Sepia: Semantic Parsing for Named Entities

by

Gregory A. Marton

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
[February 2004]
August 2003

Author .....
Department of Electrical Engineering and Computer Science
August 29, 2003

Certified by .....
Boris Katz
Principal Research Scientist
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Sepia: Semantic Parsing for Named Entities

by

Gregory A. Marton

Submitted to the Department of Electrical Engineering and Computer Science
on August 29, 2003, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

People's names, dates, locations, organizations, and various numeric expressions, collectively called Named Entities, are used to convey specific meanings to humans in the same way that identifiers and constants convey meaning to a computer language interpreter. Natural Language Question Answering can benefit from understanding the meaning of these expressions because answers in a text are often phrased differently from questions and from each other. For example, "9/11" might mean the same as "September 11th" and "Mayor Rudy Giuliani" might be the same person as "Rudolph Giuliani".

Sepia, the system presented here, uses a lexicon of lambda expressions and a mildly context-sensitive parser to create a data structure for each named entity. The parser and grammar design are inspired by Combinatory Categorial Grammar. The data structures are designed to capture semantic dependencies using common syntactic forms. Sepia differs from other natural language parsers in that it does not use a pipeline architecture. As yet there is no statistical component in the architecture.

To evaluate Sepia, I use examples to illustrate its qualitative differences from other named entity systems, I measure component performance on Automatic Content Extraction (ACE) competition held-out training data, and I assess end-to-end performance in the Infolab's TREC-12 Question Answering competition entry. Sepia will compete in the ACE Entity Detection and Tracking track at the end of September.

Thesis Supervisor: Boris Katz
Title: Principal Research Scientist

Acknowledgments

This thesis marks the beginning of a beautiful road, and I have many people to thank for what lies ahead.

- To Boris Katz for his guidance and patience over the past two years,
- To Jimmy Lin, Matthew Bilotti, Daniel Loreto, Stefanie Tellex, and Aaron Fernandes for their help in TREC evaluation, in vetting the system early on, and in inspiring great thoughts.
- To Sue Felshin and Jacob Eisenstein for their thorough reading and their help with writing this and other Sepia documents,
- To Deniz Yuret for giving me a problem I could sink my teeth into, and the opportunity to pursue it,
- To Philip Resnik, Bonnie Dorr, Norbert Hornstein, and Tapas Kanungo for many lessons in natural language, in the scientific process, and in life,
- To my mother and father for a life of science and for everything else,
- To all of my family for their support, encouragement, prayers, and confidence,
- To Michael J.T. O'Kelly and Janet Chuang for their warm friendship,

To each of you, and to all of those close to me, I extend my most heartfelt thanks. I could not have completed this thesis but for you.

Dedication

To my nephew Jonah; may the lad make the best use of his Primary Linguistic Data!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

People's names, dates, locations, organizations, and various numeric expressions, collectively called Named Entities, are used to convey specific meanings to humans in the same way that identifiers and constants convey meaning to a computer language interpreter. Natural Language Question Answering can also benefit from understanding the meaning of these expressions because answers in a text are often phrased differently from questions and from each other. For example, "9/11" might mean the same as "September 11th" and "Mayor Rudy Giuliani" might be the same person as "Rudolph Giuliani".

Sepia[1], the system presented here, uses a lexicon of lambda expressions and a mildly context-sensitive parser to create a data structure for each named entity. Each lexical item carries a Scheme lambda expression, and a corresponding signature giving its type and the types of its arguments (if any). These are stored in a lattice, and combined using simple function application. Thus Sepia is a full parser, but lexicon development has been geared toward the named entity understanding task.

1.1 Named Entities and Question Answering

There are several applications for better named entity understanding within natural language question answering, the Infolab's primary focus. Question answering is

[1] SEmantic Processing and Interpretation Architecture

similar to information retrieval (IR), but where IR evokes images of Google, with lists of documents given in response to a set of keywords, question answering (QA) tries to give concise answers to naturally phrased questions.

Question answering research has focused on a number of question types including

factoid questions - simple factual questions with named entity answers:

Q: "When did Hawaii become a state?"

A: *August 21, 1959*

definition questions - questions with essentially one query term (often a named entity) to be defined:

Q:"What is Francis Scott Key famous for?"

A: *On Sept. 14, 1814, Francis Scott Key wrote "The Star-Spangled Banner" after witnessing the British bombardment of Fort McHenry in Maryland.*

list questions - questions asking for lists, often of named entities:

Q: "List musical compositions by Aaron Copland."

A: *Fanfare for the Common Man, Appalachian Spring, Billy the Kid and Rodeo*

possibly based on a sentence like:

American composer Aaron Copland (1900-1990) experienced several distinct musical phases during his fruitful career, but is best known for his compositions for Fanfare for the Common Man, Appalachian Spring, Billy the Kid and Rodeo

1.2 Motivation

Naturally people are most interested in things that have names, associated dates, and measures, so named entities are ubiquitous in people's questions and their answers, as shown in the examples above. The first step towards making use of these relatively closed-form language phenomena was simply finding bounds or extent (start and end) and one of a few gross categories for each identified named entity. For the Hawaii example above, once a question answering system knows to look for a date, it can search for entities categorized as dates in the text, just as if they were keywords. This is indeed a good start, but for several reasons is not enough.

One reason is that information must often be combined from several sources to answer a question. This is one of the stated goals of ARDA's AQUAINT Question Answering Program for the next two years. As a example, consider the question, "How many years passed between Lincoln's and Kennedy's assassinations?" It may be relatively easy to find the dates of Lincoln's and Kennedy's assassinations in a corpus, but there may be no simple sentence that tells the difference in years between the two. In this case we have to *understand* the points in time that the dates refer to, and be able to subtract them.

Another reason is that it is often important to know when two names refer to the same thing. Another stated goal of AQUAINT is to better "fuse, evaluate, and summarize" the information that answers the user's question. In the Hawaii example, it is better to present the one solid answer than to present "August of 1959", "1959", and "August 21, 1959" as three separate answers, each with nearly one-third confidence. To go beyond dates, many heuristics are useful for names as well, and it is possible to guess that "the Francis Scott Key Memorial Bridge", "Baltimore's Key Bridge", and "the F.S. Key Bridge" could very well refer to the same structure and that sentences about that bridge would not likely describe their namesake. It may be more difficult to decide heuristically that "Aaron Copeland" is very likely to be a misspelling of "Aaron Copland", but such knowledge would be useful in constructing a list of American composers!

Third, understanding a named entity's internal structure and its immediate context can help in finding its boundaries as well. Consider the sentence,

> Like Gould, Pratt has become very influential among young musicians, says Oppens, the John Evans Distinguished Professor of Music at Northwestern University.[2]

The part "John Evans ... University" is a description of Oppens, and it would be a mistake to retrieve this sentence in search of information about Evans. Moreover, any further parsing would get confused if Evans were not known to modify Professor. The syntactic pattern "<person> Distinguished Professor of <subject> at <university>" is thus a valuable structure to teach the lexicon. To relate it to Oppens, we would also need to understand apposition, which is beyond the immediate scope of this thesis, but is a rich source of future work within the Sepia architecture.

Less common but less benign examples that whet the imagination include understanding that "two to three hundred miles" is a single measure, that "Ronald and Nancy Reagan" probably refers in part to Ronald Reagan, that "scores of a hundred and ninety-nine respectively" does not include the number 199, and that there are two separate people in the sentence, "I told Howard Dean would win."

Finally, the salient motivation for this work is to play with precisely the kind of difficult phrases discussed above. Since taking my first semantics class, I have wanted a system on which to experiment with natural language semantics as if they were computer programs, with a compiler to tell me what combinations worked and what didn't. Sepia is exactly such a compiler, and named entities are a rich, well explored, and somewhat restricted test domain. In generating a competitive broad coverage named entity understanding package that will be useful to a larger question answering system, I will thus also have the opportunity to explore lexical semantics from an implementer's point of view.

In Chapter 2, I review related work and show support for these motivations.

[2]AQUAINT document APW20000329.0104

1.3 Approach

Sepia consists of a mildly context-sensitive parser partially implementing Steedman's Combinatory Categorial Grammar [59] formalism, and a lexicon geared towards named entity understanding. A lexical entry consists of a string or regular expression, an associated Scheme lambda expression representing its meaning, and a function signature, called a `category`, to guide the parser in semantic application.

The Scheme expressions are evaluated as part of the parsing process, and can influence that process in a variety of ways. As they are combined using simple function application (Scheme `apply`), their primary goal is to compositionally build a data structure representing an understanding of the text they refer to. Secondly, if a function application fails or returns false, the resulting partial parse is pruned. Finally, the lambda expressions in the lexicon in JScheme have full access to the parser's Java runtime environment[3], so they can change or add tokens, modify function signatures (categories), and temporarily change the lexicon. With this special access, the semantics of a particular lexical item can accomplish arbitrary, perhaps linguistically motivated structural changes in parsing, for example corrections or reanalysis.

Perhaps the most important decision in Sepia's implementation is to forgo a pipelined architecture. Rather than having separate modules (such as sentence breaking, tokenization, part-of-speech tagging, etc.) arranged in a pipeline with each module feeding its output to the next, Sepia starts with unannotated text, assigns interpretations where possible directly from the lexicon, and uses simple function application rules to combine those interpretations into larger structures. There is only one stage of processing, so it cannot commit to an incorrect interpretation "early on in the process." In the Dean example above, both "Howard Dean" as a unit and "Howard" and "Dean" separately would continue to be considered until the end of parsing. At that point the named entity constituents of the largest partial parses are reported.

In Chapter 3, I describe the system design and implementation decisions in more

[3] a shared environment in the programming language sense—the JScheme language provides access to all the classes and methods available in Java. This is not an instance of safe programming: the data, in the form of lexical items, are specifically intended to execute as code.

detail.

1.4 Evaluation

The principal challenge in Sepia, as in any knowledge-based system, is in building a lexicon. This manual task is widely noted as difficult, time-consuming, and slow to adapt to a new language or domain. Yet knowledge engineering still has a large role in most natural language tasks. I hope to show throughout the evaluation that Sepia is an intuitive and effective tool for knowledge engineering.

In Chapter 4, I will walk through a set of illustrative examples to showcase strengths and weaknesses of the approach, and discuss some difficult problems in named entity semantics. The examples will show that the lexicon language is both simple and powerful.

In Chapter 5, I show an intrinsic evaluation on named entity understanding against a set of guidelines I first had in mind when creating Sepia, and also against the much more stringent ACE guidelines. I go on to describe two ways that Sepia was actually integrated into two components of the Infolab's TREC-12 / AQUAINT Question Answering competition entry, and the results of extrinsic evaluation of Sepia via the end-to-end system.

1.5 Contributions

In this thesis I present Sepia, a non-pipelined parser for compositional semantics in natural language, and a lexicon geared toward understanding named entities. This is a fully implemented parser that realizes the Combinatory Categorial Grammar formalism in a new way. The principal distinction from other parsers is that the meaning or semantics of each element in the lexicon is not a simple data structure, but a small program in JScheme, a functional programming language. The parser can thus be entirely language-independent and task-independent. This lexicon format is intuitive and powerful, allowing lexicon developers to write rules that compositionally

assemble complex data structures to represent language.

I focus my lexicon building effort in named entity understanding because I believe it is a task in natural language that can greatly benefit from incorporating bottom-up compositional information with top-down constraints on the semantics of the entity that best fits into the larger sentence. Most named entity understanding systems preclude top-down information from benefiting named entity understanding because they look for named entities as a separate and prior step to parsing larger structures. In Sepia I show one way to avoid the pipelining problem by integrating all processing stages.

Finally, with Sepia, I make available to the Infolab a semantic named entity understanding system to be integrated into larger question answering systems. The system has been integrated with two question answering components, and I discuss lessons learned from those integrations.

Chapter 2

Related Work

In exploring the related literature, I will first relate Named Entity identification to Question Answering. Second, I will review the late 1990's Message Understanding Conference (MUC) evaluations of named entity identification, and the related Information Extraction (IE) literature. Third, I will review semantic parsing strategies and systems, and motivate Sepia's design decisions as a content extraction platform. Finally, I will describe a new hybrid between broad coverage named entity identification and information extraction, called Automatic Content Extraction (ACE), and summarize some goals that Sepia must address in order to do well in the ACE task.

2.1 The role of named entities in question answering

The Text REtreival Conference (TREC) Question Answering (QA) Track [63, 62] provides a forum for evaluating and comparing question answering systems. TREC systems try to find answers to factoid and definition questions, and more recently to list questions, in a large corpus of English language newspaper articles. The crucial role that named entity identification can play has been recognized from the outset of the competition. Srihari [58] *et al.* of Cymfony Co., winners of the TREC-8 (1998) QA track, found that 80% of the 200 questions in the track asked for a named entity

response (a person, location, etc.).

More recently LCC, the winner of the TREC-2002 QA competition, performed a detailed error analysis [53] in which they estimated that without named entity identification, their system would lose 68% of its baseline performance, while their named entity identifier only caused 3.1% of their total error due to recall and 4.9% due to precision. Their largest source of error was in question analysis, where their system failed to identify the correct category of the required answer.

The conclusion that named entity identification technology is good enough is misleading. According to their analysis 67.5% of questions in the TREC-2002 QA question set required simple factual answers found within the corpus. 2002 was in fact the first year that not all questions were of that nature: an additional 27.9% of questions required simple reasoning. It is not possible to answer "Is Qatar bigger than Rhode Island" if we only know that "1045 square miles" and "11,437 sq. km." are areas. The existing technology may be good enough for the simpler task, but not for upcoming tasks that require inference.

Both of these successful QA systems used some semantics in their named entity identification, and had many more identification classes than the MUC or ACE evaluations (see Sections 2.2 and 2.5) require. Cymfony's finite-state Textract system went as far as building "a mini-CV" for person entities, recording name, subtype (religious person, military person, etc.), gender, age, affiliation, position, birth time and place, spouse, parents, children, origin, address, phone, fax, email, and a generic "descriptors" field. In the coarser-grained MUC-7 named entity task, Textract performed at 91.24% F-measure.

It is not clear that named entity identification is even always crucial to an application. Semagix, Inc., has put no small effort into creating a regular-expression-based semantic suite of tools for their content management application [32], for example. But when they compared their named entity identifier, which performed at 76% F-measure, to a manually generated perfect named entity identification for a particular set of content needs, the end-to-end performance did not change [34]!

Sepia had similar results in end-to-end evaluation in question answering, as pre-

sented in Section 5.2, and the lesson to be learned is that how one uses a named entity understanding system is as important as how well it performs.

2.2 Named Entity Identification

The Sixth and Seventh Message Understanding Conferences (MUC) [33] evaluated named entity [14] identification and information extraction [13] tasks in 1997 and 1998. Named entities to be marked included the categories: organization, person, location, date, time, currency, and percentage. It could be said that some semantics were required. Metonymy was the most significant embodiment of semantics: artifacts (buildings, desks, copies of a newspaper) were not to be marked, so that The White House should be marked as an organization when it announces something, but not when someone takes a tour in it; "The Wall Street Journal" was not to be marked if it referred to a copy of the newspaper, but was markable when referring to the organization. Groups (e.g., Republicans), and non-currency or percentage numbers were also not to be marked. These are indeed semantic distinctions, but there were so few instances of these distinctions that it is not clear that semantics were being meaningfully evaluated. Some systems which simply picked the most common case performed very well. Furthermore, the categories are very coarse, and of little use in inference. In results, the clear winners were statistical systems, performing near human levels, and scores got monotonically poorer as systems became more knowledge intensive.

In the previous section, I motivated the need for the harder, knowledge-based approach, namely that semantics are required for inference in question answering. The following describes both the knowledge-based and statistical systems in MUC and outlines lessons to be learned for Sepia.

On the purely statistical end are three maximum entropy models. Edinburgh LTG's system [50], NYU's MENE system [11] and Kent Ridge Labs' system [67] performed at 93, 92, and 78 percent F-measure, respectively. These systems identified bounds and picked a type for each entity, but gave no further semantic information

about the entity's contents. The states and transitions in a finite state machine can be interpreted as a set of rules much more easily than the maximum entropy models above can, but no effort was made to assign rule-like interpretations to Nymble, BBN's Hidden Markov Model (HMM) system[52]. That system performed very well across languages, with case and punctuation removed, and with only 50,000 words of training data [7, 8, 51].

The lesson to be learned is that statistical approaches are robust to many forms of noise and unseen entities, are easy to train, and should be used as (at least) a guiding force for any modern named entity identifier. I have eschewed statistical methods in Sepia partly because it is clear that they can only help (and so are part of future work!), and partly to make a scientific point about how well a purely knowledge-based system can be expected to work.

Pattern matching was another popular paradigm, and the Infolab's Blitz system [42], developed around the same time, took a pattern matching approach. Pattern matching has the advantage that it is relatively easy to extract somewhat more semantics (salutation, first name, last name; value of a unit of measure, its units, and its normalized form). The best of these, NetOwl [44], achieved 91.6% F-measure on MUC-7. In MUC-6 (where results were slightly higher), the FASTUS [2, 38] system achieved 94% F-measure. In the FASTUS error analysis, however, Appelt *et al.* pointed out that they had no systematic approach to separating the various roles of named entities—whether they were artifacts or not. If failing to distinguish roles only accounted for part of a six percent loss in MUC-6 performance, then one must wonder how much the semantic capacities of any of the systems were really tested in either evaluation.

Both systems boasted rapid development, so perhaps the lesson from their good performance is that knowledge is the most useful when it is easy to enter, easy to understand, and easy to test.

Dekang Lin's approach [46] was unique and achieved the best result of the more semantic systems. He used his minimalist parser, MINIPAR [47], to find dependency parses of the text to be annotated. He then compiled "collocation" statistics—counts

of participation in relation triples for each word within the corpus being annotated—to make a decision with a naive Bayes classifier. This approach yielded 86.37% F-measure on named entity identification. In contrast, the LOLITA [28] system depended greatly on fully parsing sentences, building a logical representation for them, and doing inference. It proved to be brittle, with only 76.43% F-measure. The lesson, found elsewhere in the literature as well, is that being robust to partial parses leads to better performance than requriring full parses in many sentences.

The FACILE [9] system had, in many ways, a similar inspiration to Sepia, but fared badly, at only 82.5% F-measure. FACILE used a context-sensitive grammar with regular-expression-style operators to compositionally combine attribute vectors. The failure was most likely in the opacity of the notation. Aside from heavy use of abbreviation in the grammar rules, a list of attributes for a constituent proved insufficient (from their description) for coreference resolution and template building. The parser was expanded with hooks to let the features influence subsequent parsing, but that strategy separates the resolution action from the part of the grammar indicating resolution, and is thus harder to understand. As if to prove the point, the system was under development for 20 months before its first deployment in this setting.

The LaSIE-II [37] system attained 85.83% officially, but for the IE tasks, its authors added a small amount of knowledge, boosting named entity performance to 90%. LaSIE was built on the General Architecture for Text Engineering (GATE), a modular pipelined architecture that makes it easy to do large-scale software engineering right. In order, the modules included a tokenizer, gazetteer, sentence breaker, part-of-speech tagger, POS-based morphology analyzer, a ten-stage named entity CFG parser, in one step a phrase structure CFG parser and Quasi-Logical Form generator, a coreference analyzer, a discourse interpreter, and a domain-specific template writer. By modularizing the task in this way, the Sheffield team was able to use others' work, such as the Brill part of speech tagger, and efficiently divide resources.

The modularity blessing of GATE is also its curse. Note the position of the gazetteer before the sentence breaker or part of speech tagger. In the previous year's competition, there were sentence boundary and part-of-speech errors on words in the

gazetteer, but because the sentence had already been split, the gazetteer could not recover. As reordered for the following year (as listed above), the sentence breaker and part-of-speech tagger treat gazetteer matches as opaque tokens. The gazetteer might thus recognize a name which should have been overridden by an actual sentence boundary, but this is less common. Similarly, in the ten-stage named entity grammar, it was impossible to correctly identify both "Julian Hill" as a person and "Pearl Harbor" as a place at the same time. If known first names stage ran first, both would be marked as person; if the location cue stage ran first, both would be locations. In their words, they lacked a "controlled propagation of ambiguity". That propagation is unbounded in Sepia—both possibilities are allowed—and I will rely on the semantics of the context to determine which entity is of which type.

The Message Understanding Conferences stopped partly because some systems were achieving near-human performance at the task, and partly because the task was so far from the much more useful task of understanding the *meanings* of the named entities identified. Many participants chose to pursue the more semantic Information Extraction task, described in the next section, and following that, the two have been fused into a new named entity understanding task focused on meaning: the Automatic Content Extraction conference.

2.3 Information Extraction Systems

Information Extraction (IE) is the task of using a natural language (most often news) corpus to fill a predetermined template appropriate to the content of the document. This is a domain specific task that was also addressed in the Message Understanding Conference (MUC) competitions. The general consensus for IE is that knowledge engineering approaches have significant advantages over pure machine learning approaches, but new learning approaches may be gaining ground.

In MUC-6, the CIRCUS [27] team described the shortcomings of their automatic dictionary acquisition tool with respect to the Template Extraction task:

> Unfortunately, CRYSTAL's CN [Concept Node] definitions offer little help

> with noun phrase analysis, since they operate at a relatively coarse level of granularity with respect to a complex noun phrase. The CNs that CRYSTAL induces are designed to locate relevant noun phrases for an extraction task, but they do not help us understand where to look inside a given noun phrase for the relevant information.

This indicates that sentence-level features can disambiguate the boundaries of the targets needed for information extraction, but they must still have access to the contents of those noun phrases for successful information extraction. Sepia aims to combine both the sentence level and named entity level understanding into a single system so the two levels can help disambiguate each other.

In integrating the two levels, one must keep in mind that the performance of separately evaluated systems may not reflect the performance of the system as a whole. In reviewing FASTUS and another system, TextPro, during an information extraction tutorial [3], Douglas Appelt and David Israel voiced two familiar morals, one advocating partial parsing, the other end-to-end integration of components. Full parsing, especially in new domains where training data is expensive, often breaks, so that an IE engine must take advantage of any partial analyses robustly. It matters not, they say, that full sentence analyses are broken or underspecified—that is not the greatest source of end-to-end error. The second moral emphasizes the point: improving the performance of any IE component, measured with tests of that component, does not necessarily translate into better end-to-end performance.

On the partial parsing angle, the LaSIE team also offered a lesson: in building the 150 named entity rules and 400 phrase-structure rules, they had initially bootstrapped from the Penn Treebank, automatically acquiring syntactic structures, then annotating them with semantics.

> The resulting analyses were poor, and the effort of manually annotating the rules with features for semantic interpretation substantially reduced the benefits of the grab-and-run approach to grammar acquisition.

Instead, they ended up rebuilding the rules from scratch for this system, using an

iterative refinement approach on the MUC-7 training data. This sort of iterative refinement approach is what we are using in preparing Sepia for the Automatic Content Extraction competition next month.

Probabilistic methods were not competitive in information extraction when MUC-6 was held, but there is now one promising probabilistic IE system. The ALICE system [12] has shown promising results on the MUC-4 information extraction task, largely because it uses the Collins parser's grammatical relations as features. So at least on a particular domain, information extraction competitive with knowledge based methods might be usefully learned.

The lesson I take for Sepia from the initial success of the knowledge-based approaches and this new hope for a statistical approach is that the knowledge based approaches serve to locate the most salient features for a learning system to use, and the most useful data structure for it to build. My goal is to build Sepia with a flexible data structure that will serve as a new vehicle for machine learning algorithms to identify relevant semantics. In Section 3.3.13, I discuss how Sepia might be immediately augmented with statistics for an improved hybrid approach.

Yorick Wilks argues that the TREC Question Answering track is the one to watch in the debate between traditional AI approaches to Information Extraction and statistical language modelling approaches [65], in part because indexing by template might be a good way to answer questions about common events in the corpus. The Question Answering track may also be where the two ideologies can most easily work together to find the most likely meaningful answer to a question.

2.4 Semantic parsing

In the majority of parsing systems, any attempt at semantics takes the form of unification over a vector of attribute–value pairs [55, 19, 5]. Unification in this case means bringing all attributes of both attribute vectors into a new encompassing attribute vector. Unification *fails* if there are incompatible attribute values. Just what values are compatible is up to the linguistic theory they implement, and it takes work to

make the various unification-based grammars compatible so that they can be directly compared [6].

One instance of a unification-based robust parser is the prolog-implemented Gemini parser [23]. Like Sepia, Gemini uses the same formalism throughout its processing and interleaves its semantics into the logical forms as partial parses are built during processing, dropping partial parses that do not create a new well-formed logical form. The logical forms built in Gemini were originally used to recognize spoken language utterances in a flight corpus, but has been expanded with 50,000 new lexical items to become part of the QUARK multi-document multi-media question answering system [64].

A unification model can support a lambda calculus: one or more attributes can be assigned lambda expressions as values, and "compatibility" might be defined as whether the two expressions can combine. Combinatory Categorial Grammar (CCG) is ideal for supporting just this sort of processing [59], but Grok [5], a CCG parser developed with Steedman at Edinburgh, does not, in fact, do that. Its logical forms are devoid of lambda calculus, again using only feature vectors.

Instead of using unification over typed feature vectors as a basic semantic structure, Sepia uses the CCG formalism with Scheme expressions for lexical semantics. The simple combination rule is now function application and the data structures associated with a lexical item are small pieces of Scheme code, thus forming a lambda calculus.

In CCG, the lexicon contains triples: a word form, an associated category that tells the parser what categories the entry can combine with, and a lambda expression for the meaning of the word form. Parsing is done by function application, composition, coordination, and type raising, each of which translate into simple transformations on the lambda expressions involved. CCG is mildly context-sensitive because it lets each lexical item specify a finite number of non-terminals to expect. For someone familiar with context free grammars, this is akin to being able to specify not simply a string of terminals and nonterminals on the right hand side, but instead a structured tree of them. Even though CCG is mildly context-sensitive, Jason Eisner [25] showed that

there is a normal-form which allows efficient parsing with the Earley algorithm [24]. Statistical parsing is also possible, with full semantics, in this framework [36], and Hockenmaier was able to bootstrap a statistical grammar from the Penn Treebank in LaSIE style [35]. CCG is also very good at accounting for linguistic data and their interpretations, especially coordination and gapping structures. Extensions such as Set-CCG [4] and ordered Set-CCG [39] can account for some difficult cross-linguistic variation. Thus the formalism is a very attractive one for semantic processing in general. By implementing the named entity semantics in this formalism, I hope to be able to interface with the best practices of higher level semantics and full statistical parsing at a later stage.

The use of thematic roles as semantics is also common [30, 26, 20]. Acquisition of semantic lexicons for determining thematic roles has been explored by Cindi Thompson [61] and Bonnie Dorr [22] (whose lexicon for English is available [21]). Parsers which label thematic roles include MINIPAR [47], and Gildea and Jurafsky's system [30]. I find this work inspirational, and useful in methods to automatically acquire a semantic lexicon, but its application to named entity identification is more distant. It will be interesting to see any examples where a thematic parse will cause the system to perform well or poorly. The benefit of using the CCG system is that it is easy to put in a small bit of thematic parsing and show its effect on named entity understanding when the need arises.

Finally, Lenat's Cyc [45] system is slowly coming into its own, with work by O'Hara [54] to map Framenet and Penn Treebank roles to Cyc structures, and the newly Free variant OpenCyc [17]. Commonsense projects such as Cyc and OpenMind [60] could provide much needed additional data and a source of semantics for a parser like Sepia. Perhaps Sepia lexicons should be developed as an open source commodity!

2.5 Automatic Content Extraction

Most recently, NIST has launched the Automatic Content Extraction (ACE) [57] evaluations on Entity Detection and Tracking (EDT), and Relation Detection and Categorization (RDC). The tasks are very similar to the MUC tasks, though with finer control over metonymy, and more natural classes. EDT is what I have been calling named entity understanding, as opposed to the relatively shallow task of named entity identification, which requires only bounds and type. RDC goes a step further and requires recognition of relations between named entities, such as one being located at, having a structural (e.g., part of), or a role relation (e.g., founder) to another entity. ACE also requires that simple nominals and pronouns be tagged, not only name mentions. Overall, this is a much more semantic set of tasks.

LaSIE has jumped the GATE [16][1], and the resulting information extraction system, ANNIE [15], is publicly available and currently being adapted for the upcoming ACE competition [31]. It would be interesting to compare the performance of the hand-crafted rules in its MUSE named entity semantics module to the hand-crafted rules generated for Sepia. I expect that many of the rules would be similar, and the primary differences would be less attention to ordering in Sepia, and fewer special cases to fix bad rule orderings.

The GATE MUSE system reports accuracies in the 80% range on ACE. Other systems from MITRE, BBN, SRI, IBM, and NYU report earlier results with accuracies around 75% on the ACE site. This fall's evaluation will be the first not marked as a "pilot" evaluation and thus if I understand correctly, the first with citable publications.

One of the major challenges in the ACE competition will be dealing with lack of case and inappropriate punctuation, especially in the broadcast news texts. One article in the training collection opens,

> grief engulfed as tree, a one day after a fire in a cable car filled can skiers killed at least 155 people. it happened inside a tunnel high in the austrian

[1] Their joke, not mine. But it *is* cute.

alps.

By the end of the second sentence we as humans were able to decipher "as tree, a" to mean "Austria", but I still cannot guess what might have produced "filled can skiers". The Austrian Alps are easier to restore because they are unique words despite capitalization, but several organizations, such as "NOW", present a greater challenge.

That challenge can be called "truecasing", and Lita *et al.* [48] present a statistical approach to truecasing arbitrary text from which they report an increase in ACE EDT mention detections[2] on Automatic Speech Recognized broadcast news (e.g., the excerpt above) from 5% F-measure to 46% after truecasing, and a 26% improvement in F-measure overall in named entity understanding. I would thus like to incorporate some truecasing measure into Sepia as well for broadcast news, if time permits before the actual ACE competition in four weeks, because it would help easily overcome some of the constraints of noisy data.

[2]detection of coreferring entities

Chapter 3

Sepia's Design

The Sepia system consists of two primary parts: the parser and the lexicon. The parser is a partial implementation of a CCG parser, and aims to be language-independent, pushing all language-specific knowledge to the lexicon. The lexicon is not a simple list of words, but a set of strings or regular expressions associated with semantics in a lambda calculus (specifically, in JScheme).

Every semantics, or `logical form`, in the lexicon must have an associated function signature, or `category`, that guides the parser in applying functions to matching arguments. This is used, for example, to specify subcategorization information of verbs for their arguments,[1] and of adjectives for the nouns they can modify. If a function application is attempted and fails (for example because of knowledge outside the lexicon), then no new combined constituent is constructed.

In this chapter, I will detail the design and implementation decisions embodied in Sepia, and hope to give a clear understanding of the lexicon format and of how the parsing mechanism uses the lexicon, so that examples in the upcoming sections will be fully understandable.

[1] or of the arguments for the verbs they may have various roles with. Whichever does the specifying the other has to agree that it matches the specification, so the which is the head and which is the modifier seems akin to deciding whether the Earth revolves around the Sun or vice versa, which is to say a matter of interpretation.

3.1 No Pipeline

The most common natural language processing architecture is a pipeline of modules where each has access only to the final decisions of the ones before it [49, 16]. One module will identify sentence boundaries, the next will tokenize, and so on. A mistake in an early module is usually irreparable by a later one. This means that there is no way to incorporate top-down understanding into the lowest level decisions.

The following are examples where a state-of-the-art named entity identifier would cause a parsing failure in a later module:

- "They scored *a hundred and ninety-one* respectively."[2]
- "I told *Harold Dean* would win."
- "Mr. and *Mrs. Hodson* and their children..."

In the first case the word *respectively* can have no meaning, because the coordination it requires has been analyzed as part of a number. By having access to the intermediate candidates *a hundred* and *ninety-one* separately, Sepia can decide to use *and* as a coordination rather than part of a larger numeric expression.

In the second case, *told* cannot find a good argument structure. If Sepia had lexical entries for *told* and other verbs, the intermediate representation with *Harold* and *Dean* as separate names would be available to them as a possibility. Adding broader language coverage including verbs is a goal for future work.

In the third case, a later coreference module would not be able to associate a pronoun "their" with the Hodson couple because only one person is recognized. Sepia (with the current lexicon) sees *Mr.* and *Mrs.* as functions that take a person as an argument, and add title information to that person's record. One meaning of *and* combines adjacent meanings that have like categories into a new meaning that has the same category. It will create a new one-place function that will map the

[2]parsing numeric expressions seems unpopular: it is not discussed in MUC, and ANNIE does not make the attempt beyond marking "hundred", "ninety", and "one" separately. Numbers of this sort do fall under the common definition of a named entity, and ACE requires "five astronauts" to be marked, but does not expand on the requirement.

two underlying one-place functions over their argument. The result will be a data structure that contains two people with a name *Hodson*, one of which has a title *Mr.*, and the other has a title, *Mrs.*.

Controlled propagation of ambiguity starts at the lowest levels. Sepia does not expect any form of preprocessing before it sees a text.[3] The parser makes no decisions about what interpretation of a partial parse is correct until it finishes parsing. Therefore it cannot make an incorrect decision due to lack of contextual information *in a particular stage of processing.* All decisions take into account both bottom-up and top-down information from the whole text.

Of course the system may fall prey to missing vocabulary, to a combinatory explosion, or to an incorrect decision at the end of processing. The only problem unique to Sepia is the combinatory explosion, on which the bet is that semantic pruning (see Section 3.3.4) will be sufficient.

It is possible to propagate ambiguity and semantics between separate modules in a pipeline explicitly, but there are drawbacks. Much of the practical usefulness of pipelining is that very different modules can be interchangably replaced and evaluated so long as they have the same input and output formats, but the more information one passes between modules in a pipe, the more difficult it is to make any component fit.

Furthermore, it's not clear what information to pass: top-down information is likely to be in the form of a constraint, and the probability ordering of the options given the constraint might be different than the order was without the constraint. Thus passing the top k options even with original probabilities is not enough, because the answer that matches constraints might be unlikely (and therefore surprising, interesting!). Even passing all of the options is not theoretically enough, because useful information about the structures and their likelihoods then needs to be duplicated in the later module in order to evaluate the possibilities under the new constraint. Thus

[3]though truecasing and removal of other markup before processing would both very likely help, given that we have been writing lexicon for newspaper text. Of course I believe that these processes should also ideally become part of a system like Sepia, but there is only so far that one can push an agenda before hitting a deadline.

to fully incorporate top-down constraints, one has to duplicate the prior module in the later one.

Sepia is still impoverished because it does not yet have an infrastructure to use probabilities for lexical items, but the thrust of this work is in showing a proof-of-concept architecture that can integrate the modules, and thus best reap the benefits when probabilities are later added.

Though it has no pipeline, Sepia can serve as part of a pipeline. Sepia is built with standoff annotation in mind, and there is a wrapper that adds Sepia-recognized entities as annotations to an XML standoff document. This interface was used to integrate Sepia with two modules in the Infolab's TREC 12 / AQUAINT Question Answering competition entry.

3.2 Working Example

Throughout this chapter I will use examples to illustrate each idea. The example in Figure 3-1 will serve in multiple explanations, so I begin by summarizing it.

The default tokenization is shown in parentheses, but two items from the lexicon, *O'* and *'s* are also recognized as tokens. At the same time, any multi-word lexical entries will also be recognized, (e.g., a pre-existing lexical entry for "Margaret O'Hanlon", were she famous). Lexicon-based tokenization is discussed in Section 3.4.2.

Hanlon is recognized not as a word in the lexicon, but by a regular expression pattern. All tokens are tested against each pattern in the lexicon. The special $*i*$ notation allows the semantics to have access to the groups a match generates. In this example, there is no difference between $*0*$ and $*1*$ because the first set of parentheses in the pattern happens to capture the whole string. Regular Expression entries are discussed in Section 3.3.8.

O' is a function that requires a `capitalized_word` and generates a `common_lastname`. It applies rightward (a backslash in the category would indicate leftward application) to *Hanlon* to generate the string "O'Hanlon". The new category `common_lastname`

Example: Margaret O'Hanlon's Bar&Pub

Default tokenization: (Margaret) (((O)('))(Hanlon)((')(s))) ((Bar)(&)(Pub))

Lexicon:

```
(import "edu.mit.nlp.sepia.AtValSet")
(import "edu.mit.nlp.sepia.Coordination")

;: Margaret : common_female_firstname
*0*

;: 's : possessive
#t

;: /^([A-Z][a-z]+)$/ : capitalized_word
*1*

;: O' : common_lastname / capitalized_word
(lambda (w) (string-append *0* w))

;: :common_female_firstname: : firstname
(AtValSet. 'name *0* 'gender 'female)

;: :common_lastname: : person \ firstname
(lambda (p) (.append p 'name *0*))

;: Bar : organization \ person \ possessive
;: Pub : organization \ person \ possessive
(lambda (possessive)
  (lambda (namesake)
    (AtValSet. 'type 'food-service
               'name (list namesake "'s" *0*))))

;: & : * / * \ *
(lambda (back)
  (lambda (forw)
    (Coordination. 'and back forw)))
```

Figure 3-1: An example and a lexicon.
Disclaimer: entries are meant to be syntactically correct and illustrative, but not realistic.

and its new logical form, the string "O'Hanlon", are assigned to the existing token *O'Hanlon.*

Margaret is first recognized as a `common_female_firstname`. That category can be promoted to `firstname` using a category promotion rule. Here the *0* still refers to the entirety of what has matched, but that is now a Scheme object (not guaranteed to be a string). In this case the new promoted logical form is a variant on a feature structure, a set of attributes associated with lists of values. Category promotion is discussed in Section 3.3.9. Complex logical forms including the Attribute-Value-Set are discussed in Section 3.3.11.

O'Hanlon is promoted to a function that applies to the `firstname` *Margaret.* The `firstname` *Margaret*'s underlying object is an AtValSet (see Section 3.3.11). This semantics appends the string "O'Hanlon" to the `name` field of that AtValSet and returns the resulting new structure. That new structure is housed in a new token *Margaret O'Hanlon* which is created on the fly with category `person`. The creation of new tokens and its impact on parsing strategy is discussed in Section 3.4.1.

The word & is a coordinate conjunction that combines two tokens to either side that have the same category (function signature), and creates a new token spanning both that has the same category as its arguments. The `Coordination` logical form data type automatically adds semantics for mapping applications over their arguments to create a new `Coordination` of `organizations` when "Bar&Pub" is applied. Coordinations and other complex logical forms are discussed in Section 3.3.11.

Bar alone will also combine with *Margaret O'Hanlon's*, to generate a parse for only *Margaret O'Hanlon's Bar*, but a final heuristic after parsing is finished will remove partial parses entirely subsumed by larger ones. This final heuristic is discussed in Section 3.4.5.

In the following sections, I will first discuss general design principles, and then address issues of implementation.

3.3 Lexicon Design

A lexical item consists of a word or regular expression and its "definition". To a computer, a definition must be a function or data structure, and the language I have chosen to build definitions in is JScheme [1]. Each function and each data structure has an associated function signature, called a `category`, used to guide the parser in applying the right functions to the right data structures.

3.3.1 Separation of Language from Parsing

Combinatory Categorial Grammar (CCG) is a formalism for semantic parsing, pioneered by Mark Steedman, that can capture many previously problematic phenomena across languages in a compositional structure [59]. The key idea is that the parser be very simple and language-independent, following pure function application rules, and that the lexicon contain all language-specific rules, word meanings, and associations. This separation is a central theme in Lexical Semantics, and is a central design guideline in Sepia. Moreover, keeping the parser as simple as possible makes it easier for a lexicon builder to predict what effect his changes will have, and so makes his job easier.

A CCG parser uses a `category` associated with every logical form to guide its processing, and this category is all that the parser knows about each semantics, or `logical form`. The category indicates whether each logical form is a function or a data structure. Functions have arguments, whereas data structures do not; if a category has slashes in it, it is a function, otherwise it is a data structure. A category with slashes—a category indicating a function—is a list of categories separated by slashes. The first category specifies the type of the function's result. After that, a slash indicates the direction in which an argument is expected (a forward slash (/) indicates rightward in the string, and a backslash (\) indicates leftward), and the category which follows indicates the required type of the argument. The lexical entry for the verb "slept" (see Figure 3-2) might specify that it requires an animate argument to its left.

```
;: slept : VP \ animate
```

Figure 3-2: The verb "slept" requires an animate subject.

The arguments are filled from right to left[4], so that one may assume left-associativity in the category. Thus one logical form for "King" that was intended to recognize "King of Spain" might be specified as shown in Figure 3-3.

```
;: King : person / country / of
```

Figure 3-3: A category of the title King, as it might try to recognize "King of Spain". Note that arguments are left-associative.

Sepia's CCG parser will attempt to apply a function to all constituents of a matching category in the correct relative direction.

3.3.2 Using JScheme for Semantics and as an Interface to the World

I chose JScheme as the language of Sepia semantics for several reasons. Lisps are acknowledged to be the most intuitive way to express functional programs such as the lambda expression semantics proposed by Steedman. I wanted to use Java for the parser to make Sepia most easily usable by others. I also wanted to enable the semantics to make use of others' Java programs as resources for building meaningful semantics. For example one might imagine a particular function wanting to call out to Google to decide whether "Dana Farber" was male or female.[5] JScheme is a Scheme with a strong Java interface both in being called from Java and in calling out to other

[4]The rightmost argument corresponds to the outermost lambda in the logical form.

[5]only to find, of course, that it's an organization! For the curious, it is a cancer institute founded by industrialist Charles Dana and Dr. Sidney Farber.

Java modules. The strong type checking that Scheme provides has been helpful in developing initial lexicons.

3.3.3 Scheme Expressions vs. Feature Structures

Grok [5] and other existing semantic parsers use a vector of attribute-value pairs as their base semantics, and often perform unification as part of parsing. I think unification on attributes can be very helpful in syntax checking, and won't rule it out of future versions of Sepia. However, using full Scheme expressions for the semantics offers several advantages. The first is that the parser becomes simpler and more transparent to the lexicographer. The second is that it becomes possible to communicate meaningfully with other expert systems to make decisions about a meaning.

3.3.4 Pruning Failed Partial Parses

One of the primary uses of unification is to prune parses which have conflicting features, interpreted as meaning. Sepia can also prune partial parses because of a meaning conflict, but the possible conflict is more generic: if any semantics returns `#null` or `#f` as its result, or exits via an exception, then that partial parse is dropped.

For example, the syntax need not specify that in "two one hundred dollar bills", there are two separate numbers, namely "two" and "one hundred", because the semantics can recognize that these cannot conjoin the same way that the ambiguous "thirty one hundred dollar bills" can. The Scheme expression that would combine "two" and "one hundred" simply needs to return failure.

The key difference between this pruning strategy and pruning in other parsers is that a parse in Sepia is not pruned because it is unlikely. Rather, it's pruned because it's semantically not possible or plausible.

Statistical pruning can complement such direct meaning-based pruning by assigning likelihood to individual logical forms. The most likely parses which are still available after processing may often be correct.

3.3.5 Multiple Entries with a Single Semantics

To avoid unnecessary redundancy in the lexicon, it is possible to specify multiple entry words and categories that all share a single semantics. The format is simply to specify all the lines of word and category before the block of semantics. For example one might use the following to specify multiple names:

```
;: Mary : common_female_firstname
;: Margaret : common_female_firstname
;: Michael : common_male_firstname
*0*
```

The three names have different strings, and different categories, but share the same simple semantics (*0* indicates the actual string matched—one of their names).

3.3.6 String Entries vs. Headwords

In a traditional lexicon, inflectional variants of a word (fly, flew, flown, flying, flies) are grouped under a headword (fly), and their definition given with the assumption that the inflectional transformations in meaning are known. Handling headwords would involve handling morphology in general, and this is not a problem I have considered as part of Sepia. That decision makes Sepia difficult to use for morphologically productive languages like German, Hungarian, or Turkish. Sepia would not essentially change if a morphological analyzer were put in place of the tokenizer, but that would have been more difficult to implement.

String entries may use the *0* variable to retrieve the actual string that triggered them (as in the example in Section 3.3.5) rather than repeating the string in the semantics. This usage is similar to (and inspired by) the use of the *0* variable for pattern entries, as discussed in Section 3.3.8.

3.3.7 Multiword Entries

Multiword lexical items are desirable because many noun compounds, e.g. "World Cup", verb-particle constructions, e.g. "took off" (as an airplane or idea), and other phrases have noncompositional or idiomatic meanings. There are also entries which span token boundaries as currently defined in Sepia, but are parts of words. For example *'s* as a possessive seems to encode a semantic unit. See Section 3.4.2 for implementation details of multi-token lexical entries.

3.3.8 Pattern Entries

Regular Expressions are a common way to recognize classes of strings that may have a particular closed form. Allowing regular expressions in the lexicon is an intuitive way to recognize, for example, the string "28" in the text being processed and convert it into the number 28. The lexical entry for converting a sequence of digits to a number might be:

```
(define number-format (java.text.NumberFormat.getInstance))

;: /^(\d+)$/ : digits
(.parse number-format *0*)
```

Similarly, currency or date recognition can take advantage of Java's library of date and currency parsers, and immediately transform the text into a form that has semantics to the program. Numbers, dates, and currencies can be readily compared with other objects of the same type, or converted to other formats.

The lexicographer need not even be careful to only try to parse legal numbers or legal values in other formats. If a DateFormatException (or any other exception) is thrown, the partial parse will simply be abandoned, as discussed in Section 3.3.4.

Special variables of the format $*i*$, where i is a number 0 or greater, are available to the semantics of pattern entries. These refer to the groups captured by the regular

expression. In the example above, *0* refers to the entire match (as group(0) does in Java, perhaps as $0 does in perl). *1* would be equivalent, as the parentheses capture the whole token.[6]

3.3.9 Category Promotion

It is often useful to specify an is-a hierarchy for things in the world. A verb might require that its agent be animate, or a first name might want to combine with a last name. WordNet has an extensive is-a hierarchy which many find useful, and which will likely be useful in specifying semantics of a broader coverage lexicon.

For Sepia's parser, the only notion of type is the category system, and so it is useful to be able to say that any object of one type is also necessarily an object of another type. In the *Margaret O'Hanlon's Bar&Pub* example (Figure 3-1), a `common_female_firstname` is necessarily a `firstname`. Presumably a `common_male_firstname` would be as well. A `lastname` can then specify that it wants to combine with a `firstname`, whatever sort of `firstname` its argument might underlyingly be.

The notation for this category promotion is like any other lexical entry, except that in the place of a string to be recognized is the category to be promoted, inside a pair of colons.[7] The semantics specify the transformation involved in promoting the type from the base category to the higher category, and have access to the actual object of the base category through the *0* variable. The use of the *0* variable to mean "whatever this rule refers to" is inspired by the use of *0* in pattern rules (Section 3.3.8) and used here in a metaphoric sense.

3.3.10 Wildcard Matching Category

Coordinations such as "and" and "or" can apply to any category, including complex categories that have arguments. Rather than handling coordination specially in the

[6]Please refer to the `java.util.regex.Matcher` javadoc, specifically the `group(int)` function for implementation details.

[7]A text can unfortunately fool the parser into thinking that a category is present by happening to have a category name between colons, but the semantics would likely not get too far. In any case, Sepia is not designed with security in mind.

parser, I chose to create a wildcard category label * that would:

- match any other category, and
- when applied to a real category, would cause the wildcards in the resulting category to be filled with the matched category.

Consider the lexical entry

```
;: and : * / * \ *
(lambda (back) (lambda (forw) (Coordination. 'and back forw)))
```

When applied leftward to a category `a`, the new logical form will have category `a / a`. Similarly, when applied to a category `a / b`, the new logical form will have category `(a / b) / (a / b)`. Because categories are left-associative, that is just the same as `a / b / (a / b)`

In the "Bar&Pub" example, *Bar* and *Pub* both have category `organization \ person \ possessive`. When & is applied to *Bar*, it creates a new function with category `organization \ person \ possessive / (organization \ person \ possessive)`. This new category can apply directly to *Pub*, and the result is a `Coordination` that has the same type each constituent started out with. The `Coordination` class takes care of the correct combination of `map` and `apply` to create a result of application that is a `Coordination` of the two objects that would have been created had each of *Bar* and *Pub* applied separately. This may not always be the desired semantics for coordination, and I will address that in the next chapter.

There is another type of wildcard character that also matches anything, but does not do anything special when matched, as * does. This wildcard is denoted with a period.

3.3.11 Complex Logical Form Data Structures

A semantics cannot live on strings and alists[8] alone. Some formal semantics use a Typed Feature Structure, a form of association list, to represent meaning. Unification over a feature structure[9] has been a useful tool in combinatory semantics. Sepia provides a variant on a feature structure called an AtValSet (Attribute-Value Set) as one possible type of logical form, provides a Coordination as another, and leaves open the possibility of other types.

AtValSets are the most common complex logical forms in the named entity lexicon. These are HashMaps keyed on Scheme Symbols whose values are LinkedLists of Objects. Append and prepend are defined to append or prepend objects or lists of objects to the list associated with a symbol. Append and prepend also support entire AtValSets, in which case they perform the appropriate list-appends key by key. AtValSets support deep copying so long as their contents do as well. They also support an asymmetric match operation that requires an argument AtValSet to have all but not only the same properties.

Coordinations contain a simple ordered list of Objects and a coordination type. They support a match operation as a set would (without regard to order) by calling the match operator on the element data types. They handle mapping over a single argument if their contents are functions, or mapping an applying function over multiple arguments otherwise. As a disclaimer, this is simply the most obvious interpretation of coordination, and I discuss problems with this form of coordination in Chapter 4.

Though I have not implemented them, I imagine that Lexical Conceptual Structures [21] might be another useful type of complex logical form data structure to represent paths, causality, and event relations, for which simple sets of attributes seem impoverished.

Each data structure discussed above is language-independent, which is why I provide them as part of the Java framework. Complex logical forms like Person might

[8] `alist` is a lispism for "association list", akin to the Perl hash or the Java Map

[9] Typed Feature Structure Unification is essentially the process of merging two Typed Feature Structures (sets of attributes and their values, where those values may also be Typed Feature Structures) while ensuring that there is no unresolvable conflict or disagreement between them.

extend[10] AtValSet. These can be created in JScheme as part of the lexicon. Such data structures might provide valuable special handling of some attributes (such as deciding on feminine gender for "Bobby-Sue" despite ambiguous or conflicting cues) in a language-specific way.

3.3.12 Execution Environment

The JScheme semantics are evaluated in an execution environment which is initially empty. Even in an empty environment, the Scheme logical forms would have access to the Java environment through JScheme's JavaDot notation, but a lexicon can provide additional definitions to its logical forms.

Any Scheme that appears in a lexicon file before the first lexical entry is evaluated at load time, and treated as part of every lexical entry's environment. This was hinted at in the NumberFormat example in Section 3.3.8, but was not explicitly discussed. The `number-format` variable there would become available to any logical-form in any lexicon file. It is also common to see Java-like `import` statements here to give more readable access to classes like AtValSet and Coordination, as appear in the first two lines of the Pub example lexicon (Figure 3-1).

3.3.13 Stochastic Parsing

A major design decision in the current version of Sepia is conscious inattention to statistical methods in parsing. The motivation is to see how far purely knowledge-based methods can be pushed. I do not see a conflict between stochastic and knowledge based parsing, and in fact would like to attempt an integration of probabilities on logical forms associated with lexical items, and thus on partial parses. It is easy to see how stochastic parsing would be implemented in this framework: as a set of probabilities of a logical form being associated with its string, pattern, or category promotion. However, if the probabilities were made available to the semantics, and parsing decisions were made during processing based on the intermediate probabili-

[10] extend in the very concrete Java subclassing sense

ties, then it would be difficult to separate the effect of the statistics from the effect of the richer semantics in the final results.

When statistics are incorporated into Sepia, they will not be visible in the lexicon. Humans are not good at specifying probabilities or at understanding their impact. Probabilities should come from some form of training, and be stored separately from the human-readable semantic lexicon.

3.4 The Parsing Process

3.4.1 Tokenization

Every parser must have a smallest unit of meaning, and Sepia primarily uses word boundaries and boundaries between sequences of letters and numbers for its smallest division points. Specifically, strings are first split on `/\s/`, then into substrings matching `/([0-9]+|[a-zA-Z]+|[^a-zA-Z0-9])/`.

Tokens are stored in a lattice data structure that makes access to adjacent nodes efficient. The initial set of tokens also includes any multitoken sequences present in the lexicon; thus the tokenizer can be said to be lexically aware, even though it remains language independent. One can argue that it is not language-independent because it is tied to a particular character set. But adapting the tokenizer to another character set would be trivial, involving a change to the two regular expressions mentioned above. A more fundamental change would be to break on morphemic boundaries, as discussed in Section 3.3.6.

Tokens are "activated", added to a parse queue, when they are first created, and when a new token is created adjacent to them. The parser evaluates tokens, attempting to combine them with adjacent ones, in the order that they are enqueued[11]. Logical forms can themselves generate new tokens, in particular they can create tokens representing linguistic empty categories, or they could use this feature to simulate

[11]This is an area for further research. It might make sense in some cases to explicitly promote an ordering, as for a hyphenated sequence of words

morphological analysis.[12] No lexical entries make use of this flexibility at the moment, however.

3.4.2 Multitoken Lexical Entries with Prefix Hashes

Sequences of tokens are recognized by precompiling from the lexicon a pair of prefix hashes, one for entries spanning word boundaries (the first regular expression above) and one hash for entries spanning within-word boundaries (the second regular expression). These contain, for every multi-token lexical entry, all token subsequences starting at the beginning of the entry.

For *Margaret O'Hanlon*, were it a single lexical entry, the word prefix hash would contain *Margaret*, and the subword prefix hash would contain *O* and *O'* (with the apostrophe).

During tokenization, which happens left to right, if a current sequence of tokens (including the base case of one token) is in the prefix hash, then it is kept in consideration as a possible multi-token entry. If any of these finds a match in the lexicon, a token is generated.

3.4.3 Initial Lexical Assignment

Each token may be associated with several categories in the lexicon, and each category may have several logical form functions. All logical forms associated with all matches to a token are initially postulated as possibilities. Moreover if a category can be promoted, it will be, and the promoted logical forms also activated. Category promotion was discussed in Section 3.3.9.

3.4.4 Parsing

As these logical forms combine through application, new tokens are generated with new categories and logical forms. When a new token is inserted into the lattice, all

[12] A pattern might recognize a word ending and generate two tokens of the appropriate boundaries for the stem and affix.

adjacent tokens are activated, and have a chance to combine with the new token.

This strategy is almost equivalent to an n^3 chart parsing strategy, but allows extra flexibility by letting logical forms to influence the course of parsing, as discussed in Section 3.4.1.

When a logical form combines with another and yields a null or false result (including failures due to Scheme errors), the partial parse is not added to the lattice, and parsing continues. This sort of semantic pruning was discussed in Section 3.3.4.

3.4.5 Final Decision Among Partial Parses

When parsing is complete, some set of named entities must be chosen to report as discovered entities. Sepia finds all partial parses that are not entirely subsumed by strictly larger partial parses, and reports all the constituents that compositionally generated this partial parse which are also marked with a named entity type.

3.5 Design for Error

To Err is Human, and to err in writing computer programs is the rul$_e$. Sepia's design makes it brittle, not only in the sense that it cannot know about every lexical item, but also in the sense that it is easy to write logical forms badly. There are some common types of errors that Sepia lexicons are prone to, and various ways of mitigating their effects.

3.5.1 Underlying Data Type of Arguments is Invisible

All logical forms must take into account the types of their arguments, if any, and there is little standardization. Arguments may be base scheme types, arbitrary Java classes, or any of the complex logical forms discussed in Section 3.3.11, and a lexicographer's only clue as to the type of object a function will receive is the category it can apply to. It is very easy to make a type error of this sort. Use of Hungarian-like category naming conventions (e.g., categories names which are expected to be strings begin

with "s", AtValSets with "avs") is useful but not foolproof.

3.5.2 Category Promotion Can Cause Infinite Loop

Category promotion is another source of error. One could accidentally create two rules that promote to the same category but have different underlying data types, which would make engineering functions to apply to that category difficult. One can also accidentally create a category promotion loop, which will cause the parser to enter an infinite loop.

3.5.3 Fault Tolerance

Sepia tolerates faults in the lexicon by limiting the damage they can cause. The infinite loop will eventually cause an OutOfMemoryException, which will cause the partial parse to be abandoned, and the rest of the execution will go on, and the memory reclaimed. A type error or any other exception will have similar effect.

3.5.4 Testplans

Sepia also provides a testing infrastructure that allows a lexicon developer to create testplans whose testcases specify a string, a part of that string to be recognized, the category, and a Scheme expression to build the logical form that the recognized object must match. Testcases may also be negative, where it is checked that the substring given does not have a logical form interpretation with the category specified and which matches the logical form given. As the test proceeds, errors and derivations for failed parses are shown. The testplan infrastructure helps lexicon developers quickly find and eradicate most bugs.

3.5.5 Short Semantics

Finally, a mitigating factor for bugs in the lexicon is that most semantics (so far) tend to be fewer than three lines long, and there is very little repetition of code. The

amount of context a lexicographer needs when writing a particular piece of semantics is small (the arguments, products, and sister functions that create similar products). These attributes make it relatively painless (I daresay even fun) to write lexicon for Sepia.

Chapter 4

Illustrative Examples

In the previous chapter, I showed how Sepia's parser evaluates lexical items to form compositional data structures. In this chapter, I will detail the proof-of-concept lexicon I created, and show how it interprets some interesting examples. In creating this lexicon, I have tried to remain true to my intuitions about the meanings of the words, and will clearly mark those cases in which I was forced to depart from those intuitions.

The basic challenge throughout this chapter is to capture enough ambiguity that the correct answer (or the set of correct answers) is always available, while reducing the number of available incorrect or inplausible possibilities. The goal is to show that Sepia can capture important elements of meaning in a useful, intuitive, and compositional way.

I begin with an overview of the contents of the initial lexicon, then show solutions to common problems in named entity understanding, and finally review problems that Sepia cannot handle. Throughout, I will refer to the Automatic Content Extraction (ACE) evaluation in which we plan to compete in four weeks. Many of the examples are taken directly from ACE guidelines, and I will discuss how Sepia will change to handle them.

12315	total fixed list entries
3712	female first names
1371	male first names
1024	last names
746	famous people
430	title cues
64	genderless first names
1655	locations (incl. cities)
401	countries
126	states (incl. abbreviations)
69	provinces
25	regions
13	U.S. territories
1884	universities
466	government agencies
264	sports mascots
65	news agencies

Table 4.1: Distribution of fixed lists in the lexicon

4.1 Overview of the Lexicon

The current lexicon was built in a relatively ad-hoc manner to fit examples. In the coming month, we will be preparing for the Automatic Content Extraction (ACE) evaluation, and will extend the lexicon in a data-driven process of iterative refinement.

The current lexicon contains 20195 total lines, including comments and whitespace as well as fixed lists and rules. There are only a few known verbs, and the coverage generally does not reflect what is required for the ACE task. The distribution of fixed lists is shown in Table 4.1.

For ACE, we will separate the fixed lists into a database, and we plan to add large lists of famous people, organizations, facilities, and locations, as well as a set of verb frames from PropBank [43] and Bonnie Dorr's publicly available LCS lexicon [21].

There are 12581 total unique keys in the final constructed lexicon. This does not mean that there were exactly 266 combinatory rules (including cue words), as one might guess by subtracting the number of fixed list entries. It simply means that there were at least 266 unique strings, patterns, or promoted categories that

```
;: de : person  first_name / capitalized_word
;: de : person  capitalized_word / capitalized_word
;: du : person  first_name / capitalized_word
;: du : person  capitalized_word / capitalized_word
   :
;: al : person  first_name / capitalized_word
;: al : person  capitalized_word / capitalized_word
;: ben : person  first_name / capitalized_word
;: ben : person  capitalized_word / capitalized_word
(lambda (lastname)
   (lambda (firstname)
     (.append (.append firstname 'name *0*) lastname)))
```

Figure 4-1: Particles that appear in mid-name. The eight separate rules take 11 lines of code, and result in four tokens specified.

rules were associated with. Figure 4-1 shows how a single string may have multiple associated rules.[1]

Of the 12581 keys in the lexicon, 58% were single words without punctuation. The breakdown by key type is shown in Table 4.2.

7248[2]	single-word
4789	multi-word
558	contain sub-word token sequences
20	are regular expression patterns

Table 4.2: Breakdown of single-token, multi-word, and multi-subword lexical entry keys.

[1]note also that some of the fixed list entries themselves may have been duplicate strings with different meanings, and may have been the same as some cues, so the subtraction is meaningless.

[2]not 7214 because 34 multi-word tokens also contain a first or last word that can break into sub-word tokens, e.g., John Boyle O'Reilly

4.2 Mechanics

This section of examples is intended to illustrate the mechanics of the parser and lexicon, showing how a set of lexical items can combine to capture both surface and meaning phenomena in simple cases. The section will also introduce some of the tools the current lexicon uses for entity recognition.

4.2.1 Function Application

Function application and category promotion are Sepia's bread and butter. Turning "a hundred and ninety-one days" into a duration of 191 day units is a straightforward process. We start with the lexicon fragment in Figure 4-2. The lexical assignment is as follows:

```
a => 1:socard_ones
hundred => 100:socard_hunds,
           {type=>cardinal; value=>100}:quantity,
           (λ a\socard_ones . (* a 100)),
           (λ a\socard_teens . (* a 100))
and => (λ u/socard_teens .
          (λ h\socard_hunds
            (if (equal? 0 (% h 100)) (+ u h))))
ninety => 90:socard_tens
          (λ q/socard_ones (if (> *0* 10) (+ *0* q)))
          (λ h/hyphen (λ q/socard_ones (if (> *0* 10) (+ *0* q))))
- => '():hyphen
one => 1:socard_ones
days => (λ q\quantity {unit=>day; value=>q})
```

While for a class in semantics, I would show several pages of derivation, here I will simply outline the steps the derivation takes:

```
;: a : socard_ones
1
;: one : socard_ones
1
;: two : socard_ones
2

;: ninety : socard_tens
90

;: :socard_tens: : socard_teens / socard_ones
(lambda (q) (if (> *0* 10) (+ *0* q)))
;: :socard_tens: : socard_teens / socard_ones / hyphen
(lambda (h) (lambda (q) (if (> *0* 10) (+ *0* q))))

;: - : hyphen
()

;: hundred : socard_hunds
100

;: hundred : socard_hunds \ socard_ones
;: hundred : socard_hunds \ socard_teens
(lambda (a) (* a 100))

;: and : socard_hunds \ socard_hunds / socard_teens
(lambda (u)
  (lambda (h)
    (if (equal? 0 (% h 100))
        (+ u h) )))

;: :socard_hunds: : quantity
(AtValSet. 'type 'cardinal 'value *0*)

;: days : duration \ quantity
(lambda (q) (AtValSet. 'unit 'day 'value q))
```

Figure 4-2: The lexicon required to parse "a hundred and ninety-one days"

1. *hundred* can apply to *a* to create `100:socard_hunds`. That is, a new token *a hundred* is generated, with category `socard_hunds` that has semantics `100`.

2. *ninety* can apply first to the hyphen, then to *one* to create a new token *ninety-one* with category `socard_teens` that has semantics `91`.

3. The special numerical non-coordination *and* can apply first to `91:socard_teens`, then to `100:socard_hunds` to create the new token *a hundred and ninety-one* with category `socard_hunds` that has semantics `191`.

4. The category `socard_hunds` can be promoted, and during promotion, a new object is generated based on the old one, still assigned to token *a hundred and ninety one* but with category `quantity` and semantics `{type=>cardinal; value=>191}`.

5. *days* now has a `quantity` to its left, and so it can apply, and create the largest token, *a hundred and ninety-one days* with category `duration`, that has semantics `{unit=>day; value=>{type=>cardinal; value=>191}:quantity}`.

Note that the end result is a nested structure of Attribute Value Sets, and that the inner AtValSet retains its category (denoted after the colon). If either the category `quantity` or the category `duration` were markable in a task, they would both be available to the parser's final selection algorithm.

4.2.2 Context Sensitivity

It is sometimes useful to find text between paired elements. A book title might often appear in title case (non-stopwords capitalized) between quotes.

Figure 4-3 shows a lexicon that would capture a series of title-case words between quotes. It would capture *"The Wizard of Oz"* by successively consuming `capwords`, `stopwords`, and `names` (not defined here) from left to right, appending them to a list, until it found a `title_end`, a capitalized word with an end quotation.[3]

[3]One may wish to allow non-capitalized words to also end titles, or quotations in general. The point here is that one can capture pairs, including pairs with certain properties, not the specific set of properties one may wish to capture in any domain.

```
;: of : stopword
;: a : stopword
;: the : stopword
*0*

;: /[A-Z][a-z]+('s)?[,:]?/ : capword
*0*

;: /"([A-Z][a-z]+('s)?[,:]?)/ : title_start
(list *1*)
;: /([A-Z][a-z]+\W?)"\W?/ : title_end
*1*

;: :title_start: : title_start / capword
;: :title_start: : title_start / stopword
;: :title_start: : title_start / name
;: :title_start: : name / title_end
(lambda (w) (append *0* (list w)))
```

Figure 4-3: A lexicon to capture Title-case words between quotes

```
;: /^[^\(\)]+$/ : nonparen
*0*
;: /\(/ : openparen
;: /\)/ : closeparen
'()

;: :openparen: : parenthesized / closeparen
(lambda (c) *0*)
;: :openparen: : openparen / nonparen
;: :openparen: : openparen / parenthesized
(lambda (w) (append *0* (list w)))
```

Figure 4-4: A lexicon to capture nested parentheses

Occasionally, it is useful to capture nested pairs, such as possibly nested parentheses. Figure 4-4 shows a lexicon to capture anything as nested lists from a set of nested parentheses.

Finally, though it's trivial to count cardinalities of lists in the semantics (Scheme is, after all, Turing-complete), it is worth noting that the category grammar itself is context sensitive, so long as standard function composition is available as an operator. Figure 4-5 shows a simple grammar to recognize the canonical non-context-free language $a^n b^n c^n$.

4.2.3 Extent Ambiguity

"John Smith" is a simple combination of a list-recognized first name and a list-recognized last name. "Sun He" on the other hand, would be very difficult to recognize as a name unless there were some solid cues, such as "Mr. Sun He met ...". The current lexicon, as I have written it, would not allow "He" as a last name even in this context, but this is a mistake of the lexicographer.

There are cases though, where names are ambiguous despite the most complete lexicon. Figure 4-6 shows a lexicon complete for parsing "I gave Mary Jane Frank's address", but two derivations are possible. "Mary Jane" may be the recipient, and

Two required rules:

base case: `b : done \ a / c`
generator: `b : (done \ a \ a / c) / (done \ a)`

Derivation for $n = 4$:

a a a a b b b b c c c c	apply last b to first c
a a a a b b b (done \ a) c c c	apply generator from last b
a a a a b b (done \ a \ a / c) c c c	apply to first c
a a a a b b (done \ a \ a) c c	compose generator with the partial derivation
a a a a b (done \ a \ a / c \ a) c c	compose generator with the partial derivation
a a a a (done \ a \ a / c \ a / c \ a) c c	apply to last a
a a a (done \ a \ a / c \ a / c) c c	apply to first c
a a a (done \ a \ a / c \ a) c	apply to last a
a a (done \ a \ a / c) c	apply to first c
a a (done \ a \ a)	apply to first c
a a (done \ a \ a)	apply to last a
a (done \ a)	apply to last a
done	

Figure 4-5: Categorial Grammar is context sensitive with composition.

"Frank's address" the gift, or "Mary" the recipient and "Jane Frank's address" the gift.

It cannot be that "Mary Jane" is the recipient and "Jane Frank" is who the address pertains to. The reason is that when "Mary Jane" is consumed by "gave", the new token becomes "gave Mary Jane", and that can only apply to adjacent objects.

For "Jane" to have both roles, there would have to be two objects whose head was "address". One would be "Jane Frank's address" and the other would be "Frank's address". Only the latter is adjacent to "gave Mary Jane", and so only the latter can be a candidate for application.

Similarly, "Mary Jane Frank" becomes a partial parse token of type person, but once "gave" consumes "Mary Jane Frank", the adjacent element is a function of category `NP / NP \ person`, and "gave Mary Jane Frank" would require a simple NP.

At the end, the parser chooses the two largest parses, which are ambiguous, and marks all four subsumed named entities. It does not mark which pairs are consistent, so to an outside observer, or to a subsequent module, it looks as if "Jane" could be used twice. This is one kind of error to be careful of when passing more than one possible parse in a pipeline, and should be better addressed in Sepia's future work, as part of Sepia's goal to serve as a component in a larger question answering system.

4.2.4 Nested Entities

I showed in the first example that an `AtValSet` may hold another `AtValSet` as one of its values, and return it as a candidate if it turns out to be one of the largest spanning logical forms. This feature can be directly applied to the Automatic Content Extraction (ACE) task.

Two of the ACE guidelines require that:

- "Provo, Utah" evokes two entities: the city of Provo in Utah, which is the whole string, and the state of Utah itself. Thus ACE asks that the both nested annotations be present.

```
;: I : person
(.get *context* 'speaker)

;: Mary Jane : firstname
;: Mary : firstname
;: Jane : firstname
;: Frank : firstname
;: Frank : lastname
*0*

;: :firstname: : person
(AtValSet. 'name *0*)

;: :firstname: : person / lastname
(lambda (l) (AtValSet. 'name (list *0* l)))

;: 's : NP / NP \ person
(lambda (p) (lambda (o) (AtValSet. 'name o 'belongs-to p)))

;: address : NP
(AtValSet. 'name *0*)

;; gave : S \ person / NP / person
(lambda (recipient)
   (lambda (gift)
     (lambda (giver)
       (Event. 'give 'agent giver 'patient gift 'receiver
recipient))))
```

Figure 4-6: Lexicon for "I gave Mary Jane Frank's address"

- The adjective "Russian", as in "Russian fighter plane" evokes the Geo-Political Entity (GPE) "Russia"; therefore, even though the fighter plane itself is not to be marked, "Russian" is, as a GPE.

Both of these can be readily implemented in Sepia. For Provo, the rule that combines a likely (or known) city name with its state must simply keep the state argument as a component of itself:

```
;: :city: : GPE / city / comma
(lambda (comma) (lambda (state) (.append *0* 'in state)))
```

"Russian" doesn't have an explicit country name nearby, but it can create one:

```
;: Russian : person_adj
(AtValSet. 'group #t 'from (.setCategory (AtValSet. 'type 'country
'name "Russia") (Category. "GPE")))
```

At the same time, ACE make two counterintuitive requirements:

- "Russian", when used as a person ("seven Russians were among..."), is judged *not* to evoke a Geo-Political Entity.
- names such as "The New York Times" are atomic, and do not evoke a Geo-Political Entity such as "New York" in this case.

Whatever the reasoning, the distinction is easy to implement—we simply do not create the referent GPE. If a semantics later wants to look up the relevant GPE, it

can simply ask the lexicon.

```
;: Russian : person
(AtValSet. 'nationality "Russia")
;: Russians : person
(AtValSet. 'group #t 'nationality "Russia")
;: The New York Times : organization
(AtValSet. 'type 'news 'name *0*)
```

The psychological reality of including a single level of referent (an actual **AtValSet within another**) or only including references (a way to find the other **AtValSet**) is beyond the scope of this thesis, but either choice can be effectively modelled.

4.2.5 Type Ambiguity

Entities in the ACE task are often ambiguous as to their category depending on their role in the sentence context. The canonical example is that of the White House. It is considered an organization when it is said to take action, "The White House vetoed the bill", but a facility when the physical structure is in question, "Visitors streamed through the White House."

One difficulty is in differentiating the physical path from the metaphoric one: "Bills streamed through Congress and the White House in a marathon effort to pass new clean air regulations today."[4] One can argue that bills do pass trough the White House facility on their way to becoming law, but there is no facility called "Congress".

A possible solution is to allow an ambiguous entity to be truly ambiguous (specifying both types) in isolation. If one of the categories is used in a larger construction, the subsumed category that is not used will be pruned. In the White House veto example, the lexicon in Figure 4-7 allows both the facility and organization interpretations, but when "vetoed" applies, it creates a subsuming element using the organization. If

[4]A reader points out another possibility: "Bills streamed through the White House on the first annual Clinton Impersonation Day"

```
;: the White House : organization
;: the White House : facility
(AtValSet. 'name *0* 'type 'government)
;: vetoed : event  organization / document
:
```

Figure 4-7: Entities can have multiple categories. If one type is used in a larger context, it will be preferred.

it stores the organization's logical form within itself, then that logical form will be marked, and the subsumed facility will not.

That strategy does not help for entities that are ambiguous between syntactically similar types like person and organization. It is very difficult to tell how "Dana Farber" should be marked in many contexts without *a priori* knowledge that it is an organization. The disambiguating information may be in a coreferenced nominal or pronoun, but coreference is not yet in place in Sepia.

4.2.6 Coordination

Coordination under CCG combines two (or more) logical forms with like categories to form a single logical form that contains all the others, has the same category as each of its constituents, and specially handles application to properly map functions over a list of arguments, or to map a list of functions over a single argument, depending on whether the coordination is in a function or argument position.

The phrase "two or three hundred miles" can be processed by a lexicon similar to

the one presented in Section 4.2.1:

```
;: two : socard_ones
2
;: three : socard_ones
3

;: or : * / * \ *
(lambda (x)
   (lambda (y)
     (Coordination. 'or x y)))

;: hundred : socard_hunds \ socard_ones
(lambda (a) (* a 100))

;: :socard_hunds: : quantity
(AtValSet. 'type 'cardinal 'value *0*)
;: :socard_ones: : quantity
(AtValSet. 'type 'cardinal 'value *0*)

;: miles : length \ quantity
(lambda (q) (AtValSet. 'unit 'mile 'value q))
```

Because both socard_ones and socard_hunds are promoted to **quantity**, there are three possible ways for the coordination *or* to apply. It can apply to:

- *two*:socard_ones and *three*:socard_ones.
- *two*:quantity and *three*:quantity,
- *two*:quantity and *three hundred*:quantity.

One of those ambiguities is real, and one is an artifact of category promotion. The

model correctly predicts that *two* cannot coordinate with *three hundred miles* because the latter is not a bare quantity, but a length.

4.3 Challenges

In this section I describe problems with the current state of the lexicon, with Sepia's architecture, and most fundamentally with CCG as a parsing formalism in terms of what it cannot capture.

4.3.1 A Hundred and Ninety One Revisited

In the first chapter, I introduced an example, "Their scores were a hundred and ninety-one respectively." In the first section of this chapter I showed how to process "a hundred and ninety-one" into 191. The challenge is to understand that there are two separate numbers, 100 and 91, in the sentence containing "respectively".

The adverb "respectively" intuitively changes a verb's default way to map conjoined subjects and objects from an all-pairs mapping to an ordered list mapping. "respectively" thereby causes the verb to require that both its subject and object be coordinations of the same cardinality.

Sepia's lexicon does not yet have verbs, so I am forced to use a counter-linguistic solution for now, noting that unlike other systems, there is a solution! I make the simplifying assumption that "respectively" will appear after the verb (as opposed to "They respectively scored a hundred and ninety-one.") and create two patterns which catch a conjunction:

```
;: respectively : * / *
;: respectively : * \ *
(lambda (c) (if (instanceof? c Coordination)
      (.setRespectiveMapping c)))
```

This works because of two key features in Sepia: In the first place, if the argument

(which can be of any category) is not a Coordination logical form, then the semantics "fails", returning nothing, and the partial parse is dropped (Section 3.3.4). Thus the numeric interpretation of *a hundred and ninety one* cannot combine with *respectively* because it is underlyingly an `AtValSet`, not a `Coordination`.

Second, in the final parser decision, partial parses which wholly subsume other partial parses occlude the subsumed partial parse (Section 3.4.5). Because *a hundred and ninety-one respectively* strictly encloses *a hundred and ninety-one*, the latter will not be considered for combination. This applies transitively to any combination with *a hundred and ninety-one*: unless the combination finds another way to use *respectively*, it will be smaller than a combination which does use *respectively*, so the correct interpretation will win.

I hope to revisit this issue after developing a verb lexicon to explore a more linguistically satisfying solution. Even so, it is satisfying to have such problems in an implemented system.

4.3.2 Overgeneration of Coordination

One problem I came across in developing a lexicon for Sepia is that coordination overgenerates in some instances. In order to parse "Mr. and Mrs. Hill", I specify "Mr." and "Mrs." as functions over a name.

```
;: Mr. : person / name
;: Mrs. : person / name
(lambda (name) (.append name 'title *0*))
```

Mr. then conjoins with *Mrs.* to create a new function that takes a single name and generates a coordination of people: Mr. Hill and Mrs. Hill.

"Jimmy and Rosalyn Carter" can be conjoined two ways. Common first names look for a last name to apply to, so they are functions that can conjoin. A common last name looks for a first name to apply to, and finds the conjoined firstnames as

arguments.[5]

But the same mechanism causes problems:

- The meeting between Mr. Hill and DeGeneres yesterday...
- Prince Charles and Fergie

Clearly there is no relevant Mr. DeGeneres, nor a Prince Fergie.

In fact, English has special forms to indicate that a title may apply to multiple names: "Drs. Brown and Whitman"

This problem was a surprising finding for me, but is apparently well known in the literature. Colin Phillips gives one comprehensive account of such overgeneration phenomena, and suggests that parsing in linear order (strictly left to right) is a solution [56].

I was very happy to find this literature after the fact, because it indicates that Sepia was useful in finding one theoretical problem with CCG theory, and will hopefully be useful in establishing the solution and finding other linguistic problems. I look forward to implementing Colin Phillips' solution as future work.

4.3.3 Coordinated Nominals

One goal in the ACE task is to understand nominals, including coordinated nominal expressions. A nominal must generate as many entities as it has heads.

- two heads ⇒ two entities: "Two accountants and three linguists"
- one head ⇒ one entity: "several descriptive and generative linguists"

The first case will generate two entities via the mechanism previously described.

Assuming that *descriptive* and *generative* are functions on *linguists*, the second phrase would create a `Coordination` equivalent to "several descriptive linguists and several generative linguists". This is clearly not what is intended in the ACE guidelines, and arguably not what is meant by the speaker. Again, the coordination would

[5]...and generates another version of the same coordination. As an aside, because the two coordinations are `.equal()`, the `HashSet` that stores logical forms will silently drop one.

have to have a different mapping (apply both functions serially to create a single new object) than usual. It is not clear how one would know which mapping to specify for any given function, or what the mechanism for specifying it in the lexicon might be. Perhaps * is too general a category, and each type of entity should specify the particular flavors of coordinations that may apply to it. Then adjectives like "French" and "German" could specify that they unlike other adjectives should apply in parallel: "the French and German soccer teams", "the old and young inhabitants", "the first and second year students".

4.3.4 Noncontiguous Constituents

As mentioned in Section 4.3.1, there is not currently a satisfactory treatment of verbs and adverbs. This can get in the way of recognizing nominal expressions that happen to have an adverb in the middle, e.g., "I met some people yesterday who love chess"

Ideally we would recognize "people who love chess" as a nominal expression. This is impossible in the current framework because elements can only combine with immediately adjacent ones, so "people who love chess" simply cannot form a constituent. One solution that would be supported would be an "ignore-me" marker. The semantics for an adverb could include adding links in the lattice around it, thereby making it possible for the normal rules to apply as if the element weren't there. Another might be an explicit movement, as the lattice also allows remove and insert operations, while preserving adjacent link structures. As our team moves to cover more of language, including pronouns and quantifiers, more sophisticated structural transformations may become necessary.

4.3.5 Mention Detection and Coreference

Sepia should find named, nominal, and pronominal mentions of a named entity within a text using the semantic features of the entities in question. The semantic `match` operator, now primarily used to ensure that parsed logical forms match the logical forms specified in their test cases, is one possible way of determining which entities

should be coreferenced. The `match` operator defines a form of entailment. It is asymmetric and transitive. For an `AtValSet`, a matches b if b has every key in a, though it may have more, and for every key has at least the values in a in the same order. For a `Coordination`, a matches b if the coordinations have the same type and a matching set of elements in any order. As a first approximation, entities which match either way should be marked as possibly coindexed. But crucially, if two entities match a third but do not match each other, then they must not be coindexed, and the index of the entity they both match will be in question.

4.3.6 Real Ambiguity

Consider the sentence, "On August twenty first, two thousand protesters assembled at Hawaii's state capitol building." A transcribed speech corpus, for example, might have numbers spelled out in this way. It is not clear whether there was an unspecified number of protesters there in the year 2000, or whether the current year was assumed, and the protesters numbered 2000. Whether a speech recognizer was able to recognize enough of a pause to generate a comma after "two thousand" or not is an unreliable way to decide between interpretations.

Similarly, for the ACE task, "...the State Department spokesman Uno Little..." must be marked as two entities (as if it were an apposition), rather than one entity for, "...told State Department spokesman Uno Little that...". They argue that there might be a missing comma after "spokesman" in the first example. It is not clear that the two are coindexed at all: consider, "the State Department spokesman Uno Little spoke with yesterday".

In the first case, if Sepia recognized the possible date including year, and it recognized the phrase "two thousand protesters" as a group of people, it would still not output "August twenty first" as a possible date because it is entirely subsumed by the larger date. Only if it understood the whole sentence (incorporating both date and protesters into a single top-level partial parse) would the smaller date become available to a later stage in processing (if any).

The second case is similar: the only way "Uno Little" would be considered a name

mention without the title would be if a larger constituent than "State department spokesman Uno Little" included both. In this case though, it's a feature rather than a bug.

4.3.7 Domain Recognition and an Adaptive Lexicon

Creating domain-specific lexicons and recognizing the domain of a particular text segment has been a very successful approach to increasing performance. This is difficult to do as part of Sepia, though Sepia could certainly be called with separate lexicons in different instances. This approach is particularly helpful in recognizing the metonymy in a sports-text instance like "Boston defeated Miami 21 to 15". I hope to develop some alternative in Sepia that will accomplish similar behavior without a strong separation of lexicons.

Chapter 5

Evaluation on Question Answering

In the previous chapter, I showed examples of how Sepia handles individual test cases. In this chapter, I will show how Sepia has performed in intrinsic and extrinsic evaluations related to query expansion, and how I approached the new TREC list question set.

In a test of performance on the manually annotated question set from the 9th, 10th, and 11th TREC competitions, Sepia annotated named entities with an 85% F-measure against a MUC-like annotation standard[1]. A detailed error analysis is reported below; the character of the errors suggests that lexicon development time is the constraining factor.

As we prepare for the upcoming Automatic Content Extraction common evaluation, the lexicon remains under active development. The same set of questions as used above, annotated according to much more stringent ACE guidelines, and using a lexicon three weeks newer, today yielded 25% recall, up from 14% yesterday.

In end-to-end evaluation within a question answering system, I integrated Sepia into two question answering components. In query expansion a few simple strategies on identified names marginally improved document ranking. In list question answering, Sepia was used to identify named entities in retrieved passages, but there were too few relevant questions to make a statistical analysis. In both end-to-end anal-

[1]This cannot be directly compared against the MUC-6 results, where other knowledge-based systems scored 76% to 86% F-Measure, because this is a different corpus, without official judgements.

yses, as predicted in Related Work, the common problem was that the task could not be greatly improved by even the best Sepia results, partly because named entity understanding is not the limiting factor, and partly because only very shallow understanding was actually used.

Sepia's results are promising, though not yet competitive with more established systems. I hope to improve Sepia's performance significantly in preparation for the ACE competition, and will integrate Sepia into our larger question answering systems in ways that will hopefully have greater impact on end-to-end performance.

5.1 Named Entities in Queries

The first numerical evaluation of Sepia was an intrinsic evaluation, designed to test how well Sepia performs in the task for which it was created. Having created a lexicon for Sepia based on AQUAINT documents, I chose to evaluate that lexicon on a different but related corpus: the set of TREC questions. By evaluating absolute performance on the same corpus that I later used for the query expansion test, I hoped to shed light on specific error types in both.

5.1.1 Experimental Protocol

The annotated corpus was the set of 1321 questions from the 9th, 10th, and 11th TREC competitions. I was developing the lexicon on actual documents from the AQUAINT corpus, but had not looked at the questions as part of development prior to August 8th.[2]

Until August 8th, I had been developing Sepia lexicon based on the AQUAINT corpus, and aimed to correctly understand only name-level expressions[3]. Expressions were categorized in a manner more consistent with the Message Understanding

[2]I did see these questions during last year's TREC competition, but not in the intervening 11 months, and did not remember more than a handful of famous examples.

[3]an ACE term indicating that the name of the entity should appear in the named entity mention (Presidential hopeful Howard Dean"), as opposed to a nominal phrase with the same referent ("the Presidential hopeful") or a coreferring prominal reference ("he")

Conference (MUC) guidelines than with the newer Automatic Content Extraction guidelines. In particular, markable categories included locations, geo-political entities (though these were labelled as locations, and country adjectives were not marked), persons (only singular, no groups of people represented as person), and organizations, but not facilities.

The Sepia evaluation run used Sepia (parser and lexicon) in the version that was used for this year's TREC competition, dated August 8th under revision control. The full question list, Sepia's responses, and annotator judgements are shown in Appendix B. Where an entity had incorrect bounds, the error was counted only once rather than as one missed entity and one spurious entity.

5.1.2 Results

There are 1321 official questions from the previous three years of the TREC competition (numbered 200–1890, with gaps). There were 776 ground-truth entities and 711 recognized entities, of which 631 were correct.

Precision: 628 / 711 = 88.33 %

Recall: 628 / 777 = 80.82 %

F-Measure: (p+r)/2 = 84.57 % height

The 83 incorrectly marked entities can be broken down as follows:

Miscategorization

There was no case where Sepia marked something where there was no entity, but it sometimes marked the wrong entity type or marked an entity that was not markable (such as a facility). There were in fact several facility keywords (such as gate, gates, stadium, and house) which were cues for organization and should not have been. There was a cue word "Star" for newspapers of that common name.

None of these took into account any context other than what is marked.

```
person:  Where is [Santa Lucia]?
```

person: Who owns the [St. Louis Rams]?

person: Where is [Ocho Rios]?

person: When was the battle of [Shiloh]?

person: What city does the [Tour de France] end in?

person: ...member of the [Pink Floyd] band?

location: ...[Bill Gates]...

location: ...[New Coke]...

location: ...''Brave [New World]''?

location: ...sing at the [Lincoln] Memorial?

location: American patriot Paul [Revere]?

location: What are the two houses of the [Legislative branch]?

location: [Which mountain] range in ...

location: [Which river] runs through...

organization: Where did the [U.S. Civil War] begin?

organization: When did [Yankee Stadium] first open?

organization: Where was the movie, ''Somewhere in [Time]'' filmed?

organization: What was the original name before ''[The Star] Spangled Banner''?

organization: What is another name for the [North Star]?

organization: What is the street address of the [White House]?

organization: When was the [White House] built?

organization: ...first female [United States Representative]?

organization: Where is the [Shawnee National Forest]?

Missing rule to combine correctly recognized parts

Sepia would often correctly identify a city and country adjacent to one another with a comma between, but I had forgotten to add a rule that combined names with countries, as I combined names with states. Thus examples like the following were

marked incorrect:

[loc: Madrid] , [loc: Spain]

[loc: London] , [loc: England]

[loc: Bombay] , [loc: India]

[loc: Rome] , [loc: Italy]

Windsor, [loc: Ontario]

Parsippany, NJ?

[loc: Kentucky] [loc: Horse Park]

There was one instance of a name not having a hyphen-combination rule:

[person: Rosanne Rosanna]-[person:Dana]

Name heuristic made incorrect decision

The only statistical code in the system was explicitly specified by the lexicon: a unigram count from the AQUAINT corpus for how often a word appeared capitalized versus uncapitalized serves as a naive heuristic for whether something is a good name. If Sepia had had more relations, it would have made sense to simply posit all reasonable names and let the grammar decide whether two words should be a name or not. By resorting to this simple heuristic, Sepia made predictable mistakes.

In each line below, Sepia tagged the bracketed portion as a name, but decided that the other portion appeared in lower case often enough that it was unlikely to be part of a name, but might instead be a headline word. This heuristic is not reasonable in the question analysis genre, where capitalized words rarely appear except as part of a proper name, but I did not make any attempt to adapt to the new genre.

Babe [Ruth]

[Henry] Clay

[Aaron] Burr[4]

[4]A burr is a dental instrument.

[Jude] Law

[Nicholas] Cage

[Billy] the Kid

[Winnie] the Pooh

[Sharon] Stone

[Prince Andrew] and Fergie (missed Fergie)

Teddy [Roosevelt]

[Marco] Polo

[Robert] Frost

[Tom] Cruise

In these cases the location appears only because the name did not subsume both tokens, again because the word was otherwise often lower case.

[Judy] [location: Garland]

[Paul] [location: Revere]

In one case, the heuristic overgenerated:

''What university was [person: Woodrow Wilson President] of?

Punctuation bug

A bug I had not noticed before this experiment had to do with recognizing entries that consisted of multiple sub-tokens when those sub-tokens did not make up all of a larger token.

In this case the recognizer for initials did not recognize them because there was no space between "C." and "S.".

C.S. [Lewis]

[Scarlett O]'Hara's

U.S.?

Washington D.C.?

Parsippany, NJ?

[Mount St]. Helens? (Helens not recognized as a capitalized-word!)

However when I fixed this bug, a new incorrect person appeared:

`Who is the evil [H.R. Director] in "Dilbert"?`

Missing Cue Words

There were many location and organization keywords missing which would have aided recognition. After this evaluation, we looked at lists of such cue words compiled statistically from a large corpus, and we would have done well to incorporate such lists.

`agency` and `company` were missing organization cues. Less obvious missing organization cues would be `United` x or the whole set of sports terms `Series`, `Cup`, `Crown`, `Bowl`, `Derby`, etc.

Location cue words, such as `Alley`, `Mountains`, and `galaxy` would be useful, but even more useful might be encompassing rules like `battle of <location>` or directly for questions `Where is <location>`

I had not noticed a person's relations in looking at the training corpus, so `<person>'s son, wife, mother,` etc. were not in the lexicon.

`Mother` was not an available title, so "Mother Angelica" was missed. "Vasco da Gama" was missed because I did not have a linking "da", despite having "de", "du", "di", and many others as possible name linking cues.

`How did <person> die?` occurred twice in the questions, and my name heuristic had passed over both people. A variant on this was common in the training corpus, but as a negative example in headlines, I had to avoid `<title> Dies` as a person.

Similarly, Sepia could incorporate both internal information about sports team name structures and explicit external cues like "sports team", from examples like,

`What kind of a sports team is the [loc: Wisconsin] Badgers?`

`What kind of sports team is the Buffalo Sabres?`

Missing List Items

Gazetteers of people, fictional characters, important artifacts, and other items

difficult to identify heuristically would have helped. There is little other way to know that a "Declaration of Independence" is not an unknown declaration from a place called "Independence", or that "Winnie the Pooh" is a fictional character, and not a person, while "Billy the Kid" is a person.

The North and South poles are probably best entered as list items, but then it becomes impossible to recognize "North and South Poles". Instead, there should be very specific rules that indicate what sorts of geographic poles one might talk about.

Additional list items would likely be useful in the following missed cases as well:

```
Who is Darth Vader's son?
When is Snoop Dog's birthday?
How did Mahatma Gandhi die?
How long did Rip Van Winkle sleep?
When was the Triangle Shirtwaist fire?
What is the Islamic counterpart to the Red Cross?
How tall is the [Sears] Building?
Who invented the instant Polaroid camera?
Where was the first McDonalds built?
How much are tickets to Disney World?
Where did Kublai Khan live?
What year was the first [Macy's] Thanksgiving Day Parade held?
What is Buzz Aldrin's real first name?
What award did [Sterling North]'s book "Rascal" win in 1963?
What TV series did Pierce Brosnan play in?
```

5.1.3 Discussion

There were two primary types of mistakes in the results above: missing lexical items, and incorrect statistical name decisions. The name heuristic was meant to be the most trivial possible guess, and remained in the lexicon longer than intended because it worked surprisingly well for the trivial amount of effort it took to write. In replacing it, I would like to use a more principled approach.

The missing lexical items are easy to add, but perhaps harder to find. A process of iterative refinement works well for finding cue words in the training corpus, but it might be well complemented by a statistical tool that found words similar to the cue words in a large corpus of the same genre, in an unsupervised way. The lists of candidate cue words, their relative frequencies, and their concordance data from the corpus might serve as an effective way to discover the most relevant cue words and surrounding syntactic alternations.

Building larger lists of cue words will increase recall, but will also increase overgeneration. For example, "The Star" may be a wonderful name for a newspaper, but is inappropriate in the quoted string, " "The Star Spangled Banner" ". In Sepia, the most effective way to curb overgeneration is to have rules to understand the larger context of the candidate. In this example, both having the song on file and recognizing quoted strings might help. For other examples, a knowledge of verb selections might be key: consider whether the White House as organization or facility is more likely to be built, or to have a street address, or to veto a bill, or to make an announcement. Therefore, adding verb frames from a source like PropBank or an LCS lexicon will be a high priority in creating the competition system.

In preparing for the competition system, we have annotated the same set of questions according to ACE annotation guidelines, and currently have 24.56% recall of the new annotations. While others report 70% performance, we have been creating lexicon for this task for only a matter of days.

We did not rerun this evaluation, of course, because that would be mixing training data with test data, but for a more positive outlook on cases we handle well, please see Appendix A. It contains our regression test suite, including examples of semantics we correctly parse.

5.2 Integration and End-to-End Evaluation

We integrated Sepia into two components of our TREC 12 / AQUAINT Question Answering competition entry: Query Expansion and List Questions. Sepia became

ready for integration too late to be used as part of our primary factoid question answering system. In the query expansion module, Sepia served as one of several annotators to mark phrases in the question that should be treated as unbreakable units. In the list question module, Sepia's category distinctions were linked to a question's analyzed focus, and annotated retrieved passages to find candidate list items.

5.2.1 Query Expansion

We thought that Named Entity understanding would be able to improve question answering via query expansion by identifying named entities in the query and specially treating them. With normal keywords, our boolean query system required all keywords to be present in a document, but allowed inflected and derivational morphology variants in place of a query term. When a string was identified as a named entity, these expansions were no longer allowed. When a multiword string was identified as a named entity, then the query required either the exact string of the multiword entity, or its component words in proximity of two words from each other. When a named entity was a person, the query expander allowed all first names to be dropped, but not last names.

Sepia was not the only phrase-identification component in our query expansion system. A gazetteer, Omnibase, identified many fixed strings, there was a simple part-of-speech-based heuristic noun phrase tagger, many verb-particle constructions were also tagged, and quoted strings were identified and treated similarly.

Our first experiment was to compare the query expander with no phrase identifiers with the query expander using only Sepia. Indeed, Sepia made a relative improvement over baseline. Note that Mean Reciprocal Rank (MRR) is worse when it is higher,

and Sepia in this case made the MRR worse.

	Baseline	Baseline + Sepia	Relative Improvement	p-value
Precision	2.1%	2.5%	16.6%	**0.00095**
Recall	79.8%	80.9%	1.5%	0.10
MRR	0.188	0.204	8.16%	**0.0045**

We then measured the performance of all components against the performance of all except Sepia. Here, Sepia seems to have made a marginal improvement in both recall and MRR.

	All - Sepia	All	Relative Improvement	p-value
Precision	4.25%	4.37%	2.81%	0.19
Recall	80.66%	81.41%	0.93%	**0.0021**
MRR	.204	.200	2.05%	**0.0082**

However, of the 711 Sepia answers given, the query expansion system only used 447 (others were overridden by other methods), and only 326 of the entities were multi-word. In comparing performance on only Sepia-marked questions, and comparing performance on only multi-word annotated questions there were no significant differences.

The worse MRR measure of Sepia against baseline can be best explained by noting that when Sepia annotates a phrase, the phrase's inflectional variants are no longer counted toward the document score. Thus correct documents with multiple inflected mentions would no longer be scored higher than documents without multiple mention forms, and so would lose place in rank.

The only part of Sepia's semantics that this measure used, however, was a distinction between first and last names. For future evaluations, it would be important to find more semantic ways to expand queries.

5.2.2 List Questions

Sepia was integrated into list questions by annotating candidate passages retrieved by the document retrieval engine. The list answer component, knowing about some

of Sepia's finer grained classes like for example corporate-organization, government-agency, educational-organization, news-organization, or religious-person, would use these subcategories if they appeared as the focus of a question. Unfortunately there were only three questions out of the 50 where a Sepia class was the focus of the question, where the Infolab's Omnibase gazetteer didn't already mark entities to cover the focus (as with countries and cities). These three questions were:

31:List companies that manufacture tractors

38:What are institutions that have conferred an honorary degree on Nelson Mandela?

45:Who are authors who have written books about near death experiences?

In all of these cases, a simpler part-of-speech-based noun phrase chunker found the same entities as Sepia did, so the performance did not change due to Sepia.

Results are promising in that Sepia understood many of the correct responses, and a major step forward in this task would be to integrate Sepia into the question analysis portion as well as the candidate generation portion of list question answering, so that the mapping between question and answer semantics is more easily usable.

Chapter 6

Future Work

I have mentioned opportunitites for future work throughout this thesis, but will collect them here, from the most urgent short range extensions of the existing system, to longer range research goals based on Sepia's central theme: compositional semantics without a pipeline.

6.1 ACE

The ACE competition will begin on September 29th, 2003, and each competitor will have five days to run their systems on the provided test data and return the results for testing.

There are three major extensions I will undertake for this competition: adding large gazetteer, a typographic case repair system, and a coreference system.

The first is motivated by a need to add large lists of known named entities including famous people book titles, locations, and facilities. The Infolab already has such large lists, and a well established database service for them, called Omnibase [40]. My first priority will be to add JDBC as an access method for the lexicon, and thus define large classes of entities without memory overhead. In addition, it would be useful to obtain data from the Alexandria Digital Library Project [18] with respect to known organizations and facilities.

Because poorly cased text plays such a large role in ACE, a second extension

would be an automatic case repair mechanism. We could rewrite parts of the lexicon to be less sensitive to case, or Sepia could use a preprocessing step such as Lita *et al.*'s truecaser [48]. The first would fit the theme better but the second might be more prudent, if it is possible, given the time remaining.

Finally, I hope to implement a minimal coreference system that would use the existing `match` operator to find compatible names and nominals (See Section 4.3.5), and to add a heuristic pronominal resolver.

6.2 Formalism

In the longer term, there are several fundamental changes I would like to make to Sepia's CCG parser.

The most obvious extension is addition of probability values to each lexical item, not in the visible lexicon for human consumption, but as an attribute that could be trained on a corpus. The more difficult part is to find a corpus to train the parser on, but Hockenmaier and Steedman have shown promising results in this area [36, 35].

Another extension to the formalism would be to add unification or linguistic feature-checking to a set of syntactic features that could be stored with every logical form. This would give a way independent of semantics to ensure consistency of person, number, gender, and other syntactic features. Also, by removing these from the semantics into a separate mechanism, it might become easier to "back off" from the purely syntactic constraints, as humans apparently do routinely, when they conflict with the semantic interpretations.

The third set of improvements would address some of the issues of overgeneration, psychological plausibility as a language model, and efficiency issues related to allowing more ambiguity than pipelined systems. I would like to explore and implement formalism changes such as Jason Eisner's normal form for CCG [25], Jason Baldridge's CCG formalism extension, Set-CCG [4], and parsing only in linear order [56]. I would also like to learn more about and model other psycholinguistic predictions, including for example Dependency Locality Theory [29].

6.3 Extending Sepia: Bigger Concepts and Smaller Tokens

In pursuing compositional semantics without a pipeline, it makes sense to attempt to build larger propositional structures. such as Lexical Conceptual Structures, starting from a phonemic or morphemic level. The advantage of building larger structures is simply better understanding and better pruning of incorrect hypotheses by their semantic context. The advantages of starting with smaller units of meaning than words include better portability to other languages, better robustness to noisy input such as broadcast news or even speech. and a framework in which to explore the compositional semantics of subword elements.

6.4 Integration with Vision, Common Sense

A final exciting future direction would be to use Sepia's Java interface capabilities to communicate with vision and motion understanding. In one scenario, Sepia could act as a query front end to a large video archive, translating user questions into the CLiViR formalism [41].

In another scenario, Sepia could have constant bidirectional communication with a vision and imagination system, as described in Winston's Bridge Project [10, 66]. Sepia would both accept visual-semantic constraints and generate language-semantic constraints in a common infrastructure designed to let the combined language and visual system understand things which would be meaningless or ambiguous to either component alone.

Integration with common sense applications would also be interesting. I would like to explore using the OpenCyc [17] and OpenMind [60] projects as sources of semantic plausibility. Moreover, Sepia could be used in some cases to help generate candidate generalizations like "The Secretary of State works for the Department of State" and "The Department of State is the same as the State Department" for users of a system like OpenMind to manually verify.

Chapter 7

Contributions

The goal of the Infolab is computer understanding of natural language, focusing on natural language question answering as a development application. This thesis contributes to those goals in several ways. I have:

- Framed the content-extraction task as a problem in compositional semantics.
- Designed and implemented Sepia, a pilot framework for research in compositional semantics.
- Suggested named entity understanding as a rich new source of test cases for computational semantics.
- Implemented a named entity understanding component usable in question answering for query understanding, query expansion, and identification of candidate answers.
- Integrated Sepia with our question answering framework for participation in AQUAINT/TREC 12 Question Answering evaluation.
- Identified key areas of progress necessary to do well in the upcoming Automatic Content Extraction evaluation.

I hope in the coming weeks to achieve state-of-the-art performance in named entity understanding, as measured by the ACE entity detection and tracking task, and thus

to provide a foundation for simple semantic inference in future question answering studies. The preliminary results of evaluation are promising, though not yet competitive with established systems. Integration with question answering components yielded some valuable lessons.

Appendix A

Test Cases

The following are the test cases currently in use for regression testing Sepia. They serve to illustrate in detail the current capabilities of Sepia. The format of these test cases is a four column tab separated file. In the first column is a string to be processed. In the second column is a category of entity that must have been recognized, preceded by a '!' character if the testcase is negative to ensure that the specified semantics were *not* recognized. The third column is the portion of the original string that must be recognized. The final column indicates the semantics that the recognized entity must `match` in order to be counted correct. The `match` operation for `AtValSets` requires that all of the attributes and contents in the test plan also be present in the data structure. but not vice versa. Text after a comment mark ('#') is ignored.

A.1 Person

```
TYPE person

#< some common overgeneration bugs
Panel OKs Plan for Reagan Monument By CARL HARTMAN WASHINGTON (AP) !person Reagan Monument (AtValSet.)
U.S.A. Reagan License Plates Approved LOS ANGELES (AP)           !person Reagan License Plates (AtValSet.)
but former President Ronald Reagan's spirit infuses Campaign 2000 person President Ronald Reagan (AtValSet. 'name "Reagan")
on May 24 we called a General Assembly for several reasons       !person May (AtValSet.)
on May 24 we called a General Assembly for several reasons       !person General Assembly (AtValSet.)
#>
#< standard name tests
Dr. Michael Collins person Dr. Michael Collins (AtValSet. 'name "Michael" 'name "Collins" 'title "Doctor" 'gender 'male)
Abraham Lincoln person Abraham Lincoln (AtValSet. 'name "Abraham" 'name "\"Honest Abe\"" 'name "Lincoln" 'occupation 'us-president 'gender 'male)
H. Ross Perot person H. Ross Perot (AtValSet. 'name "H." 'name "Ross" 'name "Perot")
Louis de Luca person Louis de Luca (AtValSet. 'name "Louis" 'name "de" 'name "Luca")
Louis deLuca person Louis deLuca (AtValSet. 'name "Louis" 'name "deLuca")
Noam Chomsky, Ph.D. MD. DDS person Noam Chomsky, Ph.D. MD. DDS (AtValSet. 'name "Noam" 'name "Chomsky" 'degree "Ph.D." 'degree "M.D." 'degree "D.D.S.")
Edward "Blackbeard" Teach person Edward "Blackbeard" Teach (AtValSet. 'name "Edward" 'name "\"Blackbeard\"" 'name "Teach" 'gender 'male)
W. E. B. DuBois person W. E. B. DuBois (AtValSet. 'name "W." 'name "E." 'name "B." 'name "DuBois")
David Fox-Grodzinsky person David Fox-Grodzinsky (AtValSet. 'name "David" 'name "Fox-Grodzinsky" 'gender 'male)
Mary Anne McDonald person Mary Anne McDonald (AtValSet. 'name "Mary" 'name "Anne" 'name "McDonald" 'gender 'female)
George Herbert Walker Bush III. person George Herbert Walker Bush III. (AtValSet. 'name "George" 'name "Herbert" 'name "Walker" 'name "Bush" 'generation 3)
Martin Luther King, Jr. person Martin Luther King, Jr. (AtValSet. 'name "Martin" 'name "Luther" 'name "King" 'generation 2)

# these fail at the moment.  [Rod Brooks I'd] seems to be a perfectly good name like O'kelly.  :-)
# update: I fixed the capitalied_word pattern not to do the O'kelly thing, but until I get context,
#         I don't see how I'll fix the I.  I could hack I to check to it's right, but then
#         "Sure, [Pope Gregory I]'d do that..." but then "first'd sounds pretty bad and looks worse,
#         so I might say "would" here, and save 'd for "Gregory IX'd do that".
#I told Rod Brooks I'd graduate person Rod Brooks (AtValSet. 'name "Rod" 'name "Brooks" 'gender 'male)
#I told Rod Brooks I'd graduate !person Rod Brooks I ()
I told Rod Brooks I'd graduate !person Rod Brooks I'd ()

#Katz, Boris; person Katz, Boris (AtValSet. 'name "Boris" 'name "Katz")
#Felshin, Sue R. person Felshin, Sue R. (AtValSet. 'name "Sue" 'name "R." 'name "Felshin")


I told MLK Jr. David would graduate person MLK Jr. (AtValSet. 'name "Martin" 'name "Luther" 'name "King" 'generation 2)
I told MLK Jr. David would graduate person David (AtValSet. 'name "David" 'gender 'male)
I told MLK Jr. David would graduate !person MLK Jr. David ()
Petty Officer First Class Zeus Jr. person Petty Officer First Class Zeus Jr. (AtValSet. 'name "Zeus" 'generation 2 'title "Petty Officer"))
George Herbert Steven Henry VIII person George Herbert Steven Henry VIII (AtValSet. 'name "George" 'name "Herbert" 'name "Steven" 'name "Henry" 'gender 'male 'gen
#>

# Interim Dean of the School of Electrical Engineering and Computer Science Howard Dean
# Howard E. Dean Professor of Physiology Michael McGeachie
# Mr. and Mrs. Howard Dean
# the Chamberlains
# the Deans  (?!?)

#< some of my friends with Polish and other wierd names
Michael J. T. O'Kelly person Michael J. T. O'Kelly (AtValSet. 'name "Michael" 'name "J." 'name "T." 'name "O'Kelly" 'gender 'male)
Mehmet Can Vuran person Mehmet Can Vuran (AtValSet. 'name "Mehmet" 'name "Can" 'name "Vuran" 'gender 'male)
Mlle. Else Hodson person Mlle. Else Hodson (AtValSet. 'name "Else" 'name "Hodson" 'gender 'female 'title "Miss")
Anne Wisniewski person Anne Wisniewski (AtValSet. 'name '("Anne" "Wisniewski"))
Aga Stokowski person Aga Stokowski (AtValSet. 'name '("Aga" "Stokowski"))
#Stanislaw Lem person Stanislaw Lem (AtValSet. 'name '("Stanislaw" "Lem"))
#Krzystof Gajos person Krzystof Gajos (AtValSet. 'name '("Krzystof" "Gajos"))
Pearl Harbor !person Pearl Harbor (AtValSet.)
Pearl Hill person Pearl Hill (AtValSet. 'name '("Pearl" "Hill"))
```

```
Mia Heavener person Mia Heavener (AtValSet. 'name '("Mia" "Heavener"))
#>

#< a bunch of ministers
#- four was security affairs Minister Gen. Wiranto,                    person security affairs Minister Gen. Wiranto (AtValSet. 'name "Wiranto" 'title
#-  Austrian Social Affairs Minister Elisabeth Sickl                  person Austrian Social Affairs Minister Elisabeth Sickl (AtValSet. 'name '("Elisat
#-      Serbian Agriculture Minister Jovan Babovic                    person Serbian Agriculture Minister Jovan Babovic (AtValSet. 'name '("Jovan" "Babc
#-           Civil Aviation Minister Sharad Yadav                     person Civil Aviation Minister Sharad Yadav (AtValSet. 'name '("Sharad" "Yadav")
#          Israeli Cabinet Minister Haim Ramon                       person Israeli Cabinet Minister Haim Ramon
#            Shas Cabinet Minister Eli Ishai                         person Shas Cabinet Minister Eli Ishai
#            former Chief Minister Hubert Hughes                     person former Chief Minister Hubert Hughes
#Palestinian International Cooperation Minister Nabil Shaath         person Palestinian International Cooperation Minister Nabil Shaath
#                 Defense Minister Raul Salazar                      person Defense Minister Raul Salazar
#                 Defense Minister General Anuruddha Ratwatte put    person Defense Minister General Anuruddha Ratwatte
#     South Korean Defense Minister Cho Sung-tae                     person South Korean Defense Minister Cho Sung-tae
#               Education Minister Vasyl Kremen                      person Education Minister Vasyl Kremen
#               Education Minister Yossi Sarid                       person Education Minister Yossi Sarid
#           Environmental Minister Dalia Itzik                       person Environmental Minister Dalia Itzik
# Venezuelan Environmental Minister Jesus Perez said                 person Venezuelan Environmental Minister Jesus Perez
#     Serbian Environment Minister Branislav Blazic                  person Serbian Environment Minister Branislav Blazic
#                 Finance Minister Avraham Shochat                   person Finance Minister Avraham Shochat
#          former Finance Minister Yaakov Neeman                     person former Finance Minister Yaakov Neeman
#          Indian Foreign Minister Jaswant Singh                     person Indian Foreign Minister Jaswant Singh
#      Mozambican Foreign Minister Leonardo Simao                    person Mozambican Foreign Minister Leonardo Simao
#     With Syrian Foreign Minister Farouk Sharaa recovering          person Syrian Foreign Minister Farouk Sharaa
#          Syrian Foreign Minister Farouk al-Sharaa                  person Syrian Foreign Minister Farouk al-Sharaa
#                 Foreign Minister Farouk al- Sharaa of Syria        person Foreign Minister Farouk al- Sharaa of Syria

#      the Syrian foreign minister, Farouk al-Sharaa                 person the Syrian foreign minister
#      the Syrian foreign minister, Farouk al-Sharaa                 person Farouk al-Sharaa

#         Taliban Foreign Minister Wakil Ahmed Muttawakil            person Taliban Foreign Minister Wakil Ahmed Muttawakil
#         Syria's Foreign Minister Farouk al-Sharaa                  person Syria's Foreign Minister Farouk al-Sharaa
#     when Iran's Foreign Minister announced                         person Iran's Foreign Minister
#                 Foreign Minister Benjamin Ortiz                    person Foreign Minister Benjamin Ortiz
#      Pakistan's Foreign Minister Abdul Sattar                      person Pakistan's Foreign Minister Abdul Sattar
#                 Foreign Minister Mate Granic                       person Foreign Minister Mate Granic
#      But Deputy Foreign Minister Nizar Hamdoon                     person Deputy Foreign Minister Nizar Hamdoon
#           Then-Foreign Minister Ariel Sharon                       person Then-Foreign Minister Ariel Sharon
#           Vice Foreign Minister Kim Gye Gwan.                      person Vice Foreign Minister Kim Gye Gwan
#   against former Health Minister Nkosazana Zuma in November        person former Health Minister Nkosazana Zuma
#                 Health Minister Manto Tshabalala-Msimang is        person Health Minister Manto Tshabalala-Msimang
#         But Immigration Minister Philip Ruddock said               person Immigration Minister Philip Ruddock
#               Interior Minister Yuriy Kravchenko                   person Interior Minister Yuriy Kravchenko
#               Interior Minister Moinuddin Haider                   person Interior Minister Moinuddin Haider
#           from Interior Minister Prince Nayef                      person Interior Minister Prince Nayef

#Flores, the vice-Interior Minister                                  person Flores
#Flores, the vice-Interior Minister                                  person the vice-Interior Minister

#             Information Minister Goran Matic                       person Information Minister Goran Matic
#     former Intelligence Minister Ali Fallahian                     person former Intelligence Minister Ali Fallahian
#               Industry Minister Suwat Liptapallop                  person Industry Minister Suwat Liptapallop
#                Justice Minister Amanda Vanstone                    person Justice Minister Amanda Vanstone
#         Israeli Justice Minister Yossi Beilin                      person Israeli Justice Minister Yossi Beilin
#      Australian Justice Minister Amanda Vanstone                   person Australian Justice Minister Amanda Vanstone
#         Federal Justice Minister Amanda Vanstone                   person Federal Justice Minister Amanda Vanstone
#                   Oil Minister Amer Mohammed Rashid                person Oil Minister Amer Mohammed Rashid
#                 Prime Minister Vladimir Putin                      person Prime Minister Vladimir Putin
#                 Prime Minister Tony Blair                          person Prime Minister Tony Blair
#                 Prime Minister Jean Chretien                       person Prime Minister Jean Chretien
```

```
#                    Prime Minister Atal Bihari Vajpayee                    person Prime Minister Atal Bihari Vajpayee
#                    Prime Minister Sirimavo Bandaranaike                   person Prime Minister Sirimavo Bandaranaike
#             heeded Prime Minister Hun Sen's urgings                       person Prime Minister Hun Sen
#            Russian Prime Minister Vladimir Putin                          person Russian Prime Minister Vladimir Putin
#         Belarusian Prime Minister Resigns                                 person Belarusian Prime Minister Resigns
#            Israeli Prime Minister Ehud Barak                              person Israeli Prime Minister Ehud Barak
#             Indian Prime Minister Rajiv Gandhi                            person Indian Prime Minister Rajiv Gandhi
#            India's Prime Minister Atal Bihari Vajpayee                    person India's Prime Minister Atal Bihari Vajpayee
#           Tuvalu's Prime Minister Ionatana Ionatana applied               person Tuvalu's Prime Minister Ionatana Ionatana
#      former Indian Prime Minister Rajiv Gandhi                            person former Indian Prime Minister Rajiv Gandhi
#       South Korean Prime Minister Kim Jong-pil appointed                  person South Korean Prime Minister Kim Jong-pil
#        Thai Deputy Prime Minister Supachai Panitchpaki,                   person Thai Deputy Prime Minister Supachai Panitchpaki
#             former Prime Minister Viktor Chernomyrdin                     person former Prime Minister Viktor Chernomyrdin
#             former Prime Minister Shimon Peres                            person former Prime Minister Shimon Peres

#      warned former Prime Minister and Nobel Peace Laureate Shimon Peres   person former Prime Minister and Nobel Peace Laureate Shimon Peres

#             Former Prime Minister Margaret Thatcher                       person Former Prime Minister Margaret Thatcher
#      Former Deputy Prime Minister Serhiy Tyhypko                          person Former Deputy Prime Minister Serhiy Tyhypko
#          Thursday, Prime Minister Lionel Jospin said                      person Prime Minister Lionel Jospin
#               then-Prime Minister Shimon Peres                            person then-Prime Minister Shimon Peres
#               Production Minister Juan Falconi                            person Production Minister Juan Falconi
#On Wednesday, State Property Minister Farit Gazizullin                     person State Property Minister Farit Gazizullin
#           Egyptian Water Minister Mahmoud Abu Zeid                        person Egyptian Water Minister Mahmoud Abu Zeid

#    by Papua New Guinea's Minister for Bougainville Affairs Michael Hill   person Papua New Guinea's Minister for Bougainville Affairs Michael Hill


# ; ; ; ; ; ; ; ; ; ; ; ; ; ;   HEADLINES   ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; yeah, these are hard.  I don't really expect to get them.
##             in the Prime Minister's office                               person the Prime Minister
##                    Iran Minister Chided for UK Visit                     person Iran Minister
##                Iranian Minister Visits Britain                           person Iranian Minister
##        Natural Resources Minister To Meet With Striking Miners           person Natural Resources Minister
##  S. Korea Approves Prime Minister SEOUL, South Korea                     person Prime Minister
## a pair of occupations this guy had:
##                          Minister Cop Charged in Death                    person Minister

##       Ex Japan Education Minister Dies TOKYO                             person Ex Japan Education Minister
##             Wife of IOC President Dies at 67                             person Wife of IOC President


## "Deputy Minister" is the name of a horse:  (ouch)
## The top yearling to sell on Tuesday was hip number 599 , a dark bay or brown colt by Deputy Minister out of the Seattle Slew mare Slew Boyera
#>

# former senator and astronaut John Glenn

# combine with occupations and nationalities
astronaut person astronaut (AtValSet. 'occupation 'astronaut)
Russian astronaut person Russian astronaut (AtValSet. 'occupation 'astronaut)
astronaut John Glenn person astronaut John Glenn (AtValSet. 'occupation 'astronaut)
American astronaut John Glenn person American astronaut John Glenn (AtValSet. 'occupation 'astronaut 'from "The United States of America")

# combine with subject-extracted RC's
The first student to do a presentation was Alex. person The first student to do a presentation (AtValSet. 'occupation 'student 'definite #t)
The first student doing a presentation was Alex. person The first student doing a presentation (AtValSet. 'occupation 'student 'definite #t)
The first student who is doing a presentation will win a prize. person The first student who is doing a presentation (AtValSet. 'occupation 'student 'definite #t)
The first student that is doing a presentation will win a prize. person The first student that is doing a presentation (AtValSet. 'occupation 'student 'definite #
The first student who does a presentation will win a prize. person The first student who does a presentation (AtValSet. 'occupation 'student 'definite #t)
The first student who did a presentation will have to present again. person The first student who did a presentation (AtValSet. 'occupation 'student 'definite #t)
```

A.2 Locations (including GPEs)

```
TYPE location

New Mexico location New Mexico (AtValSet. 'name "New Mexico" 'type 'us-state)
New York location New York (AtValSet. 'name "New York")
NY location NY (AtValSet. 'name "New York")
Tex. location Tex. (AtValSet. 'name "Texas")
P.R. location P.R. (AtValSet. 'name "Puerto Rico")

Ukraine location Ukraine (AtValSet. 'name "Ukraine")
US location US (AtValSet. 'name "The United States of America")
USA location USA (AtValSet. 'name "The United States of America")
U.S. location U.S. (AtValSet. 'name "The United States of America")
U.S.A. location U.S.A. (AtValSet. 'name "The United States of America")

Chongqing location Chongqing (AtValSet. 'name "Chongqing" 'location "China" 'population 30000000 'type 'foreign-city)
Zliha city "zliha"
Chernihiv city "chernihiv"

Sun location Sun (AtValSet. 'name "Sun" 'type 'star)
Earth location Earth (AtValSet. 'name "Earth" 'type 'planet 'rank 3)

Atlantic Ocean location Atlantic Ocean (AtValSet. 'name "Atlantic" 'type 'water 'subtype 'ocean)

Mediterranean Sea location Mediterranean Sea (AtValSet. 'name "Mediterranean Sea" 'type 'water 'subtype 'sea)

Zambezi river "zambezi"
Alabama-Coosa river "alabama-coosa"

905 Main Aly street-address 905 Main Aly (905 ((("Main") alley) . street))
905 Main Aly. street-address 905 Main Aly. (905 ((("Main") alley) . street))

Cambridge, MA city-address Cambridge, MA (("Cambridge" . city) ("Massachusetts" . state))

#905 Main St Cambridge, MA address (((905 ((("Main") strt) . street)) . street-address) ("cambridge" . city))

Genesee River location_water ("Genesee")
The Genesee River location_water ("Genesee")

Lake Ontario location Lake Ontario (AtValSet. 'name (list "Lake" "Ontario") 'type 'water 'subtype 'lake)
Conesious Lake location Conesious Lake (AtValSet. 'name '("Conesious" "Lake") 'type 'geological 'subtype "Lake")
Conesious Lakes location Conesious Lakes (AtValSet. 'name (list "Conesious" "Lakes") 'type 'geological 'subtype "Lakes")


Mount Fishy location Mount Fishy (AtValSet. 'name '("Mount" "Fishy") 'type 'geological 'subtype "Mount")
Fishy Mountain location Fishy Mountain (AtValSet. 'name '("Fishy" "Mountain") 'type 'geological 'subtype "Mountain")


Cape Canaveral location Cape Canaveral (AtValSet. 'name '("Cape" "Canaveral") 'type 'geological 'subtype "Cape")
Sundevil Stadium location ("Sundevil")
Nashville International Airport location ("Nashville" "International")

TYPE gpe

Ukraine gpe Ukraine (AtValSet. 'name "Ukraine" 'type 'country)
Johnston Atoll gpe Johnston Atoll (AtValSet. 'name '("Johnston" "Atoll") 'type 'us-territory)
MO gpe MO (AtValSet. 'name "Missouri" 'type 'us-state)
Alb. gpe Alb. (AtValSet. 'name "Alberta" 'type 'province)
American gpe American (AtValSet. 'name "The United States of America" 'type 'country)
Chuukese gpe Chuukese (AtValSet. 'name "Micronesia, Federated States of" 'type 'country)
```

A.3 Organizations

```
TYPE organization

Acron Beacon Journal organization Acron Beacon Journal (AtValSet. 'type 'news)
Asia Times organization Asia Times (AtValSet. 'type 'news)
Baltimore Sun organization Baltimore Sun (AtValSet. 'type 'news)
Boston Globe organization Boston Globe (AtValSet. 'type 'news)
Cable News Network organization Cable News Network (AtValSet. 'type 'news)
Calgary Sun organization Calgary Sun (AtValSet. 'type 'news)
Channel News Asia organization Channel News Asia (AtValSet. 'type 'news)
Charleston Post Courier organization Charleston Post Courier (AtValSet. 'type 'news)
Charlotte Observer organization Charlotte Observer (AtValSet. 'type 'news)
Chicago Sun Times organization Chicago Sun Times (AtValSet. 'type 'news)
Christian Science Monitor organization Christian Science Monitor (AtValSet. 'type 'news)
Christianity Today Magazine organization Christianity Today Magazine (AtValSet. 'type 'news)
Cincinnati Enquirer organization Cincinnati Enquirer (AtValSet. 'type 'news)
Concord Monitor organization Concord Monitor (AtValSet. 'type 'news)
Daily Illini organization Daily Illini (AtValSet. 'type 'news)
Daily Telegraph organization Daily Telegraph (AtValSet. 'type 'news)
Detroit Free Press organization Detroit Free Press (AtValSet. 'type 'news)
Financial Times organization Financial Times (AtValSet. 'type 'news)
Greenville News organization Greenville News (AtValSet. 'type 'news)
Hindustan Times organization Hindustan Times (AtValSet. 'type 'news)
International Herald Tribune organization International Herald Tribune (AtValSet. 'type 'news)
Irish Examiner organization Irish Examiner (AtValSet. 'type 'news)
Irish Independent organization Irish Independent (AtValSet. 'type 'news)
Irish Times organization Irish Times (AtValSet. 'type 'news)
Jamaica Observer organization Jamaica Observer (AtValSet. 'type 'news)
Japan Today organization Japan Today (AtValSet. 'type 'news)
Kansas City Star organization Kansas City Star (AtValSet. 'type 'news)
Melbourne Herald Sun organization Melbourne Herald Sun (AtValSet. 'type 'news)
Middlesborough Evening Gazette organization Middlesborough Evening Gazette (AtValSet. 'type 'news)
Mobile Register organization Mobile Register (AtValSet. 'type 'news)
National Business Review organization National Business Review (AtValSet. 'type 'news)
National Post organization National Post (AtValSet. 'type 'news)
New Jersey Journal organization New Jersey Journal (AtValSet. 'type 'news)
New York Daily News organization New York Daily News (AtValSet. 'type 'news)
New York Times organization New York Times (AtValSet. 'type 'news)
Oakland Tribune organization Oakland Tribune (AtValSet. 'type 'news)
Orlando Sentinel organization Orlando Sentinel (AtValSet. 'type 'news)
Pacific Business News organization Pacific Business News (AtValSet. 'type 'news)
San Diego Union Tribune organization San Diego Union Tribune (AtValSet. 'type 'news)
San Francisco Chronicle organization San Francisco Chronicle (AtValSet. 'type 'news)
San Jose Mercury News organization San Jose Mercury News (AtValSet. 'type 'news)
Sky News organization Sky News (AtValSet. 'type 'news)
Straits Times organization Straits Times (AtValSet. 'type 'news)
Sydney Morning Herald organization Sydney Morning Herald (AtValSet. 'type 'news)
Taipei Times organization Taipei Times (AtValSet. 'type 'news)
The Register organization The Register (AtValSet. 'type 'news)
The Times organization The Times (AtValSet. 'type 'news)
Topeka Capitol Journal organization Topeka Capitol Journal (AtValSet. 'type 'news)
Toronto Star organization Toronto Star (AtValSet. 'type 'news)
U.S.A. Today organization U.S.A. Today (AtValSet. 'type 'news)
USA Today organization USA Today (AtValSet. 'type 'news)
Wall Street Journal organization Wall Street Journal (AtValSet. 'type 'news)
Washington Post organization Washington Post (AtValSet. 'type 'news)
Winnipeg Sun organization Winnipeg Sun (AtValSet. 'type 'news)
```

at St. Joseph's Catholic Church on East 87th Street in Manhattan and said organization St. Joseph's Catholic Church (AtValSet.)
the Feast of Pentecost, St. Patrick's Cathedral offers the Sacrament of Confirmation to organization St. Patrick's Cathedral (AtValSet.)
the archdiocese's Office for Persons with Disabilities organization archdiocese's Office for Persons with Disabilities (AtValSet.)

#As the Taxi and Limousine Commission's chairwoman , Diane McGrath-McKechnie organization Taxi and Limousine Commission (AtValSet.)

and graduated from Columbia's School of General Studies in 1960 . organization Columbia's School of General Studies (AtValSet.)

alled the Life Project, for the U.S. Department of Labor in the early 1970s . organization Life Project (AtValSet.)
alled the Life Project, for the U.S. Department of Labor in the early 1970s . organization U.S. Department of Labor (AtValSet.)

at Columbia University's Bureau of Applied Social Research, the organization Columbia University's Bureau of Applied Social Research (AtValSet.)

Research, and the Vera Institute of Justice in New York and organization Vera Institute of Justice in New York (AtValSet.)
Bureau of Social Science Research in Washington organization Bureau of Social Science Research in Washington (AtValSet.)
an associate professor of sociology at John Jay College of Criminal Justice . organization John Jay College of Criminal Justice (AtValSet.)

On hearing of the reopening of the Butterfly Zone at the Bronx Zoo : The 1 organization Bronx Zoo (AtValSet.)

a leader of the Little Town Forum for the Historic Preservation of Sloatsburg , a group formed to res organization Little Town Forum for the Historic Preservation (AtValSet.)

the comme after Calif. prevents this version from working: (There is no Calif without the period, and there are not all subtokens)
#the City Council of San Leandro, Calif., reject organization City Council of San Leandro, Calif. (AtValSet.)

A.4 Restrictive Clauses

TYPE person

The first student to do a presentation was Alex. person The first student to do a presentation (AtValSet. 'occupation 'student 'definite #t)
The first student doing a presentation was Alex. person The first student doing a presentation (AtValSet. 'occupation 'student 'definite #t)
The first student who is doing a presentation will win a prize. person The first student who is doing a presentation (AtValSet. 'occupation 'student 'definite #t
The first student that is doing a presentation will win a prize. person The first student that is doing a presentation (AtValSet. 'occupation 'student 'definite
The first student who does a presentation will win a prize. person The first student who does a presentation (AtValSet. 'occupation 'student 'definite #t)
The first student who did a presentation will have to present again. person The first student who did a presentation (AtValSet. 'occupation 'student 'definite #t

A.5 Numbers

```
TYPE quantity
one quantity one (AtValSet. 'value 1)
two quantity two (AtValSet. 'value 2)
three quantity three (AtValSet. 'value 3)
two and two quantity two (AtValSet. 'value 2)
two and two !quantity two and two (AtValSet. 'value 4)
two and two !quantity two and two (AtValSet. 'value 22)
two - two !quantity two - two (AtValSet. 'value 4)
two - two !quantity two - two (AtValSet. 'value 22)
 one  quantity one (AtValSet. 'value 1)
bonehead !quantity one (AtValSet. 'value 1)
one's quantity one (AtValSet. 'value 1)
eleven quantity eleven (AtValSet. 'value 11)
seventeen quantity seventeen (AtValSet. 'value 17)
nineteen quantity nineteen (AtValSet. 'value 19)
twenty quantity twenty (AtValSet. 'value 20)
twenty's quantity twenty (AtValSet. 'value 20)
ten - fifteen quantity ten (AtValSet. 'value 10)
ten - fifteen quantity fifteen (AtValSet. 'value 15)
twenty one quantity twenty one (AtValSet. 'value 21)
twenty-two quantity twenty-two (AtValSet. 'value 22)
twenty - two quantity twenty - two (AtValSet. 'value 22)
hundred and one quantity hundred and one (AtValSet. 'value 101)
#one twenty four quantity one twenty four (AtValSet. 'value 124)
#one twenty four quantity* one twenty four (AtValSet. 'value 120)
#twenty twenty quantity twenty twenty (AtValSet. 'value 2020)
#six seventeen quantity six seventeen (AtValSet. 'value 617)
two zero four quantity two zero four (AtValSet. 'value 204)
two oh four quantity two oh four (AtValSet. 'value 204)
#nineteen twenty quantity nineteen twenty (AtValSet. 'value 1920)
#nineteen forty-two quantity nineteen forty-two (AtValSet. 'value 1942)
seventy five quantity seventy five (AtValSet. 'value 75)
three hundred and forty two quantity three hundred and forty two (AtValSet. 'value 342)
thirty thousand quantity thirty thousand (AtValSet. 'value 30000)
thirty two hundred quantity thirty two hundred (AtValSet. 'value 3200)
thirty-two hundred sixty three quantity thirty-two hundred sixty three (AtValSet. 'value 3263)
three hundred thousand quantity three hundred thousand (AtValSet. 'value 300000)
three hundred and fifty thousand quantity three hundred and fifty thousand (AtValSet. 'value 350000)
twenty-three hundred and forty two quantity twenty-three hundred and forty two (AtValSet. 'value 2342)
three thousand seven quantity three thousand seven (AtValSet. 'value 3007)
three thousand and seven quantity three thousand and seven (AtValSet. 'value 3007)
three million and seven quantity three million and seven (AtValSet. 'value 3000007)
three trillion and seven quantity three trillion and seven (AtValSet. 'value  3000000000007)
thirty three trillion and seven quantity thirty three trillion and seven (AtValSet. 'value 33000000000007)
two oh three million three hundred and forty two thousand four hundred and ninety-six quantity two oh three million three hundred and forty two thousand four hund
(AtValSet. 'value 203342496)
#Two Zero Three MILLION Three Hundred and Forty two thousand four HUNDRED AND NINETY-six (AtValSet. 'value quantity 203342496)
in one shot quantity one (AtValSet. 'value 1)
I shot one quantity one (AtValSet. 'value 1)
I shot one! quantity one (AtValSet. 'value 1)
One shot me! quantity One (AtValSet. 'value 1)
I shot one. quantity one (AtValSet. 'value 1)
On one, shot. quantity one (AtValSet. 'value 1)

######################################################################
# additional test cases

sixty eight quantity sixty eight (AtValSet. 'value 68)
a hundred quantity a hundred (AtValSet. 'value 100)
```

```
seven hundred quantity seven hundred (AtValSet. 'value 700)
fifteen hundred quantity fifteen hundred (AtValSet. 'value 1500)
twenty three hundred quantity twenty three hundred (AtValSet. 'value 2300)
five hundred three quantity five hundred three (AtValSet. 'value 503)
nineteen hundred four quantity nineteen hundred four (AtValSet. 'value 1904)
twenty one hundred six quantity twenty one hundred six (AtValSet. 'value 2106)
three hundred fifteen quantity three hundred fifteen (AtValSet. 'value 315)
fourteen hundred thirteen quantity fourteen hundred thirteen (AtValSet. 'value 1413)
eighty three hundred eleven quantity eighty three hundred eleven (AtValSet. 'value 8311)
six hundred twenty four quantity six hundred twenty four (AtValSet. 'value 624)
eleven hundred seventy seven quantity eleven hundred seventy seven (AtValSet. 'value 1177)
ninety nine hundred ninety nine quantity ninety nine hundred ninety nine (AtValSet. 'value 9999)
seven hundred and one quantity seven hundred and one (AtValSet. 'value 701)
eight hundred and twelve quantity eight hundred and twelve (AtValSet. 'value 812)
nine hundred and forty one quantity nine hundred and forty one (AtValSet. 'value 941)
eleven hundred and six quantity eleven hundred and six (AtValSet. 'value 1106)
sixteen hundred and twelve quantity sixteen hundred and twelve (AtValSet. 'value 1612)
thirteen hundred and fifty five quantity thirteen hundred and fifty five (AtValSet. 'value 1355)
twenty five hundred and nine quantity twenty five hundred and nine (AtValSet. 'value 2509)
thirty two hundred and twelve quantity thirty two hundred and twelve (AtValSet. 'value 3212)
ninety six hundred and sixty four quantity ninety six hundred and sixty four (AtValSet. 'value 9664)
thirteen oh four quantity thirteen oh four (AtValSet. 'value 1304)
fourteen zero nine quantity fourteen zero nine (AtValSet. 'value 1409)

######################################################################
# sets

# TYPE set
#
#ones set ones 1
#twos set twos 2
#tens set tens 10
#twenties set twenties 20
#seventeen seventies set seventeen seventies 1770


# ordinals

TYPE quantity

first quantity first (AtValSet. 'type 'ordinal 'value 1)
second quantity second (AtValSet. 'type 'ordinal 'value 2)
third quantity third (AtValSet. 'type 'ordinal 'value 3)
fourth quantity fourth (AtValSet. 'type 'ordinal 'value 4)
fifth quantity fifth (AtValSet. 'type 'ordinal 'value 5)
sixth quantity sixth (AtValSet. 'type 'ordinal 'value 6)
seventh quantity seventh (AtValSet. 'type 'ordinal 'value 7)
eighth quantity eighth (AtValSet. 'type 'ordinal 'value 8)
ninth quantity ninth (AtValSet. 'type 'ordinal 'value 9)
tenth quantity tenth (AtValSet. 'type 'ordinal 'value 10)
eleventh quantity eleventh (AtValSet. 'type 'ordinal 'value 11)
twelfth quantity twelfth (AtValSet. 'type 'ordinal 'value 12)
thirteenth quantity thirteenth (AtValSet. 'type 'ordinal 'value 13)
fourteenth quantity fourteenth (AtValSet. 'type 'ordinal 'value 14)
fifteenth quantity fifteenth (AtValSet. 'type 'ordinal 'value 15)
sixteenth quantity sixteenth (AtValSet. 'type 'ordinal 'value 16)
seventeenth quantity seventeenth (AtValSet. 'type 'ordinal 'value 17)
eighteenth quantity eighteenth (AtValSet. 'type 'ordinal 'value 18)
nineteenth quantity nineteenth (AtValSet. 'type 'ordinal 'value 19)
twentieth quantity twentieth (AtValSet. 'type 'ordinal 'value 20)
thirtieth quantity thirtieth (AtValSet. 'type 'ordinal 'value 30)
```

```
fortieth quantity fortieth (AtValSet. 'type 'ordinal 'value 40)
fiftieth quantity fiftieth (AtValSet. 'type 'ordinal 'value 50)
sixtieth quantity sixtieth (AtValSet. 'type 'ordinal 'value 60)
seventieth quantity seventieth (AtValSet. 'type 'ordinal 'value 70)
eightieth quantity eightieth (AtValSet. 'type 'ordinal 'value 80)
ninetieth quantity ninetieth (AtValSet. 'type 'ordinal 'value 90)
# using compound_quantity made from 'hundredth' and sodp_ones
hundredth quantity hundredth (AtValSet. 'type 'ordinal 'value 100)
a hundredth quantity a hundredth (AtValSet. 'type 'ordinal 'value 100)
one hundredth quantity one hundredth (AtValSet. 'type 'ordinal 'value 100)
two hundredth quantity two hundredth (AtValSet. 'type 'ordinal 'value 200)
three hundredth quantity three hundredth (AtValSet. 'type 'ordinal 'value 300)
four hundredth quantity four hundredth (AtValSet. 'type 'ordinal 'value 400)
five hundredth quantity five hundredth (AtValSet. 'type 'ordinal 'value 500)
six hundredth quantity six hundredth (AtValSet. 'type 'ordinal 'value 600)
seven hundredth quantity seven hundredth (AtValSet. 'type 'ordinal 'value 700)
eight hundredth quantity eight hundredth (AtValSet. 'type 'ordinal 'value 800)
nine hundredth quantity nine hundredth (AtValSet. 'type 'ordinal 'value 900)
# using compound_quantity made from 'hundredth' and sodp_teens
eleven hundredth quantity eleven hundredth (AtValSet. 'type 'ordinal 'value 1100)
twelve hundredth quantity twelve hundredth (AtValSet. 'type 'ordinal 'value 1200)
seventeen hundredth quantity seventeen hundredth (AtValSet. 'type 'ordinal 'value 1700)
# using compound_quantity made from ones_quantity and sodp_tens
fifty eighth quantity fifty eighth (AtValSet. 'type 'ordinal 'value 58)
ninety sixth quantity ninety sixth (AtValSet. 'type 'ordinal 'value 96)
# using compound_quantity made from ones_quantity and sodp_hunds
one hundred seventh quantity one hundred seventh (AtValSet. 'type 'ordinal 'value 107)
thirteen hundred ninth quantity thirteen hundred ninth (AtValSet. 'type 'ordinal 'value 1309)
# using compound_quantity made from teens_quantity and sodp_hunds
one hundred thirteenth quantity one hundred thirteenth (AtValSet. 'type 'ordinal 'value 113)
twenty seven hundred twelfth quantity twenty seven hundred twelfth (AtValSet. 'type 'ordinal 'value 2712)
fourteen hundred nineteenth quantity fourteen hundred nineteenth (AtValSet. 'type 'ordinal 'value 1419)
# using compound_quantity made from tens_quantity and sodp_hunds
nine hundred twentieth quantity nine hundred twentieth (AtValSet. 'type 'ordinal 'value 920)
seventeen hundred ninetieth quantity seventeen hundred ninetieth (AtValSet. 'type 'ordinal 'value 1790)
eighty six hundred seventieth quantity eighty six hundred seventieth (AtValSet. 'type 'ordinal 'value 8670)
# more advanced cases
one second helping quantity second (AtValSet. 'type 'ordinal 'value 2)
eleventh quantity eleventh (AtValSet. 'type 'ordinal 'value 11)
twenty-second quantity twenty-second (AtValSet. 'type 'ordinal 'value 22)
twenty seventh quantity twenty seventh (AtValSet. 'type 'ordinal 'value 27)
seven hundred and seventy seven thousand seven hundred seventy seventh quantity seven hundred and seventy seven thousand seven hundred seventy seventh
(AtValSet. 'type 'ordinal 'value 777777)
# I fail on this because I decided not to have hundred-omitting in the cardinals
#seven seventy seventh quantity seven seventy seventh (AtValSet. 'type 'ordinal 'value 777)


# spelled out fractions testplan

TYPE quantity

# singular ones fractions
one half quantity one half (AtValSet. 'type 'expression 'value '(/ 1.0 2))
two half !quantity two half (AtValSet. 'type 'expression 'value '(/ 2.0 2))
one third quantity one third (AtValSet. 'type 'expression 'value '(/ 1.0 3))
three third !quantity three third (AtValSet. 'type 'expression 'value '(/ 3.0 3))
one quarter quantity one quarter (AtValSet. 'type 'expression 'value '(/ 1.0 4))
two quarter !quantity two quarter (AtValSet. 'type 'expression 'value '(/ 2.0 4))
one fourth quantity one fourth (AtValSet. 'type 'expression 'value '(/ 1.0 4))
four fourth !quantity four fourth (AtValSet. 'type 'expression 'value '(/ 4.0 4))
one fifth quantity one fifth (AtValSet. 'type 'expression 'value '(/ 1.0 5))
```

```
five fifth !quantity five fifth (AtValSet. 'type 'expression 'value '(/ 5.0 5))
one sixth quantity one sixth (AtValSet. 'type 'expression 'value '(/ 1.0 6))
six sixth !quantity six sixth (AtValSet. 'type 'expression 'value '(/ 6.0 6))
one seventh quantity one seventh (AtValSet. 'type 'expression 'value '(/ 1.0 7))
seven seventh !quantity seven seventh (AtValSet. 'type 'expression 'value '(/ 7.0 7))
one eighth quantity one eighth (AtValSet. 'type 'expression 'value '(/ 1.0 8))
eight eighth !quantity eight eighth (AtValSet. 'type 'expression 'value '(/ 8.0 8))
one ninth quantity one ninth (AtValSet. 'type 'expression 'value '(/ 1.0 9))
nine ninth !quantity nine ninth (AtValSet. 'type 'expression 'value '(/ 9.0 9))

# singular tens fractions
one tenth quantity one tenth (AtValSet. 'type 'expression 'value '(/ 1.0 10))
three tenth !quantity three tenth (AtValSet. 'type 'expression 'value '(/ 3.0 10))
one twentieth quantity one twentieth (AtValSet. 'type 'expression 'value '(/ 1.0 20))
three twentieth !quantity three twentieth (AtValSet. 'type 'expression 'value '(/ 3.0 20))
one thirtieth quantity one thirtieth (AtValSet. 'type 'expression 'value '(/ 1.0 30))
three thirtieth !quantity three thirtieth (AtValSet. 'type 'expression 'value '(/ 3.0 30))
one fortieth quantity one fortieth (AtValSet. 'type 'expression 'value '(/ 1.0 40))
three fortieth !quantity three fortieth (AtValSet. 'type 'expression 'value '(/ 3.0 40))
one fiftieth quantity one fiftieth (AtValSet. 'type 'expression 'value '(/ 1.0 50))
three fiftieth !quantity three fiftieth (AtValSet. 'type 'expression 'value '(/ 3.0 50))
one sixtieth quantity one sixtieth (AtValSet. 'type 'expression 'value '(/ 1.0 60))
three sixtieth !quantity three sixtieth (AtValSet. 'type 'expression 'value '(/ 3.0 60))
one seventieth quantity one seventieth (AtValSet. 'type 'expression 'value '(/ 1.0 70))
three seventieth !quantity three seventieth (AtValSet. 'type 'expression 'value '(/ 3.0 70))
one eightieth quantity one eightieth (AtValSet. 'type 'expression 'value '(/ 1.0 80))
three eightieth !quantity three eightieth (AtValSet. 'type 'expression 'value '(/ 3.0 80))

# singular teens fractions
one twelfth quantity one twelfth (AtValSet. 'type 'expression 'value '(/ 1.0 12))
two fourteenth !quantity two fourteenth (AtValSet. 'type 'expression 'value '(/ 1.0 14))
one eighteenth quantity one eighteenth (AtValSet. 'type 'expression 'value '(/ 1.0 18))
nine sixteenth !quantity nine sixteenth (AtValSet. 'type 'expression 'value '(/ 9.0 16))

# singular hundred fractions
one hundredth quantity one hundredth (AtValSet. 'type 'expression 'value '(/ 1.0 100))
three hundredth !quantity three hundredth (AtValSet. 'type 'expression 'value '(/ 3.0 100))

# plural ones fractions
ninety halves quantity ninety halves (AtValSet. 'type 'expression 'value '(/ 90.0 2))
twenty two halves quantity twenty two halves (AtValSet. 'type 'expression 'value '(/ 22.0 2))
three thirds quantity three thirds (AtValSet. 'type 'expression 'value '(/ 3.0 3))
ninety thirds quantity ninety thirds (AtValSet. 'type 'expression 'value '(/ 90.0 3))
two fourths quantity two fourths (AtValSet. 'type 'expression 'value '(/ 2.0 4))
eighty fourths quantity eighty fourths (AtValSet. 'type 'expression 'value '(/ 80.0 4))
twenty four fourths quantity twenty four fourths (AtValSet. 'type 'expression 'value '(/ 24.0 4))
four hundred and eight fourths quantity four hundred and eight fourths (AtValSet. 'type 'expression 'value '(/ 408.0 4))
five fifths quantity five fifths (AtValSet. 'type 'expression 'value '(/ 5.0 5))
five hundred five fifths quantity five hundred five fifths (AtValSet. 'type 'expression 'value '(/ 505.0 5))
one hundred twenty five thousand seven hundred and fifty fifths quantity one hundred twenty five thousand seven hundred and fifty fifths (AtValSet. 'type 'express
six sixths quantity six sixths (AtValSet. 'type 'expression 'value '(/ 6.0 6))
six hundred fifty four sixths quantity six hundred fifty four sixths (AtValSet. 'type 'expression 'value '(/ 654.0 6))
twelve thousand three hundred and six sixths quantity twelve thousand three hundred and six sixths (AtValSet. 'type 'expression 'value '(/ 12306.0 6))
seventy sevenths quantity seventy sevenths (AtValSet. 'type 'expression 'value '(/ 70.0 7))
two hundred ten sevenths quantity two hundred ten sevenths (AtValSet. 'type 'expression 'value '(/ 210.0 7))
four eighths quantity four eighths (AtValSet. 'type 'expression 'value '(/ 4.0 8))
twenty four eighths quantity twenty four eighths (AtValSet. 'type 'expression 'value '(/ 24.0 8))
nine hundred ninety nine ninths quantity nine hundred ninety nine ninths (AtValSet. 'type 'expression 'value '(/ 999.0 9))

# plural tens fractions
nine tenths quantity nine tenths (AtValSet. 'type 'expression 'value '(/ 9.0 10))
```

```
nineteen twentieths quantity nineteen twentieths (AtValSet. 'type 'expression 'value '(/ 19.0 20))
twenty thirtieths quantity twenty thirtieths (AtValSet. 'type 'expression 'value '(/ 20.0 30))
twenty seven fortieths quantity twenty seven fortieths (AtValSet. 'type 'expression 'value '(/ 27.0 40))
one hundred eight fiftieths quantity one hundred eight fiftieths (AtValSet. 'type 'expression 'value '(/ 108.0 50))
one hundred twelve sixtieths quantity one hundred twelve sixtieths (AtValSet. 'type 'expression 'value '(/ 112.0 60))
one hundred ninety seventieths quantity one hundred ninety seventieths (AtValSet. 'type 'expression 'value '(/ 190.0 70))
one hundred seventy five eightieths quantity one hundred seventy five eightieths (AtValSet. 'type 'expression 'value '(/ 175.0 80))
one thousand four hundred six ninetieths quantity one thousand four hundred six ninetieths (AtValSet. 'type 'expression 'value '(/ 1406.0 90))

# plural teens fractions
five elevenths quantity five elevenths (AtValSet. 'type 'expression 'value '(/ 5.0 11))
fifteen twelfths quantity fifteen twelfths (AtValSet. 'type 'expression 'value '(/ 15.0 12))
thirty thirteenths quantity thirty thirteenths (AtValSet. 'type 'expression 'value '(/ 30.0 13))
thirty five fourteenths quantity thirty five fourteenths (AtValSet. 'type 'expression 'value '(/ 35.0 14))
one hundred five fifteenths quantity one hundred five fifteenths (AtValSet. 'type 'expression 'value '(/ 105.0 15))
one hundred sixteen sixteenths quantity one hundred sixteen sixteenths (AtValSet. 'type 'expression 'value '(/ 116.0 16))
one hundred twenty seventeenths quantity one hundred twenty seventeenths (AtValSet. 'type 'expression 'value '(/ 120.0 17))
one hundred forty five eighteenths quantity one hundred forty five eighteenths (AtValSet. 'type 'expression 'value '(/ 145.0 18))
one thousand nine hundred eleven nineteenths quantity one thousand nine hundred eleven nineteenths (AtValSet. 'type 'expression 'value '(/ 1911.0 19))

# plural hundreds fractions
twenty four hundredths quantity twenty four hundredths (AtValSet. 'type 'expression 'value '(/ 24.0 100))

# compound fractions

three twenty firsts quantity three twenty firsts (AtValSet. 'type 'expression 'value '(/ 3.0 21))
sixteen thirty seconds quantity sixteen thirty seconds (AtValSet. 'type 'expression 'value '(/ 16.0 32))
twenty one forty thirds quantity twenty one forty thirds (AtValSet. 'type 'expression 'value '(/ 21.0 43))
one hundred five fifty fourths quantity one hundred five fifty fourths (AtValSet. 'type 'expression 'value '(/ 105.0 54))
one hundred fourteen ninety fifths quantity one hundred fourteen ninety fifths (AtValSet. 'type 'expression 'value '(/ 114.0 95))
one hundred twenty seventy sixths quantity one hundred twenty seventy sixths (AtValSet. 'type 'expression 'value '(/ 120.0 76))
one hundred sixty five eighty sevenths quantity one hundred sixty five eighty sevenths (AtValSet. 'type 'expression 'value '(/ 165.0 87))
one hundred twenty eighths quantity one hundred twenty eighths (AtValSet. 'type 'expression 'value '(/ 100.0 28))
one hundred twenty eighths quantity one hundred twenty eighths (AtValSet. 'type 'expression 'value '(/ 120.0 8))
one thousand one hundred and eleven fifty ninths quantity one thousand one hundred and eleven fifty ninths (AtValSet. 'type 'expression 'value '(/ 1111.0 59))

# hyphen usage
three-fourths quantity three-fourths (AtValSet. 'type 'expression 'value '(/ 3.0 4))




# -*- fundamental -*-

# ****************************************************************************
# *** combined number forms

TYPE quantity

25 quantity 25 (AtValSet. 'type 'number 'value (.longValue 25))
5,000 quantity 5,000 (AtValSet. 'type 'number 'value (.longValue 5000))
1,234,567,890 quantity 1,234,567,890 (AtValSet. 'type 'number 'value (.longValue 1234567890))
-5,000 quantity -5,000 (AtValSet. 'type 'number 'value (.longValue -5000))
3.2% quantity 3.2% (AtValSet. 'type 'percent 'value 0.032)
four percent quantity four percent (AtValSet. 'type 'percent 'value '0.04)
12-15 quantity 12-15 (AtValSet. 'type 'range 'start (AtValSet. 'type 'number 'value (.longValue 12)) 'end (AtValSet. 'type 'number 'value (.longValue 15)) 'modifi
#-9-15 quantity -9-15 (AtValSet. 'type
50 million quantity 50 million (AtValSet. 'type 'cardinal 'value (.longValue 50000000))
```

A.6 Units

Note. the lexicon format for units is a prior version, and there is no third field to specify a substring in context. The whole string in the first field must be matched.

```
TYPE money
TYPE socard

a dollar money 1
two dollars money 2
two dollar money 2
2 dollar money 2
2-dollar money 2
2 dollars money 2
$2 money 2
2-dollars money* 2

a hundred dollar bill money 100

# this one is different due to np-internal stucture that needs to know
# about bills, and needs to assume that bill is a noun and that it's
# the head, or that hundred dollar can be an adjective but not a noun
# phrase or some such.  ugh.
a hundred dollar bills money 100

twenty two hundred dollar bills money 2200
twenty two hundred dollar bills money 200
twenty two hundred dollar bills socard 20

a two hundred dollar bill money 200
ten hundred dollar bills money 1000
a twenty-one hundred dollar bill money 2100
21 hundred-dollar bills money 2100
twenty-one hundred dollar bills money 2100
```

A.7 Dates

Note, the lexicon format for units is a prior version, and there is no third field to specify a substring in context. The whole string in the first field must be matched.

```
TYPE date
TYPE rte
TYPE time
TYPE year
TYPE month
TYPE day-in-month
TYPE day-of-month
TYPE month-of-year
TYPE period
TYPE quarter
TYPE half
TYPE third


1992 year 1992
nineteen ninety-two year 1992
third quarter quarter 3
the third quarter quarter 3
third quarter of 1991 period 1
the fourth quarter ended Sept. 30 quarter 4
the fourth quarter ended Sept. 30 day-of-month ((9 . month) (30 . day-in-month))
the three months ended Sept. 30 period 3
the three months ended Sept. 30 day-of-month ((9 . month) (30 . day-in-month))
first-half profit half 1
fiscal 1989's fourth quarter period 1
4th period period 4
1975 World Series year 1975
February 12th day-of-month ((2 . month) (12 . day-in-month))
Tuesday, February 12th date ((2 . day) ((2 . month) (12 . day-in-month)))
February 12, 1997 date (((2 . month) (12 . day-in-month)) 1997)
February 12, 8 A.M. day-of-month ((2 . month) (12 . day-in-month))
February 12, 8 A.M. time 8
shortly after the 4th of May date 1
noon time "12:00"
twelve o'clock noon time "12:00"

TYPE rte-time
last night rte-time 1
yesterday evening rte-time 1

5 p.m. EST time 5
5 pm EST time 5
5pm EST time 5

the first half of fiscal 1990 period 1
October 1994 period (((10 . month) . month) (1994 . year))
1994-1998 period (1994 1998)
1940s period 1940
1940's period 1940

1943s period 1943

later this month rte 1
earlier last year rte 1
```

```
1970 through 1986 period (1970 1986)
monday through friday period ((1 . day) (5 . day))
April through June period ((4 . month) (6 . month))
April - June period ((4 . month) (6 . month))

03/13 day-of-month 1303
03-13 day-of-month 1303
03/11/1996 date 19961103
03-11-1996 date 19961103
03-11-96 date 961103
03/11/96 date 961103
summer period "Summer"
last summer rte ("Summer" . period)
last summer's party rte ("Summer" . period)
```

A.8 Questions

These are 346 of the TREC questions, annotated according to ACE guidelines, on which we have begun to develop for ACE. This is the only testplan for which we do not pass all uncommented test cases. In particular, we currently pass 86 of them.

```
# We are having trouble deciding whether "many people" or "many casinos" or "many counties" are markable as generics in the context of "how many".

TYPE people
TYPE quantity
TYPE person
TYPE location
TYPE organization
TYPE facility
TYPE gpe
TYPE event
TYPE title

201: What was the name of the first Russian astronaut to do a spacewalk? person the first Russian astronaut to do a spacewalk (AtValSet. 'occupation 'astronaut 'f
201: What was the name of the first Russian astronaut to do a spacewalk? person Russian astronaut (AtValSet. 'occupation 'astronaut 'from "Russia")
201: What was the name of the first Russian astronaut to do a spacewalk? gpe Russian (AtValSet. 'type 'country 'name "Russia")
202: Where is Belize located? gpe Belize (AtValSet. 'type 'country)
204: What type of bridge is the Golden Gate Bridge? facility bridge (AtValSet. 'type 'bridge 'generic #t)
204: What type of bridge is the Golden Gate Bridge? facility the Golden Gate Bridge (AtValSet. 'type 'bridge)
205: What is the population of the Bahamas? gpe the Bahamas (AtValSet. 'type 'country)
206: How far away is the moon? location the moon (AtValSet. 'type 'astronomy)
208: What state has the most Indians? person Indians (AtValSet. 'group #t)
210: How many dogs pull a sled in the Iditarod? organization the Iditarod (AtValSet. 'type 'competition)
215: Who is the leader of India? person the leader of India (AtValSet. 'occupation 'government)
215: Who is the leader of India? gpe India (AtValSet. 'type 'country)
216: What is the primary language of the Philippines? gpe the Philippines (AtValSet. 'type 'country)
219: What is the population of Japan? gpe Japan (AtValSet. 'type 'country)
220: Who is the prime minister of Australia? person the prime minister of Australia (AtValSet. 'occupation 'government)
220: Who is the prime minister of Australia? gpe Australia (AtValSet. 'type 'country)
221: Who killed Martin Luther King? person Martin Luther King (AtValSet. )
223: Where's Montenegro? gpe Montenegro (AtValSet. 'type 'country)
225: Who is the Greek God of the Sea? person the Greek God of the Sea (AtValSet. 'type 'deity)
225: Who is the Greek God of the Sea? gpe Greek (AtValSet. 'type 'country 'name "Greece")
226: Where is the Danube? location the Danube (AtValSet. 'type 'river)
231: Who was the president of Vichy France? person the president of Vichy France (AtValSet. 'occupation 'government)
231: Who was the president of Vichy France? gpe Vichy France (AtValSet. 'type 'country)
237: Name one of the major gods of Hinduism? person the major gods of Hinduism (AtValSet. 'type 'deity 'group #t)
237: Name one of the major gods of Hinduism? organization Hinduism (AtValSet. 'type 'religious)
238: What does the abbreviation OAS stand for? organization OAS (AtValSet. )
240: How many years ago did the ship Titanic sink? facility Titanic (AtValSet. 'type 'vehicle)
242: What was the name of the famous battle in 1836 between Texas and Mexico? gpe Texas (AtValSet. 'type 'state)
242: What was the name of the famous battle in 1836 between Texas and Mexico? gpe Mexico (AtValSet. 'type 'country)
245: Where can you find the Venus flytrap? !location Venus (AtValSet. 'type 'astronomy)
246: What did Vasco da Gama discover? person Vasco da Gama (AtValSet. )
247: Who won the Battle of Gettysburg? gpe Gettysburg (AtValSet. 'type 'city)
247: Who won the Battle of Gettysburg? event Battle of Gettysburg (AtValSet. 'type 'battle)
249: Where is the Valley of the Kings? location Valley of the Kings (AtValSet. )
250: Where did the Maya people live? person the Maya people (AtValSet. 'group #t)
251: How many people live in Chile? gpe Chile (AtValSet. 'type 'country)
254: What is California's state bird? gpe California (AtValSet. 'type 'state)
256: Who is buried in the great pyramid of Giza? facility the great pyramid of Giza (AtValSet. 'type 'monument)
256: Who is buried in the great pyramid of Giza? gpe Giza (AtValSet. 'type 'city)
260: What does NAFTA stand for? !organization NAFTA (AtValSet. )
262: What is the name of the longest ruling dynasty of Japan? person Japan (AtValSet. 'group #t 'occupation 'government)
262: What is the name of the longest ruling dynasty of Japan? gpe Japan (AtValSet. 'type 'country)
263: When was Babe Ruth born? person Babe Ruth (AtValSet. 'occupation 'sports)
```

```
264: Who wrote the Farmer's Almanac? !organization Farmer's Almanac (AtValSet. )
266: Where was Pythagoras born? person Pythagoras (AtValSet. )
268: Who killed Caesar? person Caesar (AtValSet. )
270: Where is the Orinoco? location Orinoco (AtValSet. 'type 'river)
273: Who was the first U.S. president ever to resign? person the first U.S. president ever to resign (AtValSet. 'occupation 'government)
273: Who was the first U.S. president ever to resign? gpe U.S. (AtValSet. 'type 'country)
274: Who invented the game Scrabble?
# we are not sure about this case
275: About how many soldiers died in World War II? person many soldiers (AtValSet. 'group #t 'occupation 'military)
275: About how many soldiers died in World War II? event World War II (AtValSet. 'type 'battle)
276: How much money does the Sultan of Brunei have? person the Sultan of Brunei (AtValSet. 'occupation 'government)
276: How much money does the Sultan of Brunei have? gpe Brunei (AtValSet. 'type 'country)
277: How large is Missouri's population? gpe Missouri (AtValSet. 'type 'state)
278: What was the death toll at the eruption of Mount Pinatubo? location Mount Pinatubo (AtValSet. 'type 'mountain)
280: What's the tallest building in New York City? gpe New York City (AtValSet. 'type 'city)
280: What's the tallest building in New York City? facility the tallest building in New York City (AtValSet. 'type 'building)
281: When did Geraldine Ferraro run for vice president? person Geraldine Ferraro (AtValSet. 'occupation 'government)
# not sure if we want to mark "vice president" as a person
281: When did Geraldine Ferraro run for vice president? person vice president (AtValSet. 'occupation 'government)
283: Where is Ayer's rock? location Ayer's rock (AtValSet. )
285: When was the first railroad from the east coast to the west coast completed? facility the first railroad from the east coast to the west coast completed
(AtValSet. 'type 'railroad)
285: When was the first railroad from the east coast to the west coast completed? location the east coast (AtValSet. 'type 'region)
285: When was the first railroad from the east coast to the west coast completed? location the west coast (AtValSet. 'type 'region)
286: What is the nickname of Pennsylvania? gpe Pennsylvania (AtValSet. 'type 'state)
288: How fast can a Corvette go?
289: What are John C. Calhoun and Henry Clay known as? person John C. Calhoun (AtValSet. 'occupation 'government)
289: What are John C. Calhoun and Henry Clay known as? person Henry Clay (AtValSet. 'occupation 'government)
290: When was Hurricane Hugo?
291: When did the Carolingian period begin?
292: How big is Australia? gpe Australia (AtValSet. 'type 'country)
292: How big is Australia? location Australia (AtValSet. 'type 'continent)
293: Who found Hawaii? gpe Hawaii (AtValSet. 'type 'state)
293: Who found Hawaii? location Hawaii (AtValSet. 'type 'island)
295: How many films did Ingmar Bergman make? person Ingmar Bergman (AtValSet. )
298: What is California's state tree? gpe California (AtValSet. 'type 'state)
301: Who was the first coach of the Cleveland Browns? person the first coach of the Cleveland Browns (AtValSet. 'occupation 'sports)
301: Who was the first coach of the Cleveland Browns? organization Cleveland Browns (AtValSet. 'type 'sports)
302: How many people die from snakebite poisoning in the U.S. per year? gpe U.S. (AtValSet. 'type 'country)
303: Who is the prophet of the religion of Islam? person the prophet of the religion of Islam (AtValSet. 'occupation 'religion)
303: Who is the prophet of the religion of Islam? organization the religion of Islam (AtValSet. 'type 'religious)
304: Where is Tornado Alley? location Tornado Alley (AtValSet. 'type 'region)
304: Where is Tornado Alley? facility Tornado Alley (AtValSet. )
308: How many home runs did Babe Ruth hit in his lifetime? person Babe Ruth (AtValSet. 'occupation 'sports)
310: Where is the bridge over the river Kwai? facility the bridge over the river Kwai (AtValSet. 'type 'bridge)
310: Where is the bridge over the river Kwai? river the river Kwai (AtValSet. 'type 'river)
311: How many Superbowls have the 49ers won? organization 49ers (AtValSet. 'type 'sports)
311: How many Superbowls have the 49ers won? organization Superbowls (AtValSet. 'type 'competition)
312: Who was the architect of Central Park? location Central Park (AtValSet. 'type 'park)
312: Who was the architect of Central Park? facility Central Park (AtValSet. 'type 'park)
314: What is Alice Cooper's real name? person Alice Cooper (AtValSet. )
#315: Why can't ostriches fly?
#316: Name a tiger that is extinct?
317: Where is Guam? gpe Guam (AtValSet. 'type 'country)
318: Where did Bill Gates go to college? person Bill Gates (AtValSet. )
320: Where is Romania located? gpe Romania (AtValSet. 'type 'country)
321: When was the De Beers company founded? organization the De Beers company (AtValSet. 'type 'business)
322: Who was the first king of England? person the first king of England (AtValSet. 'occupation 'government)
322: Who was the first king of England? gpe England (AtValSet. )
324: What is California's capital? gpe California (AtValSet. 'type 'state)
325: What is the size of Argentina? gpe Argentina (AtValSet. 'type 'country)
```

327: When was the San Francisco fire? event the San Francisco fire (AtValSet. 'type 'disaster)
327: When was the San Francisco fire? gpe San Francisco (AtValSet. 'type 'city)
328: What was the man's name who was killed in a duel with Aaron Burr? person the man's name who was killed in a duel with Aaron Burr (AtValSet.)
328: What was the man's name who was killed in a duel with Aaron Burr? person Aaron Burr (AtValSet.)
329: What is the population of Mexico? gpe Mexico (AtValSet. 'type 'country)
331: How hot is the core of the earth? location earth (AtValSet. 'type 'astronomy)
331: How hot is the core of the earth? location the core of the earth (AtValSet.)
332: How long would it take to get from Earth to Mars? location Earth (AtValSet. 'type 'astronomy)
332: How long would it take to get from Earth to Mars? location Mars (AtValSet. 'type 'astronomy)
336: When was Microsoft established? organization Microsoft (AtValSet. 'type 'business)
339: What was the ball game of ancient Mayans called? people ancient Mayans (AtValSet. 'group #t)
341: How wide is the Atlantic Ocean? location Atlantic Ocean (AtValSet. 'type 'ocean)
345: What is the population of Kansas? gpe Kansas (AtValSet. 'type 'state)
350: How many Stradivarius violins were ever made?
357: What state in the United States covers the largest area? gpe the United States (AtValSet. 'type 'country)
359: Where is Melbourne? gpe Melbourne (AtValSet. 'type 'city)
360: How much in miles is a ten K run?
362: What is the capital of Burkina Faso? gpe Burkina Faso (AtValSet. 'type 'country)
362: What is the capital of Burkina Faso? gpe the capital of Burkina Faso (AtValSet. 'type 'city)
363: What is the capital of Haiti? gpe Haiti (AtValSet. 'type 'country)
363: What is the capital of Haiti? gpe the capital of Haiti (AtValSet. 'type 'city)
364: How many people lived in Nebraska in the mid 1980s? gpe Nebraska (AtValSet. 'type 'state)
365: What is the population of Mozambique? gpe Mozambique (AtValSet. 'type 'country)
366: Who won the Superbowl in 1982? organization Superbowl (AtValSet. 'type 'competition)
367: What is Martin Luther King Jr.'s real birthday? person Martin Luther King Jr. (AtValSet.)
368: Where is Trinidad? gpe Trinidad (AtValSet.)
368: Where is Trinidad? location Trinidad (AtValSet. 'type 'island)
369: Where did the Inuits live? person the Inuits (AtValSet. 'group #t)
372: When was the Triangle Shirtwaist fire? event the Triangle Shirtwaist fire (AtValSet. 'type 'disaster)
373: Where is the Kalahari desert? location the Kalahari desert (AtValSet. 'type 'desert)
375: What ocean did the Titanic sink in? facility the Titanic (AtValSet. 'type 'vehicle)
376: Who was the 33rd president of the United States? person the 33rd president of the United States (AtValSet. 'occupation 'government)
376: Who was the 33rd president of the United States? gpe the United States (AtValSet. 'type 'country)
377: At what speed does the Earth revolve around the sun? location the Earth (AtValSet. 'type 'astronomy)
377: At what speed does the Earth revolve around the sun? location the sun (AtValSet. 'type 'astronomy)
378: Who is the emperor of Japan? person the emperor of Japan (AtValSet. 'occupation 'government)
378: Who is the emperor of Japan? gpe Japan (AtValSet. 'type 'country)
380: What language is mostly spoken in Brazil? gpe Brazil (AtValSet. 'type 'country)
381: Who assassinated President McKinley? person President McKinley (AtValSet. 'occupation 'government)
382: When did Muhammad live? person Muhammad (AtValSet.)
387: What year did Montana become a state? gpe Montana (AtValSet. 'type 'state)
388: What were the names of the three ships used by Columbus? facility the three ships used by Columbus (AtValSet. 'type 'vehicle)
388: What were the names of the three ships used by Columbus? person Columbus (AtValSet.)
389: Who was the 21st U.S. President? person the 21st U.S. President (AtValSet. 'occupation 'government)
390: Where was John Adams born? person John Adams (AtValSet.)
391: Who painted Olympia?
394: What is the longest word in the English language?
397: When was the Brandenburg Gate in Berlin built? facility the Brandenburg Gate in Berlin (AtValSet. 'type 'gate)
397: When was the Brandenburg Gate in Berlin built? gpe Berlin (AtValSet. 'type 'city)
398: When is Boxing Day?
399: What is the exchange rate between England and the U.S.? gpe England (AtValSet.)
399: What is the exchange rate between England and the U.S.? gpe U.S. (AtValSet. 'type 'country)
400: What is the name of the Jewish alphabet? person Jewish (AtValSet. 'group #t)
402: What nationality was Jackson Pollock? person Jackson Pollock (AtValSet.)
403: Tell me what city the Kentucky Horse Park is near? facility the Kentucky Horse Park (AtValSet.)
404: What is the state nickname of Mississippi? gpe Mississippi (AtValSet. 'type 'state)
407: What is Black Hills, South Dakota most famous for? gpe Black Hills, South Dakota (AtValSet. 'type 'city)
407: What is Black Hills, South Dakota most famous for? gpe South Dakota (AtValSet. 'type 'state)
408: What kind of animal was Winnie the Pooh?
411: What tourist attractions are there in Reims? gpe Reims (AtValSet. 'type 'city)
412: Name a film in which Jude Law acted. person Jude Law (AtValSet.)

413: Where are the U.S. headquarters for Procter and Gamble? facility the U.S. headquarters for Procter and Gamble (AtValSet. 'type 'building)

413: Where are the U.S. headquarters for Procter and Gamble? organization Procter and Gamble (AtValSet. 'type 'business)

413: Where are the U.S. headquarters for Procter and Gamble? gpe U.S. (AtValSet. 'type 'country)

414: What's the formal name for Lou Gehrig's disease? person Lou Gehrig (AtValSet.)

415: What does CNN stand for? organization CNN (AtValSet. 'type 'news)

416: When was CNN's first broadcast? organization CNN (AtValSet. 'type 'news)

417: Who owns CNN? organization CNN (AtValSet. 'type 'news)

418: What is the name of a Salt Lake City newspaper? gpe Salt Lake City (AtValSet. 'type 'city)

418: What is the name of a Salt Lake City newspaper? organization a Salt Lake City newspaper (AtValSet. 'type 'news 'generic #t)

422: When did Princess Diana and Prince Charles get married? person Princess Diana (AtValSet.)

422: When did Princess Diana and Prince Charles get married? person Prince Charles (AtValSet.)

426: What format was VHS's main competition?

428: Where is Logan International located? facility Logan International (AtValSet. 'type 'airport)

429: What university was Woodrow Wilson President of? person Woodrow Wilson (AtValSet.)

430: Where is Basque country located? location Basque country (AtValSet. 'type 'region)

431: What does CPR stand for?

433: Who was Darth Vader's son? person Darth Vader's son (AtValSet. 'gender 'male)

433: Who was Darth Vader's son? person Darth Vader (AtValSet.)

435: How did Bob Marley die? person Bob Marley (AtValSet.)

436: What instrument is Ray Charles best known for playing? person Ray Charles (AtValSet.)

437: What is Dick Clark's birthday? person Dick Clark (AtValSet.)

440: Where was Poe born? person Poe (AtValSet.)

441: What king was forced to agree to the Magna Carta?

442: What's the name of Pittsburgh's baseball team? organization Pittsburgh's baseball team (AtValSet. 'type 'sports)

442: What's the name of Pittsburgh's baseball team? gpe Pittsburgh (AtValSet. 'type 'city)

444: Where is the location of the Orange Bowl? organization Orange Bowl (AtValSet. 'type 'competition)

445: When was the last major eruption of Mount St. Helens? location Mount St. Helens (AtValSet. 'type 'mountain)

446: What is the abbreviation for Original Equipment Manufacturer?

448: Where is Rider College located? organization Rider College (AtValSet. 'type 'educational)

449: What does Nicholas Cage do for a living? person Nicholas Cage (AtValSet.)

450: What does caliente mean (in English)?

451: Where is McCarren Airport? facility McCarren Airport (AtValSet. 'type 'airport)

452: Who created "The Muppets"? title "The Muppets" (AtValSet.)

453: When is Bastille Day?

454: What is the Islamic counterpart to the Red Cross? organization the Islamic counterpart to the Red Cross (AtValSet.)

454: What is the Islamic counterpart to the Red Cross? organization the Red Cross (AtValSet. 'type 'charity)

457: Where is Webster University? organization Webster University (AtValSet. 'type 'educational)

458: What's the name of a golf course in Myrtle Beach? facility a golf course in Myrtle Beach (AtValSet. 'generic #t)

458: What's the name of a golf course in Myrtle Beach? gpe Myrtle Beach (AtValSet. 'type 'city)

459: When was John D. Rockefeller born? person John D. Rockefeller (AtValSet.)

460: Name a Gaelic language. person Gaelic (AtValSet. 'group #t 'name "Irish")

461: Who was the author of the book about computer hackers called "The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage"?
person (AtValSet. 'occupation 'author)

461: Who was the author of the book about computer hackers called "The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage"?
title "The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage" (AtValSet.)

464: Where are the headquarters of Eli Lilly? facility the headquarters of Eli Lilly (AtValSet. 'type 'building)

464: Where are the headquarters of Eli Lilly? organization Eli Lilly (AtValSet. 'type 'business)

465: Where did Hillary Clinton graduate college? person Hillary Clinton (AtValSet.)

466: Where is Glasgow? gpe Glasgow (AtValSet. 'type 'city)

467: Who was Samuel Johnsons's friend and biographer? person Samuel Johnsons's friend and biographer (AtValSet. 'organization 'friend 'organization 'biographer)

467: Who was Samuel Johnsons's friend and biographer? person Samuel Johnsons (AtValSet.)

469: Who coined the term "cyberspace" in his novel "Neuromancer"? title "Neuromancer" (AtValSet. 'type 'book)

470: Who is the president of Bolivia? person the president of Bolivia (AtValSet. 'occupation 'government)

470: Who is the president of Bolivia? gpe Bolivia (AtValSet. 'type 'country)

471: What year did Hitler die? person Hitler (AtValSet.)

472: When did the American Civil War end? event the American Civil War (AtValSet. 'type 'war)

473: Who created the character of Scrooge? person the character of Scrooge (AtValSet. 'type 'character)

475: What is the salary of a U.S. Representative? person a U.S. Representative (AtValSet. 'occupation 'government 'generic #t)

476: Name one of the Seven Wonders of the Ancient World. location one of the Seven Wonders of the Ancient World (AtValSet. 'type 'monument 'generic #t)

476: Name one of the Seven Wonders of the Ancient World. location the Seven Wonders of the Ancient World (AtValSet. 'type 'monument 'group #t)

478: What is the airport code for Los Angeles International? facility Los Angeles International (AtValSet. 'type 'airport)

479: Who provides telephone service in Orange County, California? gpe Orange County, California (AtValSet. 'type 'county)

479: Who provides telephone service in Orange County, California? gpe California (AtValSet. 'type 'state)

480: What is the zip code for Fremont, CA? gpe Fremont, CA (AtValSet. 'type 'city)

480: What is the zip code for Fremont, CA? gpe CA (AtValSet. 'type 'state)

481: Who shot Billy the Kid? person Billy the Kid (AtValSet.)

482: Who is the monarch of the United Kingdom? person the monarch of the United Kingdom (AtValSet. 'occupation 'government)

482: Who is the monarch of the United Kingdom? gpe the United Kingdom (AtValSet. 'type 'country)

484: What is the name of the Lion King's son in the movie, "The Lion King"? person the Lion King's son (AtValSet. 'gender 'male)

484: What is the name of the Lion King's son in the movie, "The Lion King"? person the Lion King (AtValSet. 'occupation 'government 'gender 'male)

484: What is the name of the Lion King's son in the movie, "The Lion King"? title "The Lion King" (AtValSet. 'type 'movie)

485: Where is Amsterdam? gpe Amsterdam (AtValSet. 'type 'city)

487: What was the name of the movie that starred Sharon Stone and Arnold Schwarzenegger? person Sharon Stone (AtValSet.)

487: What was the name of the movie that starred Sharon Stone and Arnold Schwarzenegger? person Arnold Schwarzenegger (AtValSet.)

488: What continent is Bolivia on? gpe Bolivia (AtValSet. 'type 'country)

491: Where did Woodstock take place? event Woodstock (AtValSet. 'type 'concert)

493: What is Betsy Ross famous for? perso Betsy Ross (AtValSet.)

494: Who wrote the book, "The Grinch Who Stole Christmas"? title "The Grinch Who Stole Christmas" (AtValSet. 'type 'book)

495: When did Aldous Huxley write, "Brave New World"? person Aldous Huxley (AtValSet. 'occupation 'author)

495: When did Aldous Huxley write, "Brave New World"? title "Brave New World" (AtValSet. 'type 'book)

496: Who wrote the book, "Song of Solomon"? title "Song of Solomon" (AtValSet. 'type 'book)

497: Who portrayed Jake in the television show, "Jake and the Fatman"? person Jake (AtValSet.)

497: Who portrayed Jake in the television show, "Jake and the Fatman"? title "Jake and the Fatman" (AtValSet. 'type 'tv)

498: Who portrayed Fatman in the television show, "Jake and the Fatman"? person Fatman (AtValSet.)

498: Who portrayed Fatman in the television show, "Jake and the Fatman"? title "Jake and the Fatman" (AtValSet. 'type 'tv)

499: Where is Venezuela? gpe Venezuela (AtValSet. 'type 'country)

500: What city in Florida is Sea World in? gpe Florida (AtValSet. 'type 'state)

500: What city in Florida is Sea World in? facility Sea World (AtValSet. 'type 'attraction)

501: What is the population of Ohio? gpe Ohio (AtValSet. 'type 'state)

502: What is one of the cities that the University of Minnesota is located in? gpe one of the cities that the University of Minnesota is located in (AtValSet. 'type 'city 'generic #t)

502: What is one of the cities that the University of Minnesota is located in? gpe the cities that the University of Minnesota is located in (AtValSet. 'type 'city 'group #t)

502: What is one of the cities that the University of Minnesota is located in? organization the University of Minnesota (AtValSet. 'type 'educational)

503: What kind of sports team is the Buffalo Sabres? organization the Buffalo Sabres (AtValSet. 'type 'sports)

503: What kind of sports team is the Buffalo Sabres? gpe Buffalo (AtValSet. 'type 'city)

504: Who is the founder of the Wal-Mart stores? person the founder of the Wal-Mart stores (AtValSet.)

504: Who is the founder of the Wal-Mart stores? organization the Wal-Mart stores (AtValSet. 'type 'business)

505: What city is Massachusetts General Hospital located in? facility Massachusetts General Hospital (AtValSet. 'type 'hospital)

507: When did the California lottery begin? organization the California lottery (AtValSet. 'type 'governmental)

508: Where is Las Vegas? gpe Las Vegas (AtValSet. 'type 'city)

509: When was Beethoven born? person Beethoven (AtValSet.)

510: What's the name of the tiger that advertises for Frosted Flakes cereal?

511: Where is Tufts University? organization Tufts University (AtValSet. 'type 'educational)

512: What movie did Madilyn Kahn star in with Gene Wilder? person Madilyn Kahn (AtValSet.)

512: What movie did Madilyn Kahn star in with Gene Wilder? person Gene Wilder (AtValSet.)

513: When did the royal wedding of Prince Andrew and Fergie take place? person Prince Andrew (AtValSet.)

513: When did the royal wedding of Prince Andrew and Fergie take place? person Fergie (AtValSet.)

515: For what disease is the drug Sinemet used as a treatment?

517: What province is Edmonton located in? gpe Edmonton (AtValSet. 'type 'city)

518: In what area of the world was the Six Day War fought? event Six Day War (AtValSet. 'type 'war)

519: What is the zip code for Parsippany, NJ? gpe Parsippany, NJ (AtValSet. 'type 'city)

519: What is the zip code for Parsippany, NJ? gpe NJ (AtValSet. 'type 'state)

520: When was the first Barbie produced?

522: What does EKG stand for?

#523: What is Chiricahua the name of? location Chiricahua (AtValSet.)

524: Where did Wicca first develop? organization Wicca (AtValSet. 'type 'religious)

527: When was the NFL established? organization the NFL (AtValSet. 'type 'sports)

530: What other name were the "Little Rascals" known as? title "Little Rascals" (AtValSet.)

533: Who won the rugby world cup in 1987? organization the rugby world cup (AtValSet. 'type 'competition)

534: Where is Windsor Castle? facility Windsor Castle (AtValSet. 'type 'building)

536: What is the population of the United States? gpe the United States (AtValSet. 'type 'country)

541: What was the purpose of the Manhattan project? organization the Manhattan project (AtValSet.)

542: What is the most common kind of skin cancer in the U.S.? gpe the U.S. (AtValSet. 'type 'country)

545: Who wrote the song, "Silent Night"? title "Silent Night" (AtValSet. 'type 'song)

547: When was the Hoover Dam constructed? facility the Hoover Dam (AtValSet. 'type 'dam)

#548: What's the most famous tourist attraction in Rome? facility the most famous tourist attraction in Rome (AtValSet. 'type 'attraction)

548: What's the most famous tourist attraction in Rome? gpe Rome (AtValSet. 'type 'city)

549: At Christmas time, what is the traditional thing to do under the mistletoe?

550: What is the Pennsylvania state income tax rate? gpe Pennsylvania (AtValSet. 'type 'state)

551: What is the name of the Michelangelo painting that shows two hands with fingers touching? person Michelangelo (AtValSet. 'type 'artist)

552: What caused the Lynmouth floods? event the Lynmouth floods (AtValSet. 'type 'disaster 'group #t)

552: What caused the Lynmouth floods? location Lynmouth (AtValSet.)

554: How many zip codes are there in the U.S.? gpe the U.S. (AtValSet. 'type 'country)

555: What was the name of the Titanic's captain? facility the Titanic (AtValSet. 'type 'vehicle)

555: What was the name of the Titanic's captain? person the Titanic's captain (AtValSet. 'occupation 'captian)

557: When is the Tulip Festival in Michigan? organization the Tulip Festival in Michigan (AtValSet. 'type 'festival)

557: When is the Tulip Festival in Michigan? gpe Michigan (AtValSet. 'type 'state)

559: Who created the character James Bond? person the character James Bond (AtValSet. 'type 'character)

560: What store does Martha Stewart advertise for? person Martha Stewart (AtValSet.)

561: Name an American made motorcycle. gpe American (AtValSet. 'type 'country 'name "United States of America")

562: What Cruise Line does Kathie Lee Gifford advertise for? person Kathie Lee Gifford (AtValSet.)

563: Name a movie that the actress, Sandra Bullock, had a role in. person Sandra Bullock (AtValSet. 'occupation 'film)

564: Where is the Thomas Edison Museum? facility the Thomas Edison Museum (AtValSet. 'type 'museum)

566: What state does MO stand for? gpe MO (AtValSet. 'type 'state)

568: What state is the Filenes store located in? organization the Filenes store (AtValSet. 'type 'business)

570: What hockey team did Wayne Gretzky play for? person Wayne Gretzky (AtValSet. 'occupation 'sports)

572: How long would it take for a $50 savings bond to mature?

573: How many home runs did Lou Gehrig have during his career? person Lou Gehrig (AtValSet. 'type 'occupation 'sports)

575: What kind of a sports team is the Wisconsin Badgers? organization the Wisconsin Badgers (AtValSet. 'type 'sports)

576: What is the name of Joan Jett's band? organization Joan Jett's band (AtValSet. 'type 'musical)

576: What is the name of Joan Jett's band? person Joan Jett (AtValSet. 'occupation 'musical)

577: Can you give me the name of a clock maker in London, England? organization a clock maker in London, England (AtValSet. 'type 'business 'generic #t)

577: Can you give me the name of a clock maker in London, England? gpe London, England (AtValSet. 'type 'city)

577: Can you give me the name of a clock maker in London, England? gpe England (AtValSet.)

578: What was the name of the sitcom that Alyssa Milano starred in with Tony Danza? person Alyssa Milano (AtValSet.)

578: What was the name of the sitcom that Alyssa Milano starred in with Tony Danza? person Tony Danza (AtValSet.)

579: What is the real name of the singer, Madonna? person Madonna (AtValSet. 'occupation 'musical)

581: What flower did Vincent Van Gogh paint? person Vincent Van Gogh (AtValSet.)

582: What radio station did Paul Harvey work for? person Paul Harvey (AtValSet.)

583: What's the name of the song Will Smith sings about parents? person Will Smith (AtValSet. 'occupation 'musical)

584: What does NASA stand for? organization NASA (AtValSet. 'type 'governmental)

585: What famous model was married to Billy Joel? person Billy Joel (AtValSet.)

587: In what book can I find the story of Aladdin? person Aladdin (AtValSet.)

588: When did World War I start? event World War I (AtValSet. 'type 'war)

589: What state is Niagra Falls located in? location Niagra Falls (AtValSet. 'type 'waterfall)

590: What's the name of the actress who starred in the movie, "Silence of the Lambs"? person the actress who starred in the movie, "Silence of the Lambs" (AtValSet. 'occupation 'film 'gender 'female)

590: What's the name of the actress who starred in the movie, "Silence of the Lambs"? title "Silence of the Lambs" (AtValSet. 'type 'film)

591: Who owns the St. Louis Rams? organization the St. Louis Rams (AtValSet. 'type 'sports)

593: When was the movie, Caligula, made? title Caligula (AtValSet. 'type 'film)

594: Name an American war plane? gpe American (AtValSet. 'type 'country 'name "United States of America")

595: Where is Burma? gpe Burma (AtValSet. 'type 'country)

596: How many highway miles to the gallon can you get with the Ford Fiesta?

597: Name a novel written by John Steinbeck. person John Steinbeck (AtValSet. 'occupation 'author)

598: Who wrote the song, "Boys of Summer"? title "Boys of Summer" (AtValSet. 'type 'song)

599: What is the mascot for Notre Dame University? organization Notre Dame University (AtValSet. 'type 'educational)

602: Who manufactures the software, "PhotoShop"?

603: How many casinos are in Atlantic City, NJ? gpe Atlantic City, NJ (AtValSet. 'type 'city)

603: How many casinos are in Atlantic City, NJ? gpe NJ (AtValSet. 'type 'state)

604: What state does Martha Stewart live in? person Martha Stewart (AtValSet.)

606: What are Cushman and Wakefield known for? person Cushman (AtValSet.)

606: What are Cushman and Wakefield known for? person Wakefield (AtValSet.)

608: When was Nostradamus born? person Nostradamus (AtValSet.)

#609: When was "the Great Depression"?

612: Who created the comic strip, "Garfield"? title "Garfield" (AtValSet. 'type 'cartoon)

613: Where is the Isle of Man? location the Isle of Man (AtValSet. 'type 'island)

613: Where is the Isle of Man? gpe the Isle of Man (AtValSet.)

614: Who wrote the book, "Huckleberry Finn"? title "Huckleberry Finn" (AtValSet. 'type 'book)

615: What day is known as the "national day of prayer"?

619: What is the telephone number for the University of Kentucky? organization the University of Kentucky (AtValSet. 'type 'educational)

620: Who wrote "The Scarlet Letter"? title "The Scarlet Letter" (AtValSet.)

621: Name an art gallery in New York. facility an art gallery in New York (AtValSet. 'type 'gallery 'generic #t)

621: Name an art gallery in New York. gpe New York (AtValSet.)

#622: What type of hunting are retrievers used for?

623: What party was Winston Churchill a member of? person Winston Churchill (AtValSet. 'occupation 'government)

624: How was Teddy Roosevelt related to FDR? person Teddy Roosevelt (AtValSet.)

624: How was Teddy Roosevelt related to FDR? person FDR (AtValSet.)

625: When did the Chernobyl nuclear accident occur? event the Chernobyl nuclear accident (AtValSet. 'type 'disaster)

625: When did the Chernobyl nuclear accident occur? gpe Chernobyl (AtValSet. 'type 'city)

#626: What does SIDS stand for?

627: What's the name of the star of the cooking show, "Galloping Gourmet"? title "Galloping Gourmet" (AtValSet. 'type 'tv)

628: What city is 94.5 KDGE Radio located in? organization 94.5 KDGE Radio (AtValSet.)

629: What President became Chief Justice after his presidency? person President (AtValSet. 'occupation 'governmental)

629: What President became Chief Justice after his presidency? person Chief Justice (AtValSet. 'occupation 'governmental)

630: How tall is Kilimanjaro? location Kilimanjaro (AtValSet. 'type 'mountain)

631: Who won the nobel prize in literature in 1988? organization the nobel prize in literature (AtValSet. 'type 'competition)

632: What's the name of the Tampa newspaper? organization the Tampa newspaper (AtValSet. 'type 'news)

632: What's the name of the Tampa newspaper? gpe Tampa (AtValSet. 'type 'city)

634: What is Dr. Ruth's last name? person Dr. Ruth (AtValSet.)

636: Italy is the largest producer of what? gpe Italy (AtValSet. 'type 'country)

637: What wrestling star became "The Incredible Hulk"? title "The Incredible Hulk" (AtValSet.)

638: Who wrote "The Pit and the Pendulum"? title "The Pit and the Pendulum" (AtValSet.)

#639: Who manufacturers Magic Chef applicances?

640: When was the first Wall Street Journal published? organization Wall Street Journal (AtValSet. 'type 'news)

#641: What is the normal resting heart rate of a healthy adult?

642: Who's the lead singer of the Led Zeppelin band? person the lead singer of the Led Zeppelin band (AtValSet. 'occupation 'musical)

642: Who's the lead singer of the Led Zeppelin band? organization the Led Zeppelin band (AtValSet. 'type 'musical)

643: Who wrote "An Ideal Husband"? title "An Ideal Husband" (AtValSet.)

#645: When did the Dow first reach 2000?

646: Who was Charles Lindbergh's wife? person Charles Lindbergh's wife (AtValSet. 'gender 'female)

646: Who was Charles Lindbergh's wife? person Charles Lindbergh (AtValSet.)

647: How many miles is it from London, England to Plymouth, England? gpe London, England (AtValSet. 'type 'city)

647: How many miles is it from London, England to Plymouth, England? gpe Plymouth, England (AtValSet. 'type 'city)

647: How many miles is it from London, England to Plymouth, England? gpe England (AtValSet.)

648: Who is Secretary-General of the United Nations? person Secretary-General of the United Nations (AtValSet.)

648: Who is Secretary-General of the United Nations? organization the United Nations (AtValSet.)

649: Who played the teacher in Dead Poet's Society? person the teacher in Dead Poet's Society (AtValSet.)

650: How many counties are in Indiana? gpe Indiana (AtValSet. 'type 'state)

651: What actress starred in "The Lion in Winter"? title "The Lion in Winter" (AtValSet. 'type 'movie)

652: Where was Tesla born? person Tesla (AtValSet.)

653: What's the name of a hotel in Indianapolis? facility a hotel in Indianapolis (AtValSet. 'type 'hotel 'generic #t)

653: What's the name of a hotel in Indianapolis? gpe Indianapolis (AtValSet. 'type 'city)

654: What U.S. Government agency registers trademarks? organization U.S. Government agency (AtValSet. 'type 'governmental)

655: Who started the Dominos Pizza chain? organization the Dominos Pizza chain (AtValSet. 'type 'business)

658: Where is Kings Canyon? location Kings Canyon (AtValSet.)

659: Where is the Mayo Clinic? facility the Mayo Clinic (AtValSet. 'type 'hospital)

#660: How big is the Electoral College?

662: Where does Mother Angelica live? person Mother Angelica (AtValSet. 'occupation 'religious)

664: What is the name of a Greek god? person a Greek god (AtValSet. 'type 'deity 'generic #t)

666: What's the population of Mississippi? gpe Mississippi (AtValSet. 'type 'state)

667: What was the name of Jacques Cousteau's ship? facility Jacques Cousteau's ship (AtValSet. 'type 'vehicle)

667: What was the name of Jacques Cousteau's ship? person Jacques Cousteau (AtValSet.)

668: What are the names of Jacques Cousteau's two sons? person Jacques Cousteau (AtValSet.)

668: What are the names of Jacques Cousteau's two sons? person Jacques Cousteau's two sons (AtValSet. 'gender 'male 'group #t)

670: What does Final Four refer to in the sports world? organization Final Four (AtValSet. 'type 'competition)

671: What does Knight Ridder publish? organization Knight Ridder (AtValSet. 'type 'news)

#672: What task does the Bouvier breed of dog perform?

673: What sport do the Cleaveland Cavaliers play? organization the Cleaveland Cavaliers (AtValSet. 'type 'sports)

674: What year was the Avery Dennison company founded? organization the Avery Dennison company (AtValSet. 'type 'business)

675: What's the population of Biloxi, Mississippi? gpe Biloxi, Mississippi (AtValSet. 'type 'city)

675: What's the population of Biloxi, Mississippi? gpe Mississippi (AtValSet. 'type 'state)

676: Name a ballet company Mikhail Baryshnikov has danced for? organization a ballet company Mikhail Baryshnikov has danced for (AtValSet. 'type 'arts 'generic #t

676: Name a ballet company Mikhail Baryshnikov has danced for? person Mikhail Baryshnikov (AtValSet. 'occupation 'arts)

677: What was the name of the television show, staring Karl Malden, that had San Francisco in the title? person Karl Malden (AtValSet.)

677: What was the name of the television show, staring Karl Malden, that had San Francisco in the title? gpe San Francisco (AtValSet. 'type 'city)

678: Who was the founding member of the Pink Floyd band? person the founding member of the Pink Floyd band (AtValSet. 'occupation 'musical)

678: Who was the founding member of the Pink Floyd band? organization the Pink Floyd band (AtValSet. 'type 'musical)

679: What did Delilah do to Samson's hair? person Delilah (AtValSet.)

679: What did Delilah do to Samson's hair? person Samson (AtValSet.)

680: What's the name of the Tokyo Stock Exchange? organization the Tokyo Stock Exchange (AtValSet. 'type 'financial)

681: What actor first portrayed James Bond? person James Bond (AtValSet.)

#682: What type of horses appear on the Budweiser commercials?

685: When did Amtrak begin operations? organization Amtrak (AtValSet. 'type 'business 'type 'governmental)

686: Where is the Smithsonian Institute located? facility the Smithsonian Institute (AtValSet. 'type 'museum)

687: What year did the Vietnam War end? event the Vietnam War (AtValSet. 'type 'war)

687: What year did the Vietnam War end? gpe Vietnam (AtValSet. 'type 'country)

#688: What country are Godiva chocolates from?

689: How many islands does Fiji have? gpe Fiji (AtValSet. 'type 'country)

#690: What card company sells Christmas ornaments?

692: What year was Desmond Mpilo Tutu awarded the Nobel Peace Prize? person Desmond Mpilo Tutu (AtValSet.)

692: What year was Desmond Mpilo Tutu awarded the Nobel Peace Prize? organization the Nobel Peace Prize (AtValSet. 'type 'competition)

693: What city did the Flintstones live in? person the Flintstones (AtValSet. 'group #t)

694: Who was the oldest U.S. president? person the oldest U.S. president (AtValSet. 'occupation 'government)

695: Who was the tallest U.S. president? person the tallest U.S. president (AtValSet. 'occupation 'government)

696: Where is Santa Lucia? gpe Santa Lucia (AtValSet. 'type 'country)

697: Which U.S. President is buried in Washington, D.C.? gpe Washington, D.C. (AtValSet. 'type 'city)

697: Which U.S. President is buried in Washington, D.C.? gpe D.C. (AtValSet.)

698: Where is Ocho Rios? organization Ocho Rios (AtValSet. 'type 'festival)

699: What year was Janet Jackson's first album released? person Janet Jackson (AtValSet. 'occupation 'musical)

894: How far is it from Denver to Aspen? gpe Denver (AtValSet. 'type 'city)

894: How far is it from Denver to Aspen? gpe Aspen (AtValSet. 'type 'city)

895: What county is Modesto, California in? gpe Modesto, California (AtValSet. 'type 'city)

895: What county is Modesto, California in? gpe California (AtValSet. 'type 'state)

898: When did Hawaii become a state? gpe Hawaii (AtValSet. 'type 'state)

899: How tall is the Sears Building? facility the Sears Building (AtValSet. 'type 'building)

900: George Bush purchased a small interest in which baseball team? person George Bush (AtValSet.)

901: What is Australia's national flower? gpe Australia (AtValSet. 'type 'country)

906: What is the average weight of a Yellow Labrador?

907: Who was the first man to fly across the Pacific Ocean? person the first man to fly across the Pacific Ocean (AtValSet. 'occupation 'aviator)

907: Who was the first man to fly across the Pacific Ocean? location the Pacific Ocean (AtValSet. 'type 'ocean)

908: When did Idaho become a state? gpe Idaho (AtValSet. 'type 'state)

913: What year did the Titanic sink? facility the Titanic (AtValSet. 'type 'vehicle)

914: Who was the first American to walk in space? person the first American to walk in space (AtValSet. 'from "United States of America")

916: What river in the US is known as the Big Muddy? gpe the US (AtValSet. 'type 'country)

916: What river in the US is known as the Big Muddy? location the Big Muddy (AtValSet. 'type 'river)

#919: Who developed the Macintosh computer?

921: What imaginary line is halfway between the North and South Poles? location the North and South Poles (AtValSet. 'group #t)

922: Where is John Wayne airport? facility John Wayne airport (AtValSet. 'type 'airport)

923: What hemisphere is the Philippines in? gpe the Philippines (AtValSet. 'type 'country)

924: What is the average speed of the horses at the Kentucky Derby? organization the Kentucky Derby (AtValSet. 'type 'competition)

925: Where are the Rocky Mountains? location the Rocky Mountains (AtValSet. 'type 'mountain 'group #t)

928: When did John F. Kennedy get elected as President? person John F. Kennedy (AtValSet. 'occupation 'government)

929: How old was Elvis Presley when he died? person Elvis Presley (AtValSet.)

930: Where is the Orinoco River? location the Orinoco River (AtValSet. 'type 'river)

933: How many Great Lakes are there? location Great Lakes (AtValSet. 'type 'lake 'group #t)

940: How long did Rip Van Winkle sleep? person Rip Van Winkle (AtValSet.)

943: What is the name of the chocolate company in San Francisco? organization the chocolate company in San Francisco (AtValSet. 'type 'business)

943: What is the name of the chocolate company in San Francisco? gpe San Francisco (AtValSet. 'type 'city)

#946: Which comedian's signature line is "Can we talk"?

950: When did Elvis Presley die? person Elvis Presley (AtValSet.)

951: What is the capital of Yugoslavia? gpe the capital of Yugoslavia (AtValSet. 'type 'city)

951: What is the capital of Yugoslavia? gpe Yugoslavia (AtValSet. 'type 'country)

952: Where is Milan? gpe Milan (AtValSet. 'type 'city)

954: What is the oldest city in the United States? gpe the oldest city in the United States (AtValSet. 'type 'city)

954: What is the oldest city in the United States? gpe the United States (AtValSet. 'type 'country)

955: What was W.C. Fields' real name? person W.C. Fields (AtValSet.)

956: What river flows between Fargo, North Dakota and Moorhead, Minnesota? gpe Fargo, North Dakota (AtValSet. 'type 'city)

956: What river flows between Fargo, North Dakota and Moorhead, Minnesota? gpe Moorhead, Minnesota (AtValSet. 'type 'city)

956: What river flows between Fargo, North Dakota and Moorhead, Minnesota? gpe North Dakota (AtValSet. 'type 'state)

956: What river flows between Fargo, North Dakota and Moorhead, Minnesota? gpe Minnesota (AtValSet. 'type 'state)

958: What state did the Battle of Bighorn take place in? event the Battle of Bighorn (AtValSet. 'type 'battle)

962: What day and month did John Lennon die? person John Lennon (AtValSet.)

963: What strait separates North America from Asia? location North America (AtValSet. 'type 'continent)

963: What strait separates North America from Asia? location Asia (AtValSet. 'type 'continent)

964: What is the population of Seattle? gpe Seattle (AtValSet. 'type 'city)

965: How much was a ticket for the Titanic? facility the Titanic (AtValSet. 'type 'vehicle)

967: What American composer wrote the music for "West Side Story"? title "West Side Story" (AtValSet.)

968: Where is the Mall of the America? facility the Mall of the America (AtValSet. 'type 'building)

970: What type of currency is used in Australia? gpe Australia (AtValSet. 'type 'country)

971: How tall is the Gateway Arch in St. Louis, MO? facility the Gateway Arch in St. Louis, MO (AtValSet. 'type 'monument)

971: How tall is the Gateway Arch in St. Louis, MO? gpe St. Louis, MO (AtValSet. 'type 'city)

971: How tall is the Gateway Arch in St. Louis, MO? gpe MO (AtValSet. 'type 'state)

973: Who was the first governor of Alaska? person the first governor of Alaska (AtValSet. 'occupation 'government)

973: Who was the first governor of Alaska? gpe Alaska (AtValSet. 'type 'state)

976: Who was elected president of South Africa in 1994? gpe South Africa (AtValSet. 'type 'country)

977: What is the population of China? gpe China (AtValSet. 'type 'country)

978: When was Rosa Parks born? person Rosa Parks (AtValSet.)

981: Who was the first female United States Representative? person the first female United States Representative (AtValSet. 'gender 'female 'occupation 'governmen

983: What country did Ponce de Leon come from? person Ponce de Leon (AtValSet.)

984: The U.S. Department of Treasury first issued paper currency for the U.S. during which war? organization U.S. Department of Treasury
(AtValSet. 'type 'governmental 'type 'financial)

984: The U.S. Department of Treasury first issued paper currency for the U.S. during which war? gpe U.S. (AtValSet. 'type 'country)

987: What year did Canada join the United Nations? gpe Canada (AtValSet. 'type 'country)

987: What year did Canada join the United Nations? organization the United Nations (AtValSet.)

988: What is the oldest university in the US? organization the oldest university in the US (AtValSet. 'type 'educational)

988: What is the oldest university in the US? gpe the US (AtValSet. 'type 'country)

989: Where is Prince Edward Island? gpe Prince Edward Island (AtValSet. 'type 'province)

990: Mercury, what year was it discovered? location Mercury (AtValSet. 'type 'astronomy)

998: What county is Phoenix, AZ in? gpe Phoenix, AZ (AtValSet. 'type 'city)

998: What county is Phoenix, AZ in? gpe AZ (AtValSet. 'type 'state)

1001: What is the Ohio state bird? gpe Ohio (AtValSet. 'type 'state)

1002: When were William Shakespeare's twins born? person William Shakespeare's twins (AtValSet. 'group #t)

1002: When were William Shakespeare's twins born? person William Shakespeare (AtValSet.)

1003: What is the highest dam in the U.S.? facility the highest dam in the U.S. (AtValSet. 'type 'dam)

1003: What is the highest dam in the U.S.? gpe the U.S. (AtValSet. 'type 'country)

1006: What is the length of the coastline of the state of Alaska? gpe the state of Alaska (AtValSet. 'type 'state)

1007: What is the name of Neil Armstrong's wife? person Neil Armstrong's wife (AtValSet. 'gender 'female)

1007: What is the name of Neil Armstrong's wife? person Neil Armstrong (AtValSet.)

1008: What is Hawaii's state flower? gpe Hawaii (AtValSet. 'type 'state)

1009: Who won Ms. American in 1989? organization Ms. American (AtValSet. 'type 'competition)

1010: When did the Hindenberg crash? facility the Hindenberg (AtValSet. 'type 'vehicle)

1012: What was the last year that the Chicago Cubs won the World Series? organization the Chicago Cubs (AtValSet. 'type 'sports)

1012: What was the last year that the Chicago Cubs won the World Series? organization the World Series (AtValSet. 'type 'competition)

1013: Where is Perth? gpe Perth (AtValSet. 'type 'city)

1014: What year did WWII begin? event WWII (AtValSet. 'type 'war)

1017: Who discovered America? location America (AtValSet. 'type 'continent)

1020: How wide is the Milky Way galaxy? location the Milky Way galaxy (AtValSet. 'type 'astronomy)

1025: Who was the abolitionist who led the raid on Harper's Ferry in 1859? person the abolitionist who led the raid on Harper's Ferry in 1859 (AtValSet. 'occupation 'abolitionist)

1025: Who was the abolitionist who led the raid on Harper's Ferry in 1859? event the raid on Harper's Ferry in 1859 (AtValSet.)

1025: Who was the abolitionist who led the raid on Harper's Ferry in 1859? facility Harper's Ferry (AtValSet.)

1029: What is the major fault line near Kentucky? location the major fault line near Kentucky (AtValSet.)

1029: What is the major fault line near Kentucky? gpe Kentucky (AtValSet. 'type 'state)

1030: Where is the Holland Tunnel? facility the Holland Tunnel (AtValSet. 'type 'tunnel)

1031: Who wrote the hymn "Amazing Grace"? title "Amazing Grace" (AtValSet. 'type 'song)

1032: What position did Willie Davis play in baseball? person Willie Davis (AtValSet. 'occupation 'sports)

1035: What is the name of Roy Roger's dog? person Roy Roger (AtValSet.)

1036: Where are the National Archives? facility the National Archives (AtValSet.)

1040: Who is a German philosopher? person a German philosopher? (AtValSet. 'occupation 'philosopher 'from "Germany" 'generic #t)

1041: What were Christopher Columbus' three ships? facility Christopher Columbus' three ships (AtValSet. 'type 'vehicle 'group #t)

1041: What were Christopher Columbus' three ships? person Christopher Columbus (AtValSet.)

#1044: What is another name for vitamin B1?

1047: When was Algeria colonized? gpe Algeria (AtValSet. 'type 'country)

1049: What continent is Egypt on? gpe Egypt (AtValSet. 'type 'country)

1050: What is the capital of Mongolia? gpe the capital of Mongolia (AtValSet. 'type 'city)

1050: What is the capital of Mongolia? gpe Mongolia (AtValSet. 'type 'country)

1052: In the late 1700's British convicts were used to populate which colony? person British convicts (AtValSet. 'from "United Kingdom" 'group #t)

1056: When is hurricane season in the Caribbean? location the Caribbean (AtValSet. 'type 'region)

1057: Where is the volcano Mauna Loa? location the volcano Mauna Loa (AtValSet. 'type 'mountain)

#1058: What is another astronomic term for the Northern Lights?

1059: What peninsula is Spain part of? gpe Spain (AtValSet. 'type 'country)

1060: When was Lyndon B. Johnson born? person Lyndon B. Johnson (AtValSet.)

1063: Who founded American Red Cross? organization American Red Cross (AtValSet.)

1064: What year did the Milwaukee Braves become the Atlanta Braves? organization the Milwaukee Braves (AtValSet. 'type 'sports)

1064: What year did the Milwaukee Braves become the Atlanta Braves? organization the Atlanta Braves (AtValSet. 'type 'sports)

1068: Where is the Shawnee National Forest? location the Shawnee National Forest (AtValSet. 'type 'forest)

#1069: What U.S. state's motto is "Live free or Die"?

1070: Where is the Louvre? facility the Louvre (AtValSet. 'type 'museum)

1073: How far is Pluto from the sun? location Pluto (AtValSet. 'type 'astronomy)

1073: How far is Pluto from the sun? location the sun (AtValSet. 'type 'astronomy)

1074: What body of water are the Canary Islands in? location the Canary Islands (AtValSet. 'type 'island 'group #t)

1076: Where is the Euphrates River? location the Euphrates River (AtValSet. 'type 'river)

1079: Who is the Prime Minister of Canada? person the Prime Minister of Canada (AtValSet. 'occupation 'government)

1079: Who is the Prime Minister of Canada? gpe Canada (AtValSet. 'type 'country)

1080: What French ruler was defeated at the battle of Waterloo? event the battle of Waterloo (AtValSet. 'type 'battle)

1080: What French ruler was defeated at the battle of Waterloo? gpe Waterloo (AtValSet. 'type 'city)

1082: Where did Howard Hughes die? person Howard Hughes (AtValSet.)

#1083: What is the birthstone for June?

1084: What is the sales tax in Minnesota? gpe Minnesota (AtValSet. 'type 'state)

1085: What is the distance in miles from the earth to the sun? location the earth (AtValSet. 'type 'astronomy)

1085: What is the distance in miles from the earth to the sun? location the sun (AtValSet. 'type 'astronomy)

1087: When was the first Wal-Mart store opened? facility the first Wal-Mart store (AtValSet. 'type 'building)

1090: What currency is used in Algeria? gpe Algeria (AtValSet. 'type 'country)

1094: What is the name of the satellite that the Soviet Union sent into space in 1957? gpe the Soviet Union (AtValSet. 'type 'country)

1095: What city's newspaper is called "The Enquirer"? organization The Enquirer (AtValSet. 'type 'news)

1099: Where is the volcano Olympus Mons located? location the volcano Olympus Mons (AtValSet. 'type 'mountain)

1100: Who was the 23rd president of the United States? person 23rd president of the United States (AtValSet. 'occupation 'government)

1100: Who was the 23rd president of the United States? gpe the United States (AtValSet. 'type 'country)

1104: What year did the United States abolish the draft? gpe the United States (AtValSet. 'type 'country)

1106: What province is Montreal in? gpe Montreal (AtValSet. 'type 'city)

1107: What New York City structure is also known as the Twin Towers? gpe New York City (AtValSet. 'type 'city)

1107: What New York City structure is also known as the Twin Towers? facility the Twin Towers (AtValSet. 'type 'building)

1109: What is the most frequently spoken language in the Netherlands? gpe the Netherlands (AtValSet. 'type 'country)

#1115: What year was the Mona Lisa painted?

#1116: What does "Sitting Shiva" mean?

1117: What is the electrical output in Madrid, Spain? gpe Madrid, Spain (AtValSet. 'type 'city)

1117: What is the electrical output in Madrid, Spain? gpe Spain (AtValSet. 'type 'country)

1118: Which mountain range in North America stretches from Maine to Georgia? location North America (AtValSet. 'type 'continent)

1118: Which mountain range in North America stretches from Maine to Georgia? gpe Maine (AtValSet. 'type 'state)

1118: Which mountain range in North America stretches from Maine to Georgia? gpe Georgia (AtValSet. 'type 'state)

1120: What is the population of Nigeria? gpe Nigeria (AtValSet. 'type 'country)

1122: Where is the Grand Canyon? location the Grand Canyon (AtValSet.)

1124: What year did the U.S. buy Alaska? gpe the U.S. (AtValSet. 'type 'country)

1124: What year did the U.S. buy Alaska? gpe Alaska (AtValSet. 'type 'state)

1125: What is the name of the leader of Ireland? person the leader of Ireland (AtValSet. 'occupation 'government)

1125: What is the name of the leader of Ireland? gpe Ireland (AtValSet. 'type 'country)

1128: What are the two houses of the Legislative branch? organization the two houses of the Legislative branch (AtValSet. 'type 'governmental 'group #t)

1128: What are the two houses of the Legislative branch? organization the Legislative branch (AtValSet. 'type 'governmental)

1130: In Poland, where do most people live? gpe Poland (AtValSet. 'type 'country)

1132: What is the location of the Sea of Tranquility? location the Sea of Tranquility (AtValSet. 'type 'sea)

#1134: What French province is cognac produced in?

1139: What is the longest suspension bridge in the U.S.? facility the longest suspension bridge in the U.S. (AtValSet. 'type 'bridge)

1139: What is the longest suspension bridge in the U.S.? gpe the U.S. (AtValSet. 'type 'country)

1146: What did Edward Binney and Howard Smith invent in 1903? person Edward Binney (AtValSet.)

1146: What did Edward Binney and Howard Smith invent in 1903? person Howard Smith (AtValSet.)

1150: What is the depth of the Nile river? location the Nile river (AtValSet. 'type 'river)

Appendix B

Query Judgements

This appendix shows the full judgements for the named entity evaluation. This was done in two segments, one segment where the annotations differed between sepia only and all phrase annotators, and one segment where no other annotator marked phrases in that question. therefore they did not differ.

203 How much folic acid should an expectant mother get daily?
212 Who invented the electric guitar?
214 How many hexagons are on a soccer ball?
220 Who is the prime minister of [sepia-location: Australia]?
221 Who killed [sepia-person: Martin Luther King]?
225 Who is the Greek God of the Sea?
231 Who was the president of Vichy France?
246 What did Vasco da Gama discover?
247 Who won the Battle of [sepia-location: Gettysburg]?
249 Where is the Valley of the [sepia-organization: Kings]?
252 When was the first flush toilet invented?
255 Who thought of teaching people to tie their shoe laces?
261 What company sells the most greeting cards?
263 When was Babe [sepia-person: Ruth] born?
275 About how many soldiers died in World War II?
278 What was the death toll at the eruption of [sepia-location: Mount Pinatubo]?
280 What's the tallest building in [sepia-location: New York City]?
281 When did [sepia-person: Geraldine Ferraro] run for vice president?
284 What is the life expectancy of an elephant?
289 What are [sepia-person: John C. Calhoun] and [sepia-person: Henry] Clay known as?
290 OK When was Hurricane [sepia-person: Hugo]?
295 How many films did [sepia-person: Ingmar Bergman] make?
296 What is the federal minimum wage?
304 Where is Tornado Alley?
308 How many home runs did Babe [sepia-person: Ruth] hit in his lifetime?
312 Who was the architect of [sepia-location: Central Park]?
318 Where did [sepia-location: Bill Gates] go to college?
327 When was the [sepia-location: San Francisco] fire?
328 What was the man's name who was killed in a duel with [sepia-person: Aaron] Burr?
333 What is the name of the second space shuttle?
337 What's the average salary of a professional baseball player?
339 What was the ball game of ancient Mayans called?
341 How wide is the [sepia-location: Atlantic Ocean]?
357 What state in the [sepia-location: United States] covers the largest area?
362 What is the capital of [sepia-location: Burkina Faso]?
367 What is [sepia-person: Martin Luther King Jr].'s real birthday?
372 When was the Triangle Shirtwaist fire?
376 Who was the 33rd president of the [sepia-location: United States]?
381 Who assassinated [sepia-person: President McKinley]?
389 Who was the 21st [sepia-organization: U.S. President]?
393 Where is your corpus callosum?
394 What is the longest word in the English language?
398 When is Boxing Day?
399 What is the exchange rate between [sepia-location: England] and the U.S.?
402 What nationality was [sepia-person: Jackson Pollock]?
comb ~~405~~ Tell me what city the [sepia-location: Kentucky] [sepia-location: Horse Park] is near?
405 Who used to make cars with rotary engines?
408 What kind of animal was [sepia-person: Winnie] the Pooh?
412 Name a film in which [sepia-person: Jude] Law acted.
414 What's the formal name for [sepia-person: Lou Gehrig]'s disease?
418 What is the name of a [sepia-location: Salt Lake City] newspaper?
422 When did [sepia-person: Princess Diana] and [sepia-person: Prince Charles] get married?
428 Where is [sepia-person: Logan] International located? OK
429 What university was [sepia-person: Woodrow Wilson President] of? X
433 Who was Darth Vader's son?
435 How did [sepia-person: Bob Marley] die?
441 What king was forced to agree to the Magna Carta?
442 What's the name of [sepia-location: Pittsburgh]'s baseball team?
443 What is the chemical formula/name for napalm?
444 Where is the location of the Orange Bowl?
445 When was the last major eruption of [sepia-location: Mount St]. Helens?
446 What is the abbreviation for Original Equipment Manufacturer?
448 Where is [sepia-organization: Rider College] located?
449 What does [sepia-person: Nicholas] Cage do for a living?
451 Where is [sepia-organization: McCarren Airport]?
452 Who created "The Muppets"?
453 When is Bastille Day?
454 What is the Islamic counterpart to the Red Cross?
456 What is the busiest air travel season?
457 Where is [sepia-organization: Webster University]?
458 What's the name of a golf course in [sepia-location: Myrtle Beach]?
461 Who was the author of the book about computer hackers called "The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage"?
467 Who was [sepia-person: Samuel Johnsons]'s friend and biographer?
469 Who coined the term "cyberspace" in his novel "Neuromancer"?
472 When did the American Civil War end?
474 Who first broke the sound barrier?
476 Name one of the Seven Wonders of the Ancient World.
478 What is the airport code for [sepia-location: Los Angeles] International?
479 Who provides telephone service in Orange County, [sepia-location: California]?
480 What is the zip code for [sepia-location: Fremont, CA]?
481 Who shot [sepia-person: Billy] the Kid?
482 Who is the monarch of the [sepia-location: United Kingdom]?
484 What is the name of the [sepia-person: [sepia-person: Lion King]]'s son in the movie, "The Lion King"?
486 How many states have a "lemon law" for new automobiles?
487 What was the name of the movie that starred [sepia-person: Sharon] Stone and [sepia-person: Arnold Schwarzenegger]?
490 Where did guinea pigs originate?
494 Who wrote the book, "The Grinch Who Stole Christmas"?
495 When did [sepia-person: Aldous Huxley] write, "Brave [sepia-location: New World]"?
496 Who wrote the book, "Song of [sepia-person: Solomon]"?
497 Who portrayed [sepia-person: [sepia-person: Jake]] in the television show, "Jake and the Fatman"?
498 Who portrayed Fatman in the television show, "[sepia-person: Jake] and the Fatman"?
500 What city in [sepia-location: Florida] is Sea World in?
502 What is one of the cities that the [sepia-organization: University of Minnesota] is located in?
503 What kind of sports team is the Buffalo Sabres?
505 What city is [sepia-organization: Massachusetts General Hospital] located in?
506 Who reports the weather on the Good Morning [sepia-location: America] television show?
508 Where is [sepia-location: Las Vegas]?
511 Where is [sepia-organization: Tufts University]?
512 What movie did Madilyn Kahn star in with Gene Wilder?
513 When did the royal wedding of [sepia-person: Prince Andrew] and Fergie take place?
514 What cereal goes "snap, crackle, pop"?
518 In what area of the world was the Six Day War fought?
519 What is the zip code for Parsippany, NJ?

577 lines
108 mistakes
329 marked ents
53 missing
14 miscat
12 other
16 not enough name
6 combine
4 token bug

971 How tall is the Gateway Arch in [sepia-location: St. Louis, MO]?
972 How much does the human adult female brain weigh?
978 When was [sepia-person: Rosa] Parks born?
981 Who was the first female [sepia-organization: United States Representative]?
984 The [sepia-organization: [sepia-location: U.S.] Department of Treasury] first issued paper currency for the U.S. dur
ing which war?
987 What year did [sepia-location: Canada] join the United Nations?
989 Where is [sepia-location: Prince Edward Island]?
993 What is the longest major league baseball-winning streak?
1002 When were [sepia-person: William Shakespeare]'s twins born? but OK
1009 Who won [sepia-person: Ms. American] in 1989? —wrong
1012 What was the last year that the [sepia-organization: Chicago Cubs] won the World Series?
1015 What is the diameter of a golf ball?
1020 How wide is the Milky Way galaxy?
1023 What is the gestation period for a cat?
1029 What is the major fault line near [sepia-location: Kentucky]?
1030 Where is the [sepia-location: Holland] Tunnel? OK
1031 Who wrote the hymn "Amazing Grace"?
1036 Where are the [sepia-organization: National Archives]?
1039 What is the longest bone in the human body?
1041 What were [sepia-person: Christopher Columbus]' three ships?
1044 What is another name for vitamin B1?
1048 What baseball team was the first to make numbers part of their uniform?
1050 What is the capital of [sepia-location: Mongolia]?
1057 Where is the volcano Mauna Loa?
1063 Who founded American Red Cross?
1064 What year did the [sepia-organization: Milwaukee Braves] become the [sepia-organization: Atlanta Braves]?
1066 When is the summer solstice?
1068 Where is the [sepia-organization: Shawnee National Forest]?
1069 What [sepia-location: U.S.] state's motto is "Live free or Die"?
1074 What body of water are the [sepia-location: Canary Islands] in?
1076 Where is the [sepia-location: Euphrates River]?
1078 What is natural gas composed of?
1079 Who is the Prime Minister of [sepia-location: Canada]?
1080 What French ruler was defeated at the battle of Waterloo?
1084 What is the sales tax in [sepia-location: Minnesota]?
1089 What city has the zip code of 35824?
1094 What is the name of the satellite that the [sepia-organization: Soviet Union] sent into space in 1957?
1095 What city's newspaper is called "[sepia-organization: The Enquirer]"?
1098 What is the melting point of copper?
1100 Who was the 23rd president of the [sepia-location: United States]?
1101 What is the average body temperature?
1103 What is the effect of acid rain?
1104 What year did the [sepia-location: United States] abolish the draft?
1105 How fast is the speed of light?
1107 What [sepia-location: New York City] structure is also known as the Twin Towers?
1116 What does "Sitting Shiva" mean?
1122 Where is the [sepia-location: Grand Canyon]?
1139 What is the longest suspension bridge in the U.S.?
1146 What did [sepia-person: Edward Binney] and [sepia-person: Howard Smith] invent in 1903?
1149 What planet is known as the "red" planet?
1153 Mexican pesos are worth what in [sepia-location: U.S.] dollars?

1154 Who was the first African American to play for the [sepia-organization: Brooklyn Dodgers]?
1155 Who was the first Prime Minister of [sepia-location: Canada]?
1158 How old was [sepia-person: Joan] of Arc when she died?
1161 What is the capital of [sepia-location: Ethiopia]?
1165 What is the difference between AM radio stations and FM radio stations?
1186 What is the life expectancy of a dollar bill?
1199 Where is the [sepia-location: Savannah River]?
1200 Who was the first woman killed in the [sepia-location: Vietnam] War?
1205 Where is the Eiffel Tower?
1218 What date was [sepia-person: Dwight D. Eisenhower] born?
1227 What is the name of [sepia-person: William Penn]'s ship?
1228 What is the melting point of gold? fac. 1.1 g, but OK
1229 What is the street address of the [sepia-organization: White House]?
1233 What is the percentage of water content in the human body?
1239 Who painted the ceiling of the [sepia-location: Sistine Chapel]?
1245 What is the location of [sepia-location: Lake Champlain]?
1249 Who wrote "The Devine Comedy"?
1250 What is the speed of light?
1251 What is the width of a football field?
1252 Why in tennis are zero points called love?
1253 What kind of dog was Toto in the Wizard of Oz?
1256 What is the only artery that carries blue blood from the heart to the lungs?
1257 How often does Old Faithful erupt at [sepia-organization: Yellowstone National Park]?
1260 What color does litmus paper turn when it comes into contact with a strong acid?
1263 What soviet seaport is on the [sepia-location: Black Sea]?
1269 When did [sepia-location: North Carolina] enter the union?
1274 Who killed [sepia-person: John F. Kennedy]?
1275 Who was the first vice president of the U.S.?
1278 How old was the youngest president of the [sepia-location: United States]?
1279 When was [sepia-person: Ulysses S. Grant] born?
1284 Who invented the instant Polaroid camera?
1291 What is the sales tax rate in [sepia-location: New York]?
1296 When is Father's Day?
1299 What city's newspaper is called "[sepia-organization: The Star]"?
1302 When was the [sepia-location: Boston] tea party?
1304 Which [sepia-location: U.S.A.] president appeared on "Laugh-In"?
1306 What is the capital of [sepia-location: Zimbabwe]?
1321 Where is the tallest roller coaster located?
1326 Where are the British crown jewels kept?
1327 Who was the first person to reach the North Pole?
1336 Which country has the most water pollution?
1338 Who is the actress known for her role in the movie "Gypsy"?
1339 What breed of hunting dog did the [sepia-location: Beverly] Hillbillies own?
1341 Who was the first African American to win the Nobel Prize in literature?
1342 When is [sepia-person: St. Patrick]'s Day?
1346 What is the active ingredient in baking soda?
1351 Where was the first golf course in the [sepia-location: United States]?
1358 In which state would you find the Catskill Mountains?
1361 What chain store is headquartered in [sepia-location: Bentonville, Arkansas]?
1373 Which country gave [sepia-location: New York] the Statue of Liberty?
1380 What city is also known as "The Gateway to the West"?
1382 What is the source of natural gas?

1643 Who founded [sepia-location: Rhode Island]?
1645 How much is the international space stations expected to cost?
1651 What is another name for the [sepia-organization: North Star]?
1652 When did the [sepia-location: United States] enter World War II?
1653 How do you say "French fries" in French?
1657 What do the French call the [sepia-location: English Channel]?
1658 What year was [sepia-person: Robert] Frost born?
1660 What is [sepia-person: Elvis Presley]'s middle name?
1661 What does "E Pluribus Unum" mean?
1662 When was [sepia-location: Jerusalem] invaded by the General [sepia-person: Titus]? OK
1664 Who was the first person to make the helicopter?
1665 When did [sepia-person: Marian Anderson] sing at the [sepia-location: Lincoln] Memorial?
1667 What is the abbreviation for the [sepia-location: London] stock exchange?
1668 What is the oldest national park in the U.S.?
1669 How tall is [sepia-location: Mount McKinley]?
1671 Where is Big [sepia-person: Ben]?
1672 What Latin American country is the leading exporter of sugar cane?
1673 What time of year does the peregrine falcon breed?
1675 What group sang the song "Happy Together"?
1677 How did Micky Mantle die?
1680 What country was ruled by [sepia-person: King Arthur]?
1684 What card game uses only 48 cards?
1685 What is the most populous city in the [sepia-location: United States]?
1687 What president declared Mothers' Day?
1689 What war is connected with the book "Charge of the Light Brigade"?
1691 Where was the movie "Somewhere in [sepia-organization: Time]" filmed?
1692 How tall is the Eiffel Tower in [sepia-location: France]?
1697 Where is the Statue of Liberty?
1699 What city is [sepia-organization: Purdue University] in?
1701 Where was [sepia-person: President Lincoln] buried?
1702 Whose business slogan is "Quality is job 1"?
1703 What award did Sterling North's book "Rascal" win in 1963?
1704 What is the normal pulse rate?
1706 What is the beginning date for the Hershey foods company?
1714 What province in [sepia-location: Canada] is [sepia-location: Niagara Falls] located in?
1715 How much vitamin C should you take in a day?
1716 Who was the first Triple Crown Winner?
1718 When was the [sepia-organization: White House] built?
1721 How far is the pitchers mound from home plate in softball?
1723 What is [sepia-person: Madonna]'s last name?
1725 Where did [sepia-person: David Ogden Stiers] get his undergraduate degree?
1728 When did [sepia-organization: Yankee Stadium] first open?
1730 What is [sepia-person: Mark Twain]'s real name?
1731 How often does the [sepia-location: United States] government conduct an official population census?
1732 What are the opening words of the Declaration of [sepia-location: Independence]?
1734 How do you say "pig" in Spanish?
1735 What city is [sepia-organization: Southwestern University] in?
1740 What is the [sepia-location: Stanley Cup] made of?
1741 What author wrote under the pen name "Boz"?
1743 Which state has the longest coastline on the [sepia-location: Atlantic Ocean]?
1744 What car company invented the Edsel?
1746 Who stabbed [sepia-person: Monica Seles]?

1748 When were the [sepia-location: Los Angeles] riots?
1751 Where is [sepia-location: Mesa Verde] [sepia-location: National park]?
1753 When was the [sepia-location: Vietnam] Veterans Memorial in [sepia-location: Washington, D.C.] built?
1754 When did the [sepia-location: Persian Gulf] War occur?
1755 What was the profession of American patriot [sepia-person: Paul] [sepia-location: Revere]?
1757 When did the battle of Iwo Jima take place?
1758 What is the "Sunflower State"?
1759 Who wrote "Fiddler on the Roof?"
1760 Where was C.S. [sepia-person: Lewis] born?
1762 What is the name of [sepia-person: Scarlett O]'Hara's house?
1767 When was the first Ford Mustang made?
1768 What TV series did Pierce Brosnan play in?
1769 Who is the owner of the [sepia-organization: St. Petersburg Times]?
1772 Who invented the cotton gin?
1773 What was [sepia-person: Aaron Copland]'s most famous piece of music?
1777 Who did [sepia-person: Scott Bakula] play in "American Beauty"?
1778 When did [sepia-person: Walt Disney] die?
1780 Who has the most no hitters in major league baseball?
1781 What year did "Snow White" come out?
1785 What body of water does the [sepia-location: Euphrates River] empty into?
1793 What is the world's tallest office building?
1797 How did Adolf Hitler die?
1799 What is the life expectancy of the average woman in [sepia-location: Nigeria]?
1802 How tall is [sepia-person: Tom] Cruise?
1808 What island did the [sepia-location: U.S.] gain after the Spanish American war?
1809 When was the [sepia-location: Buckingham] Palace built in [sepia-location: London], [sepia-location: England]?
1810 Where are the British Crown jewels kept?
1813 When were the first postage stamps issued in the [sepia-location: United States]?
1816 How old must you be to become President of the [sepia-location: United States]?
1818 Where did [sepia-person: Golda Meir] grow up? China
1821 Who were the architects who designed the Empire State Building?
1829 How tall is the [sepia-location: CN] Tower in [sepia-location: Toronto]?
1830 What city does the [sepia-person: Tour de France] end in?
1831 What is the name of [sepia-person: Abbott] and [sepia-person: Costello]'s famous routine?
1832 What did [sepia-person: Walter Cronkite] say at the end of every show?
1837 What year was [sepia-location: Ebbets Field], home of [sepia-organization: Brooklyn Dodgers], built?
1841 What's the final line in the [sepia-person: Edgar Allen Poe] poem "The Raven?"
1844 When was the Hellenistic Age?
1853 Where was the [sepia-location: Andersonville] Prison?
1854 What was [sepia-person: William Shakespeare]'s occupation before he began to write plays?
1857 What is the length of Churchill Downs racetrack?
1860 What Broadway musical is the song "The Story is Me" from?
1861 Where was [sepia-location: Bill Gates] born?
1863 Who said "I have not begun to fight!"?
1868 What capital is on the [sepia-location: Susquehanna River]?
1872 How did [sepia-person: Eva Peron] die?
1874 When was the battle of [sepia-person: Shiloh]?
1882 What nationality is [sepia-person: Sean Connery]?
1883 How much of the ozone layer is depleted?
1888 What year was the light bulb invented?
1889 How old is the Red Pyramid?
1890 Who is the author of the poem "The Midnight Ride of [sepia-person: Paul] [sepia-location: Revere]?"

same 202 Where is [sepia-location: Belize] located? Where is [sepia-location: Be
lize] located?
same 204 What type of bridge is the [sepia-location: Golden Gate Bridge]? What type of bridge is the [
sepia-location: Golden Gate Bridge]?
same 205 What is the population of the [sepia-location: Bahamas]? What is the population of th
e [sepia-location: Bahamas]?
same 208 What state has the most [sepia-organization: Indians]? What state has the most [sep
ia-organization: Indians]?
same 215 Who is the leader of [sepia-location: India]? Who is the leader of [sepia-
location: India]?
same 216 What is the primary language of the [sepia-location: Philippines]? What is the primary language
of the [sepia-location: Philippines]?
same 219 What is the population of [sepia-location: Japan]? What is the population of [s
epia-location: Japan]?
same 223 Where's [sepia-location: Montenegro]? Where's [sepia-location: Mon
tenegro]?
same 242 What was the name of the famous battle in 1836 between [sepia-location: Texas] and [sepia-location: Mexico]? What
was the name of the famous battle in 1836 between [sepia-location: Texas] and [sepia-location: Mexico]?
same 245 Where can you find the [sepia-location: Venus] flytrap? Where can you find the [sepi
a-location: Venus] flytrap?
same 250 Where did the [sepia-person: Maya] people live? Where did the [sepia-person:
Maya] people live?
same 251 How many people live in [sepia-location: Chile]? How many people live in [sep
ia-location: Chile]?
same 254 What is [sepia-location: California]'s state bird? What is [sepia-location: Cal
ifornia]'s state bird?
same 262 What is the name of the longest ruling dynasty of [sepia-location: Japan]? What is the name of the long
est ruling dynasty of [sepia-location: Japan]?
same 268 Who killed [sepia-person: Caesar]? Who killed [sepia-person: Ca
esar]?
same 273 Who was the first [sepia-location: U.S.] president ever to resign? Who was the first [sepia-loc
ation: U.S.] president ever to resign?
same 276 How much money does the Sultan of [sepia-location: Brunei] have? How much money does the Sult
an of [sepia-location: Brunei] have?
same 277 How large is [sepia-location: Missouri]'s population? How large is [sepia-location
: Missouri]'s population?
same 286 What is the nickname of [sepia-location: Pennsylvania]? What is the nickname of [sep
ia-location: Pennsylvania]?
same 292 How big is [sepia-location: Australia]? How big is [sepia-location:
Australia]?
same 293 Who found [sepia-location: Hawaii]? Who found [sepia-location: H
awaii]?
same 298 What is [sepia-location: California]'s state tree? What is [sepia-location: Cal
ifornia]'s state tree?
same 301 Who was the first coach of the [sepia-organization: Cleveland Browns]? Who was the first coach of t
he [sepia-organization: Cleveland Browns]?
same 302 How many people die from snakebite poisoning in the [sepia-location: U.S.] per year? How many people die from sna
kebite poisoning in the [sepia-location: U.S.] per year?
same 310 Where is the bridge over the [sepia-location: river Kwai]? Where is the bridge over the
[sepia-location: river Kwai]?
same 311 How many Superbowls have the [sepia-organization: 49ers] won? How many Superbowls have the
[sepia-organization: 49ers] won?
same 314 What is [sepia-person: Alice Cooper]'s real name? What is [sepia-person: Alice
Cooper]'s real name?
same 317 Where is [sepia-location: Guam]? Where is [sepia-location: Gu
am]?
same 320 Where is [sepia-location: Romania] located? Where is [sepia-location: Ro
mania] located?
same 322 Who was the first king of [sepia-location: England]? Who was the first king of [s
epia-location: England]?
same 324 What is [sepia-location: California]'s capital? What is [sepia-location: Cal
ifornia]'s capital?
same 325 What is the size of [sepia-location: Argentina]? What is the size of [sepia-l
ocation: Argentina]?
same 329 What is the population of [sepia-location: Mexico]? What is the population of [s
epia-location: Mexico]?
same 332 How long would it take to get from [sepia-location: Earth] to [sepia-location: Mars]? How long would it take to ge
t from [sepia-location: Earth] to [sepia-location: Mars]?
same 345 What is the population of [sepia-location: Kansas]? What is the population of [s
epia-location: Kansas]?
same 359 Where is [sepia-location: Melbourne]? Where is [sepia-location: Me
lbourne]?
same 363 What is the capital of [sepia-location: Haiti]? What is the capital of [sepi
a-location: Haiti]?
same 364 How many people lived in [sepia-location: Nebraska] in the mid 1980s? How many people lived in [se
pia-location: Nebraska] in the mid 1980s?
same 365 What is the population of [sepia-location: Mozambique]? What is the population of [s
epia-location: Mozambique]?
same 368 Where is [sepia-location: Trinidad]? Where is [sepia-location: Tr
inidad]?
same 377 At what speed does the [sepia-location: Earth] revolve around the sun? At what speed does the [sepi
a-location: Earth] revolve around the sun?
same 378 Who is the emperor of [sepia-location: Japan]? Who is the emperor of [sepia
-location: Japan]?
same 380 What language is mostly spoken in [sepia-location: Brazil]? What language is mostly spok
en in [sepia-location: Brazil]?
same 387 What year did [sepia-location: Montana] become a state? What year did [sepia-locatio
n: Montana] become a state?
same 388 What were the names of the three ships used by [sepia-location: Columbus]? What were the names of the t
hree ships used by [sepia-location: Columbus]?
same 390 Where was [sepia-person: John Adams] born? Where was [sepia-person: Joh
n Adams] born?
same 391 Who painted [sepia-location: Olympia]? Who painted [sepia-location:
Olympia]?
same 397 When was the [sepia-location: Brandenburg Gate] in [sepia-location: Berlin] built? When was the [sepia-location
: Brandenburg Gate] in [sepia-location: Berlin] built?
same 404 What is the state nickname of [sepia-location: Mississippi]? What is the state nickname o
f [sepia-location: Mississippi]?
same 407 What is [sepia-location: Black Hills, South Dakota] most famous for? What is [sepia-location: Bla
ck Hills, South Dakota] most famous for?
same 413 Where are the [sepia-location: U.S.] headquarters for Procter and Gamble? Where are the [sepia-locatio
n: U.S.] headquarters for Procter and Gamble?
same 415 What does [sepia-organization: CNN] stand for? What does [sepia-organizatio
n: CNN] stand for?

same 895 What county is [sepia-location: Modesto, California] in? What county is [sepia-locati
on: Modesto, California] in?
same 898 When did [sepia-location: Hawaii] become a state? When did [sepia-location: Ha
waii] become a state?
same 901 What is [sepia-location: Australia]'s national flower? What is [sepia-location: Aus
tralia]'s national flower?
same 908 When did [sepia-location: Idaho] become a state? When did [sepia-location: Id
aho] become a state?
same 922 Where is [sepia-person: John Wayne] airport? Where is [sepia-person: John
Wayne] airport?
same 923 What hemisphere is the [sepia-location: Philippines] in? What hemisphere is the [sepi
a-location: Philippines] in?
same 952 Where is [sepia-location: Milan]? Where is [sepia-location: Mi
lan]?
same 956 What river flows between [sepia-location: Fargo, North Dakota] and [sepia-location: Moorhead, Minnesota]? What
river flows between [sepia-location: Fargo, North Dakota] and [sepia-location: Moorhead, Minnesota]?
same 963 What strait separates [sepia-location: North America] from [sepia-location: Asia]? What strait separates [sepia
-location: North America] from [sepia-location: Asia]?
same 964 What is the population of [sepia-location: Seattle]? What is the population of [s
epia-location: Seattle]?
same 968 Where is the Mall of the [sepia-location: America]? Where is the Mall of the [se
pia-location: America]?
same 970 What type of currency is used in [sepia-location: Australia]? What type of currency is use
d in [sepia-location: Australia]?
same 973 Who was the first governor of [sepia-location: Alaska]? Who was the first governor o
f [sepia-location: Alaska]?
same 976 Who was elected president of [sepia-location: South Africa] in 1994? Who was elected president of
[sepia-location: South Africa] in 1994?
same 977 What is the population of [sepia-location: China]? What is the population of [s
epia-location: China]?
same 983 What country did [sepia-person: Ponce de Leon] come from? What country did [sepia-pers
on: Ponce de Leon] come from?
same 988 What is the oldest university in the [sepia-location: US]? What is the oldest universit
y in the [sepia-location: US]?
same 990 [sepia-location: Mercury], what year was it discovered? [sepia-location: Mercury], w
hat year was it discovered?
same 1001 What is the [sepia-location: Ohio] state bird? What is the [sepia-location:
Ohio] state bird?
same 1006 What is the length of the coastline of the state of [sepia-location: Alaska]? What is the length of the co
astline of the state of [sepia-location: Alaska]?
same 1007 What is the name of [sepia-person: Neil Armstrong]'s wife? What is the name of [sepia-p
erson: Neil Armstrong]'s wife?
same 1008 What is [sepia-location: Hawaii]'s state flower? What is [sepia-location: Haw
aii]'s state flower?
same 1010 When did the [sepia-location: Hindenberg] crash? When did the [sepia-location
: Hindenberg] crash?
same 1013 Where is [sepia-location: Perth]? Where is [sepia-location: Pe
rth]?
same 1017 Who discovered [sepia-location: America]? Who discovered [sepia-locati
on: America]?
same 1025 Who was the abolitionist who led the raid on [sepia-person: Harper]'s Ferry in 1859? Who was the abolitionist who
led the raid on [sepia-person: Harper]'s Ferry in 1859?
same 1032 What position did [sepia-person: Willie Davis] play in baseball? What position did [sepia-per
son: Willie Davis] play in baseball?
same 1035 What is the name of [sepia-person: Roy Roger]'s dog? What is the name of [sepia-p
erson: Roy Roger]'s dog?
same 1040 Who is a [sepia-person: German] philosopher? Who is a [sepia-person: Germ
an] philosopher?
same 1047 When was [sepia-location: Algeria] colonized? When was [sepia-location: Al
geria] colonized?
same 1049 What continent is [sepia-location: Egypt] on? What continent is [sepia-loc
ation: Egypt] on?
same 1059 What peninsula is [sepia-location: Spain] part of? What peninsula is [sepia-loc
ation: Spain] part of?
same 1060 When was [sepia-person: Lyndon B. Johnson] born? When was [sepia-person: Lynd
on B. Johnson] born?
same 1073 How far is [sepia-location: Pluto] from the sun? How far is [sepia-location:
Pluto] from the sun?
same 1082 Where did [sepia-person: Howard Hughes] die? Where did [sepia-person: How
ard Hughes] die?
same 1090 What currency is used in [sepia-location: Algeria]? What currency is used in [se
pia-location: Algeria]?
same 1106 What province is [sepia-location: Montreal] in? What province is [sepia-loca
tion: Montreal] in?
same 1109 What is the most frequently spoken language in the [sepia-location: Netherlands]? What is the most frequently
spoken language in the [sepia-location: Netherlands]?
same 1115 What year was the [sepia-person: Mona Lisa] painted? What year was the [sepia-per
son: Mona Lisa] painted?
same 1117 What is the electrical output in [sepia-location: Madrid], [sepia-location: Spain]? What is the electrical outpu
t in [sepia-location: Madrid], [sepia-location: Spain]?
same 1118 [sepia-location: Which mountain] range in [sepia-location: North America] stretches from [sepia-location: Maine] to
[sepia-location: Georgia]? [sepia-location: Which mountain] range in [sepia-location: North America] stretches from [sepia-loca
tion: Maine] to [sepia-location: Georgia]?
same 1120 What is the population of [sepia-location: Nigeria]? What is the population of [s
epia-location: Nigeria]?
same 1124 What year did the [sepia-location: U.S.] buy [sepia-location: Alaska]? What year did the [sepia-loc
ation: U.S.] buy [sepia-location: Alaska]?
same 1125 What is the name of the leader of [sepia-location: Ireland]? What is the name of the lead
er of [sepia-location: Ireland]?
same 1128 What are the two houses of the [sepia-location: Legislative branch]? What are the two houses of t
he [sepia-location: Legislative branch]?
same 1130 In [sepia-location: Poland], where do most people live? In [sepia-location: Poland],
where do most people live?
same 1150 What is the depth of the [sepia-location: Nile river]? What is the depth of the [se
pia-location: Nile river]?
same 1156 How many Admirals are there in the [sepia-organization: U.S. Navy]? How many Admirals are there
in the [sepia-organization: U.S. Navy]?
same 1157 What instrument did [sepia-person: Glenn Miller] play? What instrument did [sepia-p
erson: Glenn Miller] play?
same 1163 How did [sepia-person: Janice Joplin] die? How did [sepia-person: Janic
e Joplin] die?
same 1164 What is the primary language in [sepia-location: Iceland]? What is the primary language
in [sepia-location: Iceland]?
same 1171 What year did [sepia-location: Oklahoma] become a state? What year did [sepia-locatio

misleading comma

Wilt Chamberlain] score 100 points?
same 1412 Who is the governor of [sepia-location: Colorado]? Who is the governor of [sepi
a-location: Colorado]?
same 1414 What was the length of the [sepia-person: Wright] brothers' first flight? What was the length of the [
sepia-person: Wright] brothers' first flight?
same 1416 When was [sepia-person: Wendy]'s founded? When was [sepia-person: Wend
y]'s founded?
same 1418 When was the [sepia-location: Rosenberg] trial? When was the [sepia-location
: Rosenberg] trial?
same 1419 What year did [sepia-location: Alaska] become a state? What year did [sepia-locatio
n: Alaska] become a state?
same 1421 When did [sepia-person: Mike Tyson] bite [sepia-location: Holyfield]'s ear? When did [sepia-person: Mike
Tyson] bite [sepia-location: Holyfield]'s ear?
same 1423 What is a peninsula in the [sepia-location: Philippines]? What is a peninsula in the [
sepia-location: Philippines]?
same 1424 Who won the [sepia-person: Oscar] for best actor in 1970? Who won the [sepia-person: O
scar] for best actor in 1970?
same 1425 What is the population of [sepia-location: Maryland]? What is the population of [s
epia-location: Maryland]?
same 1426 Who is the governor of [sepia-location: Tennessee]? Who is the governor of [sepi
a-location: Tennessee]?
same 1429 What was [sepia-person: Andrew Jackson]'s wife's name? What was [sepia-person: Andr
ew Jackson]'s wife's name?
same 1444 What female leader succeeded [sepia-person: Ferdinand Marcos] as president of the [sepia-location: Philippines]? What female leader succeeded [sepia-person: Ferdinand Marcos] as president of the [sepia-location: Philippines]?
same 1450 Which [sepia-location: U.S.] state is the leading corn producer? Which [sepia-location: U.S.]
state is the leading corn producer?
same 1453 Where was the first J.C. [sepia-person: Penney] store opened? Where was the first J.C. [se
pia-person: Penney] store opened?
same 1454 How much money does the [sepia-location: U.S.] supreme court make? How much money does the [sep
ia-location: U.S.] supreme court make?
same 1455 The [sepia-location: Hindenburg] disaster took place in 1937 in which [sepia-location: New Jersey town]? The [sepia-location: Hindenburg] disaster took place in 1937 in which [sepia-location: New Jersey town]?
same 1457 Who succeeded [sepia-person: Ferdinand Marcos]? Who succeeded [sepia-person:
Ferdinand Marcos]?
same 1465 What company makes [sepia-person: Bentley] cars? What company makes [sepia-pe
rson: Bentley] cars?
same 1473 When was [sepia-person: Lyndon B. Johnson] born? When was [sepia-person: Lynd
on B. Johnson] born?
same 1476 Who was the [sepia-person: Roman] god of the sea? Who was the [sepia-person: R
oman] god of the sea?
same 1480 What is the principle port in [sepia-location: Ecuador]? What is the principle port i
n [sepia-location: Ecuador]?
same 1481 What is the capital city of [sepia-location: Algeria]? What is the capital city of
[sepia-location: Algeria]?
same 1482 What county is [sepia-location: Wilmington, Delaware] in? What county is [sepia-locati
on: Wilmington, Delaware] in?
same 1486 Where did [sepia-person: Roger Williams], pianist, grow up? Where did [sepia-person: Rog
er Williams], pianist, grow up?
same 1489 What continent is [sepia-location: India] on? What continent is [sepia-loc
ation: India] on?
same 1491 What was the name of [sepia-person: Sherlock Holmes]' brother? What was the name of [sepia-
person: Sherlock Holmes]' brother?
same 1492 How old was [sepia-person: Nolan Ryan] when he retired? How old was [sepia-person: N
olan Ryan] when he retired?
same 1495 How did [sepia-person: Molly Shannon]'s mother die? How did [sepia-person: Molly
Shannon]'s mother die?
same 1496 What country is [sepia-location: Berlin] in? What country is [sepia-locat
ion: Berlin] in?
same 1498 What school did [sepia-person: Emmitt Smith] go to? What school did [sepia-perso
n: Emmitt Smith] go to?
same 1501 How much of [sepia-location: U.S.] power is from nuclear energy? How much of [sepia-location:
U.S.] power is from nuclear energy?
same 1502 What year was [sepia-person: President Kennedy] killed? What year was [sepia-person:
President Kennedy] killed?
same 1505 What is the currency used in [sepia-location: China]? What is the currency used in
[sepia-location: China]?
same 1508 What was [sepia-person: Dale Evans]' horse's name? What was [sepia-person: Dale
Evans]' horse's name?
same 1513 What is the current population in [sepia-location: Bombay], [sepia-location: India]? What is the current populati
on in [sepia-location: Bombay], [sepia-location: India]?
same 1514 What is [sepia-location: Canada]'s most populous city? What is [sepia-location: Can
ada]'s most populous city?
same 1517 What is the state bird of [sepia-location: Alaska]? What is the state bird of [s
epia-location: Alaska]?
same 1524 What is the name of the ballpark that the [sepia-organization: Milwaukee Brewers] play at? What is the name of
the ballpark that the [sepia-organization: Milwaukee Brewers] play at?
same 1525 What university did [sepia-person: Thomas Jefferson] found? What university did [sepia-p
erson: Thomas Jefferson] found?
same 1530 What is the capital city of [sepia-location: New Zealand]? What is the capital city of
[sepia-location: New Zealand]?
same 1532 What is the literacy rate in [sepia-location: Cuba]? What is the literacy rate in
[sepia-location: Cuba]?
same 1536 What city is [sepia-location: Lake Washington] by? What city is [sepia-location
: Lake Washington] by?
same 1537 How many electoral college votes in [sepia-location: Tennessee]? How many electoral college v
otes in [sepia-location: Tennessee]?
same 1540 What is the deepest lake in [sepia-location: America]? What is the deepest lake in
[sepia-location: America]?
same 1550 What is the southwestern-most tip of [sepia-location: England]? What is the southwestern-mos
t tip of [sepia-location: England]?
same 1555 When was the Tet offensive in [sepia-location: Vietnam]? When was the Tet offensive i
n [sepia-location: Vietnam]?
same 1558 How much does it cost to register a car in [sepia-location: New Hampshire]? How much does it cost to reg
ister a car in [sepia-location: New Hampshire]?
same 1564 When did Led Zeppelin appear on [sepia-organization: BBC]? When did Led Zeppelin appear
on [sepia-organization: BBC]?
same 1565 What is [sepia-person: Karl Malone]'s nickname? What is [sepia-person: Karl
Malone]'s nickname?
same 1568 How old was [sepia-person: George Washington] when he died? How old was [sepia-person: G
eorge Washington] when he died?
same 1570 What is the legal age to vote in [sepia-location: Argentina]? What is the legal age to vot
e in [sepia-location: Argentina]?
same 1572 For whom was the state of [sepia-location: Pennsylvania] named? For whom was the state of [s

uffalo, [sepia-location: [sepia-location: New York]] to Syracuse, New York?
same 1796 What year did [sepia-person: General Montgomery] lead the Allies to a victory over the Axis troops in [sepia-locatio
n: North Africa]? What year did [sepia-person: General Montgomery] lead the Allies to a victory over the Axis troops in [sepia
-location: North Africa]?
same 1798 On what continent is [sepia-location: Egypt] located? On what continent is [sepia-
location: Egypt] located?
same 1803 When did [sepia-person: Willis Haviland Carrier] make the air conditioner? When did [sepia-person: Will
is Haviland Carrier] make the air conditioner?
same 1804 [sepia-location Which river] runs through [sepia-location: Dublin]? [sepia-location: Which river
] runs through [sepia-location: Dublin]?
same 1805 Who was elected President of [sepia-location: South Africa] in 1994? Who was elected President of
[sepia-location: South Africa] in 1994?
same 1812 What was the name of the stage play that [sepia-person: A. Lincoln] died at? What was the name of the sta
ge play that [sepia-person: A. Lincoln] died at?
same 1815 What is [sepia-location: Nicaragua]'s main industry? What is [sepia-location: Nic
aragua]'s main industry?
same 1819 When did [sepia-person: Marilyn Monroe] commit suicide? When did [sepia-person: Mari
lyn Monroe] commit suicide?
same 1820 When is [sepia-location: Mexico]'s independence? When is [sepia-location: Mex
ico]'s independence?
same 1823 What number did [sepia-person: Michael Jordan] wear? What number did [sepia-perso
n: Michael Jordan] wear?
same 1828 What was [sepia-location: Thailand]'s original name? What was [sepia-location: Th
ailand]'s original name?
same 1833 How tall is [sepia-person: Mike Tyson]? How tall is [sepia-person: M
ike Tyson]?
same 1834 Which disciple received 30 pieces of silver for betraying [sepia-person: Jesus]? Which disciple received 30 p
ieces of silver for betraying [sepia-person: Jesus]?
same 1835 Who was [sepia-person: Sherlock Holmes]' arch enemy? Who was [sepia-person: Sherl
ock Holmes]' arch enemy?
same 1836 What river runs through [sepia-location: Rome], [sepia-location: Italy]? What river runs through [sep
ia-location: Rome], [sepia-location: Italy]?
same 1838 What was the name of [sepia-person: FDR]'s dog? What was the name of [sepia-
person: FDR]'s dog?
same 1840 What is the state song of [sepia-location: Kansas]? What is the state song of [s
epia-location: Kansas]?
same 1845 What province is [sepia-location: Calgary] located in? What province is [sepia-loca
tion: Calgary] located in?
same 1847 What is [sepia-person: Tina Turner]'s real name? What is [sepia-person: Tina
Turner]'s real name?
same 1848 What was the name of the plane that dropped the Atomic Bomb on [sepia-location: Hiroshima]? What was the name of
the plane that dropped the Atomic Bomb on [sepia-location: Hiroshima]?
same 1849 What is the nickname of [sepia-location: Oklahoma]? What is the nickname of [sep
ia-location: Oklahoma]?
same 1850 What is [sepia-person: Marilyn Monroe]'s real name? What is [sepia-person: Maril
yn Monroe]'s real name?
same 1851 Which country colonized [sepia-location: Hong Kong]? Which country colonized [sep
ia-location: Hong Kong]?
same 1852 When did [sepia-person: Henry VIII] rule [sepia-location: England]? When did [sepia-person: Henr
y VIII] rule [sepia-location: England]?
same 1859 What branch of the military has its academy in [sepia-location: Annapolis]? What branch of the military
has its academy in [sepia-location: Annapolis]?
same 1862 What is the GDP of [sepia-location: China]? What is the GDP of [sepia-lo
cation: China]?
same 1864 What was the name of the first child of English parents to be born in [sepia-location: America]? What was the
name of the first child of English parents to be born in [sepia-location: America]?
same 1865 What is the major crop grown in [sepia-location: Arizona]? What is the major crop grown
in [sepia-location: Arizona]?
same 1867 Which [sepia-location: Italian city] is home to the [sepia-organization: Cathedral of Santa Maria] del Fiore or the
Duomo? Which [sepia-location: Italian city] is home to the [sepia-organization: Cathedral of Santa Maria] del Fiore or the Duomo?
same 1869 In the Bible, who was [sepia-person: Jacob]'s mother? In the Bible, who was [sepia
-person: Jacob]'s mother?
same 1870 What country is [sepia-location: Pretoria] in? What country is [sepia-locat
ion: Pretoria] in?
same 1871 How much gravity exists on [sepia-location: Mars]? How much gravity exists on [
sepia-location: Mars]?
same 1873 What is the motto for [sepia-location: California]? What is the motto for [sepia
-location: California]?
same 1875 What county is [sepia-location: St. Paul, Minnesota] in? What county is [sepia-locati
on: St. Paul, Minnesota] in?
same 1879 By what nickname was musician [sepia-person: Ernesto Antonio Puente, Jr.] best known? By what nickname was musicia
n [sepia-person: Ernesto Antonio Puente, Jr.] best known?
same 1880 When was [sepia-person: King Louis XIV] born? When was [sepia-person: King
Louis XIV] born?
same 1885 What language do they speak in [sepia-location: New Caledonia]? What language do they speak
in [sepia-location: New Caledonia]?
same 1886 What are [sepia-location: Brazil]'s national colors? What are [sepia-location: Br
azil]'s national colors?
same 1893 What American general is buried in [sepia-location: Salzburg]? What American general is bur
ied in [sepia-location: Salzburg]?

804 lines
382 entities found
38 red marks

miscat: 10
no comb: 6
missed: 11
other: 7

's mom 2
X town 4

Bibliography

[1] Ken Anderson, Tim Hickey, and Peter Norvig. The jscheme web programming project http://jscheme.sourceforge.net/jscheme/mainwebpage.html.

[2] Douglas Appelt, Jerry Hobbs, John Bear, David Israel, Megumi Kameyama, Andy Kehler, David Martin, Karen Myers, and Mabry Tyson. SRI international FASTUS system MUC-6 test results and analysis. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, 1995.

[3] Douglas E. Appelt and David Israel. Introduction to information extraction technology. In *IJCAI-99 Tutorial*, 1999.

[4] Jason Baldridge. Strong equivalence of CCG and Set-CCG. In *Manuscript, Division of Informatics, University of Edinburgh*, 2000.

[5] Jason Baldridge and Gann Bierner. The GROK homepage: http://grok.sourceforge.net/, 2003.

[6] Jason Baldridge, John Dowding, and Susana Early. Leo: an architecture for sharing resources for unification-based grammars. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2002.

[7] Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. In *Proceedings of ANLP-97*, pages 194–201, 1997.

[8] Daniel M. Bikel, Richard L. Schwartz, and Ralph M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231, 1999.

[9] William J. Black, Fabio Rinaldi, and David Mowatt. Facile: Description of the ne system used for muc-7. In *Proceedings of MUC-7*, 1998.

[10] Keith Bonawitz, Anthony Kim, and Seth Tardiff. An architecture for word learning using bidirectional multimodal structural alignment. In *Proceedings of the HLT-NAACL '03 Workshop on Learning Word Meaning from Non-Linguistic Data*, 2003.

[11] Andrew Brothwick, John Sterling, Eugene Agichtein, and Ralph Grishman. NYU: Description of the MENE named entity system as used in MUC-7. In *Proceedings of MUC-7*, 1998.

[12] Hai Leong Chieu. Closing the gap: Learning-based information extraction rivaling knowledge-engineering methods. In *Proceedings of the 43rd Meeting of the Association for Computational Linguistics, Sapporo*, 2003.

[13] Nancy Chinchor and Elaine Marsh. Muc-7 information extraction task definition (version 5.1). In *Proceedings of MUC-7*, 1998.

[14] Nancy Chinchor and P. Robinson. Muc-7 named entity task definition (version 3.5). In *Proceedings of MUC-7*, 1998.

[15] H. Cunningham, D. Maynard, V. Tablan, C. Ursu, and K. Bontcheva. Developing language processing components with GATE, 2001.

[16] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: an architecture for development of robust HLT applications.

[17] Cycorp and OpenCyc.org. Opencyc: the open source version of the cyc technology, the world's largest and most complete general knowledge base and commonsense reasoning engine.

[18] Terry Smith (director). The alexandria digital library project http://www.alexandria.ucsb.edu/.

[19] C. Doran, B. A. Hockey, A. Sarkar, B. Srinivas, and F. Xei. Evolution of the XTAG system, 2002.

[20] Bonnie Dorr. The use of lexical semantics in interlingual machine translation. *Machine Translation*, 7(3):135–193, 1992.

[21] Bonnie Dorr. English verb LCS lexicon: http://www.umiacs.umd.edu/ bonnie/verbs-English.lcs, 2001.

[22] Bonnie Dorr, Gina Anne Levow, and Dekang Lin. Construction of a Chinese-English verb lexicon for machine translation. *Machine Translation, Special Issue on Embedded MT*, 17(3):1–2, 2002.

[23] John Dowding, Jean Mark Gawron, Douglas E. Appelt, John Bear, Lynn Cherny, Robert Moore, and Douglas B. Moran. GEMINI: A natural language system for spoken-language understanding. In *Meeting of the Association for Computational Linguistics*, pages 54–61, 1993.

[24] Jay Earley. An efficient context-free parsing algorithm. In *Communications of the ACM, 26(1)*, pages 57–61, 1970.

[25] Jason Eisner. Efficient normal-form parsing for combinatory categorial grammar. In *Proceedings of 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz*, 1996.

[26] Jerome Feldman. The FRAMENET home page: http://www.icsi.berkeley.edu/ framenet/.

[27] David Fisher, Stephen Soderland, Joseph McCarthy, Fangfang Feng, and Wendy Lehnert. Description of the UMASS CIRCUS system as used in MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, 1995.

[28] Roberto Garigliano, Agnieszka Urbanowicz, and David J. Nettleton. University of Durham: Description of the LOLITA system as used in MUC-7. In *Proceedings of MUC-7*, 1998.

[29] Edward Gibson. The dependency locality theory: A distance-based theory of linguistic complexity. In *Miyashita, Y., Marantz, A., O'Neil, W. (Eds.), Image, language, brain. MIT Press. Cambridge, MA.*, pages 95–126, 2000.

[30] Daniel Gildea. Automatic labeling of semantic roles. In *Proceedings of the 38th annual meeting of the Association for Computational Linguistics*, 2000.

[31] Diana Maynard Hamish. Adapting a robust multi-genre ne system for automatic content extraction. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, 2002.

[32] Brian Hammond, Amit Sheth, and Krzysztof Kochut. Semantic enhancement engine: A modular document enhancement platform for semantic applications over heterogeneous content, 2002.

[33] Donna Harman and Nancy Chinchor. *Proceedings of the Seventh Message Understanding Conference (MUC-7).* Published at http://www.itl.nist.gov/iaui/894.02/related_projects/muc/, 1998.

[34] Lynette Hirschman, Marc Light, Eric Breck, and John D. Burger. Deep Read: A reading comprehension system. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, 1999.

[35] Julia Hockenmaier, Gann Bierner, and Jason Baldridge. Extending the coverage of a CCG system. In *Proceedings of the 40th Anniversary meeting of the Association for Computational Linguistics*, 2002.

[36] Julia Hockenmaier and Mark Steedman. Generative models for statistical parsing with combinatory categorial grammar. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 335–342, 2002.

[37] K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. University of Sheffield: Description of the LaSIE-II system as used for MUC-7. In *Proceedings of MUC-7*, 1998.

[38] H. Jerry, R. Douglas, E. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson. FASTUS: A cascaded finite-state transducer for extracting information from natural-language text, 1996.

[39] Nikiforos Karamanis. Ordered set combinatory categorial grammar. In *Manuscript. Institute for Communicating and Collaborative Systems, Edinburgh. Also appears shortened in Proceedings of the 15th International Symposium on Theoretical and Applied Linguistics*, 2000.

[40] Boris Katz, Sue Felshin. Deniz Yuret, Ali Ibrahim. Jimmy Lin, Gregory Marton. Alton Jerome McFarland, and Baris Temelkuran. Omnibase: Uniform access to heterogeneous data for question answering. In *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB 2002)*, 2002.

[41] Boris Katz, Jimmy Lin, Chris Stauffer, and Eric Grimson. Answering questions about moving objects in surveillance videos. In *Proceedings of the 2003 AAAI Spring Symposium on New Directions in Question Answering*, 2003.

[42] Boris Katz, Deniz Yuret, Jimmy Lin, Sue Felshin, Rebecca Schulman, Adnan Ilik, Ali Ibrahim, and Philip Osafo-Kwaako. Blitz: A preprocessor for detecting context-independent linguistic structures. In *Proceedings of the 5th Pacific Rim Conference on Artificial Intelligence (PRICAI '98)*, 1998.

[43] Paul Kingsbury. Adding semantic annotation to the Penn TreeBank. In *Human Language Technologies Conference*, 2002.

[44] George R. Krupka and Kevin Hausman. IsoQuest, inc.: Description of the NetOwl extractor system as used for MUC-7. In *Proceedings of MUC-7*, 1998.

[45] Douglas B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[46] Dekang Lin. Using collocation statistics in information extraction. In *Proceedings of MUC-7*, 1998.

[47] Dekang Lin. Minipar—a minimalist parser. In *Maryland Linguistics Colloquium*, University of Maryland, College Park, March 12, 1999.

[48] Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kamhatla. tRuEcasIng. In *Proceedings of the annual Meeting of the Association for Computational Linguistics. Sapporo*, 2003.

[49] Edward D. Loper and Steven Bird. The natural language toolkit http://nltk.sourceforge.net/.

[50] Andrei Mikheev, Claire Grover, and Marc Moens. Description of the LTG system used for MUC-7. In *Proceedings of MUC-7*, 1998.

[51] David Miller, Sean Boisen, Richard Schwartz, Rebecca Stone, and Ralph Weischedel. Named entity extraction from noisy input: Speech and OCR. In *6th Applied Natural Language Processing Conference*, 2000.

[52] Scott Miller, Micahel Crystal, Heidi Fox, Lance Ramshaw, Richard Schwartz, Rebecca Stone, Ralph Weischedel, and the Annotation Group. Algorithms that learn to extract information—BBN: Description of the SIFT system as used for MUC-7. In *Proceedings of MUC-7*, 1998.

[53] Dan Moldovan, Marius Pasca, Sanda Harabagiu, and Mihai Surdeanu. Performance issues and error analysis in an open-domain question answering system. In *Proceedings of ACL-2002*, pages 33–40, 2002.

[54] Tom O'Hara, Roger Hartley, and Janyce Wiebe. Mapping corpus-based semantic role annotations from TreeBank and FrameNet to CG and Cyc. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, 2003.

[55] Fernando Pereira and D.H.D. Warren. Definite clauses for language analysis. *Artificial Intelligence*, 13:231–278, 1980.

[56] Colin Phillips. Linear order and constituency. *Linguistic Inquiry, to appear.*, 2003.

[57] Mark Przybocki. The ACE homepage, 2002.

[58] R. Srihari and W. Li. Information extraction supported question answering, 1999.

[59] Mark Steedman. *The Syntactic Process.* MIT Press, 2000.

[60] David G. Stork. The open mind initiative http://www.openmind.org/, 1999.

[61] Cynthia A. Thompson. Corpus-based lexical acquisition for semantic parsing.

[62] Ellen M. Voorhees. Overview of the TREC 2001 question answering track. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, 2001.

[63] Ellen M. Voorhees and Dawn M. Tice. Overview of the TREC-9 question answering track. In *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*, 2000.

[64] Richard Waldinger, Douglas Appelt, John Fry, David J. Israel, Peter Jarvis, David Martin, Susanne Riehemann, Mark E. Stickel, Mabry Tyson, Jerry Hobbs, and Jennifer L. Dungan. Deductive question answering from multiple resources. In *New Directions in Question Answering (submitted)*, 2003.

[65] Yorick Wilks. Ir and ai: traditions of representation and anti-representation in information processing.

[66] Patrick Henry Winston, Jake Beal, Keith Bonawitz, and Seth Tardiff. The bridge project http://www.ai.mit.edu/projects/genesis/, 2003.

[67] Shihong Yu, Shuanhu Bai, and Paul Wu. Description of the Kent Ridge Digital Labs system used for MUC-7. In *Proceedings of MUC-7*, 1998.