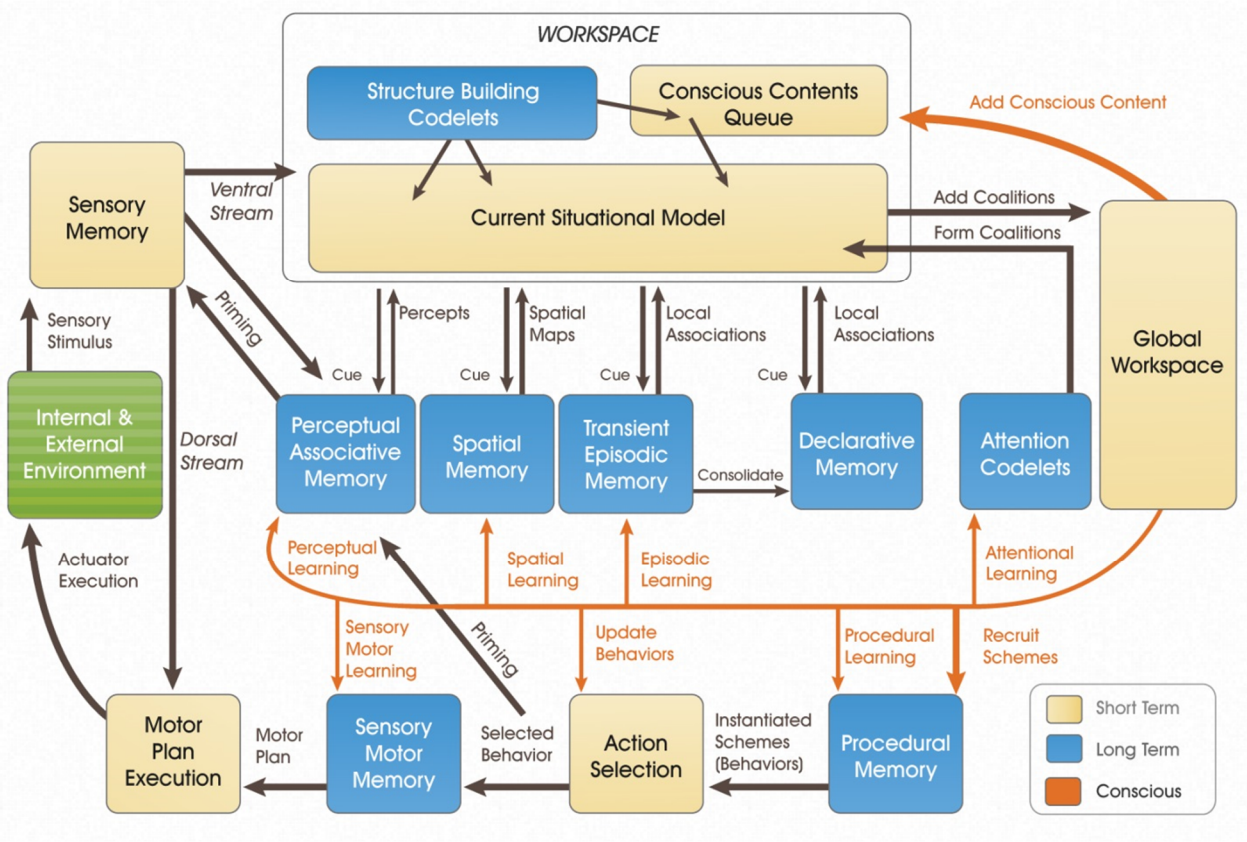


Figure 2 LIDA's Cognitive Cycle

3 OpenCog (CogPrime, OpenCogPrime)

Ben Goertzel and his collaborators have been working on OpenCog framework[5], which also uses the concept of cognitive cycle as in LIDA. It is a developed and detailed framework. The knowledge units of the system are called atoms, which can be nodes or links in a network representation. Atoms have truth and attention (which sets priority of the node) values. OpenCog has a probabilistic reasoning ability of the network representation, which is actually a hypergraph. Nodes can be conceptual, perceptual, procedural, psyche (goal and feeling of the agent) nodes while links can be logical, member, associative, execution, action, etc. It has a complex architecture as shown in Figure 3.

[5] http://opencog.org/

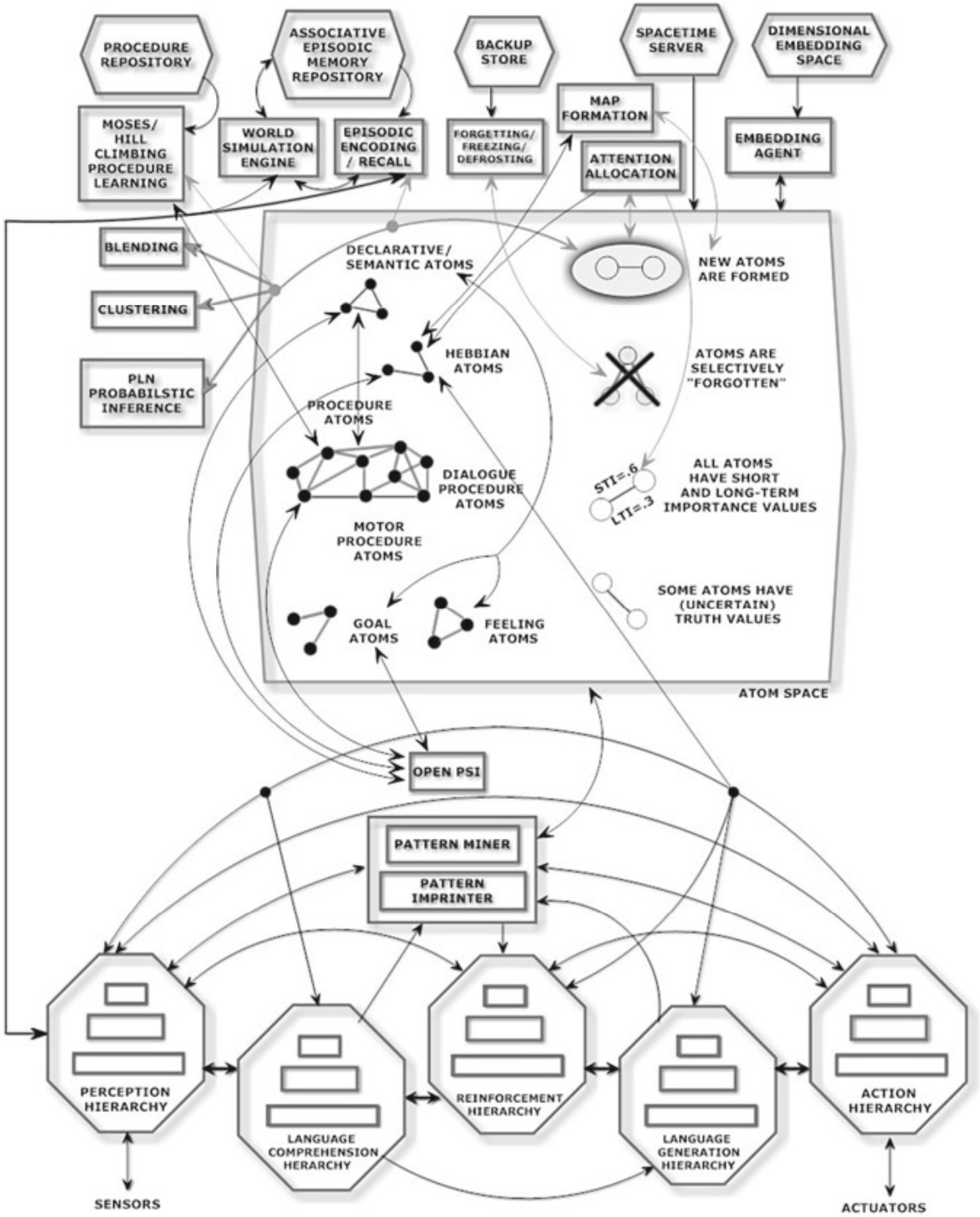


Figure 3 OpenCog architecture

OpenCog uses hand-coded rules for Natural Language Processing, which are part of a module called RelEx. The semantic parses are then mapped to *atom* structures with links. The system also has a language generation module: SegSim, which is also rule-based. Here is how it works:

1. The NL generation system stores a large set of pairs of the form (semantic structure, syntactic/morphological realization).
2. When it is given a new semantic structure to express, it first breaks this semantic structure into natural parts, using a set of simple syntactic-semantic rules.
3. For each of these parts, it then matches the parts against its memory to find relevant pairs and uses these pairs to generate a set of syntactic realizations (which may be sentences or sentence fragments).
4. If the matching has failed, then
    a. It returns to Step 2 and carries out the breakdown into parts again. But if this has happened too many times, then
    b. It recourses to a different algorithm
5. If the above step generated multiple fragments, they are pieced together, and a certain rating function is used to judge if this has been done adequately (using criteria of grammaticality and expected comprehensibility, among others). If this fails, then Step 3 is tried again on one or more of the parts; or Step 2 is tried again.
6. Finally, a "cleanup" phase is conducted, in which correct morphological forms, articles and function words are inserted.

This method works for simple cases but would probably fail frequently (according to some examples given in Goertzel et al. 2014). Moreover, it is not explained clear enough.

4 NARS

NARS[6] is a reasoning system based on a language for knowledge representation, an experience-grounded semantics of the language, a set of inference rules, a memory structure, and a control mechanism. The non-axiomatic logic is used for adaptation with insufficient knowledge and resources, operating on patterns that have the "truth-value" evaluated according to the system's "experience" with using these patterns. This approach allows for emergence of experience-grounded semantics, and inferences defined on judgments. Currently NARS is too general as shown in Figure 4 and can handle simple problems.

[6] https://sites.google.com/site/narswang/

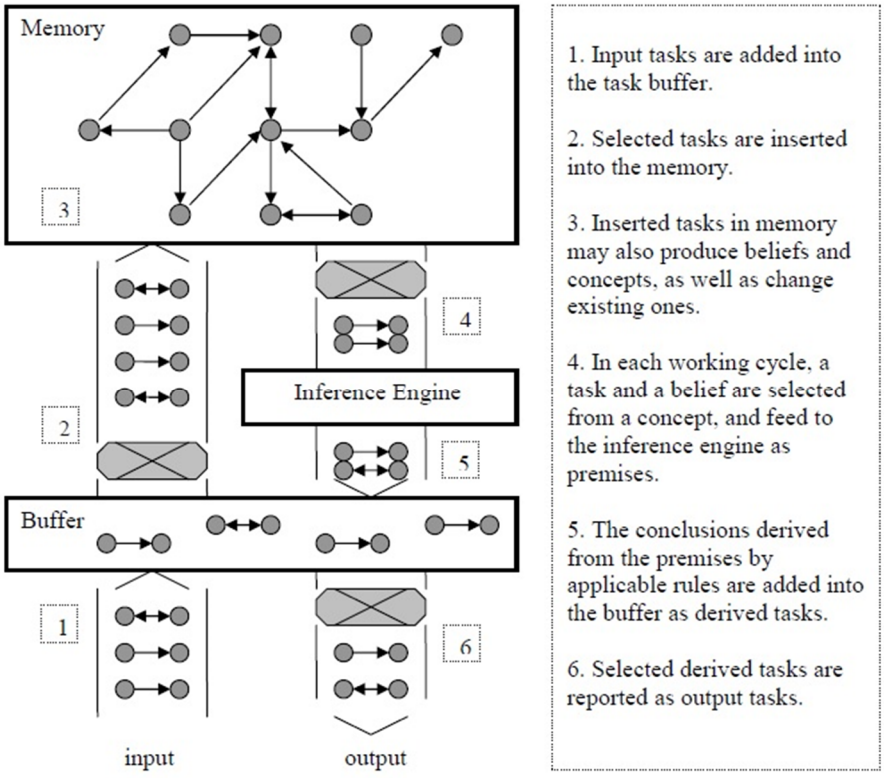


Figure 4 NARS Architecture

Yet, NARS automatically provides answers for reasoning under uncertainty, belief revision, contradiction handling, U-shaped language development, and such due to its probabilistic and truth value-dependent reasoning. A simple example for U-shaped learning of English passive voice in Narsese is given below.

// initial knowledge is input to the system.

<(*, {go}, {gone}) --> passive>.

<(*, {brake}, {broken}) --> passive>.

<(*, {add}, {added}) --> passive>.

<(*, {ask}, {asked}) --> passive>.

<(*, {need}, {needed}) --> passive>.

// state-1 response

<(*, {go}, {gone}) --> passive>. %1;0.9%

// additional information is input to the system: Append-ed operation is defined as a relation

<(*, {add}, {added}) --> append-ed>.

< (*, {ask}, {asked}) --> append-ed>.

< (*, {need}, {needed}) --> append-ed>.

< (*, {go}, {goed}) --> append-ed>.

// inductive derivation by the system

<<$x --> passive> <=> <$x --> append-ed>>.

// state-2 response

<(*, {go}, {goed}) --> passive>. %1;0.9%

//counter evidence is input to the system

(--, <(*, {go}, {gone}) --> append-ed>).

(--, < (*, {brake}, {broken}) --> append-ed>).

// *question asked to the system: What is the passive form of go?*

< (*, {go}, ?x) --> passive>.

//depending on the number of counter evidences, possible outputs by the system

<(*, {go}, {gone}) --> passive>. %0.8;0.7%

<(*, {go}, {goed}) --> passive>. %0.5;0.49%

5 ACT-R

ACT-R[7] is a theoretical framework for emulating and understanding human cognition inspired by human brain. It aims at building a system performing the full range of human cognitive tasks. ACT-R aims to describe the mechanisms underlying perception, thinking, and action. It is composed of a set of perceptual-motor modules, different memory modules, buffers, and a pattern matcher. The perceptual-motor modules basically serve as an interface between the system and the world. ACT-R has declarative memory (for factual information) and procedural memory (for procedures). ACT-R uses symbolic constructs (i.e., chunks or productions) created to describe the results of a complex operation. Such chunks may be available without re-computing the next time a similar task occurs (similar to memoization). ACT-R is a general purpose framework for psychological operations abstract reasoning, but creativity and transfer learning are still missing. A probabilistic natural language parsing model was implemented in ACT-R. Declarative memory was used as a lexicon while production rules (or grammar) were saved in procedural memory. Yet, ACT-R does not have a built-in NLP module. It provides reasoning and processing power given lexicon and grammar. The example below is taken from Lewis and Vasishth (2005):

[7] http://act-r.psy.cmu.edu/

PM content:

IP→DP VP

DP→det NP

NP→N

VP→V DP

DM content:

IP3

cat	IP
num	sing
spec	DP3
comp	VP7
tense	past
finite	finite

DP3

cat	DP
num	sing
head	*the*
comp	NP6

NP6

cat	NP
case	nom
num	sing
head	*writer*

VP7

cat	VP
num	sing-plural
tense	past
head	*surprised*
comp	DP9

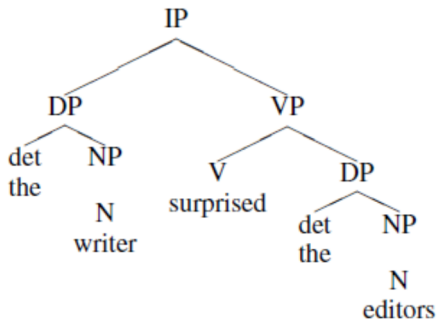
DP9

cat	DP
num	plural
head	*the*
comp	NP14

NP14

cat	NP
case	acc
num	plural
head	*editors*

The parse tree:



The advantage of using ACT-R for NLP is that intermediate phrasal constructions are automatically kept as chunks for later use, which makes parsing fast, incremental and even probabilistic.

6 SOAR

SOAR[8] is a rule-based intelligent system. SOAR stores its knowledge in form of production rules, arranged in terms of operators that act in the problem space, which is composed of a set of states that

[8] http://sitemaker.umich.edu/soar/home

represent the task at hand. It has production memory (like long-term memory) and working memory (like short-term memory). SOAR transforms every input into representations (symbols) and works on these representations. It has a production rule system (if…then rules) to modify representations or make decisions. SOAR assumes that intelligent action can be formulated as the selection and application of operators to a state, to achieve some goal.

SOAR is not strong in areas such as episodic memory, creativity, handling uncertain knowledge, and reinforcement learning. Similar to ACT-R, SOAR is originally not designed for NLP but there are some implementations. For example, Mohan et al. (2013) used SOAR to model situated language comprehension to make an agent perform commands, like "*move the large red cylinder to right of the blue triangle*". They indexed and defined all objects, properties and actions before the agent performed actions. Moreover, some independent researchers worked on NL-SOAR which integrates English grammar with WordNet data but it is no longer an active study.

ROBOCUP DATA

The Robocup dataset[9] describes robotic soccer events. The dataset was collected for the purpose of constructing semantic parsers from ambiguous supervision and consists of both "noisy" and gold labeled data. The noisy dataset was constructed by temporally aligning a stream of soccer events occurring during a robotic soccer match with human commentary describing the game. This dataset consists of pairs (x, {y0 , yk }), x is a sentence and {y0 , yk } is a set of events (logical formulas). One of these events is assumed to correspond to the comment, however this is not guaranteed. The gold labeled labeled data consists of pairs (x, y). The data was collected from four Robocup games. Several researchers used this data set to train artificial agents to win a virtual soccer game. For example, “pink10 passed the ball to pink11” has the logical form *pass(pink10, pink11)* and the action of *pass* is learned by observing the ball. The other actions can be

kick (PurplePlayer10)
pass (PurplePlayer10 , PurplePlayer11)
kick (PurplePlayer11)
pass (PurplePlayer11 , PurplePlayer10)
steal (PinkPlayer3)
turnover (PurplePlayer10 , PinkPlayer3)
kick (PinkPlayer3)
playmode (free_kick_r)

The agents are expected to observe the order of actions and opponents’ positions on the game field, and learn the successful set of strategy to win the game. Probabilistic models achieve the best performance so far. Although the virtual environment and set of semantic representations are very limited, there are very successful results for how to learn and act according to grounded meanings.

[9] http://www.cs.utexas.edu/users/ml/clamp/sportscasting/

PARSING

Parsing refers to the mental operations that establish relationships between words in sentences. Human communication requires parsing sentences to infer meanings. Thus, an artificially intelligent system with natural language processing ability is ultimately expected to analyze strings of words into a parse tree showing syntactic and/or semantic relation among the words. Thus, it is worthy to review the human parsing ability to enable an AGI system to mimic it.

The Cognitive Revolution started in 1950s and 1960s as a reaction against behaviorist descriptions of language comprehension and production. Lead by Noam Chomsky, generative linguists proposed that words in sentences should be treated as particular categories, such as verb, noun, and preposition. When sentences are processed, speakers and listeners build syntactic structures using this categorical information. Thus, the linguists placed syntax at the core of linguistics. However, psycholinguistic evidences found in the late 1970s were inconsistent with this view. The evidences showed that people make use of pragmatics or sometimes individual word meanings to cope with ambiguities in sentences. Thus, syntax does not always precede semantics. Consider the following sentences,

(1) The doctor examined …
(2) The evidence examined …

When people are asked to complete these sentences, they always add an object to (1) (for example, “the doctor examined the patient”) and turn (2) into a relative clause (for example, “The evidence examined by the detective seems reliable”). Such examples highlight constraint-based parsing theories in which words impose constraints on each other and semantic characteristics of the words in utterances may influence parsing of the utterances. Ferreira 2003) asked some college students to interpret the sentence (3).

(3) The mouse was eaten by the cheese.

Most of them ignored the syntactic information and interpreted (3) as “The mouse ate the cheese”. There also neurological evidences (Cappa, 2012) showing that semantic knowledge is distributed in human brain, and it is not always preceded by an activation of the Broca’s region, where syntax resides. Besides being more semantic, human level parsing is incremental (which leads to local ambiguities), both bottom-

up and top-down down, susceptible to "garden path[10]" sentences, has early commitment (for example, if a potential structure is found, it is added to a parse tree and revised later if needed).

Semantic Parsing in NLP

Semantic parsing can be defined as representing the meaning of a sentence in some formal knowledge representation language (or logical form) that supports automated inference. A semantic parser initially needs a formal language. Commonly used formal representations are lambda calculus (Carpenter, 1997), first order logic, natural logics (Moss, 2009), relational algebra (Liang et al., 2011), graphical models (Tellex et al., 2011), and such.

Here is an example of using lambda calculus with combinatory categorial grammar (CCG) (Steedman, 2000) to deduce *fun(Parsing*) from the sentence, "Parsing is fun" in a bottom up manner (4).

(4)

Parsing	*is*	*fun*	
NP	S\NP/ADJ	ADJ	
Parsing	*λf.λx.f(x)*	*λx.fun(x)*	>
	S\NP		
	λx.fun(x)		<
	S		
	fun(Parsing)		

To achieve such a mapping, one initially needs formal representations and categorial tags for each word. Furthermore, the cost of mapping increases in case of ambiguities. When a full mapping of a text is achieved, it is possible to use it for various purposes such as information extraction and question answering. For example, questions can be represented in this form as well (5). In this case, a system should find the best *x* satisfying the logical representation.

(5)

What is fun?	*λx.fun(x)*
What is the population of Seattle?	*λx.population(Seattle, x)*
How many people live in Seattle?	*count(λx.person(x) ∧ live(x, Seattle))*

[10] Garden path sentences are sentences which are grammatical, but which naïve subjects fail to parse. For example, "The doctor sent for the patient died." is a garden path sentence while "The flowers sent for the patient died." is not.

Besides choosing the knowledge representation formalism, a semantic parser needs to deal with two other issues: parsing method choice and learning to parse. For example, Wong and Mooney (2006) introduced a statistical parsing tool WASP[11] which produces logical representations in a meaning representation language CLANG as in (6). They made use of manually translated sentences with *synchronous context-free grammar*.

(6) If our player 4 has the ball, then our player 6 should stay in the left side of our half.

(7) ((bowner our {4})
(do our {6} (pos (left (half our)))))

Clarke et al. (2010) used a non-statistical supervised model with integer linear programming to optimize semantic constraints exposed by constituents in a sentence. Their model depends on feedbacks from human non-experts. Recently, Liang et al. (2011) have introduced *dependency-based compositional semantics* which starts with a fixed set of lexical triggers and uses trees to represent formal semantics. They use *join*, *aggregate*, *execute*, *extract*, *quantify*, and *compare* operators on nodes of trees to construct a semantic representation.

Learning aspect of a semantic parser is related to the degree of available supervision. For example, Zettlemoyer and Collins (2005) used an annotated set of straining sentences and probabilistic CCG to learn supervised semantic parsing. Some other researchers used question-answer pairs (Clarke et al., 2010), instruction-demonstration pairs observed from a virtual world (Chen and Mooney, 2011), conversation logs via dialog systems (Artz and Zettlemoyer, 2011), and visual sensors (Matuszek et al., 2012) to approximate the best parsing.

Traditionally, semantic parsing can be divided into two categories: logical and statistical parsing. Logical parsing includes using hand-written rules, tables and dictionaries to parse texts while statistical ones try to learn parsing through machine learning algorithms. Both sides have its own advantages and disadvantages. Currently, it seems that logical and statistical techniques tend to merge especially to cope with ambiguities.

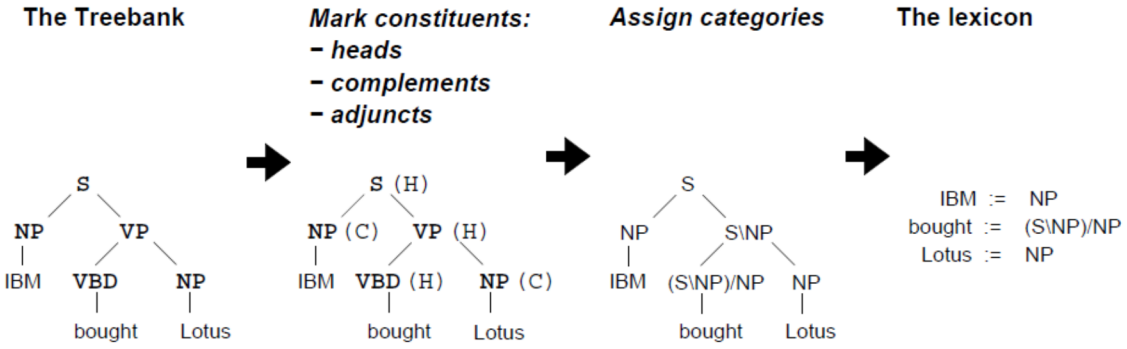
[11] The WASP is available at http://www.cs.utexas.edu/~ml/wasp/

Leading Research Groups in Semantic Parsing

There are various research groups working on improving semantic parsing. The NLP group at Stanford University, Mark Steedman's CCG team at Edinburgh University, Luke Zettlemoyer's team at University of Washington, and Raymond Mooney's AI Lab at University of Texas are the leading research groups in this area.

1. Stanford NLP Group: Their tool is called SEMPRE: Semantic parsing with execution[12] (Berant and Liang, 2014). It uses question-answer pairs and FreeBase[13] knowledge base by Google to train their parser. Thus, the SEMORE does not require annotated training set. Their parser produces output in a logical language called Lambda Dependency-Based Compositional Semantics (Liang, 2013). They use the same language for queries as well.
2. Edinburgh CCG Group: Mark Steedman introduced CCG. His group uses CCGBank[14], which is a knowledge base with CCG derivations achieved via supervised learning as in (7). There are various parsers based on CCG formalism[15]. Currently, the team is working on applying CCG to speech data with annotated intonation.

   (7)



3. Univ. Washington: Zettlemoyer and his team have been developing the University of Washington Semantic Parsing Framework[16] (Artzi and Zettlemoyer, 2013). They work closely to the Edinburgh CCG Group. They use probabilistic CCG and unification based generative lexicon.

[12] The SEMPRE is available at http://nlp.stanford.edu/software/sempre/
[13] The FreeBase is available at https://developers.google.com/freebase/
[14] The CCGBank is available at http://groups.inf.ed.ac.uk/ccg/ccgbank.html
[15] A statistical CCG parser is available at http://groups.inf.ed.ac.uk/ccg/software.html
[16] The UW SPF is available at https://bitbucket.org/yoavartzi/spf/

Their model is mildly supervised and statistical. They also claim that their model can offer some resolutions for grounded meaning problem[17] through a virtual environment.

4. AI Lab at Univ. Texas: Raymond Mooney and his team have been working on logical semantic parsers via distributional semantics (Beltagy et al., 2014). Distributional semantics is a statistical technique that represents the meaning of words and phrases as distributions over context words (Turney and Pantel, 2010). A distributional semantics is relatively easy to build from a large corpus of raw text, and provides the wide coverage that formal ontologies lack. Their parser is not restricted to a predefined ontology. However, they do not report any success rate. This team is also interested in grounded learning problem but limited to action domain. They use ambiguous supervision which contains ambiguous actions dictated by commentators.

Current and Future

Currently, logical approaches in semantic parsing rely on techniques from proof theory and model-theoretic semantics, they have strong ties to linguistic semantics, and they are concerned primarily with some problems such as inference, ambiguity, vagueness, and compositional interpretation of full syntactic parses. In contrast, statistical approaches derive their tools from algorithms and optimization, and they tend to focus on word meanings, broad notions of semantic content and grounded meanings. For example, statistical methods seem to cope well with the words whose meaning heavily depend on context while logical methods are faster at dealing with ambiguities. Currently, tools using probabilistic CCG with lambda calculus and supervised training have the best success rate in parsing. Some researchers focusing on logical parsing via large knowledge bases (e.g., Google's FreeBase) also report high success rates.

The active research groups seem to combine logical and statistical approaches to cope with the problems as suggested by Liang and Potts (2014). Unfortunately, it is hard to combine different approaches because there are many domain-specific applications with different ontologies. Besides combining approaches, some researchers work on including perceptual real-life information, such as visual and auditory one. Therefore, selection of a flexible representation language is crucial: The language should allow temporal, ambiguous, probabilistic, context dependent, sensory and compositional representations of words, phrases and sentences.

[17] The grounded meaning problem states that some words or phrases gain their meanings in their contexts other than simple unification. In other words, word meanings are anchored to the real world.

AUTOMATED TEXT SUMMARIZATION

Text summarization is the process of distilling the most important information from a source or sources to produce an abridged version for a particular user or users and task or tasks. Information overloads (or details) require us to do automated text summarization down to at most 25% of the original document. Medicine, law and finance are popular application areas of this relatively immature field. Almost all applications have *input*, *purpose* and *output* determining the summarization procedure. The procedure is usually:

- *analyzing* the input text in order to obtain a text representation,
- *transforming* obtained text representation into a summary representation
- *synthesizing* summary representation and generating an appropriate summary.

Input aspects:

Structure of the document, domain, specialization level, restriction on the language, scale, media, genre, unit, and language.

Purpose aspects:

Situation, audience, and usage.

Output aspects:

Content, format, style, production process and length.

There are different summarization approaches. Here are some frequently used approaches:

- Term frequency: Using frequent words (except functional words) as a summary.
- Location: The sentences containing the terms that occur in the title should be in the summary.
- Queue words and phrases: The sentences with “in summary, in conclusion, ultimately” should be in the summary.
- Similarity: Employ a semantic similarity method (Latent semantic analysis or a knowledge base) to rephrase your summary.
- Cohesion: If you have a set of keywords, the words co-occurring with the keywords and the words with lexical similarities should also be in the summary.
- Textual structure: Beginnings of the paragraphs are important for summarization.
- Deep Understanding: Texts are parsed into frames and templates which have predefined sentential summarizations.

Here are some automated summarization systems:

- **MEDA (**Radev at al., 2001**)**: The main idea underlying this system is the use of a centroid-based feature in combination with position and overlap with first sentence. By using a linear combination of these three features, most salient sentences are selected to be included in the summary.
- **NeATS (**Lin and Hovy, 2001**)**: It is customized for the genre of newspaper news. The main components of its architecture are: content selection, content filtering, and content presentation. The purpose of content selection is to identify the main concepts noted in the text. The techniques used in this stage are term frequency, topic signature, and term clustering. For content filtering, three different filters are used: sentence position, stigma words, and redundancy filter.
- **GISTexter (**Harabagiu and Lacatusu, 2002**)**: Its performance is different depending on whether it is working with a single document or multiple documents. In single-document summarization, the most relevant sentences are extracted and compressed according to the rules learned from a corpus of human-written abstracts. Finally, a reduction process is performed to trim the whole summary to the length of 100 words.
- **NetSum (**Svore et al., 2007**)**: This system was developed in 2007 by Microsoft's Research Department. It uses machine-learning techniques and produces fully automated single-document extracts of newswire articles based on neuronal nets.
- **Lloret (**Lloret et al., 2011**)**: It is a summarization system based on information that users are interested in. After sentence splitting and Named Entity (NE) tagging, summaries are created in two steps: (1) several features used for sentence scoring, and (2), the sentences from the document sentences are extracted, and are assigned to some categories and ranked according to their scores.

Automated summarization research is not a parsing problem. There might be more than one summary, and the problem is determining which one is a good summary. It means that despite the fact that humans know how to sum up the most important information in a text; different experts may disagree about which information is the best to be extracted. As a result of the absence of the standard human or automatic evaluation metric, comparing different systems and establishing a baseline is very difficult.

CONCLUSIONS

- It is wise to have a flexible knowledge representation system for different data modalities (linguistic, sensorial, spatio-temporal, etc.).
- Theoretically different memory types for different learnings are required for human kind. This strategy can be helpful for an AGI system.
- Researchers seem to combine knowledge-based and statistical approaches.
- Reasoning under uncertainty and belief revision are important aspects of a good system.
- The successful architectures seem to use graph based representations with nodes and arcs for reasoning. It is also a good way to deal with *impossible proposition.*
- Embedding truth values and probabilistic representations to an intelligent system is useful to deal with confidence, validity, priority or contradictions.
- Probabilistic CCG seems to achieve the best result in parsing.
- If NLP-based intelligent systems are like a toddler having recently learned how to walk, natural language production still looks like a crawling baby.
- No system seems to deal well with transferring successful strategies or useful information to other domains whereas normal humans have a high tendency to generalize or implement a successful action/reasoning/procedure to some other domains. Similarly, humans prohibit unsuccessful strategies or least-used information.

References

Artzi, Y., and Zettlemoyer, L. (2011). Bootstrapping semantic parsers from conversations. In Proceedings of the Conference on Empirical Methods in Natural Language Processing.

Artzi, Y., & Zettlemoyer, L. (2013). UW SPF: The University of Washington Semantic Parsing Framework.

Beltagy, I., Erk, K., and Mooney, R. (2014). Semantic Parsing using Distributional Semantics and Probabilistic Logic. In Proceedings of the ACL 2014.

Berant, J., and Liang, P. (2014). Semantic Parsing via Paraphrasing. In Proceedings of the ACL 2014.

Cappa, S., F. (2012). Imaging semantics and syntax. NeuroImage, 61, 427-431.

Carpenter B. (1997). Type-Logical Semantics. Cambridge, MA: MIT Press.

Chen, D. and Mooney, R. (2011). Learning to interpret natural language navigation instructions from observations. In Proceedings of the National Conference on Artificial Intelligence.

Clarke, J., Goldwasser, D., Chang, M., and Roth, D. (2010). Driving semantic parsing from the world's response. In Proceedings of the Conference on Computational Natural Language Learning.

Ferreira, F. (2003). The misinterpretation of noncanonical sentences. Cognitive Psychology, 47, 164-203.

Goertzel, B., Pennachin, C., and Geisweiller, N. (2014). Engineering General Intelligence. Atlantis Press, Paris, France.

Harabagiu, S., and Lacatusu, F. (2002). Generating single and multi-document summaries with GISTEXTER. Paper presented at the Workshop on Text Summarization (in conjunction with the ACL 2002 and including the DARPA/NIST sponsored DUC 2002 Meeting on Text Summarization). Philadelphia, PA, USA, July 11–12.

Lewis, R. L., and Vasishth, S. (2005). An Activation-Based Model of Sentence Processing as Skilled Memory Retrieval. Cognitive Science, 29, 375–419.

Liang, P. (2013). Lambda Dependency-Based Compositional Semantics. Technical Report, September 19, 2013.

Liang, P., Jordan, M. I., and Klein, D. (2011). Learning dependency-based compositional semantics. In Association for Computational Linguistics (ACL), 590–599.

Liang, P., and Potts, C. (2014). Bringing machine learning and compositional semantics together. In submission to Annual Review of Linguistics.

Lin, C. Y., and Hovy, E. (2001). Automated multi-document summarization in NeATS. In Proceedings of the human language technology HLT conference. San Diego, CA.

Lloret, E., Plaza, L., and Aker, A. (2011). Multi-document summarization by capturing the information users are interested in. In Proceedings of recent advances in natural language processing, 77–83. Hissar, Bulgaria, September 12–14.

Matuszek, C., Herbst, E., Zettlemoyer, L., and Fox, D. (2013). Learning to parse natural language commands to a robot control system. In Proc. of the 13th Int'l Symposium on Experimental Robotics (ISER).

Mohan, S., Mininger, A. H., and Laird, J. E. (2013). Towards an Indexical Model of Situated Language Comprehension for Real-World Cognitive Agents. ACS-2013, 153-170.

Moss, L. S. (2009). Natural logic and semantics. In M. Aloni, H. Bastiaanse, T. de Jager, P. van Ormondt, and K. Schulz (Eds.), Proceedings of the 18th Amsterdam Colloquium: Revised Selected Papers, Berlin: Springer.

Radev, D. R., Blair-Goldensohn, S., Zhang, Z., and Raghavan., R. S. (2001). NewsInEssence: a system for domain-independent, real-time news clustering and multidocument summarization. In Proceedings of the first international conference on human language technology research, p. 1–4.

Rapaport, W. J, (2013). Meinongian Semantics and Artificial Intelligence Humana.Mente Journal of Philosophical Studies, 2013, Vol. 25, 25-52.

Searle, J. (1980). Minds, Brains and Programs. Behavioral and Brain Sciences, 3(3), 417-457.

Steedman,M. (2000). The Syntactic Process. MIT Press.

Svore, K., Vanderwende, L., and Burges, C. (2007). Enhancing single-document summarization by combining ranknet and third-party sources. In Proceedings of the joint conference on empirical methods in natural language processing and computational natural language learning EMNLP-CoNLL, 448–457. Prague, Czech Republic, June 28–30.

Tellex, S., Kollar, T., Dickerson, S., Walter, M., Banerjee, A., Teller, S., and Roy, N. (2011). Understanding natural language commands for robotic navigation and mobile manipulation. In Proceedings of the National Conference on Artificial Intelligence.

Thrun, S. (1997). Lifelong learning algorithms. In S. Thrun and L. Y. Pratt (Ed.s), Learning to Learn, 181-209, Kluwer Academic Publishers, Boston, MA.

Turney, P., and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research (JAIR-10).

Wong, Y. and Mooney, R. (2006). Learning for semantic parsing with statistical machine translation. In Proceedings of the Human Language Technology Conference of the North American Association for Computational Linguistics.

Zettlemoyer, L. and Collins, M. (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In Proceedings of the Conference on Uncertainty in Artificial Intelligence.