The Dissertation Committee for Erik John Larson certifies that this is the approved version of the following dissertation:

Primary Semantic Type Labeling in Monologue Discourse Using a Hierarchical Classification Approach

Committee:

_____
Robert C. Koons, Supervisor

_____
Nicholas M. Asher

_____
Daniel A. Bonevac

_____
Cory F. Juhl

_____
Bruce W. Porter

Primary Semantic Type Labeling in Monologue Discourse Using a Hierarchical Classification Approach

by

Erik John Larson, B.A., M.A.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin
December 2009

Primary Semantic Type Labeling in Monologue Discourse Using a Hierarchical Classification Approach

Erik John Larson, Ph.D.

The University of Texas at Austin, 2009

Supervisor: Robert C. Koons

The question of whether a machine can reproduce human intelligence is older than modern computation, but has received a great deal of attention since the first digital computers emerged decades ago. Language understanding, a hallmark of human intelligence, has been the focus of a great deal of work in Artificial Intelligence (AI). In 1950, mathematician Alan Turing proposed a kind of game, or test, to evaluate the intelligence of a machine by assessing its ability to understand written natural language. But nearly sixty years after Turing proposed his test of machine intelligence—pose questions to a machine and a person without seeing either, and try to determine which is the machine—no system has passed the Turing Test, and the question of whether a machine can understand natural language cannot yet be answered.

The present investigation is, firstly, an attempt to advance the state of the art in natural language understanding by building a machine whose input is English natural language and whose output is a set of assertions that represent answers to certain

questions posed about the content of the input. The machine we explore here, in other words, should pass a simplified version of the Turing Test and by doing so help clarify and expand on our understanding of the machine intelligence.

Toward this goal, we explore a constraint framework for partial solutions to the Turing Test, propose a problem whose solution would constitute a significant advance in natural language processing, and design and implement a system adequate for addressing the problem proposed. The fully implemented system finds *primary specific* events and their locations in monologue discourse using a hierarchical classification approach, and as such provides answers to questions of central importance in the interpretation of discourse.

Table of Contents

Chapter 1: Introduction

The Turing Test and Natural Language Understanding

The question of whether a machine can reproduce human intelligence is older than modern computation, but has received a great deal of attention since the first digital computers emerged decades ago. Language understanding, a hallmark of human intelligence, has been the focus of a great deal of work in Artificial Intelligence (AI). In 1950, mathematician Alan Turing proposed a kind of game, or test, to evaluate the intelligence of a machine by assessing its ability to understand written natural language (Turing, 1950). But nearly sixty years after Turing proposed his test of machine intelligence—pose questions to a machine and a person without seeing either, and try to determine which is the machine—no system has passed the Turing Test, and the question of whether a machine can understand natural language cannot yet be answered.[1]

[1] Turing described his famous 'imitation game' as follows:

… the problem can be described in terms of a game which we call the 'imitation game." It is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. […] In order that tones of voice may not help the interrogator the answers should be written, or better still, typewritten. The ideal arrangement is to have a teleprinter communicating between the two rooms. […] We now ask the question, "What will happen when a machine takes the part of A in this game?" Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, "Can machines think?" (from "Computing Machinery and Thought", Alan M. Turing, 1950)

The "Turing Test", as it is now called, is usually presented in a simplified form (a form also proposed by Turing), with a single human interrogator conversing with a machine. The Turing Test has

The present investigation is, firstly, an attempt to advance the state of the art in natural language understanding (or, for our purposes equivalently, natural language processing (NLP)) by building a machine whose input is English natural language and whose output is a set of assertions that represent answers to certain questions posed about the content of the input. The machine we explore here, in other words, should pass a simplified version of the Turing Test and by doing so help clarify and expand on our understanding of machine intelligence. The investigation of machine intelligence and the machine designed and described in this work draws heavily on prior work in AI, NLP, and other cognate fields. Indeed, and as one might imagine, there are now many other functional systems that could be described (as we'll see) as solving subsets of the Turing Test. The claim in this work will be that the system (machine) described solves an interesting subset, and one in which a plausible case can be made that the performance lies on the horizon of possibilities given present theory and technology. But of course this case will require a substantial clarification of how to compare different "subsets" of the full Turing Test.

So, secondly, this investigation is an attempt to *erect a framework* that matches types of NLP systems with NLP problems that are defined on types of input, such that a general taxonomy of "hardness" is possible, with the most difficult problem Turing's

---

elicited volumes of discussion about its merits as a test of machine intelligence. A few highlights: John Searle and later Ned Block questioned whether a machine passing the test would be merely *simulating* intelligence (rather than really understanding the conversation) (Searle (1980), Block (1995)). Also, some controversy has followed the adoption of the simplified "two person" version of the game, as philosophers such as S.G. Sterrett have argued that the "Original Imitation Game Test" is not equivalent to the simplified version (Sterrett, 2000). These details are not of interest to the present discussion, however.

original, still unsolved, test. The task of passing the Turing Test then becomes that of traversing the taxonomy, and it ought to be easier to see exactly where difficulties emerge. The investigation concludes with a discussion about our prospects for passing the unconstrained Turing Test—or, in other words, reproducing human language understanding with a machine, which, it is hoped, will be more precise from the benefit of the ideas and results presented here. To begin, then, we review briefly the history of NLP research, leading up to current work.

Early Work in NLP

Some of the central difficulties facing NLP have been noted almost since its inception. Working on machine translation (MT) in the 1950s, for instance, Yehoshua Bar-Hillel noted that even simple sentences such as "The box was in the pen" often had multiple interpretations that frustrated attempts to achieve high quality machine translation of natural language[2] (Bar-Hillel, 1960).

Hillel's observations in MT were about *lexical* ambiguity in natural language. His point was that the simple view of language—a sequence of words with definitions in a particular order—adopted by researchers in MT was inadequate. Some consideration needed to be given to the structure of language, in particular to the grammatical structure of natural language sentences and how this structure affects word meaning.

[2] Hillel's 1960 addendum to his report on the status of MT research concluded that 'fully automatic high quality translation' (FAHQT) was "unattainable".

Fortunately, and at roughly the same time that Hillel's insights were emerging, Noam Chomsky work on generative grammars (then known as transformation grammars), provided a systematic method (set of rules) for the analysis of language syntax, the key element missing from the earlier lexical attempts at MT (Chomsky, 1964). The syntactic framework emerging from early linguistics hence augmented, at least in theory, the lexical approach adopted by Bar-Hillel and other MT (and by extension, NLP) theorists.

However, the confluence of linguistics research and NLP in the 1950s and 60s did not lead, at the level of systems implementation, to a uniform methodology in NLP. On the one hand, generated grammars provided the basis for automating the syntactic analysis of language. Indeed several implemented systems at this time reflected inclusion of more sophisticated analysis techniques and in particular an increased emphasis on syntactic, rather than merely lexical, analysis. For instance, Lindsay's SAD-SAM system generated syntactic parses of English sentences in a limited domain, and was regarded as a successful improvement on prior approaches (it implemented a context free grammar and used a 1,700 word English lexicon) (Lindsay, 1963).

On the other hand, the difficulties in translating linguistic theory into practice (c.f., the theory of generative grammars from Chomsky's research), resulted in the adoption of a more ad hoc, results-driven approach to NLP, typified by so-called "pattern matching" systems:

- Daniel Bobrow's STUDENT solved high school algebra problems by matching preselected patterns in sentences from textbooks. The system was effective, but brittle, and couldn't parse sentences that were rewritten but expressed the same meaning (Bobrow, 1968).
- Joseph Weizenbaum's ELIZA, perhaps the best known example of simple pattern matching, played the role of a "nondirective" or "Rogerian" therapist that engaged human patients in dialogue. (It could, of course, not pass a Turing Test, but many have commented on the odd natural feel of psychotherapeutic "sessions" with the machine) [3] (Weizenbaum, 1966).

Common to both the "new linguistic" approach to NLP and the ad hoc pattern-matchers, however, was a lack of concern for issues of scale: researchers at this stage in NLP research were occupied with showing particular results, such as a syntactic parse of a sentence, achieved on a particular, limited set of examples. Successful parses on test sentences were taken (too blithely, it's obvious in retrospect) as evidence that the model

[3] As Patrick Doyle puts it: [ELIZA] operated by matching the left sides of its rules against the user's last sentence, and using the appropriate right side to generate a response. Rules were indexed by keywords so only a few had to be matched against a particular sentence. Some rules had no left side, so they could apply anywhere with replies like "Tell me more about that." Note that these rules are "approximate" matchers. This accounts for ELIZA's major strength, its ability to say something reasonable most of the time, as well as its major weakness, the superficiality of its understanding and its ability to be led completely astray.

used was effective on the analysis of language, and hence that the extension or scaling of the approach was simply a matter of detail.

True, the infusion of ideas from research in linguistics had a large impact on NLP. Yet the purely syntactic analysis of language was, as with the lexical analysis in the early MT days, again revealed as inadequate, as early as the late 1960s (Buchanan (2005), see also Dreyfus, (1992)). The lexical ambiguity Bar-Hillel pointed out was, at root, really a *knowledge* requirement—we have to know a lot about the world to interpret natural language. Neither the early lexical analysis nor the embrace of syntactic analyses of sentence structure had accounted for the knowledge requirement: the importance of knowledge as an information source for processing natural language. This realization led to an explosion of knowledge-based work in the 1970s, from work on semantic grammars, to new knowledge representation approaches and languages such as semantic networks, to more efficient inference procedures (cf. Woods, (1975), Wilks, (1975)).

One program, in particular, showed (perhaps for the first time) that some interesting version of NLP was indeed possible, but by constraining the task such that there were tractable solutions to problems arising from ambiguity and reference (anaphora). Terry Winograd's SHRDLU (1972) interacted with a human user to manipulate blocks on a table with a simulated robot arm. The system carried on a dialogue with the user about the blocks "world", made more natural by its capability to process utterances that included conversational phenomena such as anaphora (e.g.,

pronouns) and ellipsis (Winograd, 1972). The program generated surprisingly natural dialogue with a human interlocutor, and was widely considered an advance in automated language processing. For instance, a dialogue with SHRDLU in the block world might produce a narrative as follows:

Person: Pick up a big red block.
Computer: OK.
Person: Grasp the pyramid.
Computer: I don't understand which pyramid you mean.
Person (changing their mind): Find a block which is taller than the one you are holding and put it into the box.
Computer: By "it", I assume you mean the block which is taller than the one I am holding.
Computer: OK.
Person: What does the box contain?
Computer: The blue pyramid and the blue block.
Person: What is the pyramid supported by?
Computer: The box.
Person: How many blocks are not in the box?
Computer: Four of them.
Person: Is at least one of them narrower than the one which I told you to pick up?
Computer: Yes, the red cube. etc.

SHRDLU's blocks world success suggested what became known as the "microworlds"strategy for further advances in NLP: solve the NLP problem in restricted domains, learn what works, and generalize what has been learned to larger domains[4] (Winograd, 1972).

---

[4] Minsky and Papert put the strategy as follows: *"Each model - or `micro-world' as we shall call it - is very schematic; it talks about a fairyland in which things are so simplified that almost every*

The strategy informed much of the later work in NLP in the 1970s, notably the "case frame" theories exemplified by Roger Schank's work on script theory (Shanks, 1973, Shanks and Abel, 1975). Scripts simplified the NLP problem by specifying only a set of typical (stereotypical) interactions in some predetermined situation, such as ordering food at a restaurant. In such cases, the program could invoice the "restaurant" script in order to perform disambiguation of words in a query, reference resolution of anaphora, and other natural language phenomena that, without the aid of the script, seemed hopelessly difficult. As many researchers have noted, however, subsequent research has demonstrated the difficulty in generalizing such techniques successful in microworlds approaches to more realistic domains, prompting skepticism of the approach generally (Weizenbaum, 1976, see also Dreyfus, 1992).

The failure of microworlds to scale led to a serious attempt in the 1980s to "fix" the brittleness of early systems by providing them with adequate knowledge about the world. In the Japanese "5th Generation" project as well as large scale knowledge base efforts such as MCCs Cyc project, computers' lack of "commonsense" became a popular answer to why 1970s work in NLP (and other subfields of AI for that matter) did not

---

statement about them would be literally false if asserted about in the real world. [...] Nevertheless, we feel that they [the micro-worlds] are so important that we are assigning a large portion of our effort toward developing a collection of these micro-worlds and finding how to use the suggestive and predictive powers of the models without being overcome by their incompatibility with literal truth." (Internal MIT memo Minsky & Papert, 1970; quoted in Dreyfus, 1981)

result in more general abilities to understand language (Lenat and Brown, 1984). Bar-Hillel's "box is in the pen" example, as Lenat has pointed out, was not just a knowledge limitation but more specifically a lack of simple *commonsense*: we know that the "pens" that fit in boxes are typically ink pens (not animal pens), and so it is this commonsense view of the relative sizes of objects that we talk about that must be given to a machine, if it is to understand language (Lenat and Brown, 1984).

As we might guess, actually giving computers commonsense has proven difficult in turn. We'll address the difficulty of generating commonsensical solutions with automated approaches to NLP more fully in Chapter 5; here, suffice it to note that researchers in the 1980s, while perhaps emboldened by initial successes with supplying knowledge (i.e.., assertions in a computable formal language) to computational systems that facilitated inference in domains where such knowledge applied, were later disappointed with the difficulty of scaling the knowledge-based approach to apply to larger, less "toy", domains important for real-world NLP (McDermott, 1987).

Modern NLP

In contrast to early NLP systems such as Winograd's SHRDLU or large-scale knowledge representation projects such as Cyc[5], NLP work today reflects a shift in focus towards empirical or "shallow" methods to extract information from natural language (Brill and Mooney, 1997). Whereas the early research occurred within extremely

[5] Although Cyc is often thought of as a general purpose reasoning project, its inclusion with early work on NLU is apt: Lenat proclaimed early on that the point of Cyc was to enable a computer to "read the newspaper".

restricted domains (i.e., "microworlds")[6], subfields in NLP such as Information Extraction (IE) and Information Retrieval (IR) that have emerged in the last two decades are marked by a wholly different methodology. Research in these fields involves solving much simpler problems than are required by the unrestricted Turing Test, such as extracting shallow semantic information from free text (i.e., Named Entity Recognition, or NER), but in environments that are more realistic and without the artifice of the microworlds approach.

DARPA's well-known Message Understanding Conference (MUC) competitions of the 1990s, for instance, used datasets (corpora) from standard news sources like the Wall Street Journal, but evaluated the competitor systems on a set of simplified tasks, at least by historic standards (DARPA, 1993). Likewise, the Text Retrieval Conferences (TREC) of the 1990s—the IR equivalent to the IE based competitions in MUC—used newswire data from sources like AP News and the Wall Street Journal to test on a very well-defined task: document classification[7] (TREC). Such *tasks* are remarkably restricted in comparison to much early work in NLP, but they are carried out in natural

[6] The commonsense reasoning KBs developed in the 80s were often microworlds in disguise. The Cyc KB, for instance, is organized in terms of microtheories, which are, for all practical purposes, microworlds within which reasoning can be constrained. The problem with taking the conjunction of all microtheories as a domain general KB is that it leaves unanswered the question of how to perform inter-microtheory inference, which is of course what domain generality would require. There has been work on this problem (cf. the lifting axioms of McCarthy), but it is itself unsolved and very difficult problem.

[7] Given a set of class labels L (l_1,..l_n), for each document D in a corpus C, assign the correct label l_k to D in C.

language *environments* that are remarkably *unrestricted* in comparison to earlier work[8] (Brill and Mooney, 1997).

A New Framework

This agenda shift—whatever the reasons why it occurred—makes it more difficult to view the last fifty years of work in NLP under a single rubric. Specifically, the question of "progress" is much harder to answer, since the general problem of reproducing language understanding with a machine now has so many parameters: types of tasks, algorithms and architectures, and different types of input (i.e., the different datasets used during design, development, and evaluation of NLP theories and systems). Avoiding the pessimistic inference that, as Hubert Dreyfus once remarked (referencing philosopher of science Imre Lakatos), AI is an example of a "degenerating research program", so that the practical turn witnessed since the early 1990s becomes a tacit concession that automated language understanding is just *too hard*, some firmer foundation of framework for assessing and making progress seems necessary[9] (Dreyfus, 1992). Part of the goal of this work is to begin such a project.

[8] TREC included an NLP "track" in later competitions, but, notably, the role of NLP was in extracting shallow semantics from text in order to evaluate its benefit on the core retrieval task.

[9] Norvig and Russell have remarked that there has been no real, serious attempt to solve the Turing Test in AI (Russell and Norvig, 2003). This is true (although the Loebner Prize is still awarded every year for the system that is most convincing to human testers on the Turing Test), and probably suggests that indeed the test is so hard that it is pointless to try to build a system, today, that would really pass a non-simplified version of the test.

NLP Problems in the New Framework

Most work in NLP today is work on particular NLP tasks or problems: we want to produce a syntactic parse of a sentence, or disambiguate each word in a sentence, or generate a logical representation (logical form) for a sentence; the list goes on. In the framework proposed in this investigation, we note, first, that we can view such problems as determining a class of *systems*, whose details are incidental (i.e., implementation) and whose commonalities explain why systems in this class can solve the particular type of problem. In other words, for a given problem or task T, we have a class of systems C such that, a system S is in C iff S can solve T.[10]

In this vein, we view the Turing Test as a type of problem whose solution would require a very broad class of powerful language understanding systems. What would the capabilities of a "Turing system" be? At minimum:

1. Such systems would be able to properly *interpret* dialogue utterances as produced in human conversations.
2. Such systems would be able to *generate* normal (not "odd"), human conversation utterances in the context of a conversation.

[10] Leave aside, for now, obvious questions about how we know that S "solves" T, and for what type of input, etc. This is a question that we will take up in subsequent sections.

Since 1) and 2) are entirely unrestricted (or, restricted only to normal human conversation), we have:

1. Turing systems would interpret and generate natural language for a very large class of domains (e.g., baseball, the war in Iraq, global warming, marathons, mergers and acquisitions).
2. Turing systems would be robust to informal discourse, such as relaxed grammar constraints, different orthographic conventions (all CAPS input, etc.) as well as pragmatic informality such as subject ellipsis in a "chat" and so on.
3. Turing systems would have to be *monologue*-processing systems as well. [11]

Our thesis in this work is that there exists a system that we *can* build that can solve an interesting NLP problem which, to the author's knowledge, has not yet been solved. The system will (of course) not solve many problems that our "Turing systems" can solve. But we can, given the "Turing" framework briefly outlined above, locate this system with respect to ongoing work and with respect to the ultimate NLP system: the

[11] Suppose the human participant in Turing's game asked the system:

A: Please read the following sentence and tell me what happened:

> Several shots were fired north of Baghdad today, resulting in at least one fatality and three other injuries related to the shootings.

This obviously would require monologue interpretation.

unrestricted Turing system. Specifically, in this investigation we introduce the machinery to discuss types of problems in NLP more precisely, and we introduce a system S (the "TPE" system) that solves a particular kind of *discourse interpretation* problem. The interpretation problem and the system together constitute a clear advancement in NLP.

The remainder of this chapter will provide the background for making this argument. In particular, we unpack the notion of natural language interpretation, provide a discussion of SDRT, a dynamic semantics-based theory that computes a rich relational structure for interpreting discourse, introduce the notion of *discourse wide* objects such as topics, and argue that inferring such objects is outside the purview of currently dynamic semantics based theories (including SDRT), since they are underspecified, at the limit, until the entire discourse has been processed.

The purpose of all of this background is to introduce a discourse wide problem—that of computing primary semantic types, or PSTs—discussed in Chapter 3. Chapter 2 will give the motivation for this task by fleshing out the framework introduced in this chapter. Chapter 4 provides the actual inference mechanism, including discussion of the implemented TPE system. Chapter 5 provides discussion of the challenges facing us moving from the simplified Turing problem to a more general or full version of the problem.

Discourse Interpretation

Truth Conditional Theories of Interpretation

Following Frege, Montague, and many others, *interpreting* a sentence involves assigning it a set of truth conditions (Asher and Lascarides, 2003, hereafter A&L, 2003). The assignment proceeds by first constructing a logical form (LF) for an utterance and then evaluating it with respect to a model (i.e. assigning a set of individuals to the variables in the LF such that the LF is true in the model).

The simplest case—the one in which Frege and Montague were interested in—is the single sentence:

(1) A man walked in.

In (1) the LF is (simplifying syntax): $\exists x(man(x) \wedge walk\text{-}in(x))$, where some individual is a man and walks in. Supposing that "John" walks in, then the following assignment is a true interpretation of the LF:

(1′) man(John) ^ walks-in(John)

Multi-sentence discourse complicates matters, especially with the introduction of referential phenomena such as anaphora:

(2)

a. A man walked in.

b. He ordered a beer.

Here, we have two LFs:

(2a′) $\exists x(man(x) \wedge walks\text{-}in(x))$

(2b′) $\exists y(beer(y) \wedge order(z, y))$, where z is an independent free variable.

The sentence is true under the model where z = x, but it is also true in a model where z is assigned a different individual than x (hence, some man walked in, and some other man ordered a beer).

Evans has treated this apparent shortcoming of static semantics by converting anaphoric phenomena such as pronouns as "disguised definite descriptions" as Asher and Lascarides observe (A&L, 2003). Hence, in (2b) "He" really means "the man who walked in". Following Russell's theory of definite descriptions, the "definite" conversion of the pronominal in (2a) would license only interpretations where z=x. The problem, as Heim and Kamp have pointed out (see A&L, 2003), is that the uniqueness presupposition does not only hold, as witnessed in sentences such as:

(3) When one bishop meets another bishop, he always blesses him.

(It is arguable whether the definite description treatment of anaphoric phenomena can survive such objections, but we will not pursue the matter here.) We turn now to another approach to grounding discourse interpretation in logical formalism: dynamic semantics.

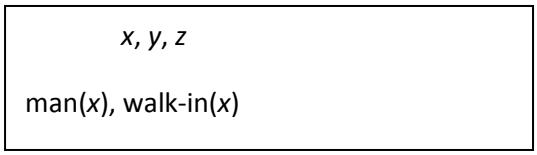
Dynamic Semantics

Dynamic semantics reformulates the interpretation problem such that each utterance in a discourse D depends for its correct interpretation on the interpretation of prior utterances in D. As such, a theory of dynamic semantics is *relational*: the interpretation of some utterance U_k depends on its relation to the interpretations of one or more utterances $U_{k\text{-}m}$, $m > 0$. The semantics of discourse is dynamic because the interpretation of an utterance is now a function of the prior utterances (A&L, 2003). The meaning of a sentence in this framework is what is known as its Context Change Potential or CCP—the potential of the context (prior utterances) to change the meaning of the sentence.

The theory of dynamic semantics is often attributed to early work on context by Kaplan (1975) and has received detailed treatments in Kamp (1981), as well as Kamp and Reyle (1993). Kamp and Reyle developed Discourse Representation Theory (DRT) on

dynamics semantics principles to provide a more adequate treatment of interpretation of anaphoric phenomena in discourse. DRT introduces the notion of an accessible antecedent to an anaphor such as a pronoun. Accessibility is defined in terms of prior context; for pronominals further conditions such as gender agreement generate the allowable set of antecedents (Kamp and Reyle, 1993). Applying the pronoun rule in DRT to the prior example, we have:

(4)



Underspecification

There are problems with the model of interpretation using DRT, however. As Asher and Lascarides point out, the assignment z = x cannot be determined by the grammar but is rather a function of the context of multi-sentence discourse *after* grammatical analysis has generated an LF (or set of LFs). In other words, the construction of the LFs in multi-sentential discourse proceeds first, then evaluation of the LFs occurs. In the case of finding the correct antecedent for a pronoun, the assignment z = x occurs after generating the set of antecedents from prior constructed LFs, using the

information sources available. DRT disguises this by resolving the antecedent during LF construction.

Reyle (1993) introduces the notion of *underspecification* to handle cases where assignment cannot be determined during LF construction. A paradigm case of underspecification is the occurrence of pronominal anaphora as in our example. Poesio puts the observation more generally, as a hypothesis about how humans must interpret discourse (to avoid generating explosively many interpretations from ambiguous expressions, see the Combinatorial Explosion Puzzle in Poesio (1994) for more discussion):

> Underspecification Hypothesis: Human beings represent semantic ambiguity implicitly by means of underspecified representations that leave some aspects of interpretation unresolved.

However broadly underspecification is construed, its effect in discourse interpretation is to cleanly separate the contribution of sentential syntax in the construction of LFs from the contextual considerations involved in inferring antecedents to anaphors (or, assignments to variables introduced by ambiguous phenomena generally). Hence, the pronoun introduces an underspecified semantic condition of the

form u = ? (rather than u = v for the original DRS) which stands for, as Asher and Lascarides point out, *u is bound to an accessible discourse referent, but syntax doesn't indicate which one* (A&L, 2003). The resulting DRSs are now syntactically determined:

(4′)

x
man(*x*), walk-in(*x*)

4(b)

y, x
beer(*y*), order(*z,y*), *z* = ?

Set union on a. and b. gives:

4(c)

x y,z
man(*x*), walk-in(*x*)

The underspecified condition $z = ?$ is resolved by selection of an accessible antecedent and application of any constraints. In the case of pronoun resolution we have number and gender constraints, such that $z = x$.

As Asher and Lascarides have shown, however, the DRT account of interpretation is inadequate (A&L, 2003). The details of the discourse phenomena DRT fails to properly analyze—from anaphora, to cases of ellipsis (VP ellipsis), and presupposition—are the subject of detailed treatment in A&L (2003), and we won't repeat it here. For our purposes it suffices to note that DRT's constraints on accessibility can both permit incorrect antecedent attachments, and fail to permit correct ones. Asher and Lascarides analyze such cases as pointing to the inadequacy of the DRT treatment of *relations* between input sentence and prior discourse context. DRT, in other words, does not provide a semantically fine-grained notion of how sentences in discourse relate such that proper constraints on attachment can be computed. The theory of SDRT addresses this shortcoming.

Segmented Discourse Representation Theory (SDRT)

Segmented Discourse Representation Theory is a dynamic semantics based theory (and hence, is compositional with regard to syntax and semantics in the Frege, Montague tradition) that introduces a number of advances over DRT. One, it introduces a set of

rhetorical relations which capture information about the relational semantics of sentences (clauses) in discourse. In contrast to the single subordination relation in DRT, SDRT introduces sets of subordinating and coordinating relations that specify the relational semantics of coherent discourse (i.e., the semantics of how sentences relate to each other to form coherent discourse). Two, in DRT the "append" procedure which updates the discourse context with an input sentence is replaced with a nonmonotonic inference procedure reflecting the nonmonotonic nature of discourse update (sentences held to be true can be seen false based on new information) as well as the necessity to ground update in inference that can consider multiple sources of information.

Asher and Lascarides provide a detailed account of SDRT; we won't reproduce the discussion here. For our purposes, we accept A&L's claim that SDRT represents a *competence* theory of sentential discourse interpretation: it specifies what competent language users must know in order to interpret discourse. This is to say, competent language users use multiple sources of information (lexical, syntactic, semantic, pragmatic) to interpret, in the truth conditional sense, a sentence such that the interpretation of each sentence is a function of the preceding sentences and involves resolving the explicit relation between the input sentence and the context (prior sentences). Hence, each LF is connected to one or more prior LFs via explicit rhetorical relations as given by a theory such as SDRT (the connection itself is an inference that is non-monotonic in the sense described by A&L, although the details of the discourse update procedure in SDRT are not of concern here).

As A&L and others have noted, discourse has a hierarchical structure (A&L (2003), see also Mann and Thompson, (1986) for discussion of this point in the development of Rhetorical Structure Theory). This structure is a function of rhetorical relation type: coordinating relations such as narration extend discourse structure on the same level, while subordinating relations such as contrast introduce a lower, subordinated, level (A&L, 2003). The resulting structure—a Segmented Discourse Representation Structure (SDRS) in the SDRT formulation—helps in the computation of complex inter-sentential phenomena such as anaphora by constraining the set of allowable attachment points for anaphors in the discourse (A&L, 2003). Hence, the hierarchical structure of discourse exposed by a relational dynamic theory such as SDRT has important advantages over other theories for purposes of interpretation: the set of attachment points for underspecified elements can be reduced when rhetorical relations are inferred between the clauses or sets of clauses in discourse.

It is common knowledge (although widely debated—see discussions on Discourse Topic by Asher (2004a), Asher (2004b), Zeevat (2004), Oberlander (2004), Stede (2004), and others) that the attachment of rhetorical relations in theories such as SDRT are not just to prior clauses, but sometimes to sets of clauses or more abstractly, to the prior topic. Asher and Lascarides give the following example:

(5)

a. One plaintiff was passed over for promotion three times.

b. Another didn't get a raise for five years.

c. A third plaintiff was given a lower wage compared to males who were doing the same work.

d. But the jury didn't believe this.

The allowable antecedents for the anaphor *this* in (5d), is, as A&L note, either the "proposition expressed by the discourse as a whole or the proposition expressed by [(5c)]". But intuitively the preferred antecedent is in fact the former, since the jury presumably was considering the prior claims as part of a case involving the mentioned plaintiffs. Hence the abstract notion of topic—not expressed by a single but rather a set of clauses in discourse—is often the preferred attachment for rhetorical relations. A&L (2003) point out that SDRT accounts for such cases. Our present interest, however, is in the notion multi-clausal semantics: the *court case* is an abstract object—in the example, the discourse topic—that plays a significant role in interpretation. How exactly such multi-clausal objects are inferred from discourse is exactly our concern. We turn now to discussion of Discourse Topic; in the final section we introduce a novel multi-clausal object, the Primary Semantic Type (PST).

Discourse Topic

The topic of a discourse is, informally, just what it is *about*: news of a suicide bombing in Iraq is about an event—a suicide bombing—occurring in a location, Iraq. As Asher notes, "… there are cohesive chunks of text that are about the same thing—that's what the notion of discourse topic is supposed to capture" (Asher, 2004b). The ontological status of DT is the subject of much discussion (see Oberlander (2004) for a view on reducing DTs, Asher (2004), Kehler (2004), Stede (2004), Zeevat(2004) for discussion of the role of DTs in discourse interpretation); for our purposes, we grant the intuitive notion of topic—what a discourse is about—as perfectly well-defined for competent language users, and raise the question of how such multi-sentential inferences about discourse arise. How, exactly, do we *infer* what a discourse is about?

The question at first blush may seem simple, but for purposes of developing automated approaches to infer DT, there are known difficulties. For one, as Asher notes, topic is often *underspecified*, in the sense that inferring a topic while processing a discourse can change as new content becomes available: "…It may not be clear what the topic is until the discourse is over" (Asher, 2004b). To put it another way, topic inference is in fact *discourse-wide*: assigning a topic to a discourse may require, at the limit, consideration of the discourse in totality—the entire set of sentences comprising it. Computation of topic then would seem to be governed by a different mechanism than specified by theories of dynamic semantics (e.g., inferring subordinating or coordinating clauses using SDRT). Indeed, inferring topic—or other abstract objects that are discourse wide—doesn't seem to fit neatly in the Frege/Montague tradition at all: one can imagine

a successful interpretation of each sentence S of a discourse D, without a successful inference about the topic expressed by D. Dynamic semantics theories like SDRT are equally vulnerable (it would seem), since the addition of rhetorical information may help solve some intersentential problems (e.g., resolving the antecedents of anaphors), but don't seem plausible candidates for illuminating topic. Hence, DT appears to require some additional inference, over and above even a completed SDRS.

This picture is troubling only insofar as we assume that the topic inference has to fall out of the machinery of SDRT (or some other dynamic semantics theory, although following Zeevat (2004) and others, we accept SDRT as the most developed theory yet of discourse interpretation, as discussed above). Since the actual *representation* of DT is relatively straightforward (introduce a set of topic labels L, and append them to discourse or discourse segments—or, modify SDRS to include a variable to be bound by a topic), the question reduces to that of *inference*, and here we argue that there is no reason to suppose that the same mechanisms accomplishing non-discourse wide inference (e.g., resolving anaphora, computing rhetorical relations) must also carry the burden of inferring DTs. In fact—and this is of central importance to the present work—it seems unlikely that this can be the case. As Asher noted (cited above), if the underspecification of topic inference may necessitate waiting until the entire discourse has been processed, *ipso facto* the within discourse inference mechanism can't suffice.

We'll accept this conclusion, and in fact, we'll extend it to include abstract objects that are discourse-wide generally. We turn now to the Constraint Framework we will use to assess the difficulty of discourse interpretation problems.

Chapter 2: The Constraint Framework

In this chapter we develop a framework for assessing the difficulty of NLU problems as a function of features of input and the sorts of questions we can ask about the input. The basic idea is that our "AI complete" problem, the Turing Test, is difficult because it is unconstrained in these senses: the input can be any instance of (written) natural language assumed understandable by native speakers, and the type of questions about the input is essentially unbounded, varying according to the interests of the human interlocutor. A constraint framework that allows us to "parameterize" aspects of natural language interpretation (in particular, to parameterize the input and the questions about the input) gives us a principled way of discussing types of natural language understanding problems relative to the "Turing problem"; correspondingly, we can talk of the types of machines required to solve different problems defined in a constraint framework. We turn now to some definitions that will be useful in building the constraint framework.

Definitions

A *dialogue* is an exchange of natural language between a speaker and an addressee for the purposes of communication. A conversation between two or more people is a common example of an instance of dialogue.

A *monologue* is a production of natural language by a speaker to one or more addressees for the purposes of communication. A news article is a common example of an instance of monologue.

We call the union of dialogues and monologues *discourse*.

Written discourse is *text*.

A set of two or more texts is a *corpus*.

A *restricted* corpus contains only text on a particular topic or set of topics. By "topic" we mean, roughly, what the text is *about*: a text that describes a suicide bombing in Iraq is about a conflict event (a suicide bombing) that occurs in the context of a war, the Iraq War. This treatment of restriction via topic is also understood as a restriction on the *domain* that circumscribes the set of entities, their attributes, and the events that occur and to which the entities are related (i.e., as agents, patients, instruments, etc.). These remarks will be clarified considerably in upcoming discussion.

Discourse that can topic-labeled by some labeling L (where each label l in L labels at least one topic) is called *restricted relative to L*.

Discourse that can be topic-labeled by some labeling L, where each label l in L maps to a term in a controlled vocabulary V, and terms in V representing classes are hierarchically defined such that the V is properly speaking an ontology O, is called *restricted relative to*

O. Any discourse that is restricted relative to O will have a *hierarchical topic labeling* that maps to concepts defined in O.

Unrestricted discourse can be about any topic (or, express entities in any domain) whatsoever.

If classes in an ontology O can be used to topic-label the text in some corpus C then O *covers* C for the topic-labeling task. In general if terms in an ontology O can be used to perform some task T on some corpus C, then O covers C for T.

Formal discourse is written with conventional rules of grammar using conventional lexicons such as the words found in a common dictionary. For instance, news articles from mainstream news providers such as CNN are written in conformance to accepted rules of prose (periods at the end of sentences, lack of slang, few if any misspellings, etc.) and consist of the words (more or less) that one expects to find in a common dictionary.

Informal discourse is any discourse that is not considered formal. For instance, an Instant Message (IM) chat between two friends may exhibit relaxed grammar (partial sentences, lack of punctuation, no capitalization) as well as a shared "slang" that consists of words not found in the dictionary, or found in the dictionary but having a nonstandard or ironic meaning (cf. the use of "bad" to mean 'good').

Input Constraints

One way to make interpreting natural language easier (or harder) is to change or *constrain* properties of the input (i.e., the instance of natural language to be interpreted). This idea by itself is nothing new, and it follows straightforwardly from the observation that different natural language phenomena pose different challenges to NL interpretation. For instance, referential phenomena such as anaphora can increase the difficulty of the interpretation task by introducing the need to infer attachment points (which often require world knowledge and reasoning about discourse coherence), and ambiguous phenomena such as polysemous words (i.e., words that have multiple senses and hence may require sense disambiguation before sentence interpretation can succeed) can increase difficulty as well, for much the same reason: by requiring multiple sources of information such as world knowledge and some understanding of what is intended to be communicated in the discourse. Other pragmatic phenomena such as metaphor, as well as phrasal structures like compound nouns with idiosyncratic or conventional meanings further complicate the interpretation task. Constraining input to reduce the occurrence of such phenomena, then, is one way of easing the difficulty of interpreting language.

Although there is a very long list of phenomena that affect interpretation difficulty, our interest here is in introducing a small but general set of distinctions at the "type" level that are useful devices for increasing or decreasing the difficulty of (computational) interpretation. That is, our interest is not in cataloging the instances of

phenomena but rather in controlling input in terms of a small set of parameters with very general application to natural language input. As such we introduce the following restrictions:

1. Restrictions on topic or domain
2. Restrictions on genre (formal or informal)
3. Restrictions on type of discourse (monologue or dialogue).

A brief discussion of (1)-(3) follows. Later in the section we will discuss the question of interpretation difficulty using these input constraints.

Input Constraint Types

Restricting the topic or domain in discourse is a s*emantic* constraint, since what is at issue is the restriction of the sorts of things (entities, properties of entities) that can be discussed. Semantic constraints are of course extremely relevant to the question of interpretation difficulty, and they perform a great deal of the work when simplifying or making more difficult a discourse for processing. For instance, the microworld strategy was an endorsement of applying strict semantic constraints: Winograd's blocks world was only about blocks and their relative positions and colors to facilitate progress (again,

the idea was that, as the problems became better understood, the constraint would be relaxed).

The formality restriction is a s*tyle* constraint, as it affects the stylistic properties of a discourse that make it recognizable as well formed and grammatical or "relaxed" in various ways. To use the example of written prose, a typical instance of formal style is a dissertation, or in general an academic paper. Other examples are: news articles from major news providers, works of non-fiction, and many examples of fiction as one might find in a typical bookstore or library. Examples of informal discourse include sketching notes as one might do making a "to do" list or writing in the margin of a book, as well as discourse generated by members of particular communities that depart from the standards in some way or other. For instance, online chat communities using a medium like Instant Message (IM) chat typically use abbreviations, jargon (depending on the community), and phrases or partial sentences which might omit sentence terminators like periods, as well as other punctuation, and orthographic abnormalities such as use of all CAPS, or all lowercase letters.

Finally, the discourse type—monologue or dialogue—is a kind of *generation c*onstraint, in so far as monologue is generated by a single author (or speaker) and dialogue requires two or more participants, a speaker and at least one addressee. In general, moving from dialogue to monologue occasions a simplifying of the

interpretation task, although this is by no means obvious. A discussion of the sense in which this is so is given later in the chapter.

The Turing Test, as has been previously discussed, is difficult in large part because the input to a Turing Test Machine is not necessarily constrained along any of the dimensions introduced above. Hence, the Turing Test may require the interpretation of utterances in an unrestricted informal dialogue (UID). This is so because the human interlocutor in a Turing Test engages the machine in a) a conversation involving dialogue that can include instances of expository discourse such as sections of monologue, where the human b) converses with the machine on any desired topic (no semantic constraint), and c) uses whatever informal style of written conversation that is useful (which is to say, the machine must not fail miserably if the human produces sentences that are not formal, grammatically correct and having no slang or other generally understood idioms.) The UID, then, is *ex hypothesi* a very difficult type of input in the general case.

By simply forming different combinations using the semantic and style constraints as well as the discourse types we can generate a number of variations on the UID:

Restricted Informal Dialogues (conversations on a particular topic or topics only)

Unrestricted Formal Dialogues (formal conversations on any topic)

Unrestricted Informal Monologues (texts such as a "blog" on any topic)

Unrestricted Formal Monologues (texts such as news articles on any topic)

Note that this simple constraint framework provides a general mechanism for exploring the power of NLU systems. In particular, we can classify the power of recent and historical systems by assigning them to the proper input class, and we can classify the (hypothetical) system that would pass a Turing Test to begin to understand the gap between extant and future (or hypothetical) systems. The goal is to achieve a more systematic treatment of the relevant issues.

Input Triples

Consider an *input triple* of the form $T_I = \{C_1, C_2, C_3\}$, where C1 is a semantic constraint, C2 is a stylistic constraint, and C3 is a generation constraint as described above. Each constituent of an input triple constitutes an attribute pair whose members have values that are binary: <restricted, unrestricted>, <formal, informal>, <monologue, dialogue>, where restricted = {0,1}, formal = {0,1}, monologue = {0,1}, dialogue =

{0,1}.[12] We can impose a partial order on input triples according to a "hardness" of interpretation criterion if it holds that, for each pair, no member of the pair can have the same difficulty. For instance, for a pair with members Ci, Cj, it is not the case that Ci generates a type of input that is as hard as Cj. Letting '<' stand for 'generates input that is easier to interpret than", the following are asserted:

(1a) Restricted < Unrestricted

(1b) Formal < Informal

(1c) Monologue < Dialogue

The interpretation of input triples is clear enough given their attributes, since the hardness of a triple will be a function of the values of each attribute. So for instance, all input triples with restricted = 1 (true) and the same values for the other two attribute pairs will specify a type of input that is easier for a machine to interpret than the same triples with value restricted = 0 (false). To put it another way, the "restricted" triples will require less sophisticated machines. The use of input triples will become clear in the development of the Constraint Framework in this chapter. A discussion of (1a)-(1c) follows.

[12] In the case of the restricted attribute, there is the third option of being unrestricted relative to some ontology O. Given that there are indefinitely many ontologies one could construct with a suitable language, there are actually an infinite number of ways that input could be relatively unrestricted in this sense. However, at the limit O would be an "ontology of everything humans can talk about" so that unrestricted relative to O would be simply 'unrestricted'. Hence, if 'A < B' means "A is less restricted than B", then if A is unrestricted, B is unrestricted relative to O, and C is restricted, we still have A < B < C for most any O.

Restricted Input

Input restrictions are central to computational approaches to natural language interpretation. Earlier approaches to NLP explicitly restricted input (e.g., microworlds). Modern work tends to define tasks which represent a subset of the full interpretation problem. Each task may be applied to a large corpus of natural language texts (cf. the Reuters dataset); however, we can view such work as applying a kind of "semantic filter" to complex discourse, such that the only properties of interest are those relevant to the specific task. To take a simple example, the Named Entity Recognition (NER) task familiar to researchers in the field of Information Extraction (IE) simply views natural language as a sequence of tokens with properties, and NL input is processed by NER systems by simply considering windows of tokens together with any features extracted from each token, and tagging them with a tag in a defined tagset (e.g., PERSON, ORGANIZATION, LOCATION) or, if not in the tagset, OUT.

We can, then, "restrict" input using a "semantic filter" approach—whatever the semantic properties of the dataset to be processed (that is, whatever the set of concepts and properties actually expressed in a particular dataset), the task is defined such that only a set of concepts (e.g., concepts about baseball playing), and the relevant relations or properties ("hitting", "catching" etc.) are allowed. In this sense, whatever the set of

domains in the input, only domains of interest pass through the filter. To take an extreme example, we might train a classifier with a binary class label set {Person, Out} such that, for each token (word) of input, the classifier only recognizes mentions of names of persons, and labels every other token "OUT". In this sense, the domain for the classifier is just the concept of a person and an identifier (the person's name) and the set of nouns, which may be names of a person in the input. We could call such a classifier "restricted to labels L, where L contains only Person and Out. This classifier would fail to recognize any other semantic class, such as, for instance, organizations, or locations (or trout, cars, books, etc.).

Input restriction in the two senses discussed are two sides of the same coin: we can restrict NL input either by selecting only discourse that expresses semantics that our computational models have been designed to process, or we can restrict NL input by taking "native" NL without cherry picking its content, but simply ignore everything our models are not designed to handle (the semantic "filter" approach). Either way, we succeed in simplifying in some or many ways the full discourse interpretation task precisely by simplifying what will be considered in the input. We can, then, speak of domain restrictions in terms of input, or in terms of what will be output (we'll cover this terrain in depth in Chapter 4).

Finally, we make the (somewhat obvious) observation that *relaxing* restrictions such that more and more features of the world (more concepts, relations between concepts, and attributes of individuals) are part of the input for some system S brings an accompanying increase in interpretation difficulty for that system (the system most "know" more and more in order to properly interpret the input). In other words, the following holds:

(1a) Restricted < Unrestricted

Input Style

Restricting a domain is straightforwardly a method for controlling the difficulty of interpretation. Perhaps less obviously, stylistic considerations such as formality affect interpretation difficulty as well. Indeed, *informal* discourse presents a number of problems for systems performing interpretation tasks on natural language input. Cases of this phenomenon are not hard to find: published results for NER system performance often include separate results for datasets missing orthographic information such as capitalization, and NLP or IE results on informal datasets such as email or IM messages typically show lower accuracy than when performed on formal structured texts (Minkov et al, 1995).

The lack of expected structure is the culprit: the presence of expected syntactic cues such as sentence terminators, capitalization to indicate proper nouns, and many other such pieces of evidence provide a syntactic foundation upon which NLU systems process natural language to expose its meaning. Eliminating such cues results on the whole in less information by which to build models of language: for instance, sequential machine learning approaches to entity recognition typically define a context "window" within a sentence, and consequently assume sentences as units of input:

(1) The earnings summary gave investors cause for concern about the future of Dell.

(2) Mr. Dell has stepped down as CEO and will assume chairman duties effective immediately.

Determining which mention of "Dell" is a person, and which an organization, is a task that is often defined on the sequence of tokens comprising the sentence. Such systems assume the prior segmentation of discourse according to sentence boundaries (Minkov et al, 1995). As one might expect, most sentence boundary detection systems rely on terminating characters like a ".", "!", or "?" to determine that a sentence boundary has been reached. Informal discourse may relax the requirement that one of these characters must occur to end a sentence, turning the otherwise straightforward task of

determining the end of a sentence into a nontrivial inference task involving advanced heuristics (Reynar and Ratnaparkhi, 1997).

Other examples of difficulties include problems introduced by lexical issues such as misspelled words, relaxed grammar conventions, and use of phrases in place of complete sentences (Minkov et al, 1995). It is difficult to say exactly how performance for a particular NLU task will degrade as a function of the style of discourse, but since syntactically informal discourse contains greater numbers of lexical and grammatical *errors*, some degradation of performance relative to discourse without such errors will in general be expected. Minkov *et al* (1997), commenting on the difficulty of identifying personal names in a corpus of email messages, make the point well:

> Informal text is harder to process automatically. Informal documents do not obey strict grammatical conventions. They contain grammatical and spelling errors. Further, since the audience is more restricted, informal documents often use group- and task specific abbreviations and are not self-contained.

In addition to difficulties raised by relaxing orthographic and syntactic rules found in formal texts, grammatical and semantic differences in informal discourse present

well-known problems for NLU. Narimaya notes, for instance, that subject ellipsis occurs more frequently with informal English (Narimaya, 2004). This has the effect of introducing an additional inference requirement into the interpretation task, since it must be inferred from context (prior discourse) what entities are getting discussed in the absence of their specific mention.

Monologue versus Dialogue

As many researchers have noted, interpreting dialogue requires, in general, a more sophisticated analysis of language (see A&L, 2003 for discussion). Hearst notes that dialogue introduces additional processing requirements such as resolution of turn taking, "grounding and repairing misunderstandings", and successful execution of initiative and confirmation strategies (Hearst, 1994). And Asher and Lascarides note that the cognitive states of dialogue participants (in particular, their beliefs and intentions) become part of the interpretation of dialogue (A&L, 2003). As they put it:

> Dialogue is different (and harder) than monologue, because with the introduction of more than one participant there emerges the possibility of

information exchange, cooperation, agreement, and disagreement.

Discourse structure must also incorporate questions and requests.

The upshot of these (fairly obvious) observations about the differences between monologue and dialogue is: for some monologue M and some system S that interprets it, translating the monologue into a dialogue such that the context expressed by M is now expressed in a dialogue D between two or more participants, tends to introduce further interpretation difficulties such that a) the performance of S on D will be less than on M, and to recover the degraded performance, some system S*, more powerful than S, must be used to process D. We turn now to a discussion of query constraints as part of the constraint framework discussed in this chapter.

Query Constraints

As discussed previously, for each *input triple* an input type is determined which then circumscribes a class of machines that can interpret that input. This section provides a discussion of constraints on *questions* that can be issued to a machine: a litmus test for inclusion in the class of machines that perform interpretation adequately given input P as determined by some input triple T given the set of triples that can be constructed from the constraints, as discussed. To begin, we can postulate a "generation oracle" (GO) such

that, whenever a machine correctly interprets a discourse or an utterance in a discourse, the GO will produce a suitable response. If a machine cannot correctly interpret an utterance, the GO will generate "I don't understand". For instance, consider some snippet of dialogue in a Turing Test:

Human: Have you ever played baseball?
Machine: Yes, when I was a kid I was in Little League, but not much beyond that.
(Machine interprets question correctly)
Machine: I don't understand. (Machine does not interpret question correctly)

The GO device is necessary to avoid entangling NL interpretation with NL generation performance, which we do not consider in the present work.

For all interpretation tests with input determined by input triples with dialogue values for the discourse type attribute, the Turing Test with GO can be used to permit open-ended questioning for interpretation purposes. A similar schema can be used for monologue testing, where the test is modified such that a particular discourse is the target of the question answer session:

Human: Please read this article http://www.cnn.com/article123.html

Machine: Ok

Machine: I'm ready (after some time has passed)

Human: What is the article about?

To avoid turning the monologue assessment test into an unintended dialogue test, we can assume another black box that correctly interprets the questions themselves. Thus, the system receives the question in a form that it can parse, so that the natural language request for the topic of the article is given by a machine-parseable query like getTopic(Article123) or what have you. We can call this black box the "Interpretation Oracle" (IO).

The Turing Test with GO, and IO for the monologue case, constitutes the upper level of difficulty on interpretation assessment, since the human questioner can ask any question of the machine at all in the dialogue input case, and any question that a competence human language user would know about the test article, in the monologue case. However, just as with input considerations raised earlier, this interpretation baseline is still too difficult to be helpful, since, presumably no machines will pass a test given in this manner for much of the types of input we have been considering.[13] We can

[13] Very likely, no systems processing unrestricted input will pass the test. For instance, consider how difficult the task becomes when the human can keep choosing articles on different topics, then questioning

pursue the same strategy, then, with regard to interpretation by considering suitable constraints on the types of questions that can be asked. The goal, again, is to provide a framework for evaluation of systems on the interpretation task given input as determined by input triples.

We want a mechanism for controlling the difficulty of interpretation by applying constraints to the questions that can be posed to a system. A question is, of course, connected to some task that the system must perform in order to provide an answer. For instance, a question "What is this story about?" assumes that the machine can perform a text classification task: given a discourse D and some set of classes C = {C1,…,Cn}, assign some Ck in C to D which best describes the content of D. For example, consider two sentences S1 and S2 in D:

S1: Bob went to the store yesterday.

S2: He made it to the bank today.

The question "Who went to the bank today?" assumes that the machine can perform an anaphora resolution task such that the pronoun "He" in S1 is resolved to the proper name antecedent "Bob" in S2. Given only S1, the question "What type of entity is the proper

---

a system to determine whether it understands the content of each article. Surely, we can't achieve this level of performance at this point.

name mention ‘Bob’?” assumes that the machine can perform entity recognition (NER), such that “Bob” is assigned the label “Person” by the system.

These questions—and the tasks that they determine for the machine—group naturally into the following:

1. Discourse Questions (tasks that are defined over the entire discourse)
2. Intersentential Questions (tasks that are defined over more than one sentence in the discourse)
3. Intrasentential Questions (tasks that are defined within the scope of a sentence in the discourse)

Discourse, Intersentential, and Intrasentential questions form the basis for the construction of different *query triples*, to be discussed next.

Query Triples

As with input constraints, we can consider different *query* constraints for the purposes of controlling the difficulty of the interpretation *task* for a machine. A *query triple* has the following form:

T_Q = {A, B, C} where S is a sentence in some discourse D and A = S1,…,S_n such that S_n is the last sentence in D, B = S_m…S_{m-k} for m >= 1, 0 < k <= n, and C = S_k for some k such that 0 <= k <= n.

Informally, each query triple defines the set of questions that can be asked based on the tasks required to answer the question. Each argument to the triple holds a Boolean value that represents whether questions of the type assigned to the argument position can be asked. For instance, in the 0^{th} argument position, the Boolean value will represent whether discourse-wide questions can be asked. In the 1^{st} position, the value represents whether intersentential questions can be asked. Likewise, in the 2^{nd} position the value represents whether intrasentential questions can be asked. Hence, T1 = {0,0,1} represents the set of questions that can be asked of a machine where an individual sentence in the input suffices to answer the question. For instance, "Is the proper name mention "Bob Smith" in sentence S2 a reference to a person?" is an allowable question given that an assessment has been constrained with triple T1.[14]

[14] Current state of the art given UID input is not more than T = {0,0,1}, and very probably extant systems on truly unrestricted, informal dialogue input would not achieve high accuracy even given questions defined within the scope of a single sentence (or utterance). As many researchers have noted, the coreference resolution problem for natural language, which requires at least T = {0,1,1}, is an unsolved problem, with reference resolution in the last MUC competitions having at 70% Fmeasure or less in spite of RFM input (newswire text on a given topic) (MUC, 1993).

It is not the present purpose to attempt to achieve a precise evaluation of current systems, and certainly the difficulty of achieving exact partitions of NLU tasks (does word sense disambiguation require discourse-wide features? Is named entity recognition best achieved with non-local evidence?) would require a thesis in itself. The purpose here is to provide the basic machinery, which allows us to consider different classes of problems for machine based on specifiable constraints on a) input and b) queries. We can set aside questions of how best to solve NLU tasks in favor of simply noting that some tasks can be addressed at the sentential level, while others quite obviously require multi-sentential or discourse-wide scope. For instance, no serious researcher in NLP would define document classification at the level of an individual sentence (although, if one did, picking the first sentence might be a reasonable heuristic!).

With these mechanisms in place, it is easy to see that the original Turing Test is described as with the triple pair {UID, T3} (where "T3" refers to the triple such that all values are "1", i.e., T3 = {1,1,1}. We note also that the relationship between input and query triples is, intuitively, inverse, so that RFM input may be processable even given a T3 query triple (for instance, Winograd's SHRDLU (1972) could perform intersentential processing such as coreference resolution given the restricted blocks world domain within which the human computer dialogue occurred).

World Knowledge and the Difficulty of NLU

An astute reader may have noticed at this point that, even granting a query constrained assessment, the interpretation task still seems hopelessly difficult. For instance, suppose a constraint given by T = {0,0,1} so that only questions with answers that can be determined by consideration of an individual sentence are allowed. But note that *any* question about the sentence can still be asked, with only the proviso that competent speakers of the language would be able to answer them. Examples such as Bar-Hillel's "The pen is in the box" expose the problem—even interpreting a single, simple sentence requires knowledge of the world (Bar-Hillel, 1960). John Haugeland (1979) gives us other examples: "I left my raincoat in the bathtub, because it was still wet.", as well as "Though her blouse draped stylishly, her pants seemed painted on.", and the literature is replete with examples of simple, single sentences where lexical or world knowledge requirements seem frustratingly high.

These central difficulties with interpretation are the subject of Chapter 5. For our purposes here, we need the constraint machinery to talk about classes of systems that can solve certain problems, but not others (or, equivalently, classes of interpretation problems that some machines can solve, but not others). For this purpose we introduce the notion of "completeness" for query constraints, such that a query constraint (i.e. given by a query triple) is said to be *complete* when any questions not violating the triple values are allowable. We can call a constraint *incomplete* whenever a specific type of question is

allowable, and no others. We will leave 'type of question' unanalyzed except to note that an incomplete constraint will use some subset of the possible questions for a given query triple, where the size of the set will be determined (presumably) by the type of task.

Some Common Definitions of NLU Tasks in the Framework

We now can define NLU tasks in terms of the query constraint framework. For instance, named entity recognition (NER), a common task in the subfield of NLP known as Information Extraction, is a task constrained by an incomplete query triple T1 = {0,0,1}, where the subset of questions correspond to the set of tags (i.e., tagset) defined for the NER task. For instance, given the tagset {Person, Organization, and Location}, the NER task for this tagset is defined in terms of questions of this form:

1. Do the tokens T_k,..,T_{k+l} for l >= 0) in sentence S_n refer to an entity of type Person?

2. Do the tokens T_k,…,T_{k+l} for l >= 0 in sentence S_n refer to an entity of type Organization?

3. Do the tokens T_k,…,T_{k+l} for l >= 0 in sentence S_n refer to an entity of type Location?

Likewise, the anaphora resolution problem is constrained by an (incomplete) T1 = {0,0,1} for the intrasentential case or T2 = {0,1,1} in the intersentential or "within document" case. For the latter, we have questions of the form:

> Are the tokens $T_k,\ldots,T_{k+l}$ for $l >= 0$ in sentence S_{n-m} for $m > 0$ an antecedent to the potential anaphor located at tokens $t_p,..,t_{p+q}$ for $q >= 0$ in sentence n?

For the former, the question form is exactly the same with the exception that there need not be a prior sentence:

> Are the tokens $T_k,\ldots,T_{k+l}$ for $l >= 0$ in sentence S_{n-m} for $m >= 0$ an antecedent to the potential anaphor located at tokens $t_p,..,t_{p+q}$ for $q >= 0$ in sentence n?

The document classification task (aka text classification, we'll use them interchangeably in this work) in the subfield of NLP known as Information Retrieval (IR) is constrained by an (incomplete) query triple T3 = {1,0,0} where the subset of questions that can be asked are of the form:

Does document instance D_n in corpus C have label l?

Does document instance D_n in corpus C have label l+1?

RFM*-<1,0,0> for O that covers newswire text

In this work, we consider a machine that is input constrained by RFM with an ontology that covers newswire articles (denoted hereafter by RFM*) from major content providers such as Reuters, the Associated Press, AFP, BBC, and others; it is query constrained by an incomplete T = <1,0,1>. This machine, called "TPE" for "Text Processing Engine" has a number of advanced features that will be discussed at length in Chapters 3 and 4. In particular, TPE:

1. Classifies text hierarchically, so that the most specific classification given a hierarchy in O is returned to label articles in the newswire dataset.
2. Classifies text according to multiple hierarchies defined in O, so that multiple labels corresponding to different concepts (e.g., for events, locations of events, persons, and organizations) are returned after processing input.
3. Uses an ontology O that covers most newswire text, making it a broad coverage, machine on a large a real-world dataset (news).

4. Answers a set of questions about the *events* expressed in the text that gives usable information about what happened as well as where it happened in nonfiction monologue (news):

    - What is the specific primary event?
    - Where did the primary event occur?

Four above is particularly significant, because it means that there is a machine that, given arbitrary newswire text (i.e., not seen before, and on a broad range of topics), can label the text with information about the main event described, and where it occurred. This result stands in contrast to either text classification systems investigated in IR, which aim for more generic labels such as topics or subjects, or within document extraction systems coming from work in IE, which extract entities (persons, organizations, locations) or events from free text but typically have no way of filtering events according to criteria such as "main event discussed". TPE, then, processes text to answer discourse wide questions that are semantically meaningful (e.g., they permit one to say "this is what was discussed, here is where it happened, and here are the main participants"). In the next chapter we begin a discussion of the task that TPE performs.

Chapter 3: Primary Semantic Types: Definition and Representation

In Chapter 2 we introduced a framework for defining NLP problems in terms of input and query constraints. We noted, briefly, an input constraint RFM*, where * denotes an ontology that covers a multi-domain dataset, together with query constraints of the form:

- What is the primary event?
- Where did the primary event occur?

We refer to a particular multi-domain dataset consisting of newswire articles from major content providers such as the Associated Press, Reuters, BBC, and others as an instance of RFM* constrained data, called the "newswire dataset" or just "NW dataset" hereafter. The claim, then, is that the problem of answering the queries for NW data is an interesting problem and that a machine built to solve a problem of this type would constitute an advance in natural language processing.

In Chapter 3 we will unpack these assertions in more detail by defining a new NLP task, Primary Semantic Type Labeling (PSTL), and a particular instance of PSTL

that specifies an event and location type for applying the query constraints above. For ease of mention we'll refer to this task as the "primary event labeling" task, or "PEL". In what follows, we investigate the representational requirements for PEL as well as offer a specific representation strategy for performing PEL on NW data. In Chapter 4 we will show how the representation strategy makes possible an inference technique—thresholded traversal of hierarchies—that enables a fine grained classification of newswire articles along multiple semantic dimensions. We introduce the PEL task in the next section by discussing, first, a related "within document" task that has received much attention from researchers in NLP recently, semantic role labeling.

Semantic Role Labeling

As noted in the Chapter 1 history section, research on NLP entered a semantic phase in the 1960s after it became obvious that world knowledge was a critical factor in the successful interpretation of natural language discourse. So-called "Case Roles" noted by Charles Fillmore (1968) and others, and the evolution of work on semantic role labeling of natural language sentences to scripts, frames and other semantic structures (e.g., Roger Schank's (1975) work on scripts), dominated work on NLP for decades. Today, Semantic Role Labeling (SRL) is a well-explored task in NLP that has been

widely adopted as a key component of empirical approaches to text interpretation (Carreras and Màrquez, 2005).

The SRL task is to identify events and their participating entities, and determine the semantic "roles" that these entities play with respect to the event. *SRL* is traditionally formulated intrasententially: given a sentence, extract the event, the participating entities, and their semantic roles. For example, in the simple sentence "John threw the ball to Mary", tagged output from a system performing SRL might be (in XML-style syntax):

<agent>John</agent><event>threw</event> the <instrument>ball</instrument> to <patient>Mary</patient>.

Since elements like "<event>" are not particularly helpful for determining what *type* of event John and Mary are participating in, given a set of event concepts in an ontology linked to lexical entries (verbs) such as "threw" (or the lexeme "throw"), we might refine the output to be more informative:

<agent>John</agent><event-throw>threw</event-throw> the <instrument>ball</instrument> to <patient>Mary</patient>.

The semantics can be further sharpened given entity recognition capabilities (e.g., from a named entity recognition (NER) system):

<agent-person>John</agent-person><event-throw>threw</event-throw> the <instrument>ball</instrument> to <patient-person>Mary</patient-person>.

If we have, in our ontology, concepts for "Person", "Throwing", and "Ball" (a subclass of GamePiece, not any round object), the SRL machine produces output that, when resolved against the concepts in the ontology, provides answers to questions of the form "who" did "what" to "whom", "when" and "where". Resolving events and event participants is, of course, of central concern to discourse interpretation.

As might be expected, work on SRL continues, and has received great attention recently, especially with the shift from manual "grammar" methods (cf. Hirst (1987), Postejovsky (1995), Copestake and Flickinger (2000)) to statistical learning methods over the last decade. Such statistical approaches to SRL have been facilitated by the development of large semantically annotated corpora like the PropBank (Kingsbury and Palmer, 2002), FrameNet (Johnson *et al*, 2003), and NomBank (Meyers et al, 2004) initiatives. SRL was included as a CoNLL-2004 and CoNLL-2005 shared task (CoNLL,

2004-5). SRL is, in short, a continuing problem of interest for researchers working on the semantic contribution to the interpretation of natural language.

As discussed, SRL is defined over input sentences in natural language: we want to identify, given some sentence S, the semantic roles expressed in S. An obvious limitation of such sentence-by-sentence analysis methods is that they fail to inform the text interpreter what a discourse is *about*. There may be, for instance, a particular event mention in a discourse that is, intuitively, central, in the sense that the purpose of the discourse is recognizably to elaborate on this particular event. For example, a news article about a crime—say, a shooting event occurring in some city—is properly about the shooting, in the sense that the introduction and elaboration of the shooting event constitutes the reason the discourse was written. In this sense, it is clear that events can function as topics in discourse, particularly if a particular event central in the discourse is construed more generally, such as, for instance "Crime" or more specifically "ViolentCrime" in the example just discussed. The connection between topic and a particular event mention in the discourse is the subject of later discussion in this chapter. For now we'll introduce the term *primary event* to mean just the particular event mention in a discourse which is central to the discourse in the sense just described.

Primary events may be expressed in a particular sentence, but their role in the discourse (i.e., as "primary") will not in general be recovered from inspection of the sentence in which they occur. Finding a primary event, then, does not reduce to

semantic role labeling. These considerations suggest that primary events are semantically underspecified in discourse, in the sense that, at the limit, it may require the interpretation of every sentence in a discourse D to correctly identify, out of the set of events expressed in D, the primary one. Asher's (2004a) observation that discourse topic (DT) is underspecified, in the sense that it "may not be clear what the topic is until the discourse is over" is germane here, since for the very reason that we expect DTs to be underspecified, primary events that have the discourse centrality condition will have this property as well.

We can know, for instance, that a particular agent performed an action (e.g., the man shot the gun) from analysis of sentence S_k in some discourse D, but we can't from this fact alone reliably infer that the event mention is primary in D (or, in the same vein, that the DT is the event or some superclass of the event). It may be the case that the shooting event is central, or it may be entirely peripheral. We conclude that some *discourse-wide* method of analysis will be necessary to reliably infer primary events; sentential analysis methods such as SRL won't suffice, because every sentence may require processing (i.e., the entire discourse) before it's possible to identify a primary event.

Primary Semantic Type Labeling (PSTL)

The move from sentential semantic analysis to discourse-wide analysis is a key feature of *primary semantic type labeling* (PSTL). We define a PSTL task as: return the set of primary types expressed in a discourse D subject to the centrality criterion. Given semantic types event and location such that an instance of an event is interpreted as answering "What happened?" in a discourse and an instance of location is interpreted as answering "Where did it occur?", we have the PEL task introduced earlier. Given a larger set of query triples, we could specify additional types (persons, organizations) and additional modifiers (e.g., time of occurrence) to form other PSTL tasks.

In addition to the centrality criteria for primary events, whenever two or more events mentioned in a discourse are hierarchically related (i.e., one event subsumes another in some discourse D, where both events are primary event candidates for D), we prefer the more specific event satisfying the centrality criterion. In general, given a hierarchy of events expressed in D, we prefer the most specific event. Hence, the PSTL task we define is to identify an event and its location in a discourse, subject to centrality and specificity criteria. We call this the *primary specific event (PSE)*: the most specific primary event given two or more candidates in D. (Specificity will be well-defined whenever there is an ontology expressing candidate events: if E1 is a subclass of E2 where {E1, E2} are the candidate primary events for D, then E1 is the primary specific event for D.) Including the specificity condition for primary events gives us a species of PSTL we can call primary specific event labeling, or PSEL.

Replacing the query constraints for primary events with ones for PSEs, we have:

- What is the primary specific event?
- Where did the primary specific event occur?

Input constraining the task to RFM* data gives us a definition of PSEL within the Constraint Framework. This task performed specifically on NW data is the subject of the current investigation. We'll return to the representational requirements for event labeling later in this chapter.

Including Context for PSEL

Event specificity is important because in many cases it results in more informative answers to the "What happened?" query we wish to answer. Unfortunately, a specific event can occur in many different circumstances or contexts: a suicide bombing in Iraq is radically different than one at a mall in America by a domestic or lone bomber. Minimally, then, it seems we want the specific event e elaborated on in some discourse D

to include a *minimal interpretive context* within which e would make sense as an answer to the "What happened?" query. Admittedly, specifying an adequate "context" will be open-ended and interest-relative, in the sense that what counts as adequate context for interpreting an event may vary from person to person and certainly changes as a function of time. Setting aside these difficulties however, a natural and relatively uncontroversial sense of context is simply *topic*, where topic can be based on current news topics or a subject-based classification scheme like those used in library science. For instance, consider this snippet from AP:

> **SYDNEY** (AFP) - A second East Timorese civilian died Saturday from injuries sustained in a clash with Australian troops **in the capital Dili**, the Australian Defence Force said.

In this example, the PSE is a military shooting. The event context, construed as topic, might be "conflict" or more specifically "ongoing armed conflict" or (better) "East Timor conflict". We now have a pair—a PSE with a suitable topic for interpreting the PSE that provides an adequate (if minimal) answer to the question "What happened?" for this particular discourse. An answer to the query given the pair can now be "there was a military shooting event in the context of the East Timor conflict", which to most human interlocutors would suffice as a reasonable answer. We will consider these pairs as

"basic" for PSEL, in the sense that a PSE is always given with a topic that provides context. The use of topics will be discussed in more detail later in this chapter.

Representational Considerations for PSEL on NW Data

In this section, we consider the representational requirements for performing PSEL on NW data. Since NW data is a species of RFM* input, we have a multi-domain dataset that must be topic-labeled (the event context) and labeled for PSEs. Prima facie, this presents us with a formidable representation task, since presumably the number of topics for such input will be large, and a fortiori the number of events that may be expressed in NW data would seem a large set indeed. The representation task, then, is to specify label sets for topics and events, such that arbitrary instances of NW data can be assigned a suitable label from the topic and the event label set. As we will see, it will prove advantageous to map topics and PSEs into a controlled vocabulary (ontology) for purposes of employing hierarchical classification techniques discussed in Chapter 4.

Topic Label Schemes - RCVI

Newswire data has been analyzed extensively in IR research. Corpora such as Reuters-21578, and the more recent release of Reuters Corpus Volume 1 (RCV1) classify articles according to content-based categories such as topic, industry, and region (Lewis *et al*, 2004). Such corpora thus provide broad topic-labeling schemes for subsets of NW data, and hence constitute important sources of information for the present investigation. RCV1, for instance, topic-labels over 800K newswire articles published by Reuters between 1996-7. The topic labeling is hierarchical, with four top-level labels: Corporate/Industrial, Economics, Government/Social, and Markets. There are 123 total topics identified, with 103 used to assign to at least one document in the corpus (Lewis et al, 2004).

RCVI has a number of features that make it a candidate for topic-labeling NW data. For one, it mostly separates topical information (what the newswire article is about) from geographical information (the main place or places to which the topic refers). This is particular important for PSEL, since identifying the PSE, and context (topic), and where it occurred are all separate tasks.[15] By contrast, popular news labeling schemes from content providers such as Yahoo! News, CNN, and many others routinely conflate topical and geographical distinctions. For instance, Yahoo! News provides the following categories (labels):

[15] Technically, PSE classification is conditional on prior topic recognition (as we will see in Chapter 4), but the label sets for PSEs, topics, and locations are non-overlapping, so they are separate in the sense that tasks do not share labels—location information is not used for topic classification, etc.

U.S.

Business

World

Entertainment

Sports

Tech

Politics

Elections

Science

Health

Most Popular

Here, topical information such as politics co-exists with explicitly geographical information such as “U.S.” or “World”.

We can detour here to make a few more observations about the Yahoo! News scheme, by way of separating it from RCVI, as well as the preferred representation for PSEL to be introduced. One, the scheme is trivially exhaustive, insofar as the categories “U.S.” and “World” form a partition: the intended extension of “World” is “all news outside the “U.S.” We also note that the scheme is not *exclusive* (only one label for each instance of news): an article about Ford Motor Company is both a business story

(Business) and a U.S. story. The scheme is thus a species of exhaustive non-exclusive categorization relative to NW data. (We might introduce the term *broad coverage cross-referenced* categorization scheme here, since a business article falling under Business will be cross-referenced under U.S. whenever the article is about a U.S. business.)

We note, finally, that the scheme is loosely based on a Subject classification scheme, reflecting the "library of science" view of information organization with subjects like Business, Politics, Sports, etc. However, categories such as "Most Popular" are not traditional subjects; they reflect, more, the consumer or market input of Yahoo! News users. We call this type of categorization *schematically heterogeneous*, including subjects such as those found (in more complete form) in schemes like the Library of Congress Classification (LCC), but including other non-subject-based categories that reflect consumer interest.

RCVI, by contrast, is with few exceptions schematically homogeneous: the topic hierarchy is distinct from "Industry" and "Region" (geography) representation. As noted above, this makes it a serious candidate for PSEL, because we need to separate what happened from where it happened (we can't, for instance, assign a "topic" such as U.S., because this would make it impossible to determine, separately, what the event context for a PSE is, and where the PSE occurred, since the same label was "overloaded" for both topic and location). Where RCVI fails, arguably, to maintain homogeneity is with topics such as "European Community" (EC), with subtopics "EC Internal Market", "EC

Corporate Policy", "EC Agriculture Policy", "EC Monetary/Economic", "EC Institutions", "EC Environmental Issues", "EC Competition/Subsidy", "EC External Relations", and "EC General". Here, the topic includes geographical information ("European"), which is locational and so arguably should be part of a geographical type (although, as we will see topics such as IraqWar seem acceptable, and subject to the same criticism). And subtopics of European Community also make reference to organizations ("EC Institutions"), which would be separate primary semantic types in the present scheme.

Notwithstanding these relatively minor quibbles, the RCVI representation seems well-suited to topic-labeling NW data. It is intended to be exhaustive, covering all newswire (even including topics such as "human interest" which we'll discuss much more in Chapter 5), and has the virtue of basing the topic scheme on distinctions that Reuters journalists make when submitting stories. Presumably, the journalists who write and submit articles are well-positioned to assign topical information, since they presumably know what the discourse is intended to be about. This makes the RCVI representation a rich source of information about intended topic for NW data.

Nevertheless, there are a number of problems with the RCVI representation scheme that argue in favor of designing, at least partially, a novel representational for topic-labeling NW data. For one, the data has become a bit outdated: spanning the years 1996-1997 only. Since newswire articles reflect current debates and circumstances,

topics will change over time, with some becoming deprecated and others becoming relevant. Entirely new topics get introduced. Two, RCVI topics seem in some cases too broad, and in others too fine-grained. For instance, "War" has no subtopics in RCVI, and so one can't, for instance, group articles by topics such as "IraqWar" or "AfghanistanWar" and so on. And also event information, such as interest rate increases in the context of the economy, are topics in RCVI. Given the current distinction between event context and PSEs, the latter would seem a more suitable home for specific economic events such as interest rate increases. But since RCVI does not distinguish between topics qua context for events, and primary specific events themselves (topics and events are effectively conflated in RCV1), there is nowhere else to represent specific events having topical importance such as interest rate hikes. They are simply subtopics of Economy. This representation may be suitable for some purposes, but for reasoning about events ("What happened"?), it seems that an explicit representation of events—apart from abstract notions such as topic or context—is superior.

Finally, RCVI is exclusive: only one topic is assigned to each instance of NW data. It is thus an exhaustive, exclusive classification scheme, since each instance of NW data must receive at least one topic label, but no more. This has a number of drawbacks for the present investigation, ranging from conceptual (is news really only about one topic? Does an article about a criminal investigation of a member of Congress receive topic Crime, or Politics, or both?) to practical (i.e., problems with inter-annotator agreement whenever two topics seem suitable: why force them to choose?). For these

reasons, our present course will diverge from RCVI in favor of a fresh scheme, which nonetheless is informed by it.

PSEL - Topics

To topic label NW data for PSEL, we develop an exhaustive, non-exclusive set of topics T such that each instance of NW data receives at least one topic from T (the total number of topics that may get assigned is effectively unbounded, but in practice newswire has one or two identifiable topics, and rarely more). Since we will need to specify subtopics, the members of T are represented as instances of a class "Topic", which is defined in a knowledge representation language having a well-defined semantics (to be discussed in the implementation section upcoming). The representation is strictly speaking non-hierarchical, since the subtopic relation is meronymic rather than one of subsumption. Hence, we have, for members of $T= t_1,\ldots,t_n$:

$isa(t_1,Topic)$

$isa(t_2,Topic)$

…

$isa(t_n,Topic)$

And, for or some t_k with subtopic t_j:

subtopic(t_k,t_j).

To date, T has the following members:

Accidents/Disasters, Business, Crime, Defense, Economy, Entertainment, Environment, Fashion, Health, Human Interest (Odd), International Relations, Legal/Judicial, Markets, Obituaries, People, Domestic Politics, Religion, Science, Technology, Terrorism, Travel, Sports, War/Unrest

In addition, the "EthicalIssues" topic currently has the following subtopics:

Abortion

Death Penalty

Drug Legalization

Environment

Euthanasia

Gun Control

Poverty

Social Security

Any instance of NW data should receive at least one label from T or a subtopic of a member of T (with the exception of any "odd" or humorous news, to be discussed in Chapter 5).

Defining Primary Specific Events - Preliminaries

The Primary Semantic Type Labeling task introduced in this chapter returns the set of primary types subject to the centrality and specificity condition. However, up until now we've helped ourselves to the notion of a semantic type, and in particular we've assumed that *events* are primary types when discussing PEL just as persons, organizations, topics, and issues, and locations are types that can be used to define other PSTL tasks. But what is a semantic type, and are events the sorts of things that qualify?

A semantic type is, at root, a concept in an ontology. A *primary semantic type* is any semantic type that, for some set of discourses (or, for some corpus of natural

language texts), has the primacy condition as discussed earlier. Hence, primary semantic types are just those concepts in an ontology that, as an empirical matter, are frequently primary in discourse. We note here that PSTs are ideally general concepts, such that they have more specific subclasses and instances that might be mentioned or implied in particular instances of discourse (i.e., they have subclasses that meet the specificity condition described earlier). Hence, "Person" is a PST, since persons are often primary in discourse (e.g., all articles discussing a politician such as Barack Obama), and also "Person" is suitably general, in the sense that particular instances of Person can be the output of a PSTL task.

We note also that abstract objects like "Topic" or "Issue" can serve as PSTs, and in general any concept in an ontology meeting the primacy condition in instances of discourse can be assigned a PST label. PSTs, in other words, can be any concept in an ontology that refers to an abstract or concrete thing in the world that might be mentioned or implied in natural language. People, Organizations, Locations, and abstract concepts like Topic or Issues are all candidate PSTs by this definition, since they are plausible nodes in an ontology of concrete and abstract objects. Likewise "Event", since events are (at least prima facie) the sorts of things that exist in the world and event mentions are ubiquitous in common language.

However, since the Event type plays such a central role in the current investigation, and since the ontological status of events has been the subject of ongoing

philosophical debate (Do events exist? Are they things in the world?), we'll detour briefly to unpack more the question of whether events are concepts in an ontology; which is to say, whether we can build PEL on the foundation of Event as a PST. In the final part of the next section we'll address the extent to which ontological questions about semantic types challenge the goals of the present work.

The Ontological Status of Events

It is, of course, clear by now that the present investigation assumes that events have a legitimate ontological status such that introducing an Event PST (i.e., as a concept in an ontology with discourse properties centrality and specificity) is not problematic. Our view is that we can talk about events as things in the world, and just as with other semantic types, events can be expressed in ontologies and referred to directly (e.g., *this* event, or *that* event mentioned in a discourse picks out events in the world like throwing a ball, or dropping a saucer, or bombing an embassy). If this seems unproblematic to the reader at this point, we agree. This view is echoed in recent work on language interpretation and analysis (A&L, 2003), and is consistent with common language usage. However, a number of philosophers such as Chisholm (1964), Kenny (1963), Strawson (1959) and many others have argued that events are troublesome denizens of an ontology, that they are not analyzable as particular things, like humans, elephants, baseball bats,

and the like. Chisholm, for instance, argues that event language in common discourse should be analyzed as "states of affairs", and that construing events as directly referable entities poses a number of problems, including accounting for so-called recurrence phenomena. Kenny similarly wishes to re-interpret events as "states" (including terminal states), and Strawson argues that events have a conceptual dependency on objects (but not vice versa) that make their inclusion in an explicit ontology problematic.

In contrast to such skeptics, however, others – notably Donald Davidson – have argued in favor of realism with regard to events: events are entities in the world that license an explicit ontology, just as ordinary objects do (Davidson, 1970). In what follows we will briefly summarize Davidson's position. Unless otherwise stated, we'll refer to this position as "Event Realism" (ER).

Davidson on Events

Davidson argues that events are individual entities such that, logically, we can introduce bound variables that take individual events as arguments. Hence, "Smith climbed Everest" can be rendered as (Exists x) (climbed(Smith, Everest, x)). Davidson argues that treating events as, as he puts it, "concrete particulars" in this sense makes sense of a range of issues in the philosophy of language and analytic philosophy

generally (Davidson, 1970). For instance, he claims that theories of action (where "actions" are presumably a subspecies of events) more or less require ER:

> Are there good reasons for taking events seriously as entities? There are indeed. First, it is hard to imagine a satisfactory theory of action if we cannot talk literally of the same action under different descriptions. Jones managed to apologize by saying 'I apologize'; but only because, under the circumstances, saying 'I apologize' was apologizing. Cedric intentionally burned the scrap of paper; this serves to excuse his burning a valuable document only because he did not know the scrap was the document and because his burning the scrap was (identical with) his burning the document (Davidson, 1969).

Further, explanation—explaining why things happen in the world—seems to require a robust notion of events that ER provides as well. As Davidson observes, events that evoke desires for explanations of why they occurred—tragedies, say, such as an avalanche—typically undergo several redescriptions in order to make clear the underlying causal mechanisms which can serve to explain or at least make clearer the occurrence of the event. Yet such different descriptions of the underlying process—the avalanche, in this case—seem to make sense only if there really is an underlying process to describe and redescribe. This is to say that, the process of explaining something seems to make sense only if there is a process (event) to explain in the first place. As Davidson puts it:

> There are rough statistical laws about avalanches: avalanches tend to occur when a heavy snow falls after a period of melting and freezing, so that the

> new snow does not bind to the old. But we could go further in explaining this avalanche –why it came just when it did, why it covered the area it did, and so forth—if we described it in still a different and more precise vocabulary. And when we mention, in one way or another , the cause of the avalanche, we apparently claim that though we may not know such a description or such a law, there must be descriptions of cause and avalanche such that those descriptions instantiate a true causal law. All this talk of descriptions and redescriptions makes sense, it would seem, only on the assumption that there are bona fide entities to be described and redescribed (Davidson, 1969).

With regard to mental events, too, as well as physical events, we see that some robust notion of events as entities (i.e., a theory like ER) is required to make sense of identity theories of mind, where mental events are identified with physiological ones. Affirming (or denying) such theories presupposes the acceptance of events as individuals. As Davidson concludes, "… for such theories to be interesting, there must be ways of telling when statements of event-identity are true" (Davidson, 1969).

Further, within the purview of theories of natural language, entailment relations between LFs constructed from natural language sentences are easy to capture when events are treated as first-order objects in an ontology. For instance, the LF for the sentence 'Sebastian strolled through the streets of Bologna at 2 a.m.' ought to entail the LF for the sentence 'Sebastian strolled through the streets of Bologna' (Davidson, 1969). But as Davidson observes, non-ER treatments of such sentences typically assign a (non-

reducible) three-place predicate to the former sentence, and a (non-reducible) two-place predicate for the latter:

(a) x strolled through y at t
(b) x strolled through y

On this formulation, it is clear that (a) entails (b), yet without the instantiation of the strolling event, non-ER theories can't capture it. Yet by constructing the LFs to include explicit terms for the strolling event, (b) is clearly entailed by (a) by conjunction elimination. The fact that such obvious entailments are difficult to capture without ER suggests that an explicit event ontology is part and parcel of a robust (logical) treatment of natural language.

Dropping ER in favor of non-ER theories of events (cf. Chisholm's "states of affairs", or Kenny's "states") also typically involves a loss of semantic information that is difficult to recover given alternative locutions that eliminate direct reference to events. As Davidson notes, locutions of the form 'brought it about that p' intended to replace direct reference to an events are semantically weaker than their original ER counterparts (Davidson, 1969). To borrow Davidson's example, 'The doctor removed the patient's appendix' is not the same as 'The doctor brought it about that the patient has no appendix', because the former case tells us directly that the doctor him or herself performed the removal, while the latter leaves open the possibility that someone else (perhaps ordered by the doctor) performed the removal. Likewise, Davidson observes

that "… 'Cass walked to the store' can't be given as 'Cass brought it about that Cass is at the store', since this drops the idea of walking" (Davidson, 1967b). In such cases, what becomes clear is that eliminating talk of events in natural language makes more difficult (or circuitous) a range of natural language modeling tasks that are relatively straightforward when ER is assumed.

This brief tour of difficulties inherited when ER is eschewed in favor of non-event descriptions of natural language suggests strongly to us that there is something clearly right-headed about assuming the existence of individual events. Davidson perhaps puts it best:

> "But the assumption, ontological and metaphysical, that there are events, is one without which we cannot make sense of much of our most common talk…" (Davidson, 1967a).

We'll take then as our starting point for defining PSEs the ER conception of events, though as we'll discuss later, we'll augment it somewhat to handle the representational requirements of PEL. The consequence of our adopting ER is that events are (ontologically speaking) perfectly acceptable PSTs, since by assumption they are individual entities which we can refer to directly. Hence, the PEL task is not "ontologically" suspicious because it requires reference to events and their locations; both "Event" and "Location" are proper PSTs.

We hope that the above (admittedly brief) discussion of Davidson's defense of ER demonstrates the plausibility of treating events as ontological entities. However, for those readers unconvinced or unsympathetic to this position, we'll note finally that, strictly speaking, even if ER is false, it is difficult to see how it poses serious problems for our inclusion of an Event PST for the purposes of PEL. This conclusion follows from the fact that PEL and other PSTL tasks are defined in the scope of *discourse* (we'll discuss this more below), and in the scope of discourse the set of types that are introduced to convey intended meaning can be treated as "brute". This is to say, whether or not ER is right, it's certainly true that in discourse events are treated as entities, and it is the semantic types that are discernible in discourse that is of concern here. To put it another way, though natural language discourse might someday be analyzed to eliminate event-talk altogether, challenging our introduction of the Event PST requires the supposition that we will stop using verbs like "walking" or "strolling" to describe walks or strolls, or introducing definite or indefinite articles to refer to events ("the walk I took yesterday"), or in many other ways using the machinery of language to refer to, as Davidson puts it, events as particulars. This seems unlikely and, at any rate, not of any immediate concern for our purposes here. We'll conclude then, by offering the following:

(1) We think ER is a perfectly defensible, and likely true, account of events that is straightforwardly consistent with the central aim of this work; namely, the definition and performance of PEL.

(2) On the supposition that ER turns out to be false (or that some readers believe it to be now), nothing substantive turns on the concession with regard to PEL. Our aim is the intelligent processing of natural language, and in so far as humans intelligently process language using (perhaps fictitious) concepts like events, so too can machines designed to approximate the same behavior.

This suffices, we hope, to put to rest any challenges to the conceptual viability of the PEL task with respect to our use of the event concept. We'll turn now to a discussion of the Event ontology that provides the representational requirements for performing PEL.

The Event Ontology

Event Classes

Recalling the analysis of the simple sentence "Smith climbed Everest" discussed above, we note that the "Davidsonian" analysis (Exists x) (climbed (Smith, Everest, x))

does not, strictly speaking, instantiate the climbing event, such that something like isa (x,Climbing) is explicit in the LF. We presume this is because Davidson is suspicious of introducing two sorts of events, instances and classes (although note his parenthetical remark to Chisholm in Events as Particulars, p. 183), when one might suffice. In this work, however, we'll worry little about this restriction: event instances and classes will figure prominently in the construction of the event ontology for PEL. We would be perfectly happy, for instance, with an LF that makes explicit mention of event classes like "Climbing" (the class of climbing events):

Exists(x) (isa(x,Climbing) ^ performedAction (Smith,x) ^
objectOfAction(Everest,x)).

We'll see later that the introduction of event classes makes possible the construction of type hierarchies, which make possible the computation of specific primary types by traversal of the hierarchies (see Chapter 4 for details). We turn now to our use of Event classes to perform PSEL.

Event classes are familiar enough. "Hurricane Katrina" is an instance of a class of **Hurricane** events, the "1999 World Series" is an instance of the class of **American Major League Baseball Games**, an IED explosion is an instance of a **Bombing**, and so

on. In the context of PEL, a discourse which mentions a particular event instance (e.g., *the* Hurricane Katrina) can also be said to mention *a* hurricane generally, and hurricanes themselves are subtypes of, say, **Natural Disasters**. Constructing an ontology of events in this fashion allow us to say of some discourse D about Hurricane Katrina that D should be classified as having a primary event **Natural Disaster**, or more specifically **Hurricane**, or (most) specifically **HurricaneKatrina**.

Individuation of Events

Davidson (1969) has noted that the question of whether two events are identical (or distinct) is a subspecies of the general problem of determining identity, and as such inherits the panoply of conundrums that questions of identity inevitably invite. As many have observed, events are often individuated by their causes, requiring the introduction of the concept of 'cause' and the conditions when, for instance, a particular cause could be cited for a particular event against a background of other possibly relevant factors to get a theory of event individuation off the ground (c.f., Salmon, 1998). Such considerations are outside the purview of the present work. For our purposes, a "minimalist" interpretation of event individuation includes only temporal and locative criteria; for any two events e and e', e = e' if and only if e occurs at the same time as e' and e and e' occur in the same place. Hence, HurricaneKatrina is a separate instance from HurricaneWilma

because Katrina occurred in August 2005, reaching the United States along the coast of Louisiana and Mississippi, and Wilma occurred in October 2005 reaching the United States along the coast of Florida. Davidson has shown that multiple change cases suggest that the present treatment is inadequate. However, such cases don't arise within the scope of PEL, and as such we'll leave such worries aside. With the admittedly minimal scheme we've adopted, all instances of classes of events in our knowledge base can be individuated. We turn now to Primary Specific Events.

Primary Specific Events

A Primary Specific Event (PSE) is necessarily an event in the Davidsonian sense discussed above, but it also must meet sufficiency conditions that are relative to the discourse where it is expressed (i.e., specificity and centrality). The discourse-relative nature of PSEs mean that, *inter alia*, the underlying event e (say, a hurricane) that gets assigned a PSE label for some discourse D may in fact fail to receive the label for some discourse D′. Hence, while hurricane events are what they are, in the objective ontological sense, PSEs are tied to particular discourses, such that the scope of any PSE is defined intra (not inter) discourse.

To give another example, an IED Bombing event refers to an improvised explosive device detonating somewhere (and at some specific time) in the world. Yet, for

some set of discourses all with at least one mention of an IED Bombing, it is possible that IED Bombing is not a PSE for any of the them, and also that some subset of the discourses license the inference to a PSE given the mention of an IED Bombing. As such, the concept of a PSE is logically and semantically distinct from that of an event; for the latter, wherever we see reference to an event, the properties of that event hold (the event is context-independent), and for the former, reference to any event is never sufficient to license attribution of a PSE. What is needed additionally is the satisfaction of a set of discourse-specific criteria; namely, that of meeting specificity and centrality given a particular discourse where a candidate event is mentioned. This suffices, we hope to explain how we intend to use the concept "event" and how events thus construed differ from PSEs.

Definition of Primary Specific Events

Given a set of event labels E such that for each $e_1,..,e_n$ we have a label for some event expressed in a discourse D, the PSE then is the e_k such that the event expressed is both central to D and most specific relative to other members of E. This specificity requirement forces upon the "flat" representation E some relational property where members of E can be compared for relative specificity. Ordinary subsumption cashes this out: e_k is most specific if for all other members of E meeting the centrality requirement,

it is not the case that e_k subsumes any of them. To represent PSEs, then, we simply translate labels for events in E into classes in a knowledge representation language, where class hierarchies can be explicitly defined. The subsumption check suffices for specificity given such a hierarchy of event classes (centrality must be computed, discussed in Chapter 4).

One unfortunate consequence of defining specificity for PSEs in terms of class subsumption is that now individual events expressed in discourse (e.g., an IED explosion) are represented as classes of events. Besides introducing a class/instance confusion into the representation, this conflation has the further unhappy result of disallowing PSEs as arguments to predicates in any language that requires instances for ground assertions. For example, if "IEDBombing" is defined as a class, then "primarySpecificEvent(D1,IEDBombing)" is an invalid triple for KR languages expecting individuals to bind the arguments for the primarySpecificEvent predicate.

Since however it is practically convenient (and ontologically sound) to represent events such as IED bombings as classes (where instances of IEDBombing would be particular IEDBombings distinguishable by specific information—who did it, where, when, etc.), an easy if perhaps inelegant solution to the class/instance problem is simply to construct corresponding instances for each class in the event hierarchy. This can be done by introducing a new name and simply asserting the new object as an instance of the class:

Class: IEDBombing

Instance: IEDBombingInst

However, perhaps an even better solution is simply to use a KR language without the first order constraint on binary properties, as is the case with the Resource Description Framework (RDF). This solution is particularly apt since, as will be shown in Chapter 4, the computational solution to PSEL offered in this work does not require complex inferences. We will discuss details of the KR language and the construction of classes and instances in the implementation section of this chapter.

A final consideration in this section is the size of the event hierarchy: how many classes must we introduce to event-cover NW data? The prima facie answer would seem to be quite a lot, since there are many possible events that can be expressed in ongoing newswire discourse reflecting happenings around the world. Our answer here is to define the top level event classes relative to the set of topics: for each topic t there are 3-5 top-level event classes. For instance, for the Accident topic we have a decomposition into "PersonAccidents", "VehicleAccidents", and "StructureAccidents". These classes are in turn subclassed, forming an accident event hierarchy and in general for each topic t_k we have an event hierarchy e_k with 3-5 nodes at each level. Hence, the number of events is at most (number of topics) * 5 * (depth of the hierarchy). Treated this way, the "topic-

relative" event hierarchies achieve adequate specificity for the purposes of selecting a PSE with a depth 5 or less hierarchy.

By way of example, the current accident event hierarchy is:

class(PersonAccident)

class(StructureAccident)

class(VehicleAccident)

subclass(AnimalAttack,PersonAccident)

subclass(SeaAnimalAttack,AnimalAttack)

subclass(LandMammalAttack,AnimalAttack)

subclass(LandReptileAttack,AnimalAttack)

subclass(Drowning,PersonAccident)

subclass(Falling,PersonAccident)

subclass(WeaponAccident,PersonAccident)

subclass(AccidentEnduringElements,PersonAccident)

subclass(AircraftAccident,VehicleAccident)

subclass(AirplaneAccident,AircraftAccident)

subclass(HelicopterAccident,AircraftAccident)

subclass(AutoAccident,VehicleAccident)

subclass(BoatAccident,VehicleAccident)

subclass(TrainAccident,VehicleAccident)

subclass(StructureCollapse,StructureAccident)

subclass(StructureFire,StructureAccident)

subclass(StructureExplosion,StructureAccident)

Given the representation outlined above, to perform PSEL it suffices to simply label or "tag" instances of NW Data with an instance of a PSE from the Event Hierarchy, given a topic. The actual computation of the most likely PSE for a given instance of NW data is the subject of Chapter 4; however the computational account rests on a prior successful determination by humans (specifically, the human annotators who create training data for the learning algorithm). If it turns out that humans encounter difficulties determining, unambiguously, a PSE for a newswire article, then we should expect that any algorithm will, at best, inherit the same limitations. Hence we need to consider the human case first. We'll treat this as an inquiry into the representation, rather than a question about the performance of humans on the PSEL task. In other words, we can assume that humans understand NW data, so the difficulties that humans may encounter identifying PSEs in such data are really questions about the nature of the problem and the representation strategy adopted. To this we turn next.

Challenges to Identifying Primary Specific Events in NW Data

We can identify at least three sources of ambiguity for the determination of PSEs in NW data. One, there is a problem identifying a PSE in discourse where the PSE seems to be a verbal or written statement offered by someone. There is, of course, a trivial sense in which newswire articles meet this condition: by definition, a news article describes a *report* of some event, which is often included in the article by way of citing the source of the information (… according to Gen. Smith). In this trivial case, we don't want to tag news articles with a "Stating" PSE (a "Stating" concept means that something is asserted by someone as true), since this would apply to the entire dataset and would therefore be uninformative. Yet, some news article do center on statements; the purpose of the article is to highlight the opinions, observations, or assessments of a particular person, such as when a general offers that violence is down in a particular region in a conflict. In this case, the PSE is the linguistic communication, the stating. The event has significance because of the person stating it (someone for whom their opinion is presumably of interest as news), and the content of the stating (a decrease in violence in a region of Iraq is a topic of interest to many people). The question becomes: what criteria are to be used to distinguish between statements that are PSEs, and ones that play a secondary and uninteresting role for the purposes of classifying the article? We call this the Linguistic Communication Threshold Problem, because the difficulty lies in determining when linguistic communication meets a "threshold" of importance and hence qualifies as a PSE in the discourse in which it appears.

Two, there is the problem of "multiple candidates". This happens when a discourse expresses multiple events but none of them clearly rank as more important or primary given the assumed discourse purpose. For instance, an article reporting a skirmish in Iraq may mention a gun fight, a rocket missile attack, and an IED explosion all as events constituting the skirmish. In this case, it is unclear which event is actually primary, and complex inferences about which candidate is somehow "primary" would seem to stretch the interpretation of the discourse beyond what was intended (i.e., by a journalist with the purpose of reporting what happened in a war zone).

Finally, there is the problem of vague or implied events. Newswire articles sometimes rely on assumed shared knowledge of prior events to discuss complex issues of continuing public interest such as nuclear energy, abortion, natural disasters etc. without clearly mentioning the context event, and not introducing any obvious PSE. Consider snippets from a Reuters article titled "Three Mile Island shows US nuclear risks, rewards":

> MIDDLETOWN, Pennsylvania (Reuters) - Four giant cooling towers loom over the Three Mile Island nuclear plant, reminders of the fears and hopes surrounding an industry that may help cut U.S. dependence on foreign oil.
>
> Two towers stand quiet, idle since a partial meltdown in a reactor almost 30 years ago in the nation's worst nuclear accident. Two others belch steam from an active reactor, providing cheap electricity to 400,000 homes.
>
> Unlike the Chernobyl disaster in Ukraine -- which will mark its 20th anniversary on April 26 -- no one died at Three Mile Island. But critics of

> atomic power raise concerns over potential terrorist threats to plants and say science has yet to provide an adequate solution for highly toxic nuclear waste. Three Mile Island owner Exelon Corp. now wants to extend its operating license as part of an industry program to keep all 103 U.S. nuclear reactors going beyond their standard 40-year licenses.
>
> New plants are also under consideration as companies hope to cash in on an expected 45 percent surge in electricity demand over the next 25 years and answer U.S. government calls to diversify sources for the world's top energy consumer.

What is the PSE? Plausibly, that Three Mile Island owner Exelon Corp. is seeking to extend its operating license. Yet, this seems not quite to meet the PSE criteria: is the purpose of this article to introduce and expound on this singular event? Consider the next sentence. Here, the prospect of building new plants—not reviving older facilities such as Three Mile Island—seems equally important given the broader context (the nuclear power option for today's energy needs). And, further in the article, we see this:

> Last month, Russia and the United States called for the world to embrace nuclear power to guarantee stable supplies of energy and cut emissions of harmful greenhouse gases.

Is this the PSE? It is difficult to say, because the real purpose of the discourse is to discuss this event and others (e.g., the status of the Three Mile Island plant) in the context of the debate about nuclear power given our current energy situation. To put it another

way, no one thing has happened; rather, a set of relevant events and considerations are offered by way of discussing the event context.

Resolution of Challenges

A number of heuristics can be introduced to resolve the challenges to identifying PSEs in NW data. First, with regard to the thresholding problem with linguistic communication, we can distinguish between statements made by people who are, in some sense, public figures whose views are of interest to the public, and statements made by relatively unknown or anonymous people. While the exact definition of a "public figure" may require further analysis, it is obvious enough when, for instance, the president gives the State of the Union address that the person giving the statement (speech) is of public interest and the communication act itself is therefore a PSE.

With regard to the multiple candidate problem, it suffices to note that the PSE classification is, like topic classification, exhaustive but non-exclusive. Hence, at least one PSE may be selected. In cases where two or more candidate PSEs have been identified without any plausible distinguishing criteria (as we have with linguistic communication), multiple PSEs can be attributed to the discourse. This will be true whenever a set of events having roughly the same import are listed by way of elaborating on a complex event occurrence. To use the example offered above, a newswire article about a skirmish in Iraq that lists the constituents of the skirmish (gun fight, rocket missile attack, IED explosion) gives us no good reason to select one of these events as the PSE, since they are all roughly of the same specificity, and in the context of skirmish

would seem to have the same import; consequently, we simply assert that the PSEs are the events reported.

The problem of vague or implied PSEs represents the most difficult of the three challenges, because here no straightforward heuristic suggests itself: highly context-dependent articles that presuppose significant familiarity with the subject matter and that fail to specify any obvious PSE are, from the standpoint of PSEL, inherently difficult. The problem is not entirely that candidate PSEs are implied, rather than expressed, in some NW articles (for instance, an article about the Iraq War may not explicitly mention the *Iraq War*, since it is assumed that the reader would know this by context). The problem is rather the vague or unspecifiable nature of some implied events, such as with the Three Mile Island article mentioned above. In such cases, it is reasonable to conclude that there is no plausible PSE: the article is not "event centered" in such a way that a PSE can be identified at all. In these cases, the only solution that suggests itself is simply to relax the exhaustive criterion for classification of NW data according to a PSE: no PSE can be identified. The article may still receive a topic (there are, to date, no known cases of NW data having no assignable topic), but there will be no PSE, and hence no location specified for the occurrence of the PSE.

This solution seems, in some sense, "against the rules" for performing PSEL on NW data, since the original task was to design a scheme that covered discourse of this type, and the present suggestion flatly fails to event-label some NW discourse whenever

there is no identifiable PSE. It this is a worry, we can approach things from the other direction and redefine the input such that "NW data" simply means all of the news discourse released by major content providers that in fact *does* have an identifiable PSE, as judged by a human reader. Then, article such as the Three Mile Island piece wouldn't constitute newswire discourse but rather some other form of news (say, opinion or background news about some topic of interest), and would fall outside the scope of the RFM* input constraints. This is a perfectly sound stratagem, but in the interests of casting the largest net over news discourse, at present the former strategy is preferred: news that fails to contain an identifiable PSE is simply not assigned one. In this case the query "What happened?" for this discourse will be left unanswered, or a stock response such as "No one event occurred, the article discusses a number of events related to the topic T…", where 'T' is the event context for the discourse, will be the reply. This has the virtue of preserving the generality of the input (all newswire articles that are or may be released by content providers) and reflects the fact that intelligent systems—like humans—should know how to answer a question with "unknown" when this is in fact the most reasonable response.

Chapter 4: PSEL Inference

As we have seen, PSEs are defined relative to hierarchies of classes, such that the most specific primary event is simply the most specific class in the event hierarchy that meets the primacy condition. For instance, consider the hierarchy:

VehicleAccident → AircraftAccident → HelicopterAccident

An article about a helicopter crash would also be an article about a vehicle accident, as well as an aircraft accident, but the PSE is "HelicopterAccident", the most specific of the classes meeting the primacy condition. A question left unanswered until now, however, is exactly how a machine such as TPE *computes* the PSE for an instance of NW data given the hierarchies defined. As we shall see, the PSE computation performed by TPE is in fact a species of the well known *document classification* task, where documents are assigned one of a set of classes on the basis of their content. By way of explaining the particular classification approach adopted for TPE to perform PSEL, we first review the basic document classification task.

Document Classification

Document classification is the task of assigning a document to a class based on document content. Typically, the class assignment is a function of the *relevance* of the document content to the class. For instance, a document with content that describes economic data may be assigned to an "Economy" class whose instances are documents describing economic issues. To automate the assignment of documents to classes given some set C of classes and documents D with content relevant to classes in C, a computational *classifier* is used. More formally:

> Given a set of document D and classes C, a classifier for a class c_i in C is a function $fi' : D \rightarrow \{0,1\}$ that approximates an unknown function $f_i: D \rightarrow \{0,1\}$ which computes the relevance of documents in D for the class c_i.

If learning methods are used to generate the classifier from data (to be discussed), the approximation function is "learned" from positive and negative examples, such that it gives a statistical estimation of the parameters of the underlying distribution of documents and classes. We say that the function represents a hypothesis learned

inductively (from provision of examples) about the relevance of a document d in D to a class c in C, which is often expressed as a real number 0 <= R >= 1 representing the probability that the class assignment is correct.

The analysis of document classification as assignment based on content *relevance* or *similarity* can be further clarified given the PSTL framework described in chapter 3. Given a document d and a set of classes corresponding to a PST (e.g., Event), a class c in C is relevant to d if the content expressed in d has PST c. For instance, if d is an article that is primarily about a train accident, then c = 'TrainAccident' is the class in C that is most relevant to d. Likewise for the other PSTs defined for PSEL (i.e., EventContext, Location).

Representation of Document Content- The Vector Space Model

The vector space model is an algebraic model for representing text documents such that the relevance of documents to each other or to a set of query terms can be determined mathematically. Modern document classification techniques very often use the vector space model (or the "term vector" model as it is also known) to represent documents to be classified. In the vector space model (hereafter VSM), each text

document is represented as a vector of *terms*, where each "term" is application dependent but is typically defined over the set of words in the document. Each term in the vector represents a dimension of the vector space V (e.g.,, the number of vectors of the basis of V). Different encoding schemes can be used but typically the occurrence of a term in a document D is given a nonzero entry in the vector for D.

The generation of vectors of terms from a corpus of documents is known as document *indexing*. Indexing occurs prior to assessing the relevance of documents and as such can be considered a preprocessing step in a document classification task. Typically, non content-bearing words (so-called "stop words") such as determiners or conjunctions (the, and) are not included in term vectors because they do not help determine relevance as they occur in nearly every document in the corpus (they are nondiscriminators). Given document vectors with suitable terms, some method of weighting the significance of the terms in the vectors is required.

Term Weighting

Intuitively, the terms (words) in a document are not all equally relevant to determining the relevance of the document to a semantic class (document classification), to other documents (clustering), or to a query (information retrieval or search). Three main considerations are typically cited as important to the determination of term

relevance (weight): the frequency factor, collection frequency factor, and length normalization. Frequency factor refers to the obvious observation that terms that occur more frequently in a given document tend to be more relevant to the determination of the meaning of the document. The collection frequency factor is relational: it is important in discriminating documents from each other. A common collection frequency factor is *inverse document frequency*, which assumes that the importance of a term is a function of the number of documents in which it occurs. The intuitive observation is: if a term occurs in a large subset of some corpus of documents, it is less important for discriminating the documents in the corpus.

Finally, length normalization techniques are typically employed during term weighting to account for the fact the documents with very large term vectors (i.e., documents of longer length) are more likely to be relevant than documents with shorter term sets. Since the length of the document itself is not indicative of relevance in the absence of other factors, normalization strategies are typically used to reduce the influence of length on the determination of relevance.

Given a set of terms, we can consider techniques that can be said to learn classifiers by provision of vectors of terms with the intended class (class label). This type of learning is known as supervised learning because the learning algorithm is provided the intended class for each vector instance. This framework is now, arguably, the

dominant approach to corpus based methods for analyzing text generally, and has proven very useful for the document classification task in particular (Mitchell, 1997).

The Supervised Machine Learning Framework

Following Mitchell (1998), we refer to *machine learning* as a general inductive procedure whereby a computer program is said to learn from examples (or "experience") E to improve its performance on a set of tasks T, according to some performance measure P. We introduce the following (minimal) terms when discussing machine learning in this sense. In *supervised learning*, all examples provided to the learning algorithm contain the desired "answer" or output. For instance, in document classification, an example is a (representation of) a document D, and when performing supervised learning the correct or most relevant class C given the content of D is provided to the learner. *Unsupervised learning* occurs without provision of class labels. *Semi-supervised learning* is a hybrid approach that uses a small subset of class labeled data (the so-called "seed") along with unlabelled data to bootstrap learning using the labeled data.

Conceptually, learning involves the generation of a function or hypothesis (called the target function) from a space of possible functions that best approximates a true (unknown) function that assigns a correct output to input from some underlying distribution we wish to model. The set of items over which a learning problem is defined

are known as *instances*, denoted by X. In the simple case, the unknown true function f can be any Boolean valued function defined over the instances X such that f: X → {0,1} (in the case of multiple classes, binary functions f are learned for each class). Instances where f(x) = 1 are positive instances. Instances where f(x) = 0 are negative instances. The ordered pair <x, f(x) > refers to an instance x and its class value f(x).

In supervised learning, positive and negative examples consisting of ordered pairs <x, f(x)> sampled from the underlying distribution are provided to the learner. The challenge in the supervised case is to estimate f given the set of examples provided. Let H be the set of all possible hypotheses the learning may consider regarding the nature of f (H is, in practice, determined by the human designer's choice of representation of instances). The successful learner finds some hypothesis h in H such that h(x) = f(x) for all x in X. That is, given some underlying distribution that is modeled by some unknown function f, the successful learner learns an approximation of f that succeeds in classifying unseen data, no in the training examples, drawn from the distribution.

In technical disciplines as with philosophical considerations in general, the problem of induction is of course relevant to the epistemic foundation of the supervised approach explicated above. After all, the hypothesis h learned from training examples has no other information other than provided by the examples in some finite subset of training data, and hence there is no analytic guarantee that application of h to unseen data not in the training data will succeed. The *inductive learning hypothesis* captures what we

typically observe nonetheless: a hypothesis that approximates the target function given a sufficiently large set of training examples tends also to approximate unseen examples from the underlying distribution. The exact details of how large the training set need be, and what type of representation to use when constructing instances, and various other statistical considerations such as how best to assess true performance, etc., are the subject of Machine Learning research considered as a subdiscipline of Computer Science. We will revisit some of these issues in this work as we proceed. We turn now to the use of supervised machine learning methods for the document classification task.

Supervised Machine Learning Approaches to Document Classification

A machine learning instance for the document classification task will be a representation of key features of a document that facilitate classifying the document according to a set of classes C. As mentioned previously, the classification is best described as a relevance relation between the content of the document and the intended semantics of the classes in C. As such, a suitable baseline representation for learning a classification function is simply the term vector model described earlier. We define an instance for learning a classification function as a vector V such that $V = \{t_1,\ldots,t_n)$ where each t as a term in some language L. Unlike in the non-learning case, a function such as cosine similarity for terms in V is not applied, but rather learned from the provision of

examples, where as discussed above an example is an instance together with the class output for the instance. In the case of document classification, positive examples are instances with the correct class, and negative examples are implicit in the sense that the instance paired with any other class labels are assumed negative. Consider the following (contrived) example:

E1: {classification, baseline, negative, positive, example, instance, learning, case,…,} ^ MachineLearning

The terms in V constitute content that is relevant to an article about machine learning, the provided class label. Example EI is thus a term vector representation of (we presume) some content discussing machine learning. The provided class label "MachineLearning" makes this a positive example.

The question remains: how do we *learn* a classifier (approximation function) from examples such as these? There are as one might expect scores of algorithms suitable for this learning task, but the general idea is that we must parameterize the terms in V such that numeric weights representing their importance or contribution to classifying the document represented by V can be determined. Then the function h will be used to *decode* unseen instances such that a most likely classification output can be

given. In what follows a few approaches will be discussed by way of explicating the general concept of learning a document classifier from data.

Naïve Bayes

The "Naïve Bayes" algorithm is a popular baseline algorithm that has been shown to perform well on many document classification tasks. The Naïve Bayes computation is as follows. A new instance is classified by assigning the most probable target value given a set of attribute values: Vmap = argmax $P(v_j| a_1,a_2,\ldots,a_n)$, where each q_k is an attribute value. For document classification, consider an instance space X consisting of an unbounded set of text documents (i.e., an unbounded set of grammatical sentences comprised of words and punctuation). To simplify exposition, suppose the learning task is to take some subset of text documents, the training set, with class labels "Politics" or "Out", where the class Politics denotes the set of documents with content relevant to the topic of Politics. Otherwise, the example is labeled "Out". The task is to learn from examples of this form a function h that accurately labels unseen text documents as "Politics" or "Out".

Given this learning task, Naïve Bayes represents documents by simply designating each word position in the document as an attribute whose value is the word at that position. For example, suppose a training corpus of 700 Politics document and 300

Out documents. Then the representation of a text document in the training corpus that begins with, say, "President Bush today…" and ends with "elections", will assign a_1=President, a_2=Bush, a_3=today and an="elections" where n is the final word in the document. The Naïve Bayes classification can then be given as:

$$VNB = \text{argmax } P(v_j) \text{ Multiply } i=1 \text{ to } n\ P(a_i|v_j)\ v_j \text{ in}\{\text{Politics,Out}\}$$

$$= \text{argmax } P(v_j)\ P(a1= \text{“President”}|\ v_j)\ P(as= \text{“Bush”}|v_j)\ P(a3 = \text{“today”}|\ v_j)\ \ v_j \text{ in}$$

$$\{\text{Politics,Out}\}\ \ \ldots\ P(an = \text{“elections”}|v_j)$$

VNB will return the classification of the document that maximizes the probability of observing the words found in the document, subject to an independence assumption (this is the "Naïve" in "Naïve Bayes"). For document classification the independence assumption can be stated as $P(a_1,\ldots,an|v_j)$ = Multiply a_1 to n $P(a_i|v_j)$: the probabilities for a word position are independent of the words in other positions given the classification v_j.

As Mitchell and many others have noted, the independence assumption for words in a discourse does not hold; the probability of observing the word "Bush" in position t will in general be greater given that the word "President" appears in position t-1. Nevertheless, in practice the naïve approach has been shown to work quite well on many document classification tasks (Mitchell, 1997). The simplifying assumption also

simplifies the complexity of training and classification, which would be an exponential time problem in its absence.

Maximum Entropy: A Log Linear Approach

While the naïve bayes algorithm is a simple (yet often very accurate) baseline for performing document classification, there are a number of supervised learning algorithms that have been shown to outperform naïve bayes on many common datasets used for testing document classification approaches (Joachims, (1998), Nigam *et al*, (1999)). We introduce here one such algorithm—Maximum Entropy—which the TPE system uses to classify text according to the PSEL task described in Chapter 3. The representation of documents with MaxEnt is much less restrictive than with Naïve Bayes: while Naïve Bayes makes use of only words and their positions, MaxEnt permits inclusion of any arbitrary word features that can be extracted from words in the document. To give one simple example: words occurring in the first paragraph of a document, or in the title of a document, can be treated as separate features. Likewise, collocations or bigrams (or trigrams) can be specified as features, and in general any transformations or groupings of the words (terms) in a document that seem promising for aiding classification can be specified. Exactly how MaxEnt makes use of features to classify documents is the subject of the next section.

Maximum Entropy: Assuming Uniform Models

Maximum Entropy is a species of probability distribution estimation technique (i.e., a statistical modeling approach) that was first applied to problems in statistical physics, and more recently has been widely used for a variety of natural language tasks, including language modeling, text segmentation, part-of-speech tagging, and prepositional phrase attachment (Nigam *et al*, 1999). The basic principle of MaxEnt is simple: model only what is known, and assume nothing about that which is unknown. To put it another way, the principle simply prescribes that we choose a statistical model consistent with all known facts, but that is otherwise as uniform as possible. This uniformity constraint is of course not specific to questions of statistical modeling; as early as the 18th century Laplace introduced the concept by (famously) declaring that, when reasoning about the world, we should consider two events equally likely when one has no information to distinguish them. The epistemic basis for MaxEnt is thus part of the historical (idealized) process of doing science generally.

Following Berger *et al*, (1996), we introduce MaxEnt as a statistical modeling technique by noting that, like with use of other statistical modeling techniques, application of MaxEnt requires the satisfaction of two core tasks. One, determine a set of statistics that captures the behavior of a random process. Two, given the statistics,

determine an accurate model of the process such that the future output of the process can be predicted. As Berger *et al* (1996) note, the first task is that of selecting features (i.e., "feature selection"), while the second is that of model selection (finding the optimal model of some data given the features selected from the first task). In what follows, we'll consider Maximum Entropy classification directed specifically at the text classification task relevant to the present work.

Using Maximum Entropy for Text Classification

Given a text classification task using the MaxEnt algorithm, we want to compute $p(y|\mathbf{x})$, where y is a member of some set of class labels $Y:\{y_1,\ldots,y_n\}$ and **x** is a vector of features derived from analysis of the target document D. In the base case, the feature vector is the set of words from D. The task is to return the label in Y that is most relevant to D subject to x, where each x_k in **x** is a constraint on the model. Such constraints constitute knowledge of the properties of the random process to be modeled. In the absence of any such constraints (features), uniformity is assumed (the Maximum Entropy principle). The following example adapted from Berger, Pietra, et al illustrates the use of Maximum Entropy for Text Classification (TC).

Consider a Maximum Entropy approach to classifying articles into a set of news event classes (i.e., to event labeling NW data). The MaxEnt model p assigns to each

previously unseen news article ***f*** an estimate ***p(f)*** of the probability that the human reader would choose ***f*** as an event class for the news document. For illustration purpose we will use a subtree of the Event Ontology concerning itself with destruction events but the approach can be generalized to include hierarchical event classification for the entire tree. We choose the set of class labels Y from our Event Ontology such that the TC task is defined for NW input, with PST Event classes as output. Choosing a subset of destruction events, we have:

Y = { IEDBombing, AirplaneCrash, Hurricane, RocketMissileAttack, SuicideBombing }.

The uniformity constraint on **p** is:

p(IED Bombing) + p(Airplane Crash) + p(Hurricane) + p(Rocket Missile Attack) + p(Suicide Bombing) = 1

This equation represents our first statistic of the process; we can now proceed to search for a suitable model that obeys this equation. There, of course, an infinite number of models **p** for which this identity holds. One model satisfying the above equation is *p (IED Bombing)* = 1; in other words, the model always predicts *IED Bombing.* Likewise, a model that predicts *Hurricane* with a probability of 1/2, and Airplane Crash with a probability of ½ will obey the uniformity constraint as well. Both models, however, seem *ad hoc* and unjustified: knowing *only* that a human reader always chooses from among these five destruction events, on what epistemic grounds is either probability distribution justified? Put another way, the two models assume more than we actually know about

the destruction event classification process. Our knowledge of the process at this point includes only the observation that the human reader will choose exclusively from among the five event types provided; hence, the intuitively appealing model is one with a uniform distribution:

$$p(\text{IED Bombing}) = 1/5$$

$$p(\text{ Airplane Crash}) = 1/5$$

$$p(\text{ Hurricane}) = 1/5$$

$$p(\text{ Rocket Missile Attack}) = 1/5$$

$$p(\text{ Suicide Bombing}) = 1/5$$

Now suppose we know from sample data that news articles are labeled *IED Bombing or Airplane Crash* with frequency 30%. We include the information by adding it as a model constraint, thus extending our original uniformity constraint to a set of two:

$$p(\text{IED Bombing}) + p(\text{Airplane Crash}) = 3/10$$

$$p(\text{IED Bombing}) + p(\text{ Airplane Crash}) + p(\text{ Hurricane}) + p(\text{ Rocket Missile Attack}) + p(\text{ Suicide Bombing}) = 1$$

There remain a large number of probability distributions consistent with the two constraints. In the absence of any other knowledge, a reasonable choice for **p** is again the

most uniform; that is, the most uniform distribution of probabilities, subject to the constraints:

$$p(\text{IED Bombing}) = 3/20$$

$$p(\text{ Airplane Crash}) = 3/20$$

$$p(\text{ Hurricane}) = 7/30$$

$$p(\text{ Rocket Missile Attack}) = 7/30$$

$$p(\text{ Suicide Bombing}) = 7/30$$

In this manner we can continue to specify constraint on **p**. For instance, from inspection of labeled NW data (by "labeled" we mean a set of examples that have been set aside and assumed "gold standard", or correctly classified) one might observe that in labels *IED Bombing* or *Hurricane* were applied to half of the NW data. The third constraint is then given by $p(\text{IED Bombing}) + p(\text{Hurricane}) = 1/2$, and the set of constraints is now:

$$p(\text{IED Bombing}) + p(\text{Hurricane}) = 1/2$$

$$p(\text{IED Bombing}) + p(\text{Airplane Crash}) = 3/10$$

$$p(\text{IED Bombing}) + p(\text{ Airplane Crash}) + p(\text{ Hurricane}) + p(\text{ Rocket Missile Attack}) + p(\text{ Suicide Bombing}) = 1$$

Finding the most uniform **p** satisfying this set of constraints is now not obvious, and to determine the most uniform **p** we now turn to the details of classification using the MaxEnt approach.

Maximum Entropy Modeling

In the simple example above we considered a random process that produces an output value *y*, a member of a finite set ***Y***. For Text Classification on NW data, we say that the process generates a class label y for each instance of NW data, an element of the set

Y: *{IED Bombing, Airplane Crash, Hurricane, Rocket Missile Attack, Suicide Bombing}.*

In generating y, the process may be influenced by complex contextual information x, a member of a finite set X. This contextual information will in general come from properties of the input (the set of NW text instances); in the example above constraints were determined only by simple statistics such as observed frequencies. To give a few straightforward examples, the information could include the words in the title of the news article, or the words in the first n paragraphs of the article (in either case we exploit a connection between the semantics of news article content and various properties of written news, like the connection between title words and article content).

Our task is now to *automatically* construct a stochastic model that accurately represents the behavior of the random process. The model will estimate the conditional probability that, given a context x, the process outputs y. We will denote by ***p(ylx)*** the probability that the model assigns to y in context x. (With a slight abuse of notation, we will also use ***p(ylx)*** to denote the entire conditional probability distribution provided by the model, with the interpretation that y and x are placeholders rather than specific instantiations. The proper interpretation should be clear from the context.) We will denote by **p** the set of all conditional probability distributions. Thus a model *p(y[x)* is, by definition, just an element of **p**.

Training Data

To study the process, we observe the behavior of the random process to collect a large number of samples (x_1,y_1), (x_2, y_2) . . . . . (X_N, Y_N). In the example we have been considering, each sample would consist of a set of text features containing content words, title words, et cetera, together with the label y in Y that the process produced. We assume at this point that such training samples have been generated by a human expert presented with a news article and asked to choose the best class label given the set of labels Y. The empirical probability distribution $\tilde{p}$ of the training examples is given by:

$$\tilde{p}(x,y) \equiv \frac{1}{N} \times \text{ number of times that } (x,y) \text{ occurs in the sample}$$

Typically, a particular pair (x,y) will either not occur at all in the sample, or will occur at most a few times, resulting in a sparse distribution.

Statistics, Features and Constraints

Our goal is to construct a statistical model of the process that generated the training sample $\tilde{p}(x,y)$. The building blocks of this model will be a set of constraints on the training sample. We noted earlier that observations of frequencies (i.e., simple statistics) that are independent of context can be used to formulate constraints; we also include statistics that depend on the conditioning information x. To take one example, we might note that, in the training sample, when the article title contains the phrase "road side bombing", the article is labeled as "*IED Bombing*" with frequency 9/10. To express this statistic as a constraint, we introduce the indicator function:

f(x,y) = { 1 if *y = "IED Bombing"* and the title of the article contains phrase *"road side bombing"*

{ 0 otherwise

The expected value of f with respect to the empirical distribution $\tilde{p}(x,y)$ is exactly the static we are interested in. We denote this expected value by:

$$\tilde{p}(f) \equiv \sum_{x,y} \tilde{p}(x,y) f(x,y) \tag{1}$$

We can express any statistic of the sample as the expected value of an appropriate binary-value indicator function ***f.*** We call such function a feature function or feature for short.

When we discover a statistic that we feel is useful, we can acknowledge its importance by requiring that out model accord with it. We do this by constraining the expected value that the model assigns to the corresponding feature function ***f***. The expected value of ***f*** with respect to the model $p(y|x)$is

$$p(f) \equiv \sum_{x,y} \tilde{p}(x) p(y|x) f(x,y) \tag{2}$$

where $\tilde{p}(x)$is the empirical distribution of x in the training sample. We constrain this expected value to be the same as the expected value of ***f*** in the training sample. That is, we require

$$p(f) = \tilde{p}(f) \tag{3}$$

Combining (1), (2) and (3) yields the more explicit equation

$$\sum_{x,y} \tilde{p}(x)p(y|x)f(x,y) = \sum_{x,y} \tilde{p}(x,y)f(x,y)$$

We call the requirement (3) a constraint equation or simply a constraint. By restricting attention to those models *p(ylx)* for which (3) holds, we are eliminating from consideration those models that do not agree with the training sample on how often the output of the process should exhibit the feature f.

To summarize, we now have a means of representing statistical phenomena inherent in a sample of data (namely, $\tilde{p}(f)$), and also a means of requiring that our model of the process exhibit these phenomena (namely $p(f) = \tilde{p}(f)$).

The Maximum Entropy Principle

Suppose that we are given *n* feature functions *fi,* constituting a set of constraints believed important to modeling the target process. We would like our model to accord with these statistics. That is, we would like ***p*** to lie in the subset ***C*** of **p** defined by:

$$\mathcal{C} \equiv \left\{ p \in \mathcal{P} \mid p(f_i) = \tilde{p}(f_i) \quad \text{for } i \in \{1, 2, \ldots, n\} \right\} \tag{4}$$

Among the models p E C, the maximum entropy concept dictates that we select the most uniform distribution. But now we face a question that has become complicated by the complexity of our set of constraints: what now is a "uniform" distribution? The mathematical measure of the uniformity of a conditional distribution $p(y|x)$ is given by the conditional entropy equation:

$$H(p) \equiv -\sum_{x,y} \tilde{p}(x) p(y|x) \log p(y|x) \tag{5}$$

Given a set $\mathcal{C}$ of allowed probability distributions, choosing the most uniform distribution now reduces to the task of choosing the model $p_\star \in \mathcal{C}$ with maximum entropy $H(p)$:

$$p_\star = \underset{p \in \mathcal{C}}{\operatorname{argmax}} H(p) \tag{6}$$

It can be shown that $p_\star$ is always well-defined; that is, there is always a unique model $p_\star$ with maximum entropy in any constrained set C .

Performance and Accuracy Considerations for MaxEnt on Text Classification

Nigam *et al* (1999) note that the maximum entropy technique as described above has been shown to reduce classification error by more than 40% when compared with Naïve Bayes. In addition, Maximum Entropy models are relatively inexpensive to construct (i.e., train from data), having a complexity bounded by the number of training instances n multiplied by the set of features m. Furthermore, as noted above MaxEnt does not suffer from independence assumptions like Naïve Bayes, and more complex features like bigrams or trigrams (in general, n-grams) can be incorporated into MaxEnt models. "Decoding", or returning the distribution of probabilities given a new instance (i.e., while applying a trained model to classifying new instances) is also inexpensive, requiring only a linear time search to return the best distribution given a new instances.

The technique, however, can suffer from the well-known phenomenon of "over fitting", where the model constructed from features in a training set does not model well the actual process intended but rather "over fits" to properties of the training set that are not representative (Nigam et al (1996), see also Mitchell (1997) for general

discussion of over fitting). Such overfitting is a general problem with empirical approaches to classification and other tasks; in the case of MaxEnt, a smoothing factor can be added (such as a Gaussian prior) that reduces the overfitting bias whenever the training sample is either too small or in some other way is not representative of the actual process to be modeled. In any case, published results on the superiority of MaxEnt to traditional classification techniques like Naïve Bayes, as well as the inherent flexibility of the approach to specifying features given knowledge of a particular process (e.g., classifying text according to the content and other features of text articles) make MaxEnt a suitable choice for the present investigation.

We note, however, that the above discussion has been generic, in the sense that we've unpacked the MaxEnt approach in the context of TC generally. For purposes of performing the subset of interest in this work, classification of Primary Specific Events or in general PSTs of interest (the PSTL task), we'll need a more powerful approach that facilitates, specifically, finding labels (concepts) that satisfy the primacy and specificity constraints discussed in Chapter 3. It is to such approaches that we now turn.

Hierarchical Learning

The DC approaches discussed thus far are informally known as "flat classification" approaches, since the label set constituting the classification targets are not related to each other but are simply a list of available classes for use in a DC task. Though, as we have seen, significant differences exist in flat classification approaches at the level of algorithms, all flat document classifiers, whatever algorithm they implement, have well-understood limitations. Namely, DC systems that are intended for large domains (e.g., the domain of newswire articles) must consider hundreds—perhaps thousands—of classes, returning a best label for one of them. For pure computational reasons this can become prohibitive; also, the accuracy of "best label" classification results given very large numbers of classes can suffer.

In contrast, hierarchical classification techniques have gained popularity in the last decade as the limitations to flat classification approaches have become more evident (Dumais and Chen, 2000). In hierarchical classification, the set of classes C:$\{c_1,..,c_n\}$ are *semantically linked*, and hence the probability that a document d has class ck can be informed by consideration of probabilities for classes in which it is linked. More specifically, if it is known that some document d should be assigned, with high probability, to some superclass c_k, the inference that d also be assigned its subclass c_{k-1} can make use of the high probability assignment of c_k. This semantic linkage between classes has been exploited by, for instance, document classification systems for the biomedicine domain, where hierarchical classification techniques are frequently employed (cf. Kiritchenko et al, 2005).

Although a number of hierarchical approaches or paradigms have been developed, we consider here *top down* hierarchical learning: the approach taken for the PSEL task. With the top down approach, classification proceeds by traversing a class hierarchy, descending to a subclass only if a classification probability for the superclass is above some threshold (typically empirically determined).

For example, a classifiers that assigns, with high probability, the class label ViolentOngoingConflict, will traverse to the subclasses (specific conflicts), and may then assign with high probability (i.e., meeting some empirically determined threshold) a class such as DarfurConflict. To take another simple example, if one knows that a document is about "Sports", one needn't bother checking all of the classes that are not subclasses of Sports; football, baseball, tennis, etc., are the only remaining relevant classes. In this manner both the computational load on systems performing DC as well as their accuracy can be reduced. Conceptually, we see that the *representational* features of a DC task can inform and guide the *inferential* (classification) task. "Flat" document classification techniques as have been seen in decades of work on DC don't exploit this powerful connection; as such, they are in general less well suited for DC on corpora such as NW data. Indeed, the specificity requirement with PSEL, suggests strongly that representation and inference should be linked hierarchically, as class "specificity" will be understood in terms of a hierarchy. As such, hierarchical classification approaches seem ideal for the PSEL task.

Outline of the Inference System for PSTL

In what follows, we will describe the implemented Inference System for TPE. TPE uses supervised machine learning techniques to train a classification model that, given an unseen document, will output a real number representing the probability that the document should be assigned a particular label, where the label corresponds to either a class or instance given the set of ontologies and instances described previously.

TPE Primary Subsystems

Training

To train a new set of models for classification (aka "production"), the system learns from examples in a *training corpus***,** where each document in the corpus has been manually annotated using terms from the ontology and knowledge base resources (hereafter, "knowledge resources"). In the current implementation, each document in the training corpus is annotated with event, person, organization, topic, issue, and location information. Example 1 illustrates the annotation information for the text content of an online news story from USA Today (annotations in bold font represent terms from the knowledge resources):

Example 1

Specter_asks_Gonzales_to_speak_on_domestic_spying_program

Event ="Eavesdropping"

Topic=”Politics"

Issue="Privacy"

Person=”AlbertoGonzales"

Organization=”National Security Agency"

Region=”NorthAmerica"

Country=”UnitedStates"

City=“Washington, D.C.”

Title=”Specter_asks_Gonzales_to_speak_on_domestic_spying_program”

Source=“USATODAY”

Date=“010806”

Type=“TopStoryWebPage"

As discussed previously, TPE uses a *top down hierarchical training* (HT) algorithm to train models that correspond to concepts in the knowledge resources. A description of the training process using the hierarchical traversal algorithm follows.

Hierarchical Training

TPE inputs the annotated files from the training corpus and extracts the trainable text from each file. It then groups the parsed text instances into "bundles" which correspond to values of the properties asserted about the text in the training corpus. For instance, the system would add the file in Example 1 to a **Topic** bundle with value "Politics". Separately, it would add the file to a **Issue** bundle with value "Privacy", an **Event** bundle with value "Eavesdropping", an **Organization** bundle with value "National Security Agency", a **Person** bundle with value "AlbertoGonzales". The location of the event is currently not trained using the training system of TPE; as a heuristic, the report location from the article is used as the event location.

The HT algorithm then traverses the concept hierarchies defined in the knowledge resources to train classification models for the bundles. Specifically, bundles with values of instances in the knowledge resources are visited first. Then classes of the instance are visited in order from *least to most general*. For each bundle, the algorithm checks if the

number of articles in the bundle meets or exceeds a threshold number for training (the threshold used currently is >= 30 articles.) If the bundle count is greater than or equal to the threshold, a classification model is learned for this bundle. For example, if there are at least 30 articles with label “Eavesdropping”, an “Eavesdropping” classification model is produced by the system.

Next, the classes and superclasses of the instance are visited. Note that the class bundle will contain more trainable text than the instance whenever there is more than one instance bundle for the class. Thus if the instance bundle did not meet the threshold, it is possible that the class bundle for that instance will meet the threshold, since the class bundle can contain text from more than one instance bundle. For example, the **GovernmentSurveillance** class bundle may contain articles from the **Eavesdropping** instance bundle as well as the **Spying** instance bundle. In this case a “GovernmentSurveillance” classification model is trained. The HT algorithm continues traversing the hierarchy in this manner until all bundles are trained up to the upper class cut off.

In addition to the HT procedure for standard models just described, “simple” classification models are learned for any instance bundles with at least 2 articles. The primary purpose of simple model training is to generate classification models for instance bundles not meeting the threshold. Such models, by themselves, may be expected to produce less accurate classifications (because of issues with overfitting due to sparse

data), but are made more accurate by using the hierarchical approach. The use of these "simple" models will be explained in the classification section.

After the HT algorithm has traversed all of the hierarchies from the knowledge resources that the user has specified, the metadata descriptions of the classification models are generated (for expository purposes a simplified syntax is provided here):

Description of: TopicClassifier

Name="TopicClassifier"

Location=../data/models

ClassifiesArticlesByType=Topic

ClassierType=MaxEntClassifier

These descriptions are used to load the trained models back into the system and to apply them to generate label descriptions during the classification phase.

Classification

To classify new data after training, the system uses the *hierarchical classification* (HC) algorithm. The HC algorithm works as follows. For each production text (article), each standard model generated during training classifies the new text. For example, the Event model produces an array of probabilities representing the likelihood that the text should receive a label representing an instance or class in the Event ontology. The model also produces an array of labels representing instances or concepts in the knowledge resources, where each label has one value from the value array (i.e., the models produce an array of label-value pairs). For each value meeting a lower bound threshold (this threshold is determined empirically by the user), the value and label for the value are saved for later consideration. If a simple model was generated during training, only the highest value is saved, and then only when it meets a user specified threshold. For instance, if the highest value for the simple model is 0.85, but the user specified threshold is .90, no simple model will be saved.

Next, the HC system performs the following sequence of steps. First, if there is no simple model label-value pair saved, then HC checks whether there is more than one label-value pair saved for a standard model. This would mean that there is more than one value from classifying the text that met the lower bound threshold.

1) If there is only one label-value pair, the value is checked against the *acceptable threshold*, which is always greater than the lower bound. For example, the acceptable threshold may be greater than or equal to 0.70 for

classifying news article text, while the lower bound is less than 0.50. (It is important to note that the acceptable threshold is determined empirically and can vary depending on application requirements and other factors that are extrinsic to the HC algorithm.) If the acceptable threshold is met, the label for this value is returned as the best label for the new text (recall that there can be more than one "best label" depending on the number of ontologies used for classification. Typically there will be multiple). If the acceptable threshold is not met, the algorithm returns a "NoGoodLabel" result and terminates.

2) If there is more than one label, the label with the greatest value is considered. If the value for this label meets the acceptable threshold, it is then compared with the rest. Otherwise the "NoGoodLabel" returned. A description of algorithm for labels meeting the acceptable threshold follows. (We refer to the label with the greatest value as GL (greatest label), and any label from the remaining labels as RL (remaining label).)
    a. For each RL, the algorithm retrieves the corresponding instance or class from the knowledge resources
    b. The algorithm checks whether RL corresponds to an instance or subclass of the class that corresponds to GL.
    c. If yes, GL is set to RL.
    d. When the RLs have been checked, the HC algorithm terminates with an RL representing either the most specific subclass of GL or an instance of

GL (or, if no instances or subclasses for GL are found, it terminates with the original GL).

If a simple model label-value pair has been saved, the standard model results are returned as described above. If the standard model result did not meet the acceptable threshold, then HC terminates as before. If the HC algorithm produced a GL corresponding to an instance, then the algorithm terminates. Otherwise, the simple model label and the GL label are checked. If the simple model label corresponds to an instance of the class that corresponds to the GL label, then the simple model label is returned. Otherwise, the GL label is returned.

Implementation

The hierarchical text classification system described previously has been fully implemented in the Java programming language and trained using a corpus of about 20K news articles (the NW dataset). It has been tested on over 1K unseen text articles to produce annotated (classified) text mapping to six different PST hierarchies. As mentioned previously, all knowledge representation was done using the RDF/RDF(S) ontology language. The subsumption checking for the HT and HC algorithms, and other ontology management duties are performed using HP Labs' Jena2 framework. The opensource MALLET machine learning toolkit was used for the training and classification performed during operation of the HT and HC algorithms (McCallum, 2002). A maximum entropy algorithm was used, with a term (word) features minus stop words. Initial results using a 10-fold cross validation on 90% train, 10% test splits of the NW corpus have shown greater than 90% performance on Topic and Issue PSTs. Event performance has been sporadic, ranging from 92% for "Harm" concepts to 50% for other concepts (e.g., business) where there is either inadequate training data or many specific concepts that the algorithm most consider. Given the variation of results, a "topic constraint" was introduced, which resulted in a significant decrease in Event classification variation.

Constraining PSEL by Topic

In Chapter 3 we introduced the notion of "topic relative" events, in the sense that for each topic (event context) a decomposition of event classes is specified. We can, then, separate the classification task into two tasks:

1. Given an article instance, assign the most likely topic to the article (topic identification)
2. Given an article instance with an assigned topic, determine the most likely event classification for the topic.

In other words, we first identify the most likely topic. Then, given this assignment, we search the subtree of the Event ontology that corresponds to this topic, returning the most likely Event class. To illustrate, consider the following snippet from AP News:

> **5 killed in helicopter crash off England**
>
> A helicopter carrying seven people crashed off the northeast English coast Wednesday night, killing at least five and setting off a major sea rescue, officials said. A pair of Royal Air Force helicopters, two lifeboats and other vessels were searching the cold waters of Morecambe Bay just east

> of the Isle of Man. Lancashire police and the Maritime & Coastguard Agency said five bodies had been found and the search was continuing for the other two people aboard. The Maritime agency said contact with the helicopter was carrying five gas rig workers and two crew was lost around 6:40 p.m.

The HC algorithm first classifies according to Topic, returning best label Accident. It then searches the subtree for Accident events: VehicleAccident → AircraftAccident → HelicopterAccident, terminating with the most specific event (subject to the threshold, currently set at 70% Fmeasure): HelicopterAccident. Thus the HC algorithm of the TPE system terminates with Topic and PSE. Currently, the TPE system returns PSEs given for the Accident topic at accuracy 92% (cross-validated F-measure using 187 test articles).

Other Performance Considerations – Feature Selection

Currently, the TPE system uses a **baseline feature set,** which is simply the set of words in the text of the news article, removing stop words. ("Stop words" are common close class words that are deemed irrelevant for classification since they typically appear in most every document regardless of the documents class: "and", "not", "or", and "but" are a few examples.) Thus the baseline feature set is simply a vector of words found in the document without regard to title words, position

(whether found in first paragraph, next paragraph, etc.) or any other consideration apart from simple occurrence in the article content.

Intuitively, the baseline feature set is suitable for a PSEL task because the words (the open class words such as nouns, verbs, adjectives, and adverbs) found in the article will tend to indicate relevant classes: articles with occurrences of "blast", "investigation", "suspect", "victims", "suicide", "bomber", etc., are likely relevant to classes such as SuicideBombing. However, inspection of instances of news such as the example above reveals that important features are left out of the baseline set. For instance, words in the title of the article are strongly indicative of the content of the article. Likewise, reported news tends to "get to the point" fairly quickly for a various pragmatic reasons (news readers typically wish to "get the gist" of the article quickly). Hence, words found in the first or second paragraph may have a special status with regard to the topic of the article. It would be helpful to capture this additional information as features. In future versions of the TPE system, an augmented feature set using, specifically, title words and perhaps paragraph boundary detection will be investigated. Nonetheless, the baseline feature set using the hierarchical classification system (HC), tested on (minimally) topics and events, has returned encouraging F-measure scores and suggests strongly that the basic approach taken is successful. Further work will include additional PSTs and, as mentioned, investigation of additional features to further increase the classification performance of the system.

Results

Topic

Training / Testing Data Used

PST	Number of articles
AccidentsAndDisasters	176
Politics	300
Business	298
Technology	60
Health	295
Science	265
Crime	72
Economy	190
Religion	90

Total Articles 1746

Total Articles Trained 1571 (90%) – Estimated time 25 second(s) to train each iteration

Total Articles Tested 175 (10%)

Result of 10 iterations of Training with Title Feature

Iteration	Accuracy	Precision	Recall	F1
1	0.8514285714285714	0.875	0.875	0.875
2	0.8685714285714285	0.8333333333333334	0.8333333333333334	0.8333333333333334
3	0.8228571428571428	1.0	1.0	1.0
4	0.8514285714285714	1.0	0.8333333333333334	0.9090909090909091
5	0.8285714285714286	0.875	0.7777777777777778	0.8235294117647058
6	0.8	0.8571428571428571	0.6666666666666666	0.75
7	0.8742857142857143	0.75	0.6	0.6666666666666665
8	0.8571428571428571	1.0	0.5454545454545454	0.7058823529411764
9	0.8171428571428572	0.8571428571428571	0.75	0.7999999999999999
10	0.8342857142857143	1.0	0.7777777777777778	0.8750000000000001
AVG	**0.840571429**	**0.904761905**	**0.765934343**	**0.823850267**

Result of 10 Iterations of Training with No Title Feature

Iteration	Accuracy	Precision	Recall	F1
1	0.8285714285714286	1.0	0.9090909090909091	0.9523809523809523
2	0.8971428571428571	0.3333333333333334	0.9090909090909091	0.8695652173913043
3	0.8457142857142858	0.8	0.6666666666666666	0.7272727272727272
4	0.8685714285714285	0.875	0.875	0.875
5	0.8742857142857143	1.0	0.8181818181818182	0.9
6	0.8514285714285714	0.8571428571428571	0.75	0.7999999999999999
7	0.8914285714285715	1.0	0.8888888888888888	0.9411764705882353
8	0.8228571428571428	1.0	0.8181818181818182	0.9
9	0.8742857142857143	0.8	0.8888888888888888	0.8421052631578948
10	0.8457142857142858	0.75	0.42857142857142855	0.5454545454545454
AVG	**0.86**	**0.898015873**	**0.795256133**	**0.835295518**

Issue

Training / Testing Data Used

PST	Number of articles
GlobalWarming	50
Immigration	58
BirdFlu	71
GayMarriage	36
DrugLegalization	39
HealthCare	37
Globalization	31
DeathPenalty	46
AnimalRights	42
Abortion	64
AIDS	32
ReligiousExtremism	31

Total Articles 537

Total Articles Trained 483 (90%) – Estimated time 6 second(s) to train each iteration

Total Articles Tested 54 (10%)

Result of 10 iterations of Training with Title Feature

Iteration	Accuracy	Precision	Recall	F1
1	1.0	1.0	1.0	1.0
2	0.9814814814814815	1.0	1.0	1.0
3	0.9814814814814815	1.0	1.0	1.0
4	0.9814814814814815	1.0	1.0	1.0
5	0.9814814814814815	1.0	1.0	1.0
6	0.9814814814814815	1.0	1.0	1.0
7	0.9814814814814815	1.0	1.0	1.0
8	0.9814814814814815	1.0	1.0	1.0
9	1.0	1.0	1.0	1.0
10	1.0	1.0	1.0	1.0
AVG	**0.987037037**	**1.0**	**1.0**	**1.0**

Result of 10 iterations of Training with No Title Feature

Iteration	Accuracy	Precision	Recall	F1
1	1.0	1.0	1.0	1.0
2	0.9814814814814815	0.5	1.0	0.6666666666666666
3	0.9444444444444444	1.0	1.0	1.0
4	0.9814814814814815	1.0	1.0	1.0

5	0.9814814814814815	1.0	1.0	1.0
6	0.9629629629629629	0.75	1.0	0.8571428571428571
7	0.9629629629629629	1.0	1.0	1.0
8	0.9814814814814815	1.0	1.0	1.0
9	0.9814814814814815	1.0	1.0	1.0
10	0.9814814814814815	1.0	1.0	1.0
AVG	**0.975925926**	**0.925**	**1.0**	**0.952380952**

Chapter 5: The Problem of Odd News

"Odd" stories, in the context of news, are stories about events that are humorous, or ironic, or just plain weird. Odd stories (hereafter we will refer to discourse of this type as "Odd news") present a couple of challenges to the approach outlined in this work. One, the content expressed by Odd news does not generally conform to a topic scheme, as outlined in Chapter 3. Odd stories are not intelligible as having a "Politics" or "Science" or "Business" context; a story about a man who is robbed for a bag of tacos, entitled "Your tacos or your life!", is not a story with context "Crime", even though the primary event in the story might plausibly be a criminal act (robbery). But it was the usefulness of topics such as "Crime" to provide a context for answering questions about what happened—a context for the primary specific event—that motivated their inclusion in the representation scheme. With Odd news, however, the topics fail to provide this context—it is not the robbery itself that constitutes the point of the story, but the robbery for tacos, and this meaning is completely lost whenever Crime is given as the context, and "Robbery" as the primary specific event. In short, Odd news cannot be classified correctly given the assumptions of TPE.

A plausible response to the problem of Odd news might be, that we simply need an "Odd" topic, or that we need some new set of topics that can provide context for stories that express humorous or ironic messages. This is, however, not possible given

the assumptions of TPE, a point that we hope to make clear in this chapter. The conclusion reached—based both on empirical considerations and on consideration of the issues—is that a) an "Odd" classifier is not plausible given the limitations of the statistical learning approach used by TPE (in general: inductive methods will be insufficient for classifying Odd news) and that b) the problem doesn't appear to be confined simply to choice of approach; it is not an engineering problem requiring a different algorithm, but instead, in the general case, a deep problem with using computation to understand natural language. In short, the inclusion of Odd news to the framework of TPE represents a transition from RFM* input to UFM input; this transition appears to be exactly the line between practical engineering systems like TPE and systems whose powers outstrip our current abilities.

The Inadequacy of Inductive Approaches

Abstracting away from many of the details at the level of particular algorithms introduced in the prior chapter, supervised learning approaches to classifying text are a species of *induction*: from a set of examples (observations) we produce a rule that, given new examples, can correctly classify them. The inductive approach works well for text classification because, given a set of texts each containing a sequence of words, it is likely that texts sharing a topic will also contain common words. As such, induction for

the purposes of text classification exploits word *frequencies*; that is how a classification rule can be generated from provision of examples.

For example, we expect that text on the topic of War will, in general, contain certain words used to describe war situations, such as "soldiers", "killed", "hostile" and so on. Such words will tend to occur in text about War regardless of the particular conflict (e.g., the Darfur conflict or the war in Iraq) because they are terms used to describe aspects of all conflicts and as such constitute strong evidence for classifying texts as on the topic of war. Supervised algorithms, such as the MaxEnt algorithm described in the prior chapter, exploit these common words by attaching to them a value that represents the word's contribution to class assignment. These parameters on word features used by inductive approaches are at root, essentially frequency based, however, regardless of the particular details at the level of algorithm: if a word does not co-occur with high frequency, regardless of the inductive approach it will not play a large role for purposes of classifying texts. We can call this the "frequency assumption" for inductive approaches to text classification.

The frequency assumption, as mentioned, is robust for text classification precisely because of the connection between common topics and shared words in discourse. However, as might be expected, the assumption will limit the usefulness of inductive approaches to text classification if ever word frequencies do not determine (or make more likely) selection of a class. This appears to be the case with "Odd" news.

The Frequency Assumption and Odd News

The inadequacy of using word frequency information to classify Odd news can be illustrated by consideration of a couple of examples. Consider the following two "Odd" stories, taken from the NW corpus:

> Your Tacos Or Your Life!
>
> FONTANA, Calif. - A hunger for carnitas nearly led to some carnage after a Fontana man was robbed of a bag of tacos at gunpoint. Police Sergeant Jeff Decker said the 35-year-old victim had just bought about $20 in tacos from a street-corner stand Sunday night and was bicycling home when the suspect confronted him and said "Give me your tacos." Decker said the suspect grabbed the bag of food, punched the victim in the face and began to flee. When the victim demanded his tacos back, the suspect pointed what appeared to be a handgun at the man and threatened to kill him before running away.
>
> Boy, 11, Bites Pit Bull To Fend Off Attack
>
> SAO PAULO, Brazil - An 11-year old boy is in Brazil's media spotlight after sinking his teeth into the neck of a dog that attacked him.
>
> Local newspapers reported on Thursday that Gabriel Almeida was playing in his uncle's backyard in the city of Belo Horizonte when a pit bull named Tita lunged at him and bit him in the left arm.

> Almeida grabbed the dog by the neck and bit back — biting so hard that he lost a canine tooth. Almeida tells the O Globo newspaper: "It is better to lose a tooth than one's life." Stonemasons working nearby chased the dog away before it could attack again.

We've noted that the first example cannot be classified correctly as a "Crime" story in spite of the fact that the PSE is in fact a criminal act (robbery). The story is not "Crime" because the discourse purpose is not to convey information about a robbery, but rather to tell a humorous story about a robbery of tacos; it is the nature of the items robbed and our understanding of typical robbery events that create a humorous or "odd" context and hence requires classifying the story as Odd rather than a Crime. In such cases the event context is not a straightforward inference from the PSE: once the PSE has been found (a robbery), we cannot simply "lookup" the event context given knowledge of the PSE.

We also note, germane to the present discussion of frequency information, that inducing the context using the frequency assumption is not feasible. The story "looks" like a crime story given an analysis of story content: "robbed", "gunpoint", "victim" and other words that will occur with high frequency in Crime articles are all present in the Odd article. The approach taken thus will misclassify such articles, having no inferential resources to discern the oddness or humor in such stories that make them candidates for Odd news classification.

The core problem here is further explicated by considering the semantic *relations* between two or more instances of Odd news, such as the two examples given above. Both are Odd news examples (as judged by Yahoo! News), though they share very little in common semantically, since robberies for tacos and defending against a dog attack by biting the dog are conceptually distinct events. As one might expect, the semantic dissimilarity in such examples spells trouble for classification techniques based on word frequency assumptions as well, since the difference in meaning of the two stories is of course largely a function of the different words used to express the different meanings. To put it another way, the stories use different words to express different meanings. What they do have in common, of course, is that both are candidates for expressing "odd" or humorous content, and hence both receive a common classification as Odd news. It is these pragmatic considerations that fall outside the scope of word frequency techniques, and hence the "oddness" in Odd news must be discovered using some other means.

Although this work has focused exclusively on a species of induction, extending ongoing work in IR using empirical learning methods, the problem of Odd news appears to be a very general problem for automated approaches to text classification. If this is in fact the case, then we can draw some conclusions about where, more precisely, the horizon of discourse interpretation is with respect to the text classification problem given different input constraints; if Odd news is generally hard in a way that resists known computational approaches, then simply extending a labeling scheme for text classification by one label, "Odd", will require a different, perhaps as yet unknown, class of machines.

It would be this gap between the two classes of machines—on the one hand those that can use inductive techniques to accurately classify NW data, and on the other, some machines that could reliably uncover intended oddness or humor in NW data, that would constitute the horizon—or at least a horizon—for current approaches to discourse interpretation with the aim of performing PSEL. We turn now to a consideration of the suitability of non-inductive approaches to solving the Odd news problem.

Knowledge Engineering Approaches

As Sabastiani (2002) and many others have noted, Knowledge Engineering (KE) approaches to text classification (and to NLP generally) dominated the field prior to the current concentration on inductive or empirical approaches. With the KE approach, rules are hand-coded by humans with expert knowledge of the domain to which the rules apply. These so-called "expert systems" are then applied to a domain, and the rules are iteratively improved by inspection of results.

KE approaches do not suffer from a limitation inherent in inductive approaches, namely, that all evidence—features or more generally facts discovered in some domain—available to inductive systems must be extracted by the system from the domain itself. In other words, features used by inductive systems are found in the data to be analyzed, or have been transformed purely automatically by the inductive system from the data.

When applying such systems to unseen data (e.g., after the classifier has been trained), the features extracted from the data must match those extracted when training (e.g., when inducing the classifier) because it is exactly these features (observed facts) that the system uses to classify the unseen data. One cannot, in other words, add knowledge to inductive systems that the system cannot automatically "observe" itself. We can call this the *empirical constraint* on such systems.

KE systems clearly do not have this limitation, because human experts encode knowledge about the domain. However, a key distinction must be made: KE systems may, for practical reasons, simply reproduce the observational horizon inherent in inductive systems. For example, for TC, simple rules with antecedents in Disjunctive Normal Form can be specified, where the truth of the DNF formula implies the assignment of the category in the consequent:

If <DNF formula> then <category>

For TC, humans familiar with the relevant corpus of texts to be classified engineer such rules, and the DNF formulas used are often satisfied by the occurrence of words or word phrases in the texts. In other words, such rules rely on no evidence other than word features, and hence have same evidentiary horizon as rules induced by learning algorithms. Sebastiani (2002) gives the following example of such a rule used by the CONSTRUE system in the 1980s:

if ((*wheat* & *farm*) **or**

(*wheat* & *commodity*) **or**

(*bushels* & *export*) **or**

(*wheat* & *tonnes*) **or**

(*wheat* & *winter* & : *soft*)) **then** WHEAT **else** : WHEAT

Fig. 1. Rule-based classifier for the WHEAT category; key words are indicated in *italic*, categories are indicated in SMALL CAPS (from Apt´e et al (1994)).

We can see here the empirical constraint in the rules used by this KE system. Unlike the inductive approach, however, such constraints are not intrinsic, and it is to KE systems that are not limited by the empirical constraint that we now turn.

KE Systems without Empirical Constraints

As researchers in Artificial Intelligence and related fields have long been aware, general knowledge about the world can be encoded in computational systems and brought to bear on solving problems. The structure of such systems typically consists in a knowledge base, or KB, where assertions are encoded in a suitable computational

language, and an inference engine or reasoner, where rules making use of the assertions in the KB generate new assertions. Such *knowledge-based systems* can contain arbitrary statements about the world, from abstruse assertions in some scientific domain to common sense statements about everyday objects and relations (e.g., "grass is green").

In the context of natural language processing, knowledge based systems would seem to hold great promise, since simply having knowledge of entities and their relations in the world—knowing how the world works and the sorts of entities in it—is a large part of successful language interpretation. Unfortunately, attempts to use such systems to improve the performance of computational systems on natural language processing tasks—among them, text classification—have largely failed. Indeed, there are reasons for believing that such systems, at least as we currently understand them, may have inherent limitations applying encoded knowledge to particular NLP problems.

Knowledge Based Systems and Inference

In this section we develop further the notion of "inference" and in particular the sort of inference typical of knowledge based systems. As one might expect, there are many variations on, for instance, knowledge representation languages (e.g., description logics, first or second order) and particular algorithms for performing inference, so we will need to abstract away from some of this detail to provide a general account of

inference. Our goal here is to address the question of whether systems not bound by the empirical constraint—that is, having access to general facts and rules encoded by humans—might interpret odd news correctly where, as we have seen, inductive and other approaches using only textual evidence (e.g., word occurrence, frequency) fail.

It is perhaps difficult to draw an exact line between "textual evidence" and "general knowledge" but roughly we can describe the former as tokens that are computationally accessible from the text (e.g., words in a sequence) and the latter as any assertions about the world generally, including axioms (rules) as well as particular facts. Addressing the power of knowledge based systems to correctly classify Odd news—to interpret instances of natural language that have pragmatic phenomena such as humor or "strangeness" recognizable by humans—then amounts to whether such systems can successfully apply general knowledge in their KBs to the problem of correctly interpreting Odd news text.

Preliminaries

We want our system to apply general knowledge to the interpretation of text such that Odd news can be correctly classified. For the purposes of this exposition we can cash out "Oddness" as simply "atypical" so that the exact task of a KB system will be to discover whenever a discourse describes atypical events (note that this should be, strictly

speaking, easier than understanding humor, since events can be atypical without being humorous, but it seems that the converse isn't true. The critique of inferring "atypicality" will then apply a fortiori to a critique of inferring oddness or humor). The problem now is: how would a system use general knowledge about the world to discover text with atypical events? That is, how would such a system use its KB to discover Odd news examples?

This is, as we mentioned, an inference question, but not unimportant is the representation of the general knowledge in the KB. The system will infer, using statements in the KB and observations in the discourse, atypicality, and how exactly statements are made available for computational inference is a representation question. To answer this question we must briefly detour into a discussion of syntax.

A Turing Machine can manipulate formal languages, where a language is formal just in case it can be exhaustively specified in virtue of its syntax alone—we do not need to know what syntactic symbols *mean* in order to compute with them. We can, in this sense, evaluate "P ^ Q" with a Turing Machine because we know—without knowing whether "P" is true or "Q" is true, that "P ^ Q" is true if and only if both are true separately. Likewise, if the symbol "→" is intended to be the material condition, then we know the truth conditions for "P → Q" without knowing whether "P" or "Q" are true, i.e., whether they refer correctly to objects in the world. We say that languages that can be

manipulated this way—in virtue of their syntactic constituents and relations alone—are formally valid, and candidates for evaluation by Turing Machines.

Assertions in a KB, then, must be encoded in a formal language. These assertions or groups of these assertions then form a template or "pattern", and inference rules matching this pattern will permit new statements (conclusions) to be generated. For example, from an assertion that "Grass is green" and a new fact "this item is an instance of grass", we can conclude that it must be green. "Grass is green" can be encoded in first-order logic, in which case it is universally quantified:

All X, if X is grass → X is green

If "→" is interpreted as the material conditional, then the fact that some object is grass licenses the inference that that object is also green. This admittedly simple example is, sans details, what we mean by knowledge based inference.

Modus ponens is not, strictly speaking, a complete inference procedure for a first-order system, since there are sets of statements (assertions) whose consequences cannot be determined using only MP. We can introduce resolution using refutation (RR), then, which is demonstrably complete for first-order systems (with of course the proviso that it

the language is semi-decidable, so that if a statement is true we can prove it using RR, but if not, it may not be provable.

As a final piece of preliminary detail, we introduce a distinction between generating the consequences of a set of statements in the KB (forward chaining) or determining, given a particular statement (query), whether that statement is in fact true (backchaining). The latter can be considered a method for *asking* the KB system whether something is true; linking this to the question of Odd news, we can describe a KB system that was engineered to answer questions about whether some instance of discourse should be classifiable as Odd News.

Russell and Norvig (1995) provide an amusing example of using the RR procedure to answer a question posed to a KB. The English version of the KB is as follows:

Jack owns a dog.

Every dog owner is an animal lover.

No animal lover kills an animal.

Either Jack or Curiosity killed the cat, who is named Tuna.

Did Curiosity kill the cat?

Converting to implicative normal form, we have:

Dog(D)

Owns(Jack,D)

Dog(y) ^ Owns(x,y) → AnimalLover(x)

AnimalLover(x) ^ Animal(y) ^ Kills(x,y) → False

Kills(Jack, Tuna) Or Kills(Curiosity, Tuna)

Cat(Tuna)

Cat(x) → Animal(x)

As we might expect, the query is whether "Curiosity killed the cat", or more precisely, whether Kills(Curiosity,Tuna) is true. Using the complete RR inference procedure, we assume the negation: Kills(Curiosity,Tuna) → False, and attempt to derive a contradiction given the implicative normal form KB above. It will suffice for our purposes to simply offer the proof steps in English. Russell and Norvig offer the following:

> Suppose Curiosity did *not* kill Tuna. We know that either Jack or Curiosity did, thus Jack must have. But Jack owns D, and D is a dog, so Jack is an animal lover. Furthermore, Tuna is a cat, and cats are animals, so Tuna is an animal. Animal lovers don't kill animals, so Jack couldn't have killed Tuna. But

> this is a contradiction, because we already concluded that Jack must have killed Tuna. Hence the original supposition (that Curiosity did not kill Tuna) must be wrong, and we have proved that Curiosity *did* kill Tuna.

This rather imaginative example nonetheless illustrates the broader point, which is that such inference procedures (this complete one, RR, and it many variants defined for FOL KBs) are sensitive only to "internal" or syntactically defined properties of the statements in the KB. The procedure doesn't care about the meaning of the objects that bind to variables, or about the relations between these objects except insofar as they are connected with an allowable logical connector. This fact, obvious, enough, makes it difficult to achieve the kind of inferential success seen with toy examples like Russell and Norvig's "Curiosity killed the cat" example. Extending the logical inference procedures—indeed syntactically defined inference at all—to more complicated examples quickly leads us into trouble. We will unpack the source of the trouble with later examples, but for now, we can point out that our "Curiosity killed the cat" example contains some rather obvious simplifications: animal owners aren't all animal lovers, and even animal lovers might, for various reasons, kill animals. Such simplifications preserve the strength of inference procedures like RR, but at the expense of carving out realistic scenarios knowledge-based systems to do more impressive reasoning.

Over the years, a number of theories have emerged as to why the "toy domain" successes of inference using logical formalisms like FOL cannot be extended to include more real-life reasoning problems, involving much larger KBs and inference that is

sensitive to contextual considerations (as human inferential practice clearly is). AI researchers like Doug Lenat, former professor at Stanford and currently head of the AI company Cycorp, have, historically, analyzed the problem as one of KB *size*: if we could encode many more facts and rules describing the everyday world, increasingly interesting (and correct) inferences about the everyday world would be possible. Besides the empirical failure of Lenat's "large KB" project to prove his hypothesis by generating so-called "common sense" reasoning using very large KBs, the hypothesis on its face is puzzling. How, for instance, would more facts and rules help determine which facts and rules are relevant for reaching true conclusions?

Reaching a conclusion from a set of premises is not equivalent, of course, to drawing the correct one, and prima facie it would seem that increasing the number of options to a procedure like RR would also increase the number of possible conclusions, of which only one (presumably) is correct. Viewed this way, large KBs tout court would seem only to exacerbate the problem of inference using a KB and an inference procedure defined for the logic of statements in that KB.

Other researchers (including, later, Lenat) have attempted to improve the prospects for syntactically-driven inference by, not necessarily increasing the KB size, but rather improving the inference procedure itself. Under this rubric we might include attempts at designing nonmonotonic reasoning engines, capable of retracting conclusions in the face of conflicting evidence. There is a large body of literature on nonmonotonic

logic (most of it generated in the 1980s and 1990s, a curious observation itself), and it is not in the scope of the present work to attempt to provide a detailed account of this project. Rather, we'll note first that aims of nonmonotonic inference—of retracting conclusions if no evidence comes to light—is a bit orthogonal to the present discussion, which involves finding, in the first place, an *inference path* in a KB using a logical syntax to a correct conclusion given a query ("correct" as viewed by humans that is). Notwithstanding the irrelevance of the nonmonotonic effort to the present issue, however, we note also that claims about the additional power of nonmonotonic inference engines on real-world problems appears dubious, at best. As computer scientist Drew McDermott (1987) put it bluntly, "…for now we must conclude that there is no appeal to non-monotonicity as a way out of some of the problems of deduction."

At any rate, the core problem that surfaces when attempting to use a KB system to perform reasoning, as alluded to above, is that contextual or relevance considerations appear necessary to ensure that syntactically-driven inference procedures will not reach absurd results. Such contextual considerations are *exogenous*, in the sense that they must be added to procedures like RR to augment (or block) certain chains of inference resulting in unwanted conclusions from occurring. The question here is whether there is some syntactic augmentation of inference that might provide necessary context. Here, it appears that the at least immediate prospects for such an addition are not encouraging.

John Haugeland (1979) years ago posed this *problem of context* with KB reasoning in terms of commonsense knowledge; the knowledge that ordinary people have about the features and happenings in the world around them. He unpacks the common sense problem using simple natural language examples where inferential tasks are defined in terms of the knowledge required to correctly resolve pronominal anaphora: "I left my raincoat in the bathtub, because it was still wet". This simple example gives us ample opportunity to explicate contextual problem with regard to KB systems using syntactically-driven inference. Here, we consider the query "Did the person put the raincoat in the bathtub because the raincoat was wet, or the bathtub?" The KB for a reasoning system (ostensibly) capable of answer a query of this type would presumably contain the concepts about bathtubs, raincoats, why people use bathtubs or wear raincoats, water (or "wetness") and so on. Given such a KB, it seems we could arrange the relations between such concepts in the KB such that a system using an inference procedure would, in fact, generate the correct conclusion (e.g., would resolve the pronoun to the intended antecedent). It is difficult to see, however, how statements that permitted this correct conclusion would be *natural* in any real sense. For instance, bathtubs are tubs designed to be filled with water, and as such are sometimes wet. So too are raincoats sometimes wet (at least on the outside).

The reasons for the system to prefer interpretations of wet raincoats rather than bathtubs will in general involve much larger considerations about how people behave with their clothing, and role of bathtubs, and so on, a vastly large sphere of facts and

assertions about common sense than would seem practicable given such an ostensibly simple query. Haugland puts the problem as follows:

> What's so daunting about this, from the designer's point of view, is that one never knows which little fact is going to be relevant next—which common-sense tidbit will make the next disambiguation "obvious." In effect, the whole of common sense is potentially relevant at any point.

He calls this intransigent feature of natural language *common sense holism*. Haugeland's analysis of the problem with natural language understanding fits well with our current investigation of why knowledge based inference seems so hard. As Haugeland points out, since we do not, in general, know "which little fact is going to be relevant next", designers of knowledge bases for the purposes of reasoning organize them in terms of themes, or subjects. Taking (as just one example) the concept of a monkey, he explains that "… the concept for 'monkey' would include not only that they are primates of a certain sort, but also a lot of "incidental" information like where they come from, what they eat, how organ grinders used them, and what the big one at the zoo throws at spectators."

Such organization of assertions in a KB permit (correct) answers to queries such as ones about whether "they" in queries such as "Did the monkeys eat the bananas because they were hungry, or because they were ripe", but remain profoundly

problematic when applied to examples such as the proffered "bathtub" one. Haugeland's critique of the problem is worth quoting in full:

> Both raincoats and bathtubs typically get wet, so that won't decide which was wet when I left my coat in the tub. People opt for the coat, because being wet is an understandable (if eccentric) reason for leaving a coat in a tub, whereas the tub's being wet would be no (sane) reason to leave a coat in it. But where is this information to be coded? It hardly seem that concepts for 'raincoat', 'bathtub', or 'is wet', no matter how encyclopedic", would indicate when it's sensible to put a raincoat in a bathtub.

Importing Haugeland's critique into our present discussion about contextual considerations with inference, we might consider it this way. How would a knowledge based system "know" when to use subject or topic based parts of the KB, and when such "templates" would be inadequate, and a larger search for more knowledge would be required? This problem of "getting additional context" for a particular chain of inference is usually formulated as the problem of performing tractable search in a large KB (where we assume, somewhere, a relevant fact will enable the correct conclusion), but regardless of this problem another emerges: how would the system know *when* to begin such a search, particularly if a typical inference is already available? How would it have this appreciation of context? We might (and many researchers do) postulate some set of heuristics that provide the missing facts for a correct inference, but upon reflection (and

trial and error) the question of how heuristics can avoid themselves becoming part of some unwanted chain of inference, given a slightly different problem, arises once again.

This admittedly brief sketch of the contextual problem with inference can be seen more clearly when applied to some of the Odd news examples mentioned earlier. For instance, with the man-robbed-for-tacos example, removing the oddness (i.e., interpreting the story as a standard robbery) fits a topic-based representation (we'll call this a schema) for criminal acts, since stories about people getting robbed are *ipso facto* stories about criminal events. But the story, of course, is not a standard crime story—a story to be classified as having topic Crime. Given the centrality of the robbery in the discourse, which is elaborated on in the first few paragraphs, it is difficult to see how any topic-based inference scheme would not classify it as such. The key evidence in the text are the items taken—tacos—which, given the severity of punishment for robbing someone by threat of force, seem to insignificant and inexpensive to explain someone's decision to commit robbery (hence, the oddness, or the humor).

A system sensitive to such observations might constrain "Crime" inferences by specifying that items to be robbed are typically valuable, and so on. This approach will be subject to easily formulated counter examples, such as criminal acts where someone steals an empty wallet and so on. With the taco example itself, the inference to a typical criminal act might be licensed by changing some of the context of the story—say, by explaining that the robber was an escaped criminal that had not found food in several

days. The point here is that avoiding the problem of selecting among many possible inference paths (or, equivalently, of having to be sensitive to many contexts) will in general require more than a topic-based representation of knowledge. In this case, however, the entire program of representing knowledge by grouping like concepts together becomes suspicious. If we cannot represent knowledge by grouping concepts together based on similarity or topicality, how exactly is knowledge of the world, necessary for performing "odd" inferences, to be represented? As Haugeland mentions, humans somehow put themselves into a situation, in order to see what is happening, what is out of place, what is proper, and so on. The question of how a machine could perform inference based on situation (or context) rather than with rules and facts organizing the world in terms of topics or subjects is just at present unanswered.

Frequencies Revisited: The Single Term Problem

Haugland's examples suggest that the meaning of a discourse can be changed radically, without making radical changes to the discourse as a whole: "… In effect, the whole of common sense is potentially relevant at *any point*." (italics added) In this section we'll take another look at the use of empirical (frequency-based) approaches to NLP, beginning with a discussion of the type of discourse models that are assumed when performing NLP tasks like interpretation using one or other inference approach.

Recall from Chapter 1 that state-of-the-art approaches to NL interpretation like SDRT (or dynamic semantics approaches generally) presuppose robust models of discourse: a discourse is comprised of a set of sentences, in turn comprised of clauses, analyzable in terms of a grammar into syntactic categories, in turn comprised of words (lexemes) terminating with singular parts of speech. In such models, prior sentences (clauses) comprise a context that informs interpretation of a current sentence; discourse phenomena like anaphora, ellipsis, presupposition, bridging and others are analyzed in terms of this multi-sentence context, and lexical, semantic, syntactic and other information sources in this context are all potentially relevant.

Note, however, that with the development of the hierarchical classification approach in this investigation, we eschewed this model in favor of a much simpler one, considering the discourse as essentially a sequence of terms (words) conjoined with a sequence of features definable without breaking the empirical assumption. This simplified model facilitated an effective approach to solving one particular type of problem of interpretation with wide scope in discourse, that of inferring discourse topic or more specifically Primary Semantic Types. By introducing a distinction between within discourse (sentential) and discourse-wide interpretation tasks, in other words, we were able to formulate an approach that could be tailored specifically to the latter without bothering with many of the details inherent in the former.

Yet, with the relaxed lexical or "bag of words" model of discourse adopted, our approach to inference was necessarily empirical (frequency-based), since singular terms (or at most n-grams) and a minimal shallow structure like paragraph breaks and title words constitute now the information source available from such a model. And this leads us to the present discussion: what we gained by adopting a simplified model of discourse for solving tasks like PSTL, we lost for solving tasks not amenable to frequency considerations. Odd news is a prime example. Here, as we've seen, frequency considerations actually *impede* inference, since it is often the very backdrop of a general theme or topic that makes a particular feature of the story humorous (stealing tacos is, after all, a story mostly about a theft. The object of the theft—tacos—is a small piece of the discourse that nonetheless accounts for the oddness or comical element). And so it is that inferences about primacy tend to be blind to specificity.

Odd news is one species of a more general set of problems of interpretation that seem to lie outside the scope of the frequency assumption. The difficulty, again, seems particularly intransigent to empirical approaches because it seems to stem from the power of single words or terms to radically change the flavor or meaning of a discourse. Single terms with such powers are poor candidates for term frequency based approaches for the obvious reason that there frequency in a discourse or set of discourses may be one.

Examples of this "single term" phenomena are not hard to find. We noted in Chapter 1 that examples like "The pen is in the box" lead early AI researchers to abandon

simple lexical approaches to machine translation, and ushered in a more explicitly semantic approach to NL interpretation. Similarly, research on dynamic semantics approaches to interpretation has uncovered the importance of "cue words" or phrases that can affect assignment of rhetorical relations: "but" can introduce a relation like Contrast, which in turn can affect the meaning of the entire multi-sentence discourse of which it is part (A&L, 2003). And more recently we discussed Haugland's (1979) critique of NLP in terms of the problem of relevance or context; we can see from his examples now that single words lie at the heart of the interpretation problem for him as well, since we needed only to consider a simple modifier like "wet" to introduce radically different interpretations in his "bathtub" example.[16] In all such examples, the empirical models used to infer discourse-wide features (like PSEs in the present case) will be insensitive to shifts in meaning from single terms (necessarily so because they are designed to generalize and not pick out particular, uncommon terms, by design).

It seems, then, that the step from RFM*, where '*' is a topic labeling for NW data *sans* Odd news, to RFM-Complete machines, where NW data can include categories such as Odd news is a large one indeed. The requirement for inference to consider context (which may be a function of the presence of a single word, as we've seen) in discourse breaks the frequency assumption, which effectively precludes use of inductive

[16] For another, slightly bizarre example, I recently recovered from the flu, and woke up in a pool of sweat. It occurred to me in this state how different things would be if I merely exchanged "blood" for "sweat" in the sentence "I woke up in a pool of sweat." We can imagine in such cases an entire discourse describing the flu, and yet with the exchange of the single word "sweat" for "blood", the discourse would be very strange indeed, though if analyzed by empirical approaches the preponderance of words indicating say "Sickness" or "Health" topic would likely make the single occurrence of "blood" irrelevant.

approaches relying on it; likewise, we've seen that KE systems with or without the empirical assumption fair no better. Exactly *how* we might design a class of systems that are capable of correctly interpreting Odd news (or more generally, RFM-Complete discourse) is unclear, and if one accepts the above critique the current horizon would seem to present a formidable challenge, indeed.

As we have seen, however, we can build systems to exploit regularities in language, and such systems can be improved to perform arbitrarily better below the contextual horizon spelled out in this chapter. Likewise for knowledge engineered approaches. It is hard to envision, however, that such systems using such techniques will soon *cross* the horizon, for the reasons given. New systems must be built, and unfortunately the details of the theoretical assumptions underlying such systems are not at present available. Whether indeed Turing machines and the classic notion of computation will ever perform such tasks is itself unclear.

Nonetheless, the incremental progress detailed in this work should, we hope, prove of continuing interest to philosophers and researchers in AI attempting to understand the types of systems that we build, and the types of problems that such systems can solve. We have outlined a framework for classifying natural language interpretation problems in terms of their difficulty (the Constraint Framework discussed in Chapter 2), and we've identified a novel interpretation problem whose solutions provides us interesting information about a discourse: what is the primary specific event?

It was argued that the class of machines that can reliably solve the PEL tasks (or PSTL tasks generally) lie at the forefront of tractable NLP, and as such the present works helps to define the horizon of research on AI, in particular on the automated understanding of natural languages, one of the most important branches of AI and one since at least the time of Turing that has inspired our imaginations. We've outlined in detail the representational requirements for a machine to perform PEL in Chapter 3, and in Chapter 4 we outlined empirical (learning) approaches to performing inference. An inference procedure was designed—hierarchical text classification—that exploits the information present in the type hierarchies (Chapter 3), facilitating a power text classification capability that terminates with the most specific, primary semantic type given the hierarchies. The approach has been implemented and tested on actual NW data, resulting in a set of accuracy statistics that vindicate the idea and demonstrate (we believe) conclusively that the theoretical ideas can be made practical and hence that the entire project succeeds in its aim, that of demonstrating the feasibility of such a machine. Finally in Chapter 5 we discussed classes of machines that lie outside the scope of the present investigation, and the specific reasons that such RFM-Complete machines seem so distinct and at present, so unreachable given our understanding of natural language interpretation. Future investigations with other insights may indeed begin to shine light on the many challenges of NL interpretation; we hope here that some of the framework and ideas may become part of this ongoing discussion.

References

Asher, N. (2004a). Discourse topic. Theoretical Linguistics, 30(2-3): 163–202.

Asher, N. (2004b). Troubles with topics: Comments on Kehler, Oberlander, Stede and Zeevat. Theoretical Linguistics. Volume 30, Issue 2-3, Pages 255–262.

Asher, Nicholas and Lascarides, Alex. Logics of Conversation: Studies in natural language processing. Cambridge University Press, Cambridge, 1 edition, 2003.

Bar-Hillel, Yehoshua, The Present Status of Automatic Translation of Languages, Advances in Computers, vol. 1 (1960), p.91-163.

Berger, A. L., Della Pietra, S. A., & Della Pietra, V. J. (1996). A maximum entropy approach to natural language processing. Computational Linguistics, 22.

Block, Ned. (1995). The Mind as the Software of the Brain. In Daniel N. Osherson, Lila Gleitman, Stephen M. Kosslyn, S. Smith & Saadya Sternberg (eds.), An Invitation to Cognitive Science. MIT Press.

Bobrow, D.G., 1968. Natural language input for a computer problem-solving system. In Minsky, pp. 146-226.

Brill, Eric and Mooney, Raymond. (1997). An Overview of Empirical Natural Language Processing, AI Magazine Volume 18 Number 4.

Buchanan, Bruce G. A (Very) Brief History of Artificial Intelligence. AI Magazine 26(4): Winter 2005, 53–60.

Carreras and L. Màrquez. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In Proc. of CoNLL-2005.

Roderick M. Chisholm. (1964). The Descriptive Element in the Concept of Action. Journal of Philosophy 61 (20):613-625.

Chomsky, Noam. (1964). Current issues in linguistic theory.

CoNLL-2004 and CoNLL-2005 Shared Tasks. Available at http://www.lsi.upc.edu/~srlconll/.

Copestake, Ann and Flickinger, Dan. (2000). An open-source grammar development environment and broad-coverage English grammar using HPSG In: Proceedings of the Second conference on Language Resources and Evaluation (LREC-2000), Athens, Greece

DARPA. (1993). Proceedings of the Fifth DARPA Message-Understanding Evaluation and Conference. San Francisco, Calif.: Morgan Kaufman.

Davidson, Donald. (1967a). 'Causal Relations', Journal of Philosophy, 64: 691–703; reprinted in Davidson 2001a.

Davidson, Donald. (1967b). 'The Logical Form of Action Sentences', in Nicholas Rescher (ed.), The Logic of Decision and Action, Pittsburgh: University of Pittsburgh Press, reprinted in Davidson, 2001a.

Davidson, Donald (1969), 'The Individuation of Events', in Nicholas Rescher (ed.), Essays in Honor of Carl G. Hempel, Dordrecht: D. Reidel, reprinted in Davidson 2001a.

Davidson, Donald. (1970) Events as Particulars. Noûs 4 (1):25-32.

Davis, Ernest (1990). Representations of Commonsense Knowledge. San Mateo, CA: Morgan Kaufmann.

Dreyfus, Hubert (1992), What Computers Still Can't Do, New York: MIT Press.

Dumais, S. and Chen, H. (2000). Hierarchical Classification of Web Content, In SIGIR 2000.

Fillmore, Charles J. (1968) "The Case for Case". In Bach and Harms (Ed.): Universals in Linguistic Theory. New York: Holt, Rinehart, and Winston, 1-88.

Haugeland, J. (1979). "Understanding natural language," Journal of Philosophy, vol. 76, pp. 619-632.

Hearst, M. (1994). Multi-paragraph segmentation of expository text. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics.

Hirst, G. (1987). Semantic Interpretation and the Resolution of Ambiguity. Cambridge University Press.

Joachims, Thorsten. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. In Machine Learning: ECML-98, Tenth European Conference on Machine Learning, pages 137-142.

Johnson, Christopher, Petruck, Miriam, Baker,Collin, Ellsworth, Michael, Ruppenhofer, Josef and Fillmore, Charles. (2003). Framenet: Theory and practice. Berkeley, California.

Kamp, H. (1981). A theory of truth and semantic representation. In T. Janssen J. Groenendijk and M. Stokhof, editors, Formal Methods in the Study of Language. Mathematical Center, Amsterdam, pages 277-322.

Kamp, H. and Reyle, U. (1993). From Discourse to Logic, volume 42 of Studies in Linguistics and Philosophy. Kluwer Academic Publishers.

David Kaplan (1975). How to Russell a Frege-Church. Journal of Philosophy 72 (19):716-729.

Kehler, Andy. (2004). Discourse Topics, Sentence Topics, and Coherence. Theoretical Linguistics 30:2-3, pp. 227--240.

Kenny, A.J.P. (1963) Action, Emotion and Will. Routledge and Kegan Paul, London.

Kingsbury, Paul and Palmer, Martha. (2002). From Treebank to PropBank. In LREC02.

Kiritchenko, S., Matwin, S., and Famili, F. (2005). Functional annotation of genes using hierarchical text categorization. Proceedings of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology.

Lenat, Douglas B. and Brown, John Seely. (1984). "Why AM and Eurisko Appear to Work." J. Artificial Intelligence 23: 269 -- 94.

Lenat, Douglas B. and Guha, R. V. (1990). Building Large Knowledge-Based Systems. Reading, Mass.: Addison-Wesley.

Lewis, D. D.; Yang, Y.; Rose, T.; and Li, F. (2004). RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, 5:361-397. http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf.

Lindsay, R.K. (1963). Inferential memory as the basis of machines which understand natural language. In Feigenbaum and Feldman, pp. 217-233.

Mann, William C. and Sandra A. Thompson. (1986). Rhetorical Structure Theory: Description and Construction of Text Structures (Technical Report No. ISI/RS-86-174). Marina del Rey, CA: Information Sciences Institute.

McCallum, Andrew Kachites. (2002). "MALLET: A Machine Learning for Language Toolkit." Available at: http://mallet.cs.umass.edu.

McCarthy, John (1990). Formalizing Common Sense. Norwood, NJ: Ablex.

McDermott, Drew. (1987) A critique of pure reason. Computational Intelligence 3(33), pp. 151–160.

Meyers, A., Reeves, R., Macleod, C. , Szekely, R., Zielinska, V. , Young, B., and Grishman, R. (2004). The NomBank Project: An Interim Report, Proc. of HLT-EACL Workshop: Frontiers in Corpus Annotation.

Minkov, E., Wang, R. C., and Cohen, W. W. (2005). Extracting personal names from emails: Applying named entity recognition to informal text. In HLT-EMNLP.

Mitchell, Tom M. (1997). Machine Learning, McGraw Hill, 1997.

Nariyama, S. (1994). Subject elipsis in English. Journal of Pragmatics.

Nigam, Kamal, Lafferty, John, McCallum, Andrew. (1999). Using Maximum Entropy for Text Classification. IJCAI'99 Workshop on Information Filtering.

Palmer, M. , Gildea, D., and Kingsbury, P. (2005). The Proposition Bank: A Corpus Annotated with Semantic Roles, Computational Linguistics Journal, 31:1.

Pustejovsky, J. (2005). The Generative Lexicon, MIT Press.

Reyle, R. (1993). Dealing with Ambiguities by Underspecification: Construction, Interpretation and Deduction. Journal of Semantics, 10, 123-179.

Reynar, J.C. and Ratnaparkhi, A. (1997). A maximum entropy approach to identifying sentence boundaries. In Proceedings of the 5th Conf. on Applications of Natural Language Processing, 16-19.

Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, NJ: Prentice Hall.

Salmon, Wesley C. (1998). Oxford University Press, USA; illustrated edition.

Schank, R. (1975). Conceptual Information Processing.

Schank, R., and Abelson, R. P. (1977). Scripts, Plans, Goals, and Understanding. Hillsdale, NJ : Lawrence Erlbaum.

Searle, John. (1980). Minds, Brains, and Programs. Behavioral and Brain Sciences 3:417-57.

Sebastiani, Fabrizio (2002). Machine learning in automated text categorization. ACM Computing Surveys, 34(1):1-47.

Stede, Manfred. (2004). Does discourse processing need discourse topics? Theoretical Linguistics. Volume 30, Issue 2-3, Pages 241–253.

Sterrett, S. G. (2000), "Turing's Two Test of Intelligence", Minds and Machines 10 (4): 541.

Strawson, P.F. (1959) Individuals. Methuen, London.

Text Retrieval Conference (TREC). Available at http://trec.nist.gov/.

Turing, A. (1950), 'Computing Machinery and Intelligence', Mind 59(236), pp. 433-460.

Weizenbaum, J. (1976). Computer Power and Human Reason: From Judgement to Calculation. San Francisco: W.H. Freeman.

Wilks, Y. (1975). A preferential, pattern-seeking semantics fornatural language inference. Artificial lntelligence, 6, pp. 53-74.

Winograd, Terry. (1972). Understanding Natural Language, (191 pp.) New York: Academic Press. Also published in Cognitive Psychology, 3:1, pp. 1-191.

Woods,W. (1975). What's in a link : foundations for semantic networks. In Bobrow, U.G. and Collins, A. (Eds.) Representation ana Understanding. Academic Press, New York, 35-82.

Zeevat, H. (2004). Asher on discourse topic. Theoretical Linguistics, 30(2-3):203-212.

Vita

Erik John Larson attended Freeman High School, Rockford, Washington. In 1990 he entered Whitworth College (now Whitworth University) in Spokane, Washington, where he received the degree of Bachelor of Arts in 1995 with majors in Philosophy and Mathematics. In 1997 he entered The University of Texas at Austin, where he received the degree of Master of Arts in 1999. While pursuing the degree of Doctor of Philosophy at The University of Texas at Austin, he has been employed as a researcher and software developer for companies in Austin, Texas and was employed as a Research Scientist Associate at The University of Texas at Austin, where he worked on problems in natural language processing.

Permanent Address: 2602 Claudia Drive, Leander, TX 78641

This manuscript was typed by the author.