Extracting Pragmatic Content from Email

Hamid Khosravi-Bardsirpour

June 1999

Submitted in accordance with the requirements

for the degree of Doctor of Philosophy to:

Department of Computer Science

University of Sheffield

بسم الله الرحمن الرحيم

This work is dedicated to the memory of my mother,

who died when I was far away from her.

Hamid

Abstract

This research presents results concerning the large scale automatic extraction of pragmatic content from Email, by a system based on a phrase matching approach to Speech Act detection combined with the empirical detection of Speech Act patterns in corpora. The results show that most Speech Acts that occur in such a corpus can be recognized by the approach. This investigation is supported by the analysis of a corpus consisting of 1000 Emails.

We describe experimental work to sort a substantial sample of Emails based on their function, which is to say, whether they contain a statement of fact, a request for the recipient to do something, or ask a question. This could be highly desirable functionality for the overburdened Email user, especially if combined with other, more traditional, measures of content relevance and filters based on desirable and undesirable mail sources.

We have attempted to apply an IE engine to the extraction of message content located in the message, in part by the use of speech-act detection criteria, e.g. for what it is to be a request for action, under the many possible surface forms that can be used to express that in English, so as to locate the action requested as well as the fact it is a request. The work may have potential practical uses, but here we describe it as the challenge of adapting an IE engine to a somewhat different, task: that of message function detection.

The major contributions are:

Defining Request Speech Act types.

The Request Speech Act is one of the most important functions of an utterance to be recognised, in order to find out the gist of a message. The present work has concentrated on three sub-types of Requests: Requests for Information, Action, and Permission.

An algorithm to recognise Speech Acts

Patterns found frequently in a domain, together with linguistic rules, make it possible to recognise most of the examples of Requests in the corpus. The results of the evaluation of the system are encouraging and suggest that, in order to avoid long-response time systems, a fast and friendly system is the right approach to implement.

Acknowledgements

I would like to express my sincere gratitude to my supervisor professor Yorick Wilks for his invaluable guidance, regular instruction, stimulating discussions and helpful suggestions throughout the period of this study. His support during the preparation of the thesis is gratefully acknowledged.

I am grateful to the other members of my panel, Dr. Phil Green and Dr. Rob Gaizauskas for their valuable guidance and suggestions.

I am also thankful to the Ministry of Culture and Higher Education (MCHE) of the Islamic Republic of Iran for the grant of scholarship and financial support.

I would like to thank all members of NLP group especially Mark Lee and Paul Woods for their valuable comments and discussions. I am also thankful to the members of technical support for their technical help and for providing me with the Email corpus.

Finally, I would like to express my sincere thanks to my wife, M. Ebrahimi and my children Sara, Hassan, and Rana, who have been a source of inspiration for me and who have given me constant love and support.

Declaration

The candidate confirms that the work submitted is his own, and that appropriate credit has been given where reference has been made to the work of others.

Contents

List of figures

List of tables

Chapter 1

Introduction

The tradition of routing messages using information retrieval (IR) techniques based on key word statistics (Guthrie and Walker and Guthrie 1994) has normally been concerned with content, in the sense of the topic, subject matter or domain relevant to the user. In this work we describe experimental work to sort a substantial sample of Emails based on their function, which is to say, whether they contain a statement of fact, a request for the recipient to do something, or ask a question. I believe this to be highly desirable aid for the overburdened Email user, especially if combined in use with other, more traditional, measures of relevance, and filters based on desirable and undesirable mail sources. Such function-based message routing naturally relates to the tradition of analysis based on speech or dialogue acts, which we describe below, an established body of philosophico-linguistic work focused upon describing and detecting such message functions.

I have adopted the overall methodology of Information Extraction (IE) [Gaizauskas and Wilks, 1998], a technology developed not to route messages but to extract content directly from text, rather than selecting relevant document subsets as IR does. I have attempted here to apply an IE engine to the extraction of message content located in the message, in part by the use of speech-act detection criteria, e.g. for what it is to be a request for action, under the many possible surface forms that can be used to express that in English, as well as to the request content itself. The work may have potential uses in a busy world, but here we describe it as the challenge of adapting an IE engine to a somewhat different, task, that of message function.

1.1 Background

Today Email is a fast and easy tool for communication which was first used between university research centres in about 1971, supported by U.S. military contracts to exchange information as quickly as possible. Later, like other software systems, its application was extended to other areas as well. Today, most universities, research centres, commerce and business areas, and even private activities get the benefit of this facility.

Scientists, politicians, businessmen, and increasingly many others spend a great deal of time every day reading, sending and replying to their Emails, and it can take hours for busy people to read hundreds of Emails, although they know that some of them have higher priority than others. Chomsky was recently quoted saying that it took him three hours a day to deal with his Email (Times Higher Education Supplement. 9/4/99). Email is a fast and easy tool for communication, of course, although this also brings problems. Such problems become more important when Email is used as input text to a Natural Language Processing (NLP) system. These problems are, in general:

1- Information overload

It is possible for a person to receive hundreds of Emails everyday. One main reason is that communication by Email is easy, fast and cheap and also that Email facilities have made it possible to send a single message to a large group of people. Also, the use of an informal language, sometimes closer to spoken language rather than written, (see section 8.3) encourages increased communication.

2- Structure and content

Email message texts differ from ordinary texts. Since Email is an informal way of communicating, it turns out to have many carelessly misspelled words (see section 8.1.1) and abbreviations which do not occur in written texts. In addition, sentence recognition (parsing), which is a general requirement for many NLP systems, is far more difficult with Emails, in part due to their lack of complete punctuation.

The final goal of research such as mine is to implement friendly and fast software as an interface which will be able to recognise and summarise the gist of a message. For the pur-

pose of this thesis, my goal is to recognise specific Speech Acts that occur in Email messages especially in a pre-determined domain, of the sort we describe here (see chapter 8).

To cope with understanding the gist of a message, as Halliday and Hasan [Halliday and Hasan, 1976] have argued, a message must be treated not just as a string of sentences, but a complex structure comprising many components, including a reader, a writer, shared knowledge of the world and a communicative situation. The approach which has been investigated in this research is a combination of phrase matching and pragmatic rules, for recognising Speech Acts that occur in a message. Sorting Emails based on Speech Acts occurring in their text, and content analysis of the text (as opposed to analysis of information like the subject line or sender's address) make it possible for the end user not only to achieve reliable filtering of Emails, but also to prioritise them according to their content.

1.2 An outline of the Thesis

Chapter two reviews definitions of pragmatics from different points of views. In addition, Speech Act theory [Searle, 1969], on which this research is partly based, is discussed. The notion of indirect Speech Acts, is introduced.

Chapter three discusses semantic representational approaches to natural language processing. These approaches share the use of templates for representing semantic features.

Chapter four presents computational approaches to Speech Acts.

Chapter five investigates more recent work in natural language processing. In its first part, three sub-areas of natural language processing, related to text processing, are discussed: Information Extraction, Information Retrieval, and Text Summarisation and Classification.

Chapter six describes earlier work directly related to Email.

Chapter seven discusses the general principles of corpus analysis. In particular, it investigates the distinction between spoken and written language and discusses an algorithm to distinguish these two genres.

Chapter eight describes the corpus used in this research and the results found from analysis

of the corpus. More than 1000 Email messages sent to the support group of the Department of Computer Science at the University of Sheffield, have been analysed. Specifications of the Email texts, which make them different from other texts, are discussed.

Chapter nine provides an analysis of requests in Email and distinguishes three types of Requests: Request-Information, Request-Action, and Request-Permission. A preliminary analysis of the Email messages from the corpus prepared for this research (see chapter 8) led us to concentrate on "Request" Speech Acts, since they occur in more than 90% of the messages.

Chapter ten describes the implemented system (Pyam). The system receives Email messages as input and performs the following tasks: it recognises all Focus Sentences that appear in the text, then prints out all request types corresponding to each focus sentence, plus more information related to the syntactic structure of the sentence.

Chapter eleven provides an evaluation of the system described in this thesis and compares Pyam's performance with Emails to natural spoken dialogue corpora.

Chapter twelve gives the conclusion and suggests future directions for investigation.

Chapter 2

Pragmatics

The theory of Speech Acts is a basis of the work presented in this thesis. In addition, pragmatics, as the main foundation for Speech Act recognition, plays a major rule in this investigation of functionality in Email texts.

2.1 Pragmatics in general

Pragmatics is concerned with the way people use language to communicate, the way that people make themselves understood and try to understand what other people say to them. Gazdar [Gazdar, 1979] defines pragmatics as follows:

> "Pragmatics has as its topic those aspects of meaning of utterances which cannot be accounted for by straightforward reference to the truth conditions of the sentences uttered. Put crudely:
>
> PRAGMATICS = MEANING - TRUTH CONDITIONS."

The use of the term "pragmatics" in linguistics was originally defined by Morris [Morris, 1938] who distinguished three branches of inquiry: syntax, semantics and pragmatics. In his definition, syntax is the study of the formal relation of signs to one another, semantics is the study of the relations of signs to the objects to which the signs are applicable, and pragmatics is the study of the relation of signs to interpreters. Levinson [Levinson, 1983] provides an alternative definition as "syntax is the study of the combinatorial properties of words and their parts, semantics is the study of meaning and pragmatics is the study of language

usage". The reason that researchers in Artificial Intelligence (AI) use the term "language understanding" is to draw attention to the fact that understanding an utterance involves a great deal more than just knowing the meaning of the words uttered and the grammatical relation between them. Understanding an utterance involves making inferences which AI researchers usually take to constitute pragmatics. Pragmatics shows its importance when syntactic and semantic theories are unable to clarify the meaning of an utterance or in some cases resolve the ambiguity of a sentence. For instance, in this example from Levinson [Levinson, 1983], a semantic theory based on logical forms which are supposed to be true or false in virtue of their forms is unable to differentiate between the two parts of the sentence below:

"Getting married and having a child is better than having a child and getting married."

The reason is that logically, the statement "A and B" is equivalent to "B and A" and it is confusing that one of those logical forms is claimed be better than its equivalent. This example brings up the difficulty of mapping many ordinary everyday utterances to logical forms using symbols such as "AND", "OR", or "NOT". The following example [Mey, 1993] shows the difficulty of distinguishing between "and" and "but" in a logical form:

A: Mary is a nice girl and she takes swimming lessons.

B: Mary is a nice girl but she is poor at tennis.

In the two sentences A and B, the use of "and" or "but" seems to add extra information, which makes a value judgment on the part of the speaker. If a sentence has the form "Mary is X and Y", the implication is that Y modifies the initial judgment or viewpoint X in a positive sense. That is, the addition of Y serves to augment or improve on the initial assessment implied by X. Analogously, the use of "but" seems to have a contrary effect on the initial assessment afforded by X. It seems only reasonable to argue that the addition of the element Y is not merely extra information of a neutral character; the pragmatic intent of such additional information with "and" or "but" is positive or negative respectively. The difference between the meanings of the above sentences shows the limitations of truth conditional semantics in representing conjunctions such as "and" and "but".

2.2 Conversational Implicature

Grice suggests that to recognise the full meaning of an utterance, four general maxims should

be considered in any rational conversation: maxims of Quality, Quantity, Relation and Manner.

1- Quantity, which is related to the quantity of information to be provided.

a-Make your contribution as informative as is required.

b-Do not make your contribution more informative than is required.

2- Quality, which is "to make your contribution true".

a- Do not say what you believe to be false.

b- Do not say that for which you lack adequate evidence.

3- Relation

Be relevant.

4- Manner, which concerns "How what is said is to be said".

a- Avoid obscurity of expression.

b- Avoid ambiguity.

c- Be brief.

d- Be orderly

Grice claims that, although these maxims should be followed, failing to obey the maxims may suggest that the hearer should try and understand the implicature intended by the speaker. Grice defined three situations in which implicatures can be recognised. The first situation is when all the maxims have been followed. For example,

A: So you really believe Lucy hates Sally?

B: I did not say that.

In the above example, A should understand that not only B did not say "Lucy hates Sally", but he does not believe it either.

In the second situation a violation in one of the maxims implies another piece of information (implicature) to be inferred by the hearer. For example,

A: At what time is the match?

B: Sometime tomorrow.

B's reply breaks the quantity maxim because it is not "as informative as required" and so A should infer that B does not know the exact time of the match.

In the third situation, the speaker breaks one or more maxims deliberately. In this example, when A is asked to write a recommendation and he writes "Dear Sir, Mr. X has a good command of English.", A is violating the quantity and relation maxims to avoid mentioning Mr.X's poor academic ability.

2.3 Speech Acts

Austin

Austin claimed that utterances are not just descriptions of states of affairs, but are used to do things in the world [Austin, 1962]. Under appropriate conditions utterances can change the mental state of the speaker and hearer. The main idea behind Speech Act theory is that each utterance, sentence or phrase, by a speaker is an action used by him to achieve part of his goal(s). So a sentence like

"I promise to be home by midnight."

could be part of a speaker's plan to get permission to go out. Austin also defined as "performatives" utterances which are not necessarily true or false, for example "I bet you five pounds".

Austin also differentiated between explicit performatives and implicit performatives. He considered those acts which perform by convention and which use the verb that names the act as explicit performatives, e.g. "betting" or "naming". One clue to recognizing this class of performative is that the verbs in these acts can follow "hereby". For example "I hereby bet you five pounds". Other utterances which can change the state of the world are considered to be implicit performatives.

To understand the exact meaning of an utterance a hearer must be in possession of appropriate related knowledge or beliefs. For example, an utterance like "Do you know the time?", might be a question or warning, depending on the beliefs of the speaker and the hearer. If it is assumed that the speaker does not know the time, this utterance can be a request as in "please

tell me the time" but in another situation if the speaker does know the time, it can be a warning that "it is becoming late".

Austin defined three distinct acts which are performed by any utterance: the **locutionary** act; the **illocutionary** act; and the **perlocutionary** act. The locutionary act refers to the absolute act of speaking, the illocutionary act refers to the act effected by the speaker and intended by him by making an utterance such as making a statement, request, promise, question, etc. and, finally, the perlocutionary act refers to the act effected by the context of the utterance, and which may not be intended by the speaker.

Searle

Searle [Searle, 1969] extended Austin's work on illocutionary acts. He argued that there are certain conditions necessary to perform and recognise any Speech Act:

i. **Normal input-output conditions obtain:** the conditions for normal speaking obtain, such as the language being comprehensible, the hearer is paying attention etc.

ii. **Propositional content:** the conditions describe restrictions and limits related to each Speech Act.

iii. **Preparatory conditions:** the basic conditions that make the Speech Act useful and relevant.

iv. **Sincerity conditions:** the speaker's actual desire to be the same as attitudes expressed by the act.

v. **Essential conditions:** the intention of the speaker in performing the act are appropriate.

For example he lists the following conditions necessary to recognise and understand the "promising" Speech Act.

1. Normal conditions must obtain for input and output.

2. The promise must have some content.

3. The promise must concern some action in future.

4. What is promised must be of advantage to the promisee.

5. The promised action must not be something that will happen anyway.

6. The promiser must be sincere.

7. The promiser intends the act of promising to place him under obligation to fulfil the promise.

8. The promiser intends the promisee to recognise that a promise is being made.

9. An utterance is a promise if and only if conditions above are met.

There is an extensive literature [Wittgenstein, 1958], [Levinson, 1983] questioning the coherence and clarity of such sets, some of which has come from within the AI/NLP tradition [Grosz and Sidner, 1986]. In particular, it is a problem that the conditions seem to oscillate between the points of view of speaker and hearer: (6) is known to the speaker but (4) can only be reliably known to the hearer and so cannot be a condition on the state of a speaker.

Searle [Searle, 1976] also extends the classification of types of illocutionary acts. He identifies five possible Speech Act types:

i. **Representatives** such as asserting, to commit the speaker to the truth of the proposition expressed by the utterance.

ii. **Directives** like requesting, to attempt to get the hearer to perform some action for the speaker.

iii. **Commissives** like promising, to commit the speaker to some future action.

iv. **Expressives** like thinking to express a mental state.

v. **Declarations** like naming, to change an institutional state of affairs.

Searle classifies four general aspects for each Speech Act: propositional content, preparatory conditions, sincerity conditions and essential conditions. The following Table demonstrates how one should distinguish between requests and assertions. In this table, S and H represent the Speaker and the Hearer respectively.

	Request	Assert
Propositional content	Future act of A of H	Proposition P
Preparatory conditions	1. H is able to do A and S believes H is able to do A. 2. It is not obvious that H will do A in the normal course of events.	1. S has justification for the truth of P. 2. It is not obvious that H knows the truth of P.
Sincerity conditions	S wants H to do A	S believes P
Essential conditions	Counts as an attempt to get H to do A	Counts as an undertaking to the effect that P represents an actual state of affairs.

Table 1: The Illocutionary acts for Request and Assert [Sea69]

2.3.1 Indirect Speech Acts

Searle [Searle, 1975] argued that there are also situations when the speaker's utterance meaning and the sentence meaning come apart in various ways, contrary to those sentences in which the speaker utters a sentence and means exactly and literally what he says. In these cases, a sentence that contains an illocutionary act can be uttered to as to perform, in addition, another type of illocutionary act indirectly. Indirect Speech Acts are cases in which one illocutionary act is performed indirectly by way of performing another. So if someone says, "Could you move over a bit please?" he or she does not expect a yes or no reply, although he or she has asked a yes/no question.The apparatus necessary to explain the indirect part of an indirect Speech Act includes a theory of Speech Acts, certain general principles of cooperative conversation and mutually shared factual background information between the speaker and the hearer, together with an ability on the part of the hearer to make inferences. By cooperative conversation, Searle refers to the four general maxims suggested by Grice [Grice, 1975].

Searle argued that in the field of indirect illocutionary acts, the area of directives is the most useful to study because ordinary conversational requirements of politeness normally make it awkward to issue straightforward imperative utterances or explicit performatives, and we therefore seek to find indirect means to our illocutionary ends. In directives, politeness is

the chief motivation for the indirectness.

Group 1: *Sentences concerning a hearer's ability to perform actions.*

Can you pass the salt?

Could you be a little more quiet?

You could be a little more quiet.

Group 2: *Sentences concerning a speaker's wish that a hearer will do an action.*

I would like you to go now.

I would be most grateful if you would help us out.

I would be very much obliged if you would pay me the money back.

Group 3: *sentences concerning a hearer's action*

Would you kindly get off my foot?

Aren't you going to eat your cereal?

Group 4: *Sentences concerning a hearer's desire or willingness to do an action.*

Would you be willing to write a letter for me?

Would it be convenient for you to come on Wednesday?

Group 5: *Sentences concerning the reasons for doing an action.*

You should leave immediately.

Why not stop here?

Group 6: *Sentences embedding one of these elements inside another.*

Would it be too much if I suggest that you could possibly make a little less noise?

In general the list of "felicity conditions" on the directive class could be summarised as follows:

Preparatory condition: H is able to perform A.

Sincerity condition: S wants H to do A.

Propositional content: S predicates a future act of H

Essential condition: Counts as an attempt by S to get H to do A

Here H, S, A are abbreviations for 'hearer', 'speaker' and 'action' or 'act'.

Regarding politeness and indirect Speech Acts, which Searle claimed to have a strong connection especially in requests, Davison [Davison, 1975] argued that politeness involves both pleasant and unpleasant things. However, indirect Speech Acts seem to be associated most of the time with bad news, unfavourable opinions, and intrusive questions.

Yet this strong connection between politeness and indirect Speech Acts may not be always true. Macaulay [Macaulay, 1996], in her research based on interviews, mentions that "politeness, which is normally associated with indirectness, would seem to have little role to play in the negotiation of interpersonal meaning between speakers. Indeed, it is desirable for interviewers to be seen as tough and hard-edged in their representation of requests for information." Since being tough, is not a general desire, one can still believe that in most form of communications, indirectness is a sign of politeness.

2.4 Summary

This chapter has presented work written from the point of the philosophy of language. The theory of Speech Acts, which is a basis of the work presented in this thesis, has been explained in this chapter. In addition, some related aspects of pragmatics, as the main foundation for Speech Act recognition, have been reviewed.

The motivations for concentrating on "Request" Speech Act types (as we shall do in what follows) are:

i. As mentioned above, Searle believes that the area of directives is the most useful to study.

ii. Request Speech Acts occur the most in the Email corpus investigated in this research (as we shall show later).

Chapter 3

Representing Semantics with Templates

3.1 Introduction

Natural Language Processing researchers have developed many different approaches for understanding utterances. This chapter reviews work on representing semantics by means of templates and the next chapter will consider templates within the new technology of Information Extraction. A template is a data structure with pre-defined slots which are to be filled in with specific kinds of information.

3.2 Wilks

Wilks [Wilks, 1964,Wilks, 1973,Wilks, 1975a,Wilks, 1975b] described a semantics-based computational system for representing natural language content. The system contained two logical and linguistic methods for expressing the content of any given utterance, and was used as translator between English and French. He argued that any system of analysis must depend on the quality of the dictionary information available to the system and he argued that his lexical entries could express semantic content.

Wilks' system first used a fragmentation technique to break up paragraph-length texts into units like clauses and phrases. A fragmented text was then represented by an interlingual structure consisting of TEMPLATES later bound together by PARAPLATES and COMMON SENSE INFERENCES. All these three items consist of SEMANTIC FORMULAS and the FORMULAS consist of SEMANTIC ELEMENTS, of which there were about eighty.

Items in semantic representation	Built from	Corresponding text items
Formula	Structured elements	English word sense
Template	Formulas	English clause simple surface item
Semantic block	Templates	English paragraph or text

Table 2: The Semantic representation of text items

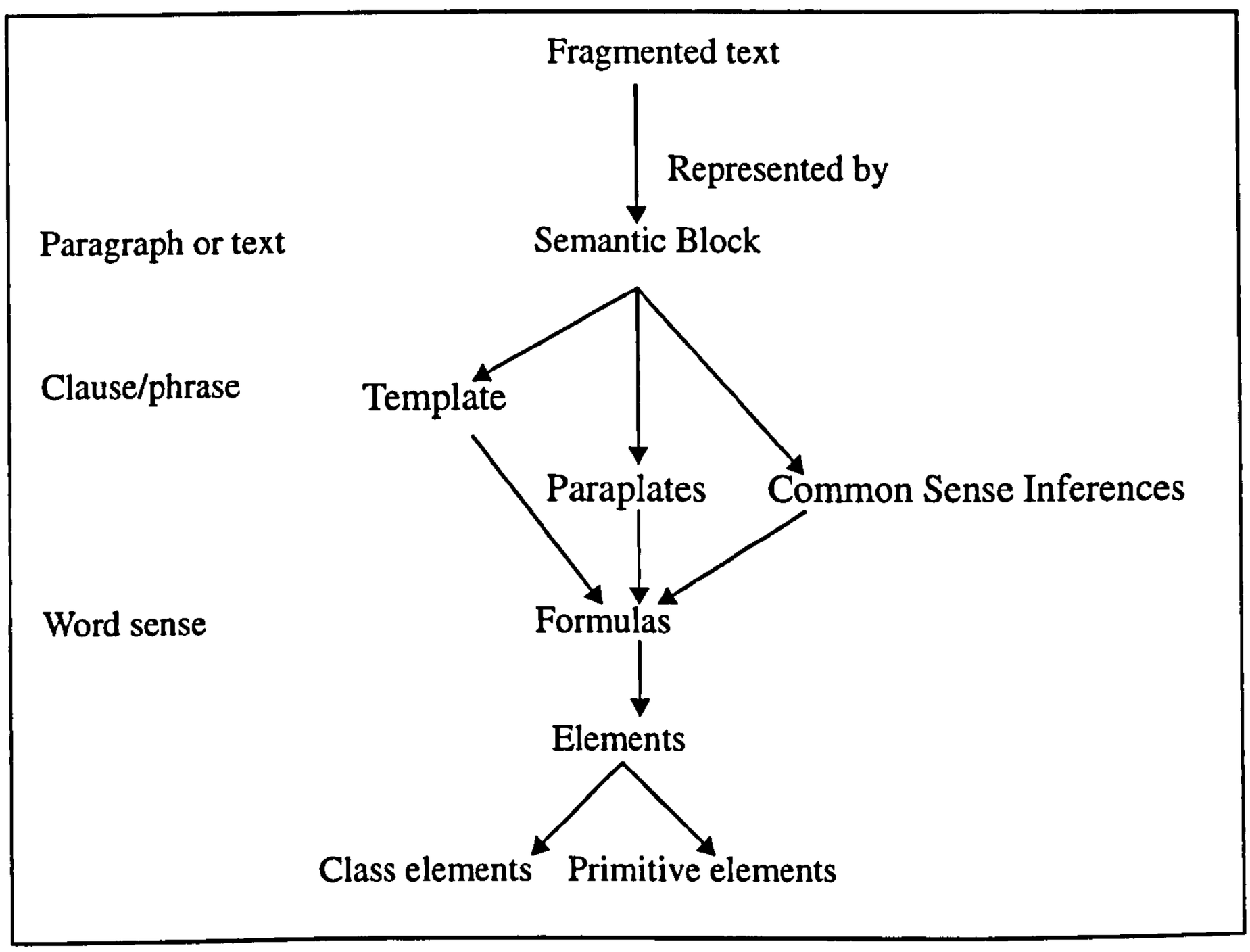


Figure 1: Hierarchical semantics for representing texts

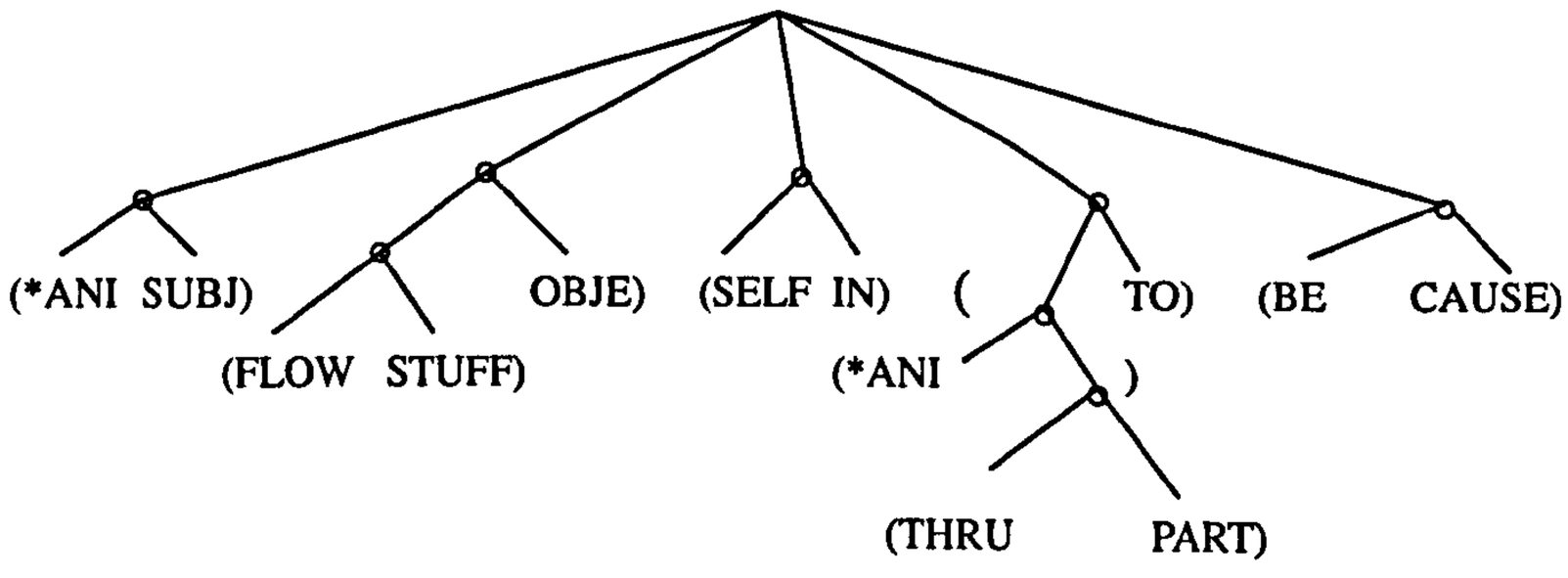
The **primitive elements** are used to express the semantic entities, states, qualities, and actions. The 80 primitives divided into five main groups: entities, actions, cases, type indicators, and qualifiers. Examples are entities: MAN (human being) or THING (physical object), actions: FORCE (compels) or BE (exists), cases: TO (direction) or LOCA (location), type indicators: KIND (being a quality) and finally qualifiers: GOOD (being acceptable) or THRU (being an aperture).

There are about 15 **Class elements** which are distinguishable from primitive elements by

an asterisk prefix to their names. For example *ANI represent the class of animate elements such as MAN or BEAST and HUM represent human elements such as MAN and FOLK.

Formulas express the senses of English words and they are constructed from elements. The head of the formula is the most important element and it appears in the right most position. The formula for the word "drink" as an action primitive is shown below:

"drink" (action) :- ((*ANI SUB) (((FLOW STUFF) OBJE) (SELF IN)
(((*ANI (THRU PART)) TO) (BE CAUSE)))))



Here is a short explanation about different parts of the formula to clarify it:

(*ANI SUB) means the agent is animate, ((FLOW STUFF) OBJE) means the object is liquid, (*ANI (THRU PART)) TO) means the direction of the action is a human aperture, and (BE CAUSE) means the action is of causing to be.

Different meanings of a word are represented by separate formulas. For instance, figure 2 below represents two different meaning of the word "grasp" in a tree form representation:

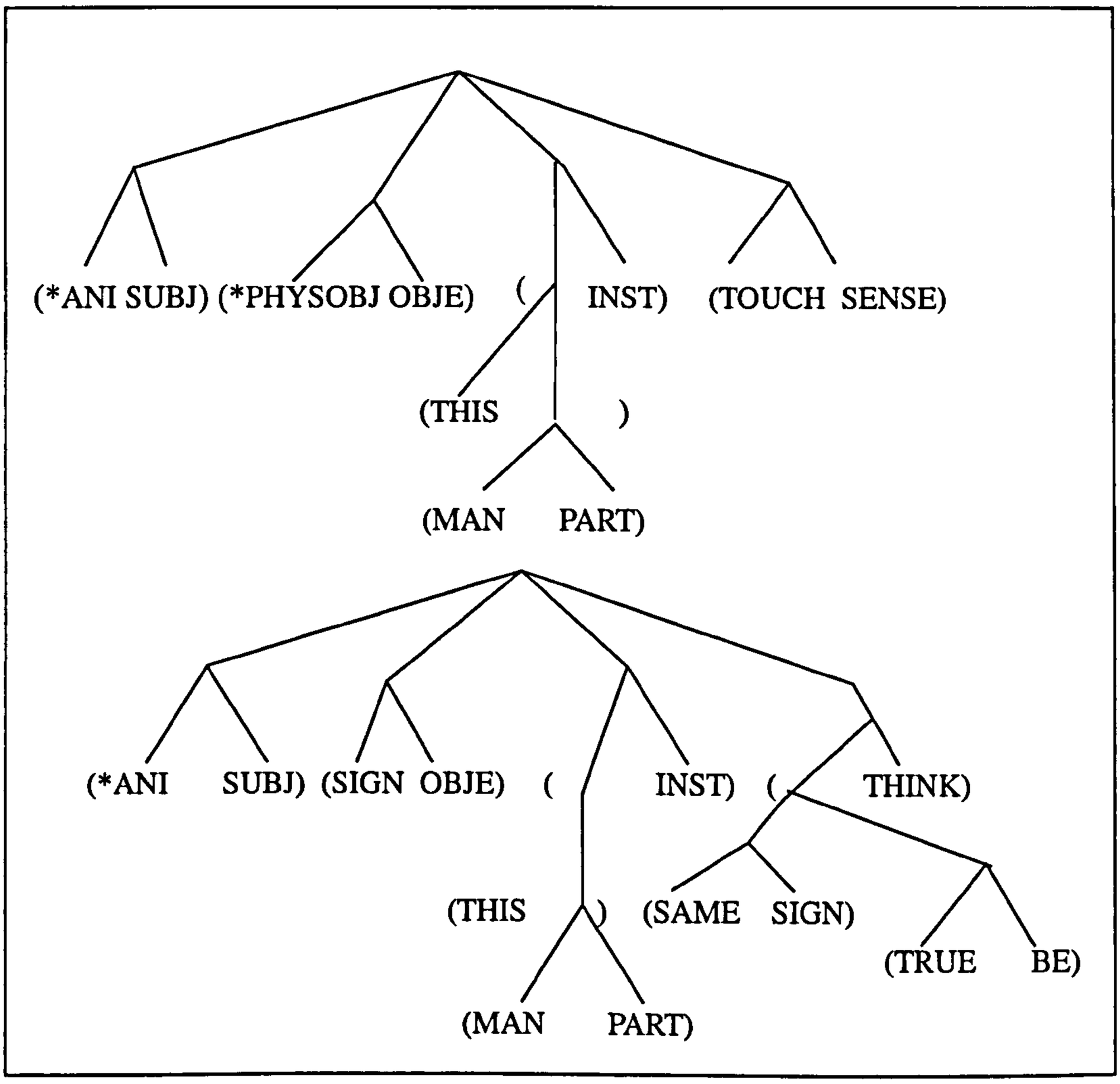


Figure 2: Semantic presentations for the word "grasp"

The upper tree implies that grasping is a kind of thinking action which is done by an animate agent and it is done with an instrument which is part of human body (i.e brain). The lower tree defines grasping as an sensing action, where the object is an physical object and the action is done with an instrument as part of human body (i.e. hand). These examples show how different sense of words can be represented.

"crook" :- ((((NOTGOOD ACT) OBJE) DO) (SUBJ MAN), which represents "a man who does bad acts".

"big" :- ((*PHYSOB POSS) (MUCH KIND))

interrogates" :- ((MAN SUBJ) ((MAN OBJE) (TELL FORCE))), which represents forcing to tell something acted by humans and to humans.

Semantic **Templates** are the third kind of semantic item in the system. Templates form the semantic representation for clause or phrase-length fragments of text. Each template is constructed of at least three formulas corresponding to an: agent, action, and object. For example, in a sentence such as "The crook drank some beer" since there are two semantic formulas for "crook", there are at least two initial representations for this sentence.

In Wilks' theory of Preference Semantics, and the computational model corresponding to it, it is assumed that the coherence of a discourse can be computed from the semantics of individual sentences, where coherence within templates is computed in terms of the preferences of templates for neighbouring formulas, so that the formula for "drink" preferring a human agent, will prefer to be in a template with the "human" formula for "crook" and not the one for the shepherd's staff.

3.3 Schank

Schank introduced a theory of language and language processing by the name of Conceptual Dependency (CD), one also based on the semantic representation of the meaning of a sentence [Schank, 1972] [Schank, 1975] [Lytinen, 1992]. He claimed that his focus is on meaning and not on syntax. Conceptual dependency theory was based on two assumptions:

1. If two sentences have the same meaning, they should be represented in the same way, regardless of the particular words used. For instance "John presented Mary with a ball" and "Mary was given a ball by John" should be represented similarly although they have different agents and different verbs.

2. Any information in the sentence that is implicit must be made explicit in the representation of the meaning of that sentence.

The meaning of a linguistic proposition is called a conceptualization, which can be active or stative. An active conceptualization can be represented as structured by: actor, action, object, and direction, source (from) destination (to) and instrument. A stative conceptualization has object, state and value slots.

Stative conceptualizations are statements with values which are defined by a large number

of scales. These numbers are usually in the range of -10 to 10 and they can be used to show the value of a state for an object. For example, in state of the health of an object, -10 indicates it is dead, while 10 indicates perfect health (and all numbers between them represent different health conditions).

In case of active conceptualizations, Schank classified actions by eleven primitive actions as follows:

1. ATRANS: The transfer of ownership, possession, or control of an object. For example one meaning of the verb "give" is to ATRANS something to someone else, while the verb "take" is to ATRANS something to oneself. Some actions might be defined by more that one primitive action. For example the verb "buy" consists of two ATRANs. An ATRANS of money and ATRANS of an object being bought.

2. PTRANS: The transfer of location of an object. For example the action "go" is to PRANS oneself to a place and "put" means PTRANS an object to a place.

3. PROPEL: The application of a physical force to an object. This primitive is used whenever any force is applied. If the force cause any movement, the action is considered as a PTRANS. For example the verbs "push" and "pull" are PROPEL actions and if the action caused a movement by an object, it is PTRANS. Most PROPEL actions are also PTRANS actions and the inference mechanism will have to decide in each case of PROPEL if PTRANS is applicable too.

4. MTRANS: The transfer of mental information between or within agents. To define as many actions as possible, Schank claims memory is partitioned into three parts. The conscious processor memorises entities which are thought of, the long term memory considers entities to store, and intermediate memory is where the current context is stored. Here are examples of different MTRANS actions:

   The verb "tell" is MTRANSing between people.

   The verb "see" is MTRANSing from the eye to the conscious processor.

   The verb "remember" means MTRANSing from long term memory to the conscious processor.

The verb "learn" is the MTRANSing of new information to the long term memory.

5. MBUILD: The construction of a thought or of new information possibly from old information by an agent. Verbs such as "decide", "consider", "imagine" are all MBUILD actions.

6. ATTEND: The act of focusing the attention of a sense organ on an object. For example, the verb "listen" is an ATTEND-ear and the verb "see" is an ATTEND-eye. ATTEND usually refers to the instrument of MTRANS.

7. SPEAK: The act of producing sound, including non-communicative sounds. Although by this definition, many objects can SPEAK, when referring to human, SPEAKing is a way of MTRANSing. The verbs "say" and "sing" are examples of human SPEAKing.

8. GRASP: The grasping of an object by an actor so that it may be manipulated. While the verbs such as "hold" or "grab" involve GRASP action, the verb "throw" involves ending with a GRASP action.

9. MOVE: The movement of a body part of an agent by that agent. Usually a MOVE action is related to the ACT of the instrumental conceptualization for other actions. In order to "throw" something, which is an GRASP action, it is necessary to MOVE an agent's arm, or in order to "kick" a ball, which is a PROPEL or possibly a PTRANS action, it is necessary to MOVE agent's foot.

10. INGEST: The taking in of an object by an animal. This actions usually refers to eating food or drinking liquid. The most common verbs describe by INGEST are "eat", "drink", and "breathe".

11. EXPEL: the expulsion of an object from the body of an agent into physical world by an agent. Although in general an object can be EXPELd if it has been INGEST before, verbs such as "cry" or "sweat" are in this category.

Among these primitive actions, (6,7,8,9,11) are designated instrumental acts and (2,3,10) are called primary physical actions. Here are some examples to show how the slots of active

and stative conceptualization slots are filled in by information from the sentences. As mentioned before active conceptualizations are represented by actor, action, object, and direction, source (from) destination (to) and instrument and stative conceptualizations have object, state and value slots.

"John went": active

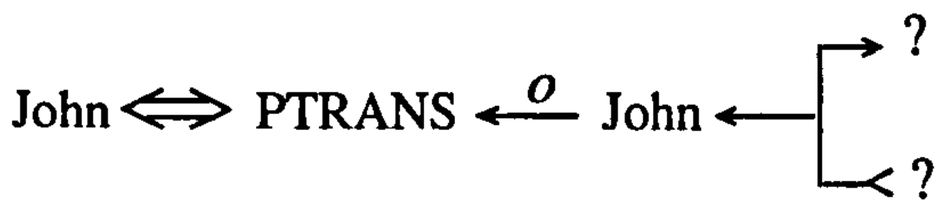
(Actor: John

Action: PTRANS

Object: John

Direction: (From: unknown

To: unknown))



"John is heavy": stative

(Object: John

State: WEIGHT

Value: over average)

"John kicked the cat": active

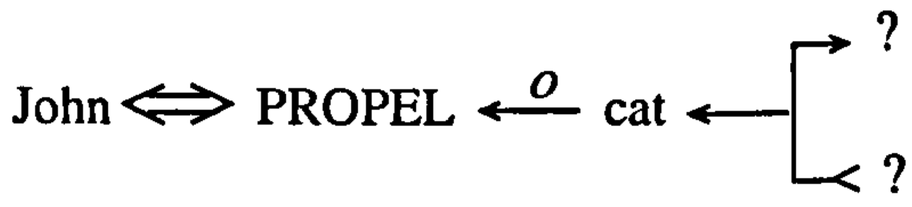
(Actor: John

Action: PROPEL

Object: cat

Direction: (From: unknown

To: unknown))



"John donated blood to the Red Cross": active

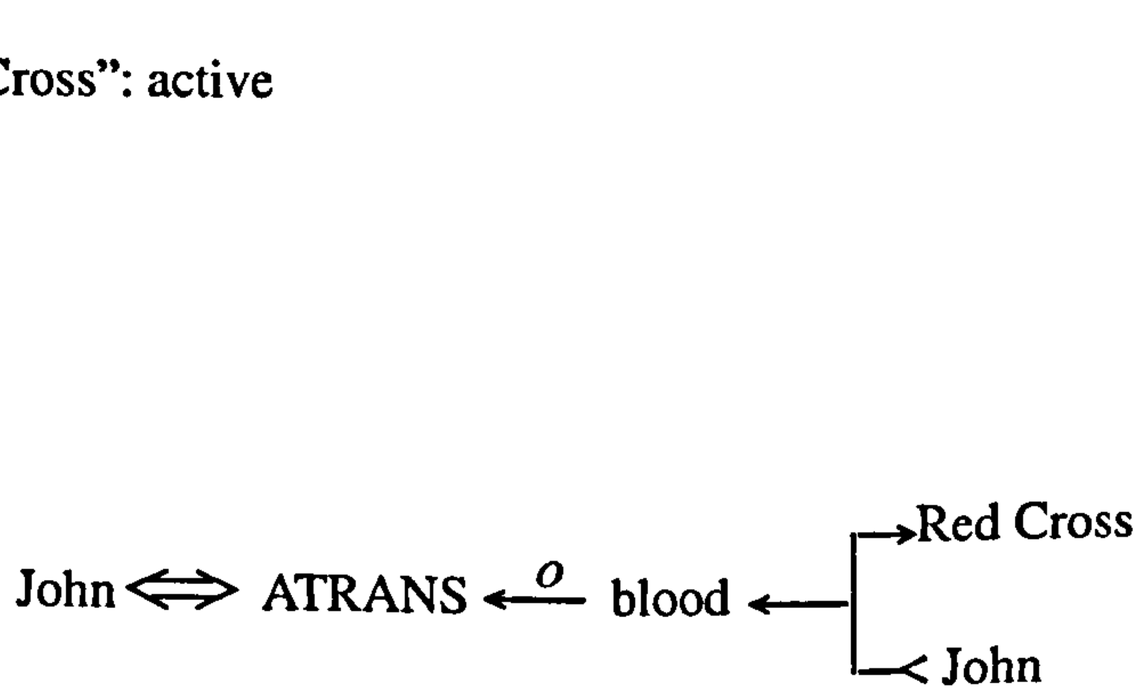
(Actor: John

Action: ATRANS

Object: blood

Direction: (From: John

To: Red Cross))



"John hit Bill with his hand": active

(Actor: John

Action: PROPEL

Object: Bill

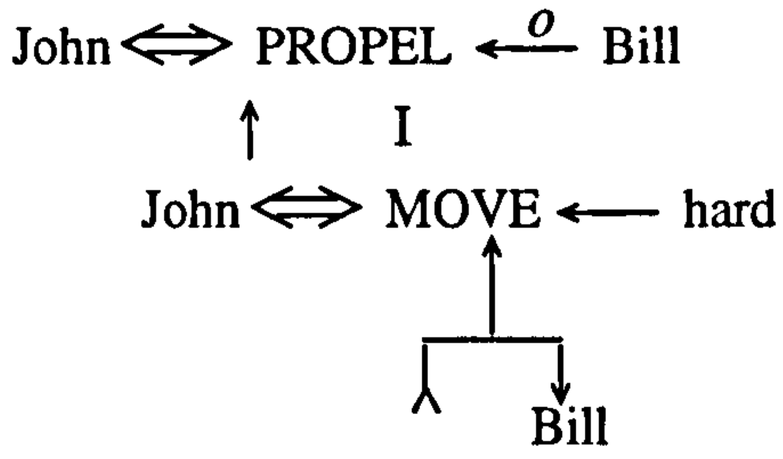
Instrument: (Actor: John

Action: MOVE

Object: hand

Direction: (From: John

To: Bill)))



This system was implemented by Riesbeck [Riesbeck and Schank, 1976] as essentially a template-system, though this term was never used, with structures for each primitive opening up slots of a certain type to be filled by entities mentioned in the text, though there was no analogue to a preference algorithm for determining how alternative fillers for slots were to be assigned, since there were no analogues to Wilks' formulas to provide alternative senses for words.

Schank also proposed (after Minsky) the notion of scripts as standard event sequences. Schank and Abelson [Schank and Abelson, 1977] defined a script as a structure that describes appropriate sequences of events in a particular context, such as going to a restaurant or going to a birthday party. Using scripts gives quick access to those events in the context of a (partially) ordered event set, which happen in a stereotypical event sequence, thus avoiding other inferences which would most likely be irrelevant ones.

A script consisted of a set of roles, common objects used, and scenes, each of which described the typical events in one portion of the script. For example, in defining a restaurant script, the roles are the customer and the waiter/waitress, and objects are restaurant and the food, and scenes are ORDER, EAT, PAY, and LEAVE.

In summary Schank argued that, by using one or more (as in the last examples) conceptual dependency structures (possibly ordered within scripts), natural language sentences can be mapped into an internal representation based on semantic structures.

3.4 Lehnert

Lehnert began as a student of Schank, but later incorporated features of Wilks' structures into her system, such as representation of noun senses and an algorithm for computing preferences, which she made learnable rather than a priori. She also reintroduced syntactic structures into Conceptual Dependency.

Lehnert [Lehnert, 1987] discussed the notion of semantic preference from a knowledge acquisition perspective, and presented a system for computing semantic preferences based on what she called dynamic memory structure. She implemented a sentence analyser, ELAN (Episodic Language Acquisition Network), which turns to the episodic memory structures that it creates as it learns. Two modes were defined for ELAN: a training mode, and a test mode. In training mode, ELAN received sentences with target case frame representations based loosely on Schank and Wilks notions for those sentences and created a memory structure called an "integration map sequence". These maps are used later during test mode. During the test mode, ELAN used integration map sequences as a source of syntactic knowledge and semantic preferences. For example in

"*Paul ate rice with chopsticks.*"

The target meaning representation was:

(EVENT(eat) AGENT(Paul) OBJECT(rice) INSTRUMENT(CHOPSTICKS))

and syntactic structure was:

(NP V NP PP)

The Integration map sequence was:

(NP/AGENT V/EVENT NP/OBJECT PP/INSTRUMENT)

In the same way sentences like "John ate pizza with Mary." and "Mary ate spaghetti with meatballs." create different integration map sequences for the form (NP V NP PP):

(NP/AGENT V/EVENT NP OBJECT PP/INSTRUMENT)

(NP/AGENT V/EVENT NP OBJECT PP/CO-AGENT)

(NP/AGENT V/EVENT NP OBJECT PP/CO-OBJECT)

Since ELAN does not operate with pre-defined knowledge structures in the form of a semantic memory or inheritance hierarchy, all semantic preferences from its training corpus are derived with the use of "binding pools". A binding pool is a collection of slot fillers that have been used in conjunction with a specific integration map at some time during training. For those three sentences mentioned above, the binding pools are:

NP/AGENT (Paul John Mary)

NP/OBJECT (rice pizza spaghetti)

PP/INSTRUMENT (chopsticks)

PP/CO-AGENT (Mary)

PP/CO-OBJECT (meatballs)

These binding pools gave the basis for semantic preferences when ELAN operated in test mode. So, if ELAN trained long enough on enough sentences, it should provide reasonable behaviour in sorting out various ambiguities. Two basic problems with this strategy were 1) the amount of training required is unreasonable and 2) memory is not being used very efficiently.

3.5 Summary

This chapter reviewed some semantic-based approaches to Natural Language Processing. The shared view in these approaches is the use of semantic templates in knowledge and meaning representation. Chapter Five, especially its Information Extraction section, also discusses templates as a way to extract information from text on a large and practical scale.

Chapter 4

A Computational approach to Speech Acts

4.1 Introduction

Many theories and computational models have been developed based on the notion of Speech Acts and their relation to plans expressed in natural language. A plan for achieving goals normally consists of sequences of actions where each action (conventionally) has its preconditions and effects. Each action might also have some simpler actions as steps. For example, in a computational environment, EDIT (Agent, File, Instrument) is an action with preconditions that the Agent is a human, the File is accessible and the Instrument is a text editor. The effect of the whole action is an edited file.

Allen [Allen, 1987] suggests that a Speech Act is successfully performed by a speaker S by saying utterance U to a hearer H if and only if:

1. The preconditions of Speech Act hold.
2. Saying U to H accomplishes the effect of Speech Act.
3. S intended that condition 2 would be the case.
4. S intended that H would recognize S's intention in condition 3.

and, from another point of view, the significance of a Speech Act depends on:

1. A Speech Act and its literal meaning
2. The sequence of Speech Acts, both before and after that particular Speech Act.
3. The general context and purpose of the dialogue.

4.1.1 Cohen, Allen, and Perrault

The work of Cohen, Allen and Perrault demonstrated that Speech Acts can be seen as plan operators within a computational model. They argued that understanding an utterance consists of recognising the underlying plan of the agent. They also suggested that a theory of Speech Acts based on plans should specify the following:

1. A planning system which consists of a formal language for describing states of the world and describing operators.
2. Definitions of Speech Acts as operators in the planning system.

They did more than simply implement Searle's theory, described earlier, by presenting a new definition for Speech Acts to make them independent of the speaker (see section 4.1.1.), since in the process of clarification and implementation they unified the point of view confusion (between H and S) that we noted earlier (section 2.5) when discussing Searle.

4.1.1.1 Cohen and Perrault

Cohen and Perrault [Cohen and Perrault, 1979] illustrate methodological issues of how Speech Acts should be defined in a plan-based theory by defining operators for the two Speech Acts requesting and informing. They define plans as sequences of actions, where each action has preconditions, effects, and bodies. Preconditions, effects and bodies are evaluated within the model of the world. In general, the preconditions of an operator should be stated from the speaker's point of view in terms of speaker's beliefs and the effects should be stated from the hearer's point of view.

They argued that a theory of Speech Acts based on plans should specify a planning system that defines Speech Acts as operators within the planning system. They introduce two classes of preconditions for all operators: CANDO.PR and WANT.PR. CANDO refers to propositions that must be true within the world model for that operator and WANT.PR formalizes a principle of intentional behaviour. They define REQUEST Speech Act as:

REQUEST (SPEAKER, HEARER, ACT)

CANDO.PR:SPEAKER BELIEVE (HEARER CANDO ACT)

AND

SPEAKER BELIEVE (HEARER CANDO ACT)

WANT.PR:SPEAKER BELIEVE (SPEAKER WANT REQUEST-INSTANCE)

EFFECT:HEARER BELIEVE (SPEAKER BELIEVE (SPEAKER WANT ACT))

Inform has been defined as:

INFORM (SPEAKER, HEARER, PROP)

CANDO.PR:SPEAKER BELIEVE PROP

WANT.PR:SPEAKER BELIEVE (SPEAKER WANT INFORM-INSTANCE)

EFFECT:HEARER BELIEVE (SPEAKER BELIEVE PROP)

As to questions, they argued that questions can be treated as requests for information. In other words, questions can be seen as a REQUEST that the hearer perform an INFORM. For wh-questions two new operators INFORMREF and CONVINCEREF were defined. They then illustrated how a plan for a wh-question can be built up using these two operators. In the same way, the plan for yes/no questions is defined by using two new operators INFORMIF and CONVINCEIF. They also described plans for multi-party Speech Acts where more than two agents are involved: for example, "ask Tom to tell me where the key is".

Cohen and Perrault then present a new definition for Speech Acts to make them independent of the speaker, which it means that for preconditions, no CANDO.PR or EFFECT should be stated as a proposition beginning with 'SPEAKER BELIEVE'. Their new definitions for INFORM and REQUEST are as follows:

INFORM (SPEAKER, HEARER, PROP)

CANDO.PR:PROP

WANT.PR:SPEAKER BELIEVE (SPEAKER WANT INFORM-INSTANCE)

EFFECT:HEARER BELIEVE (SPEAKER BELIEVE PROP)

REQUEST (SPEAKER, HEARER, ACT)

CANDO.PR:HEARER CANDO ACT

WANT.PR:SPEAKER BELIEVE (SPEAKER WANT REQUEST-INSTANCE)

EFFECT:HEARER BELIEVE (SPEAKER BELIEVE (SPEAKER WANT ACT))

4.1.1.2 Allen and Perrault

Allen and Perrault [Allen and Perrault, 1979] explained a plan-based model for natural language dialogue specialised to question-answering. They argue that a good question-answering system often needs to provide more information than strictly required by the question. This model provides the mechanisms to explain these aspects of language use:

The generation of responses that provide more information than required.

The generation of responses to sentence fragments.

The analysis of indirect Speech Acts.

Their definitions for actions, plans, and Speech Acts are very similar to those of Cohen and Perrault that I gave in the previous section.

They introduce two processes that a system must have: plan construction and plan inference. Their method for constructing a plan is backwards chaining: given a goal G, find an action A that has G as one of its effects, then evaluate the preconditions of A and, if some of these conditions are not satisfied in the initial state, they become new goals and the plan construction process repeats.

Plan inference rules are divided into three categories: rules concerning actions, rules concerning knowledge, and rules concerning planning by others. Here is their example of the treatment of indirect Speech Acts within this framework. Suppose these sentences express a question and the expected answer:

A: Do you know when the Windsor train leaves?

S: Yes, at 3:15

The goal inferred from the literal interpretation is that

A KNOWIF (S KNOWREF 'departure time').

Applying the know-positive rule, we obtain the goal

S KNOWREF 'departure time'

which enables the planer (for S) to perform the action (via the precondition-action rule)

INFORM (S, A, 'departure time')

to achieve the goal (via the action-effect rule)

A KNOWREF 'departure time'

In summary, they implemented a simple question-answering system for understanding and acting as an information clerk at train station. This system was able to distinguish the beliefs and wants of the user from its own, and could model elementary indirect Speech Acts.

4.2 Hinkelman

Hinkelman [Hinkelman, 1989] [Hinkelman and Allen, 1989] considered syntactic and semantic information so as to recognise Speech Acts as part of a general theory of plan-based reasoning. She argued that it is necessary to include pragmatic rules within any model of Speech Act interpretation so as to recognise Speech Acts correctly, usually based on the role of specific word cues (a very important notion in Information Extraction (section 5.2)). For instance:

Can you speak Spanish?

Can you speak Spanish, please?

While the first of these can be interpreted as either a request to speak Spanish or as a yes / no question, in the second one the word "please" forces the utterance to be interpreted as a request. There are other examples to show the role of such pragmatic cues in the interpretation of utterances.

Can you open the door?

Are you able to open the door?

While the first of these could be either a request or a yes / no question, the second must be yes / no question. While the above examples show the insufficiency of Speech Act theory alone to recognise a speaker's plan, surface linguistic information taken alone is not sufficient either. For example, an utterance such as "it is cold in here" might be a statement or a request from hearer to close the window. Hinkelman argued, based on the above examples, that a combination of linguistic rules plus Speech Act theory is necessary to recognise such Speech Acts correctly. Within her model, an utterance is parsed and linguistic information is used to

predict all possible Speech Acts. In cases of ambiguity, plan-based reasoning is used to select the best Speech Act from the set. So, in her model "Can you do X?" is recognised by the following steps:

> Since the mood is interrogative, the subject is "you" and the modal verb is "can", the suggested interpretation is a request to do X.

Hinkelman's model for interpreting Speech Acts integrates plan-based reasoning with syntactic and semantic analysis, and she argued that such a technique could be used even for spoken data including intonation.

4.3 The TRAINS Project

The TRAINS project is one of long-term research to develop an intelligent planning assistant that is conversationally proficient in natural language [Traum et al., 1994] [Heeman and Allen, 1995].

The TRAINS System helps a user construct and monitor plans about a railroad freight system. Since 1990 there have been several TRAINS systems developed by this team.

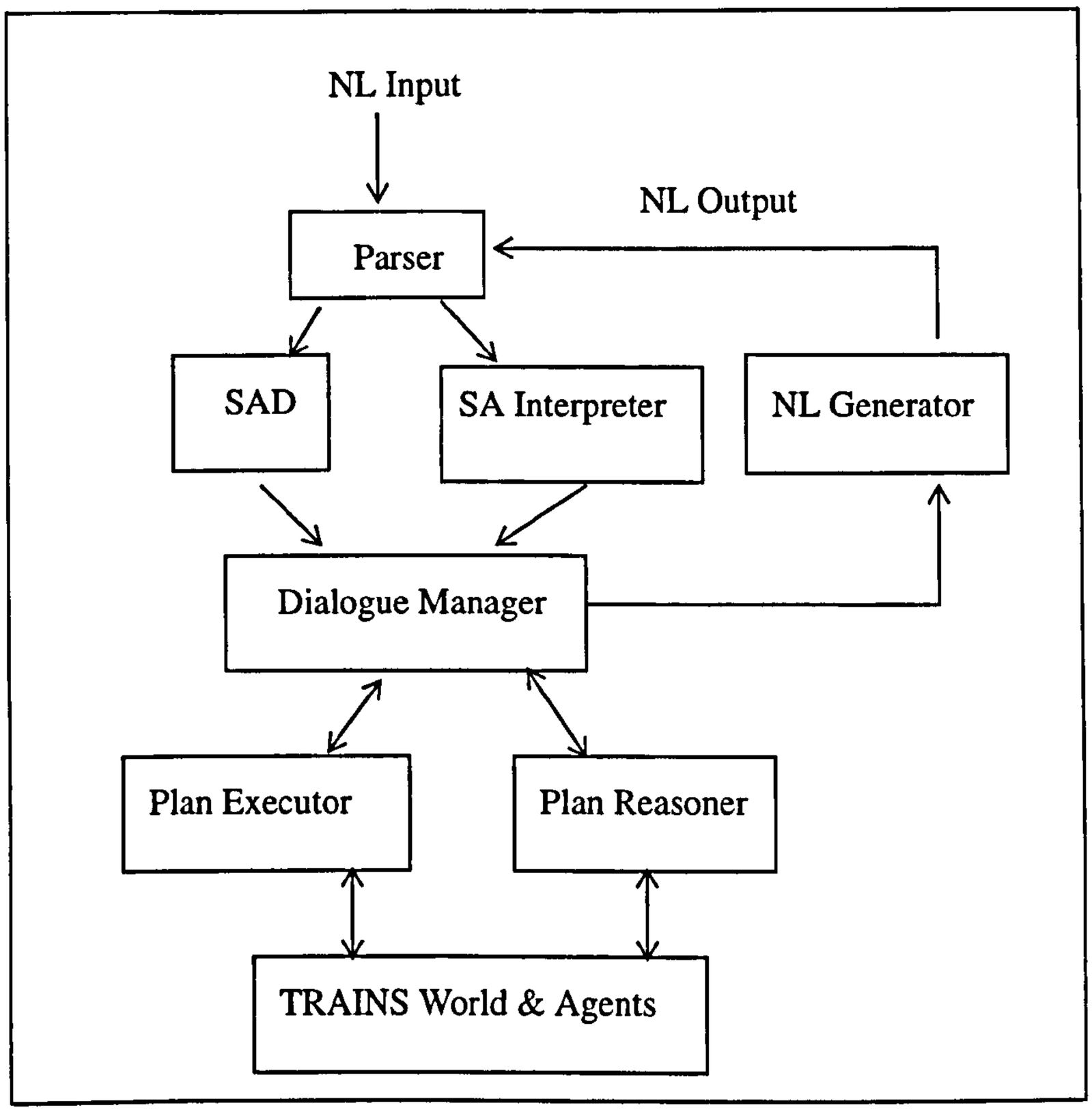


Figure 3: The TRAINS System Architecture (1993)

Parser takes an utterance and produces a representation that combines the result of syntactic analysis and lexical interpretation. Each rule in the grammar consists of a syntactic rule coupled with a corresponding semantic rule. They claimed that the parser achieved a 99% accuracy rate when run on the training data with no sentences failing to parse. [Allen et al., 1995]

Scope and Deindexing (SAD) deindexes context-dependent aspects of an utterance's content such as referential expressions. Its input is produced by the parser and its output is a set of alternative hypotheses about how to resolve the ambiguity suggested by the given context. The theory of discourse interpretation on which the deindexing module is based is called Conversation Representation Theory (CRT).

Speech Act Interpreter is responsible for determining what a speaker means by an utterance. It takes semantic interpretations of utterances and recognises which acts have been per-

formed by the speaker in constructing the utterance. The Speech Act interpreter builds a list of hypotheses about the Speech Act interpretations of an utterance. So far, only the Core Speech Acts module [Poesio and Traum, 1997] has been implemented. Speech Act recognition consists of two stages: the Speech Act Interpreter provides a list of all possible acts based on the linguistic form of utterance; then the Speech Act Pruner produces a list of acts which have actually been determined to occur based on contextual information [Traum, 1993]. The following are the core Speech Act types implemented in the system:

Inform: Speaker presents Hearer new Knowledge.

Ynq: Speaker asks Hearer to provide information Speaker is missing but suspects Hearer may know.

Check: Like a Y/N question, but Speaker already suspects the answer.

Suggest: Speaker proposes a new item as part of a plan.

Request: Like a Suggest, but imposes a discourse obligation to respond.

Accept: Speaker agrees to a proposal by Hearer.

Reject: Speaker rejects a proposal by Hearer.

Dialogue Manager (DM) maintains the flow of conversation, and the domain plan reasoner allows the system to reason about the TRAINS domain. In this system, the main goal is an executable plan agreed by both the system and the user to meet the user's goal [Traum, 1993].

On deciding which action should be done next, the Dialogue Manager considers obligations first. In the case of no obligations, it will consider the possible intentions and perform related actions if any are found. The Dialogue Manager's decision is based on the following priorities:

1. Discourse obligations resulting from Speech Act effects.

2. A general obligation not to interrupt the other's turn.

3. Intended Speech Acts. When the system has the turn but does not have any pending discourse obligations, it will perform acts which have been planned but not yet generated.

4. A general obligation to ground or fulfil expressed content.

5. Discourse goals of domain plan negotiation.

6. High-level Discourse Goals. Given no higher priority items, the system will attempt to further the conversation.

Domain Plan Reasoner provides planning and plan recognition services and performs reasoning about the state of the world. It provides an algorithm that attempts to find causal and motivational connections between potential interpretations of the current utterance and the current plan. It provides a question-answering facility about the current plan and the state of the world.The dialogue manager uses the results of plan reasoning to disambiguate Speech Acts interpretations, update beliefs and generate new conversational elements. More specific types of utterances that it interprets are:

Suggestions: utterances that suggest actions, e.g. "Send engine E3 to Dansville".

Goals: utterances that identify goals of the plan, e.g. we have to make OJ".

Plan Executor takes a plan and sends the necessary commands to the individual agents. Once a plan has been agreed to by the system and the manager, the DM hands the plan to the execution planner. It also aids the planner by making choices among a set of alternatives presented by the planner. In addition, it gathers the information that allows it to make better choices on subsequent queries. Since the TRAINS system cannot directly cause any of the events specified in the plan, it must request of the agents to cause these events.

TRAINS World is a detailed simulation of action executions by agents of world actions and is used for plan execution and monitoring.

NL Generation takes Speech Acts representations produced by the dialogue manager and converts them to natural language text.

4.4 Summary

This chapter reviewed previous research programs that considered speech acts and formalised them as planning operators in computational systems. Section 4.1.1 concentrated on Inform and Request Speech Acts, which we will deal with later in chapter 9, and section 4.2 sug-

gested a combination of pragmatic rules within any model of Speech Act interpretation, which is made use of in chapter 9.

Chapter 5

Text Processing

The amount of textual information electronically available today has made it impossible for anybody to find or extract relevant information easily. In order to overcome this problem, new technologies for information systems have been explored by various researchers. This chapter reviews different lines of such research on text processing, as a preliminary to incorporating some of their functions into the Pyam system.

5.1 Information Retrieval

As with other sub-areas of Natural Language Processing, the need for Information Retrieval increases because of the increasing availability of on-line documents on a huge scale. This technique is also known as Text Retrieval or Document Retrieval. Document Retrieval is for the user who wants to learn something by reading about it, so document retrieval must find relationships between the information needs of users and the information in the documents. Obviously, this definition distinguishes between DR and extracting specific pieces of data from a text (as in Information Extraction) as pre-defined information or an answer to a question. The main goal of any information retrieval system is to increase two evaluation factors called precision and recall. Precision is the proportion of retrieved documents that are relevant and recall is the proportion of relevant document that are retrieved.

All information retrieval systems, regardless of the techniques they employ should have the following abilities [Evan and Zhai, 1996]:

1. An ability to process a large amount of text.

IR systems deal with gigabytes of text. Processing this huge amount of information requires an efficient system both in time and space requirements.

2. An ability to process unrestricted text.

The IR task is usually involved with unrestricted text from different domains. The system should be able to manage different types of documents and be able to process unknown words, proper names and unrecognised structures.

3. A need for shallow understanding.

Although the two, above mentioned, specifications make an IR task harder, the nature of any IR task, compared with other NLP applications, such as Machine Translation or Information Extraction, requires only a shallow understanding of the text. Since the main goal of an IR system is to classify documents as relevant and irrelevant with respect to a query, a deep understanding of the text is not necessary.

IR techniques were initially developed for the retrieval of references to documents from bibliographic databases [Salton, 1983]. These techniques have also proved applicable to any sort of textual information such as technical manuals, reports of meetings and, more recently available, multimedia information systems.

One common IR task is Boolean retrieval, where the query terms are linked by logical operators (AND, OR, and NOT) together with pattern matching facilities to catch any relevant information [Willett and Ingwersen, 1994]. However, there are disadvantages to the Boolean model: the lack of control over the size of output generated by the system is one major disadvantage of this approach, especially when there is a large amount of data to search. Another disadvantage is the way that the system divides all information into two discrete categories: related and unrelated. There is no way to measure any piece of information as less or more related to the query. A similar problem arises in preparing queries: all the terms presented in a query are considered as having the same importance and the absence of a term has no value in the search.

An alternative method developed for text searching is called the Best-match searching method. This technique involves ranking a database of documents in order of decreasing sim-

ilarity to a query. The Best-match search compares a set of query terms with the set of terms corresponding to each of the documents in the database, calculates similarities between the query and each of documents, based on the terms that they have on common, and finally sorts the documents into order of decreasing similarity to the query.

This model has the following advantages over Boolean model:

1. An unstructured list of input terms is sufficient instead of having to specify Boolean relationships between the terms.

2. The end user is able to control the amount of data output, since the documents are ranked and sorted by their similarity to the input terms.

Indexing is also a well known approach to DR, one which requires an indexing language with a term vocabulary together with a method for constructing requests and document descriptions. Manual indexing uses syntactic and semantic analysis of the texts and queries. Early work on the SMART project [Salton and Buckley, 1988] [Salton and Buckley, 1990] showed that inferred keywords gave levels of retrieval performance that were comparable with those obtained from the manual application of controlled vocabularies or of phrase-based indexing.

Recently, statistical DR methods, which enhance the use of representations based on single terms, have provided significant improvements. Statistical DR methods rank documents by their similarity to the query or on an estimate of the probability of their relevance to the query, where both query and document are treated as collections of numerically weighted terms. Automatic indexing could be achieved by using statistical information about the frequencies with which terms occur. It should be noted that words which occur very frequently in documents cannot distinguish between relevant and non-relevant documents. On the other hand, those words that occur rarely in documents might be good terms for indexing, but carry the risk that they do not appear in the queries. Considering these two points, the most useful words for retrieval purposes are those with intermediate frequencies of occurrence.

Research showing the effectiveness of statistical DR methods appears promising in tests done in various environments [Lewis and Sparck Jones, 1996]. Nowadays, one advantage of

using statistical techniques is that a large number of on-line documents are available and make it easy to identify words which can be used as index terms.

There are general problems with word-based techniques which make them limited and somewhat domain dependent [Riloff and Lehnert, 1994]:

1. Different words and phrases having similar meanings is a well-known limitation of word-based techniques. For example word "make" and "produce" could refer to the same entities. Even in a domain-dependent system, this problem can cause problems.

2. Individual words have different meanings: "court", "bank", and "post" are examples of words with multiple meanings. Part of this problem can be solved in a domain dependent system: in a system dealing with financial resources, the word "bank" is more likely to refer to a financial institution or a building which people go for financial problems than to a river side.

3. Sequences of words in a phrase may have no relation to the words used individually. For example, the meaning of the phrase "pass away" is different from the meanings of "pass" and "away".

4. In some documents, it is hard to find any word or phrase good enough to serve as an indexing term, although the whole sentence or paragraph is a coherent and meaningful text. For example "an armed man took the money and fled" has the meaning of robbery without containing any word for indexing it.

5.2 Information Extraction

There are too many texts in different electronic forms for any human to read, understand, or summarise on an everyday basis. Information Extraction (IE) is a process which takes unsorted texts as input and produces pre-defined data as output. In other words, IE prepares structured information sources from any unstructured text information source. This structured information can then be used for different Natural Language Processing purposes.

In spite of a similarity at first glance between Information Extraction and Information Retrieval, these are two quite different tasks. Not only do they differ in their objectives, they

also do so in their techniques: Information Retrieval retrieves relevant documents (or parts) from collections while Information Extraction culls relevant information from the texts of individual documents. As to the techniques they employ, Information Extraction uses rule-based systems from computational linguistics and Natural Language Processing, while Information Retrieval systems are based on information theory, probability theory and statistics. The main difference between Information Extraction and Information Retrieval is that, while IR simply finds relevant texts and presents them to the user, IE systems search texts and prepare specific (usually factual) information (defined by the user) from them. For example, if a user is interested in a specific chemical item, IR systems can, in principle, collect all available texts about it, but IE systems prepare pre-defined information related to that item based on templates pre-defined in the system, such as the names of the companies that produce this item, its price and so on. The goal of IE research is to build systems that find relevant information and ignore irrelevant information. The example in figure 4 below [Califf and Mooney, 1997] illustrates how a template should be filled with relevant information.

Posting from Newsgroup

Telecommunications. SOLARIS Systems Administrator. 38-44K

. Immediate need.

Leading telecommunication firm in need of an energetic individual to fill the following position in the Atlanta office:

SOLARIS SYSTEM ADMINISTRATOR

Salary: 38-44K with full benefit

Location: Atlanta Georgia, no relocation assistance provided

Filled Template

Computer_science_job

title: SOLARIS Systems Administrator

salary: 38-44K

state: Georgia

city: Atlanta

platform: SOLARIS

area: telecommunications

Figure 4: Sample message and filled templates

Although Information Extraction appears to be a new idea, its history goes back at least as far as 1964, [Wilks, 1964]. From a Natural Language Processing point of view, IE is attractive for the following reasons:

i. Extraction tasks are well defined.

ii. IE uses real-world not artificial text.

iii. IE poses difficult and interesting NLP problems.

iv. IE performance can be compared rigorously to human performance over the same task.

5.2.1 The history of IE Systems

FRUMP was one of the earliest IE systems: implemented by DeJong [DeJong, 1979] [DeJong, 1982], FRUMP sought to match each incoming news story with a relevant script on the basis of keywords and conceptual sentence analysis, using Schank's [Schank, 1972] theory of conceptual dependency.

Before FRUMP, a project for extracting information from texts was directed by Sager [Sager, 1981], which combined surface syntax analysis and the use of templates, and was supposed to convert patient discharge summaries to a suitable data-base form.

In 1980, Dasilva and Dwiggins [DaSilva and Dwiggins, 1980] extracted satellite-flight information from reports produced by monitors around the world, but the system was restricted to single sentences and lacked a methodology for extracting complete event descriptions. In the early 1980s, Zarri [Zarri, 1983] worked on texts that described the activities of various French historical figures. The system sought to extract information about relationships and meetings between these people. Cowie [Cowie, 1983] developed an IE system that extracted canonical factual structures from field-guide descriptions of plants and animals.

The main difference between the systems developed in the 1980s and those developed more recently is the decrease in the amount of time and energy needed to collect relevant documents and to create sets of templates. For more detail about Information Extraction see [Cowie and Lehnert, 1996] [Cunningham, 1997] [Gaizauskas and Wilks, 1998].

5.2.2 The Message Understanding Conferences (MUC)

The most significance improvement in Information Extraction happened when ARPA, the US defence agency, funded research groups to pursue IE, and established a regime to evaluate the results. So far there have been seven Message Understanding Conferences for this and the last was in spring 1998.

The first Message Understanding Conference (MUC-1) was held in 1987, when twelve training reports and two unseen messages were prepared to test the systems. No specific task was defined and there was no official evaluation of the system. In 1989, the second MUC was concerned with extracting information from a small number of short naval messages [Sundheim and Chinchor, 1993]. Eight systems participated and, to evaluate the result of the systems, templates were filled manually and scoring was done by participating sites.

Two years later, in 1991, the third conference (MUC-3) was held in San Diego with fifteen systems. The domain was defined as stories about terrorist attacks in Latin American countries, and this time the database was prepared from an electronic newswire. The training sample had 1,300 texts and each system was evaluated on the basis of an unseen test set consisting of 100 new documents. For each text in the training corpus, a hand-coded template with 18 slots was prepared. Figures 5 and 6 below show a message and its corresponding filled templates.

TST2-MUC3-0069

BOGOTA, 7 SEP 89 (INFRAVISION TELEVISION CANADA1) - [REPORT] [MARIBEL OSORIO] [TEXT] MEDELLIN CONTINUES TO LIVE THROUGH A WAVE OF TERROR. FOLLOWING LAST NIGHT'S ATTACK ON A BANK, WHICH CAUSED A LOT OF DAMAGE, A LOAD OF DYNAMITE WAS HURLED AGAINST A POLICE STATION. FORTUNATELY NO ONE WAS HURT. HOWEVER, AT APPROXIMATELY 1700 TODAY A BOMB EXPLODED INSIDE A FAST-FOOD RESTAURANT.

A MEDIUM-SIZED BOMB EXPLODED SHORTLY BEFORE 1700 AT THE PRESTO INSTALLATIONS LOCATED ON [WORDS INDISTINCT] AND PLAYA AVENUE. APPROXIMATELY 35 PEOPLE WERE INSIDE THE RESTURANT AT THAT TIME. A WORKER NOTICED A SUSPICIOUS PACKAGE UNDER A TABLE WHERE MINUTES BEFORE TWO MEN HAD BEEN SEATED. AFTER AN INITIAL MINOR EXPLOSION, THE PACKAGE EXPLODED. THE 35 PEOPLE HAD ALREADY BEEN EVACUATED FROM THE BUILDING AND ONLY ONE POLICEMAN WAS SLIGHTLY INJURED; HE WAS THROWN TO THE GROUND BY THE SHOCK WAVE. THE AREA WAS IMMEDIATELY CORDONED OFF BY THE AUTHORITIES WHILE THE OTHER BUSINESSES CLOSED THEIR DOORS. IT IS NOT KNOWN HOW MUCH DAMAGES WAS CAUSED; HOWEVER, MOST OF THE DAMAGE WAS OCCURRED INSIDE THE RESTAURANT. THE MEN WHO LEFT THE BOMB FLED AND THERE ARE NO CLUES AS TO THEIR WHEREABOUTS.

Figure 5: Example of MUC-3 messages

The fourth conference, MUC-4, [ARPA, 1991] [ARPA, 1992] was held in1992 and fifteen systems participated. The domain, Latin American terrorism, and the structure of the template remained unchanged, and a significant improvement took place in the evaluation of the systems.

One year later, in August 1993, MUC-5 was held, and differed from the previous confer-

ences in many ways. For the first time that this conference became an international conference, rather than just an American one. Of the seventeen systems which participated, one was British, one Japanese, one Canadian and the fourteen remaining systems were from the US. The structure of the templates for MUC-5 were more complex than previous ones. In contrast to the "flat" templates used up to this point, the slots were allowed to have pointers to other slots, and there were eleven objects and 49 slots to be filled by participants. For the first time some of the participants worked on both English and Japanese, and the English text was prepared from the Wall Street Journal for both training and test texts. Useful additional data were provided, such as lists of countries (244), nationalities (216), international organisations (175), female forenames (4967), and male forenames (2924) among other forms of information [Gaizauskas and Wilks, 1998].

0.MESSAGE ID	TST2-MUC3-0069
1.TEMPLATE ID	1
2. DATE OF INCIDENT	(06 SEP 89) / (06 SEP 89 - 07 SEP 89)
3. TYPE OF INCIDENT	ATTACK
4. CATEGORY OF INCIDENT	?TERRORIST ACT
5. PERPETRATOR: ID OF INDIV(S)	-
6. PERPETRATOR: ID OF ORG(S)	-
7. PERPETRATOR: CONFIDENCE	-
8. PHYSICAL TARGET: ID(S)	"BANK"
9. PHYSICAL TARGET: TOTAL NUM	1
10. PHYSICAL TARGET: TYPE(S)	FINANCIAL: "BANK"
11. HUMAN TARGET: ID(S)	-
12.HUMAN TARGET TOTAL NUM	-
13. HUMAN TARGET TYPE(S)	-
14. TARGET: FOREIGN NATION(S)	-
15. INSTRUMENT: TYPE(S)	-
16. LOCATION OF INCIDENT	COLOMBIA: MEDELLIN (CITY)
17- EFFECT ON PHYSICAL TARGET(S)	SOME DAMAGE: "BANK"
18. EFFECT ON HUMAN TARGETS	-

0.MESSAGE ID	TST2-MUC3-0069
1.TEMPLATE ID	2
2. DATE OF INCIDENT	07 SEP 89
3. TYPE OF INCIDENT	BOMBING
4. CATEGORY OF INCIDENT	TERRORIST ACT
5. PERPETRATOR: ID OF INDIV(S)	"TWO MEN" / "MEN"
6. PERPETRATOR: ID OF ORG(S)	-
7. PERPETRATOR: CONFIDENCE	-
8. PHYSICAL TARGET: ID(S)	"FAST-FOOD RESTAURANT" / "PRESTO INSTALLATIONS" / "RESTAURANT"
9. PHYSICAL TARGET: TOTAL NUM	1
10. PHYSICAL TARGET: TYPE(S)	COMMERCIAL: "FAST-FOOD RESTAURANT" / "PRESTO INSTALLATIONS" / "RESTAURANT"
11. HUMAN TARGET: ID(S)	"PEOPLE" "POLICEMAN"
12.HUMAN TARGET TOTAL NUM	36
13. HUMAN TARGET TYPE(S)	CIVILIAN: "PEOPLE" LAW ENFORCEMENT: "POLICEMAN"
14. TARGET: FOREIGN NATION(S)	-
15. INSTRUMENT: TYPE(S)	-
16. LOCATION OF INCIDENT	COLOMBIA: MEDELLIN (CITY
17. EFFECT ON PHYSICAL TARGET(S)	SOME DAMAGE: "FAST-FOOD RESTAURANT" / "PRESTO INSTALLATIONS" / "RESTAURANT"
18. EFFECT ON HUMAN TARGETS	INJURY: "POLICEMAN" NO INJURY: "PEOPLE"

Figure 6: Filled templates for MUC-3 messages

MUC-6 was held in 1995 with seventeen participants and a domain of financial news stories. This time participants were allowed to choose any of the four following tasks: (i) **Named Entity** recognition (NE) which required the recognition of named entities such as organisations, persons, locations and dates, (ii) **Coreference Resolution** (CO), which required the identification of phrases in the text that referred to a person or object description in the text, (iii) **Template Element** filling, which required the filling of small scale templates and (iv) **Scenario Template** filling which required the detection of specific relations holding between template elements. Figure 7 below shows a general IE System and its components.

Named Entity recognition (NE)

Named Entity recognition is the simplest and most reliable IE technology. Named Entity systems identify all the names of people, places, organisations, dates, and amounts of money in a text. So far NE recognition can be performed at about 96% accuracy [Cunningham, 1997].

Coreference resolution (CO)

Coreference resolution involves identifying relations between entities within texts. These entities are those identified by both NE and anaphoric references to the entities. This process is somewhat less directly relevant to users than other IE tasks. In the case of text browsing, CO might be used to highlight all occurrences of the same object or provide hypertext links between them.

The main significance of this task, however, is as a building block for Template Element and Scenario Template filling. CO establishes the association of descriptive information scattered across text with the entities to which it refers. The approximate performance for CO is about 55% recall and 70% precision. Recall is the number of slot fills matched correctly divided by the total number of slot fills in the key. Precision is the total number of slot fills produced correctly divided by the total number of slot fills produced.

Template Element production (TE)

The Template Element task builds on Named Entity recognition and Coreference Resolution. In addition to locating and typing entities in documents, TE associates descriptive informa-

tion with the entities. The best system's current score is 80%, while a human normally achieves about 93%.

Scenario Template extraction

Scenario Templates are the prototypical outputs of full information extraction systems. They tie together template element entities into event and relation descriptions. Compared to other tasks in information extraction systems, scenario template extraction is difficult. The best system score is about 56%, while the normal human score is 81%. The scenario template task is domain dependent and, by definition, tied to the scenarios of interest to users.

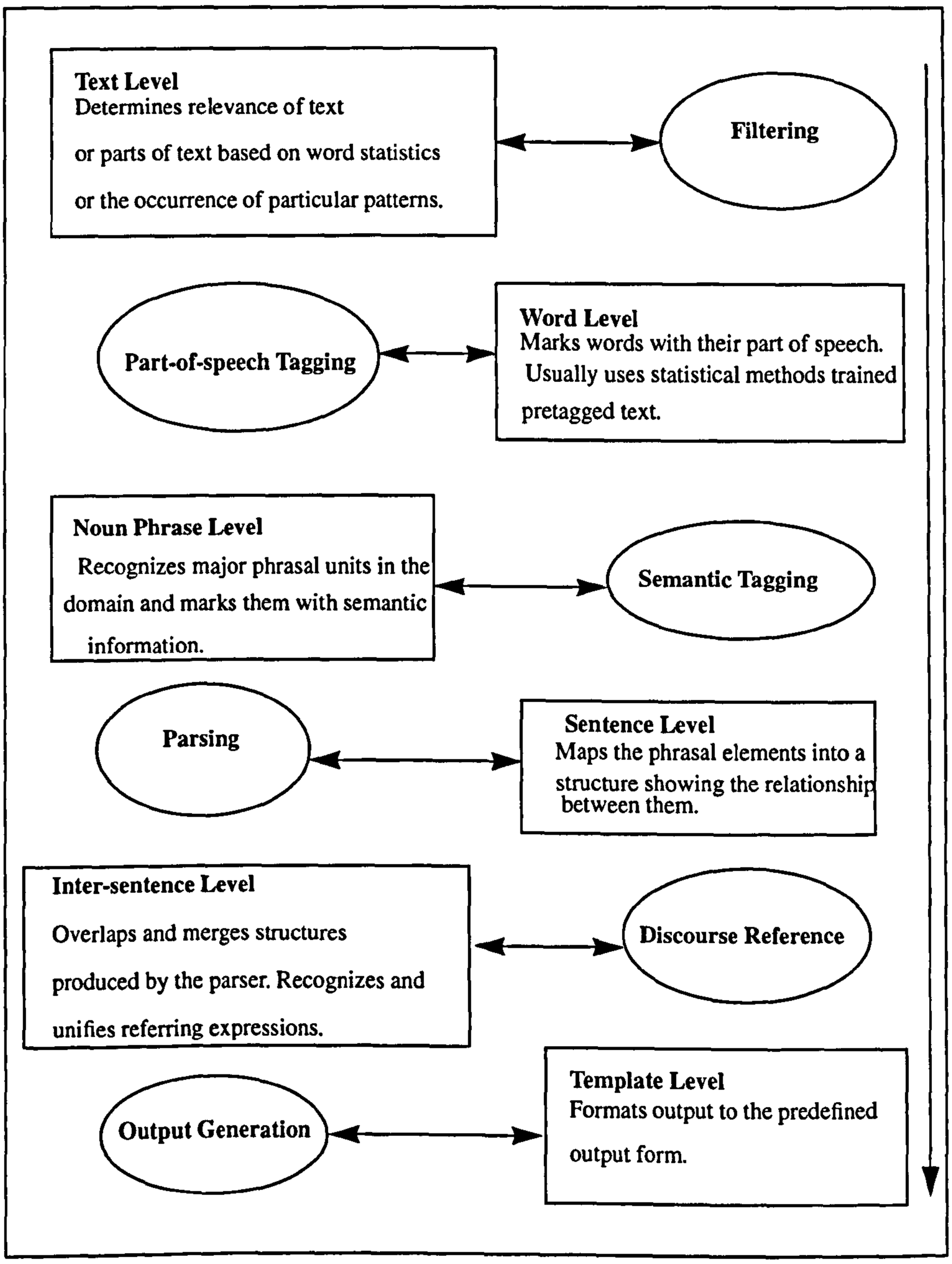


Figure 7: A general IE System

5.2.3 Information Extraction and Natural Language Processing

Information Extraction, as a sub-task of NLP, has a very close relation with other sub-tasks such as Information Retrieval (IR), Natural Language Generation (NLG), and Machine Translation (MT). While, in some cases, IE uses the results produced by other NLP systems, there are also cases where IE results are used by other NLP systems (see below).

As mentioned before, IR seems very close to IE, but has different aims and technology and some believe that IE and IR are complementary to each other [Gaizauskas and Robertson, 1997]. Since IR picks up relevant documents based on a query, and IE picks up relevant information items from a document, it follows that an IE system should be used after the application of some sort of IR system. In other words, Information Extraction may use as its input the documents found by an Information Retrieval system.

There is also a close relationship between Information Extraction and Text Generation. The two most important phases in Text Generation are preparing correct and relevant information and producing coherent text as result. Text Generation by computer, using as source filled templates with correct information, found by an Information Extraction system, is an initial and necessary step towards the first phase of a Natural Language Generation system.

Machine Translation is another sub-area of NLP which seems to have only a slight relation with IE. One possibility is first to extract interesting information from a document in a source language, and then translate the extracted information into a target language. This combination of Information Extraction and Machine Translation, which makes it possible to translate a short and formatted text instead of a full and unstructured text, is an easier task to perform than performing a full Machine Translation first. Another positive point about this combination is not only that the extracted information yields less volume of text to be translated, it also has a structured format which causes less ambiguity.

5.2.4 LaSIE

LaSIE (Large Scale Information Extraction system) [Gaizauskas et al., 1995] is the Sheffield NLP group's MUC-6 system. The system processes Wall Street Journal texts and produces results for three MUC-6 tasks: named entity (NE) recognition, co-reference resolution (CO), and template element generation (TE). Figure 8 below shows the LaSIE System Architecture. LaSIE processes texts one sentence at a time and it is done in three stages: lexical preprocessing, parsing, and discourse interpretation.

Lexical preprocessing takes as input an original text and tags the tokens with part-of-speech tags. The preprocessor is written mainly in C and C++ and uses the Brill tagger

[Brill, 1994] for part-of-speech identification. Input to the preprocessor is from an ASCII file containing a Wall-Street Journal article marked up in SGML. Morphological analysis and the matching of phrases against a list of proper names are performed in this stage. Seven file lists are used: 2600 names as organization names, 94 company designators, human names, (mainly first names), 160 title names, 100 currency units, 2000 location names, and 49 time expressions.

Parsing and semantic interpretation prepare lexical and phrasal chart edges in a chart parser. They perform a two-pass chart parsing using a special named entity grammar as well as a general grammar. The grammar rules which recognise Named Entity items are part of the noun phrase rules and the grammar used for parsing at the sentence level are derived from the Penn Tree Bank-II [Marcus et al., 1993][Krotov et al., 1998]. Finally, it selects the best parse for the current sentence, using an algorithm that is a modification of Gazdar's and Mellish's [Gazdar and Mellish, 1989] bottom-up chart parser in Prolog.

Discourse interpretation integrates the semantic representation of a set of sentences into a single model. The input to this stage is the semantic representation from the parser. Discourse processing contains four stages. In the first stage, the semantic structure generated by the parser is processed by adding its instances and attributes. In the second stage, more information is added or removed from the model, and in the third stage all new instances are compared with the previous instances so as to merge any possible pair of instances into one. Finally, the last stage allows inferences to be added to the discourse model.

So far the performance of the system at Named Entity recognition is much better than its performance at the other three tasks. While the performance of LaSIE in Named Entity recognition is about 89% Recall and 93% Precision, its result for Coreference is 54% and 70%, for Template Element filling is 68% and 74% and, finally, for Scenario Template filling 37% and 73%, respectively.

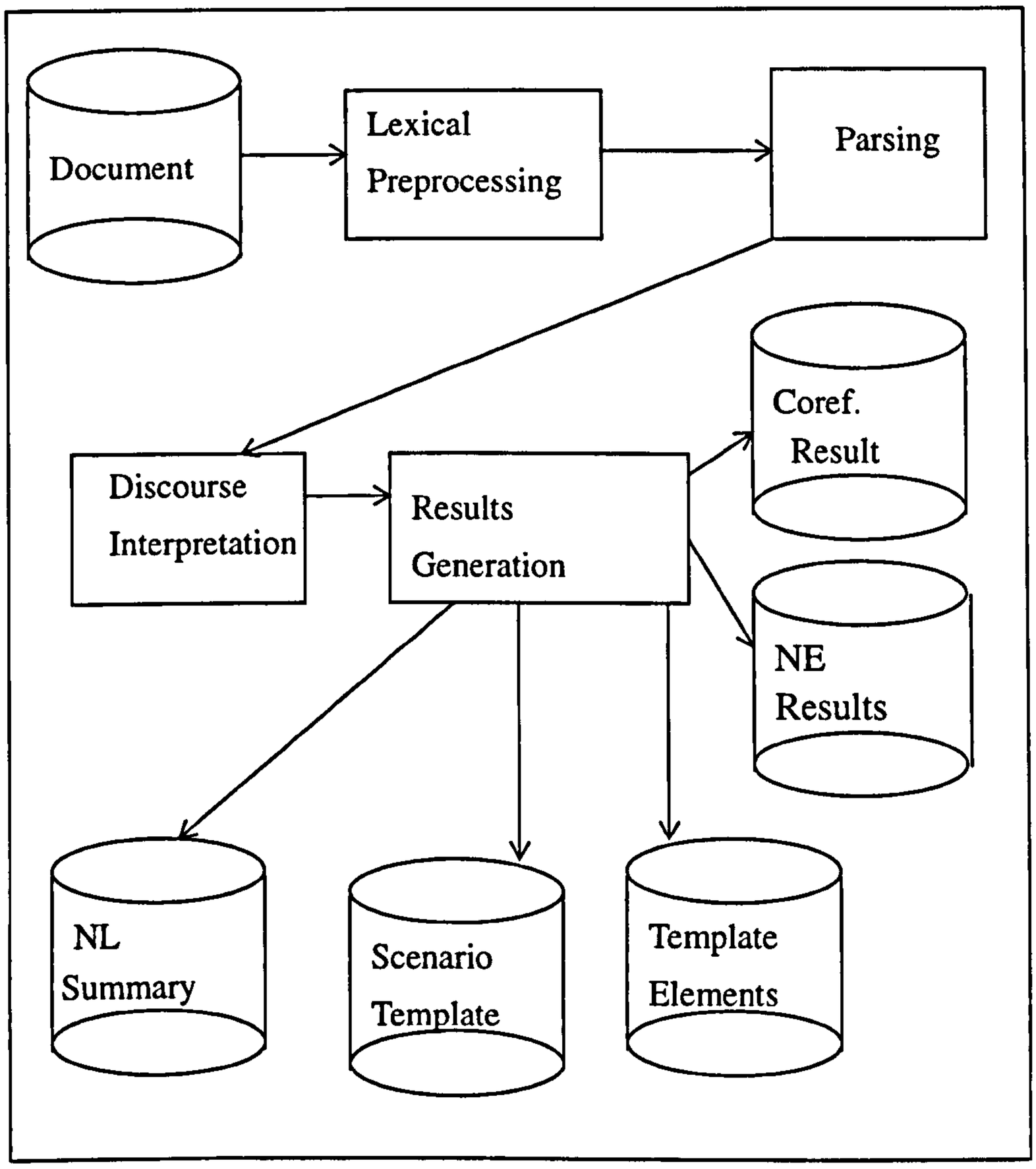


Figure 8: The LaSIE System Architecture

5.2.5 A Machine learning approach to Information Extraction

This section discusses an approach to IE which uses learning techniques to build IE systems from scratch. One of the problems with IE systems is the time and expense of building a new system: the domain dependency of such systems makes it difficult to port a system to another domain. It was for these reasons that researchers investigated applying learning methods to the construction of IE systems [Huffman, 1996].

Recently, Califf and Mooney [Califf and Mooney, 1997] implemented a system called RAPIER (Robust Automated Production of Information Extraction) which learns rules from a corpus given a set of IE templates. The rules are indexed by template and slot name and have three parts: a pre-filler pattern that must match the text immediately preceding the filler, a pat-

tern which matches the actual slot filler and, finally, a post-filler pattern which match the text immediately following. The learning algorithm for RAPIER is based on ILP (Inductive Logic Programming) and a bottom-up search which goes from specific to general. They have trained the system using a set of 100 documents paired with correctly filled templates. They claim the performance of the system gives as average precision of 83.7% and as average recall of 53.1%. They concluded that, because of difficulties related to manually constructing information extraction systems, learning methods have the potential to construct unbounded pattern-match rules using a database of texts and gold-standard filled templates.

5.3 Text Summarisation

Summarisation is finding the key ideas and facts in a document while ignoring irrelevant information. Sparck Jones [Sparck Jones, 1994] argues that coping with too much material presents all kinds of problems for an end-user and what is needed is whole text condensation, or Summarisation. Aretoulaki [Aretoulaki, 1996] defines text summarisation as the process whereby a series of ordered, cohesive, and coherent utterances is reduced to a relatively small set of propositions which convey the core messages of the corresponding parent text. Generally, summarisation is composed of two stages. Identifying the most important pieces of information in the document is the first step and putting them together to generate a coherent report is the second one. Text Summarisation has proved a difficult task because it involves most natural language processing tasks. In addition, when talking about locating important pieces of information (as part of the first stage above), different end-users have different views of the facts in the documents, which means different users might be looking for different summaries from a document.

Summarisation has always, as a key human capacity, been a challenge for NLP, one that has achieved no great success over more than twenty years. Sparck Jones believes that since texts are, in general, individual and each has its own message to convey, systems based on Information Extraction as exemplified by MUC are unsatisfactory. Spark Jones claims that [Sparck Jones, 1993] Summarisation is both a critical Natural Language Processing function and an increasingly pressing NLP task, and that assessing it in any general way will force us to reinvestigate what texts are about, which should remain a central concern for NLP.

Although there are some similarities between summarisation and abstracting, the following definitions should clarify their differences:

A summary of a text is usually more lengthy and informative than its abstract. In addition, the summary may depart from the original to a greater extent and can be less dependent on it formally. Abstracts are thought to contain mainly information about the structure, rather than the actual content, of the text. In fact, the difference between summarisation and abstracting is generally thought to be that the former is applicable to any text type while the latter is mostly associated with scientific and technical source texts. [Aretoulaki, 1996]

Most text summarisation systems to date employ standard Information Retrieval and Information Extraction techniques to decide which text parts contain relevant information to be considered in the summary. Examples of such techniques are keyword matching and lexical pattern matching. As mentioned before, Information Extraction involves the consideration of the immediate syntactic context of a keyword, in terms of part of speech and, possibly, subcategorisation restrictions. The systems based on Information Extraction techniques do not generate new text, but instead extract the most important parts of sentences, re-organise them so as to appear as a new text, the summary.

5.4 Text Classification via Information Extraction

Riloff and Lehnert [Riloff and Lehnert, 1994] describe an approach to text classification that represents a compromise between traditional word-based techniques and in-depth natural language processing. They introduce three algorithms based on Information Extraction systems to classify documents. These algorithms are: a relevancy signatures algorithm, an augmented relevancy signatures algorithm, and a case-based text classification algorithm. They have tried to follow the path that most humans used to classify documents: even for humans, some documents are difficult to classify as a relevant to a domain while others are straightforward to classify. Riloff and Lehnert declared their goal to be recognising the texts most likely to be relevant. Although there is a risk of missing some relevant texts, the advantage is good precision with the ones that have been classified as relevant texts. A second assumption based on assumed human behaviour is that a single relevant sentence, or sometimes a phrase, is often enough to classify a text as a relevant. For example, in the domain of terrorism, the cor-

pus used for MUC-4, the phrase "was shot to death" is sufficient to show a text to be relevant. Finally, as soon as the first relevant sentence is found in a document, the system ignores the rest of it and considers the text as a relevant one. Let us look at the three classification algorithms in more detail.

5.4.1. The Relevancy Signatures Algorithm

Riloff and Lehnert argue that, although keywords are one of the most important cues in identifying relevant texts, the lack of natural language context surrounding words makes it difficult to rely on particular cue words by themselves. For example, and again in the domain of terrorism, the word "dead" is not a good keyword for identifying relevant texts, but every occurrence of the phrase "was found dead" indicates a relevant text. Amusingly enough, while the word "casualties" is not a good indicator of a relevant text, "no casualties" does seem to be useful for identifying relevant texts. Natural language processing capabilities can recognise phrases more effectively by recognising also syntactic relations, such as active and passive verb constructions, conjunctions, etc. The relevancy signatures algorithm [Riloff and Lehnert, 1992] is an attempt to use natural language processing to classify texts on the basis of linguistic expressions, called "signatures". A signature is a pair consisting of a word and a concept node that it triggers, which together represent a set of linguistic expressions. For example <murdered $murder-passive$> represents all passived forms of the verb "murder" such as "was murdered", "were murdered" or "has been murdered".

The relevancy signatures algorithm has two phases. During the first phase, called the training phase, a set of relevancy signatures based on a training corpus are generated. This set would be used in the second phase, called the classification phase, to classify new texts. An important aspect of this algorithm is that a single signature in a text is enough to classify it as a relevant text.

5.4.2. The Augmented Relevancy Signature Algorithm

Although relevancy signatures identify relevant documents in a domain, there are situations in which they fail to meet the goal and occur in irrelevant texts. The examples below from [Riloff and Lehnert, 1994] exemplify this point. In the first pair of sentences below, the signa-

ture <exploded, $explosion$> is common, but the first sentence describes a terrorist event and the second sentences does not:

"A car bomb exploded."

"The foreign debt crisis exploded."

The same problem arises in the next pair of sentences: although the signature <attacked, $attack-passive$> is common between these two sentences the first is part of a terrorist report and the second is not:

"The peasants were attacked by the rebels."

"Kent Jr. was attacked by three other Pavan Prison inmates."

The obvious cue words in these two sentences for a human to identify them as relevant or irrelevant to the terrorist domain are "rebels" and "inmates". To get the benefit of the surrounding word information, some slots have been added to the signatures to improve the accuracy of the resulting classifications. An augmented relevancy signature is a combination of a signature and a slot filled in by a piece of relevant information surrounding the key phrase.

5.4.3. Case-based Text Classification

The augmented relevancy signatures algorithm is an approach to classifying texts based not only on their keywords and phrases but also from considering information available about the surrounding words. This section deals with the situation when a relevant document does not contain any specific words or phrases that are highly correlated with relevance. Some sentences contain pieces of information such that considering them all together makes it a relevant text, while, individually they do not pass any test for being a relevance keyword. Riloff and Lehnert describe how, in this situation, a case-based reasoning algorithm is used to classify texts. The first step is to create a set of cases for a document: each case represents the natural language context associated with a single sentence and has five slots: signatures, perpetrators, victims, targets, and instruments. Figure 9 shows how a sentence can be represented by this complete notion of case. The underlined words are cue words whose combination in a case structure should guide the user to classify the text as relevant or irrelevant.

Sentence:

Two vehicles were destroyed and an unidentified office of agriculture and livestock ministry was heavily damaged following the explosion of two bombs yesterday afternoon.

Case: Signatures: (<destroyed, $destruction-passive$>,
<damaged,$damage-passive$>'
<bombs, $weapon-bomb$>)

Perpetrators: nil

Victims: nil

Instrument: (BOMB)

Figure 9: A sample sentence and its resulting case-based representation

The final decision is made based on comparing the case generated for the text with all cases prepared as a case base; based on a statistical investigation [Riloff, 1993] claims the document can be classified satisfactorily.

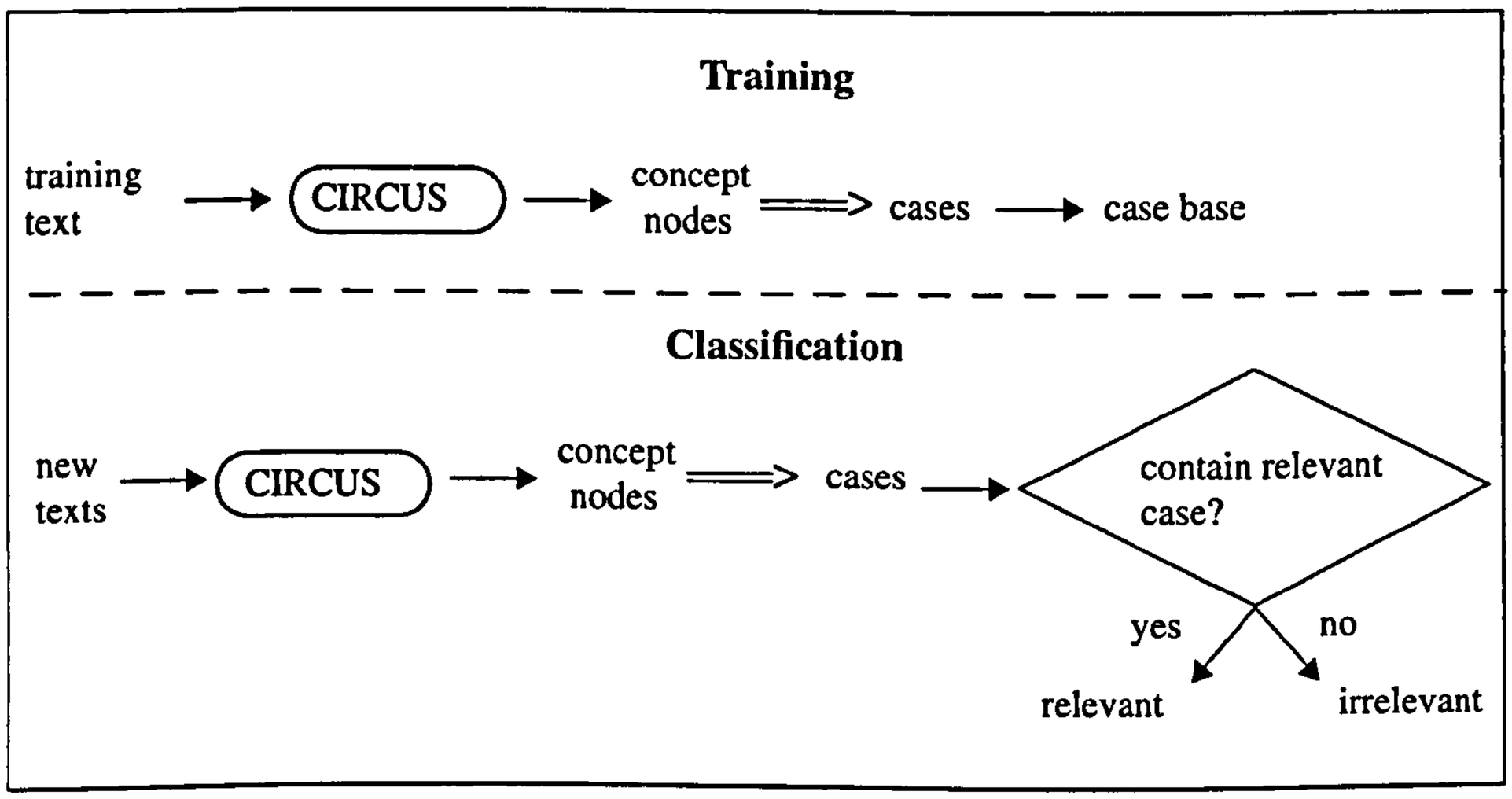


Figure 10: Flowchart for the Case-Based Text Classification Algorithm

When classifying a new document, based on the above flowchart, the concept nodes produced by CIRCUS [Lehnert, 1991] are converted into set of cases. One relevant case is sufficient to classify the text as relevant. A relevant case must satisfy following three conditions:

1. The case contains a strong relevancy index.

2. The case does not contain any "bad" signatures.

3. The case does not contain any "bad" slot fillers.

The following two sentences demonstrate how slight difference between two sentences can cause one text to be deemed a relevant and another an irrelevant text.

Sentence:

More than 100 people have died in Peru since 1980, when the Maoist Shining Path organisation began its attack and its wave of political violence.

Case: Signature: (<died, die>, <wave, $generic-event-marker$>,
<attack, $attack-noun$>
Perpetrators: (TERRORIST ORGANISATION)
Victims: (HUMAN)
Target: nil
Instrument: nil

While the above case identifies the sentence as an irrelevant text, the next sentence, which is very similar to the previous one, represents a relevant document:

Sentence:

More than 100 people have died in Peru during two attacks by the Maoist Shining Path organisation yesterday.

Case: Signature: (<died, die>, <attack, $attack-noun$>
Perpetrators (TERRORIST ORGANISATION)
Victims: (HUMAN)
Target: nil
Instrument: nil

The only difference between two above sentences is the concept node called $generic-

event-marker$ in the first sentence. Since this concept node indicates that the sentence is describing a summary event description, the sentence is considered as irrelevant.

In summary, Riloff and Lehnert claim their model uses an underlying Information Extraction system to achieve high-precision text classification. A positive point about their approach is its ability to be ported to different domains and can be used to support many applications.

The claim for the high portability of the system across domains is based on the following:

i. The classification algorithms use general statistical techniques that are independent of the domain.

ii. The concept node dictionary is the main domain-dependent knowledge for the Information Extraction system. According to Riloff and Lehnert, their system (called AutoSlog), can be used by anyone who is familiar with the domain, which makes it possible to generate automatically a concept node dictionary more simpler than with any other techniques.

5.5 Summary

This chapter has reviewed three sub-areas of NLP: Information Retrieval, Information Extraction, and Text Summarisation. Yet none of these approaches is fully suitable for extracting the gist of messages in the domain of Emails:

Information Retrieval: Since IR deals with identifying documents from a larger collection which are relevant with respect to some query, its techniques can only be used in a system to pick out relevant Email messages from a large number of messages not based on a pre-defined subject.

Information Extraction: IE techniques require a set of user defined templates which specify the information to be extracted, i.e proper names, organization names, dates, and so on. But in dealing with Email messages it is impossible to predict the relevant information involved, because the content of Emails does not fit into a clearly defined domain. A survey of the subjects of Emails sent to a computer support group shows a great variety of subjects, each of which would need different templates. In this investigation we tried to classify the Email corpus described in chapter 8 according to subject. The list below shows some of these classes

and examples of each class.

Subject Class	Elements
Students	New account, Keys
Booking	Labs, PCs, rooms
Accounts	temporary account, remove account
Hardware	Scanner, cdrom
Facilities	photocopy card, Box of papers
Software	decoding, image scanning, word
Emails	alias, mailing list, return mails
questions	posters
directories	no home directory, lost my file

Text Summarisation is considered to be a reasonable approach for long and relevant texts, but this is not usually appropriate to Email messages.

As mentioned earlier, none of the approaches described in this chapter is completely suitable for dealing with Email messages. In the next chapter we review work more directly related to Email.

Chapter 6

Previous work on Emails

This chapter describes research more directly related to our subject matter. Patel [Patel, 1990] dealt with Email messages directly and Gasparotti and Simone in their paper [Gasparotti and Simone, 1990] introduced a user interface based on Speech Act theory to interpret Email messages.

6.1 The Summarisation and Categorisation of Electronic Mail Messages

Malti Patel claims that the aim of her research was to discover how Lehnert's "Plot Unit" techniques [Lehnert, 1982] could be extended to cope with the summarisation of text in the form of electronic mail messages. The categorisation of messages was another aim of the work. Since Lehnert's "Plot Units' only dealt with stories, they could not be used to represent important features of an Email. Extending Lehnert's summarisation strategy involved both theoretical and practical considerations.

A small system called EMMY (Electronic Mail Message summarY) was implemented by Patel, and her paper discussed the necessity of using syntactic, semantic and pragmatic methods in association with plot units. The paper does not aim to model the content of a message, but the effect that the contents have on the person reading the message. Since most Email messages are not well formed syntactically [Patel, 1990 Chapter 1], Patel prefers parsing on a word-by-word basis instead of parsing the whole sentence.

The Parsing strategy adopted by EMMY

As Patel claimed since Email messages contain un-grammatical sentences, EMMY employs semantic analysis to locate the meaning of the text and turns to Word Expert Parsing (WEP) [Small and Reiger, 1982] for a word-by-word parsing method. In this algorithm, each word is examined on its own to see what information it can offer as regards understanding the sentence as a whole. If a word is ambiguous, then the words surrounding it will also be analysed to see if they can give a clue to the correct meaning of the ambiguous word.

To summarise a message, it must be fully understood and the most important point within the message must be recognised. The first criterion is met by using semantics to find the meaning of a message; after a message has been analysed, its meaning is stored in a structure known as the "Semantic Content Representation (SRC)".

Knowledge Assistants

The semantic analysis of a message is performed by means of what is called a knowledge assistant, which comes in three forms: *Meaning objects*, *Reference objects*, and *Verb knowledge objects*.

Meaning-object

There is a Meaning-object for each word containing its grammatical type, meaning, and a choose-if, where the last is called in case of ambiguity. Since a word may have different meanings, a Meaning-object stores the various meanings which a word may have together with the following information (for each meaning):

i. The grammatical type of the word.
ii. Information about the context in which this meaning is applicable.

In addition, each word, along with its meaning, has a column called Choose-if column, which deals with ambiguous words. Table 3 below illustrates how different meanings of the word "issue" are stored in the dictionary.

Dictionary words: issue issued issuing issues			
Number	Grammatical type	Meaning	Choose-if
1	Noun	Subject	Default (Subject)
2	Noun	Publication	Paper
3	Noun	Creature	Birth
4	Verb	Publish	Default (Subject

Table 3: The representation of the word "issue"

For each category, a Default row is defined, and in case that there is no clue for choosing the correct meaning.

Reference objects

Reference objects are associated only with pronouns. A Reference object consists of the gender and name of the person referred by the pronoun. "I", "me" and "you" are referred to the "sender" and "receiver" respectively, and the pronoun "it" refers to the last mentioned noun. EMMY resolves the pronouns "he" and "she" by finding the gender of the person involved in the sentence. For example in the sentence:

> Mary went to the grocer's, John went to the supermarket. Then she went to the baker's.

EMMY recognises that "she" refers to Mary; but EMMY is unable to recognise correctly that in the sentence:

> Mary said that Jenny could borrow her dress.

the pronoun "her" refers to Mary and not to Jenny.

Verb Knowledge Objects

The slots of Verb Knowledge Objects (VKO) may only be filled by important words which surround a verb, such as nouns and adverbs, where nouns are classified as either common or

abstract nouns. The reason for classifying nouns is that some verbs only make sense when linked to a common noun or to an abstract noun respectively. Different labelled slots which can appear in a Verb Knowledge Objects are:

aninoun (A,B), must be filled by an animate object. Slot "A" holds the name of the animate object, and slot "B" holds its gender.

start_allnouns (S), must be filled by words which appear before the verb in a sentence.

last_allnouns (S), must be filled by words which appear after the verb in a sentence.

common_noun (C), must be filled by a common noun.

abstract_noun (A), must be filled by an abstract noun.

For example, the verb "ask" would have a structure containing the following slots: (aninoun (A,B), aninoun (C,D), last_allnouns (L)).

The Dictionary

The dictionary is structured towards assisting EMMY to perform its tasks: different words are stored in the dictionary chiefly according to their grammatical type.

Summarisation and Categorisation Process.

The summarisation of a message involves discovering the receiver's reactions to the information within a message and placing the most important part of this information within the summary. The receiver's reaction is achieved by means of *Mental States* and a message conveys ideas and information which are represented by *Idea Links*. These two combine to form Email Units, and we will now illustrate these theoretical terms.

Mental States and Idea Links.

Mental States are used to describe the effect a piece of news has on the receiver and are termed positive (+), negative (-), and neutral (N). The five Idea Links which reflect the ideas sent by the sender on some topics are: transfer (t), receiver (r), sender (s), question (q), and request (req). Email units are each composed of a Mental State and an Idea Link, which hold together information which is to be summarised.

Email Units were introduced similar to Lehnert's plot units as follows:

- Glad to see the information.

- Unhappy with the information

- Having a neutral response to the information

and some physical activities such as:

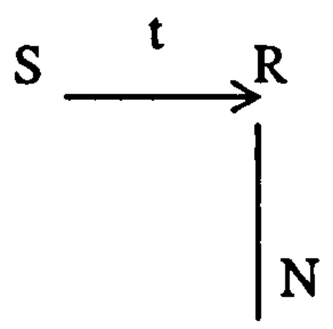
The receiver having to perform an action.

The sender having to perform an action

The sender wanting to contact the receiver.

Nine different Email Units have been described in the work.

For example, Email Unit below means that the receiver has information (t) which is of neutral (N) interest.



The Email Unit Generator separates the information on the summarisation and categorisation board into categories, Mental States, and Idea Links and then performs the appropriate action on them. After the formation of Email Units, the next stage is to select the important facts and then to produce the actual sentences of the summary. The building blocks of the summary are verbs and their Verb Knowledge Objects. Producing a summary requires the following steps: creating a summary from Idea Links, recognising actions, obtaining important information, generating a summary sentence from one VKO and finally choosing one VKO from many to form a summary sentence.

The categorisation process performs its tasks along with the summarisation process. Most categories are determined by key words and key phrases which store the category to which they belong in their dictionary. There are five categories which may be assigned to a message: Courses, Meeting, Question, Answer, and Specific Topic. The first two categories, Courses and Meeting, are assigned to a message by recognising keywords such as "course", "courses"

and "meeting" respectively. The Question and Answer categories are assigned by recognising phrases such as "Do you", "Can you" and "the question of", and "answer" respectively. Any other messages which are left unassigned by the four categories mentioned above are assigned as Specific Topic. Figure 11 illustrates a message and results reported by EMMY.

Message:

As you know our next staff meeting is going to talk place on the 5th of January from 14:00 to 17:30 in conference room Saleve.

Could you please let me know asap if you have any topic you would like to discuss so that I can prepare an agenda.

Thanks in advance for your inputs.

Sally

EMMY's initial report is:

FROM: Sally James

DATE: 1st may

meeting: staff / meeting / conference room / saleve

questions: asap / topic / agenda

specific topic: asap / topic / agenda

Number of sentences in message: 2

So far, EMMY has reported three categories along with nouns related to each category as clue words for the receiver to get an idea what the message is about. The summary produced by EMMY is:

As you know our next staff meeting to take place on the 5th of January from 14:00 to 17:30 in conference room saleve.

Could you please let sender know asap if any topic to discuss.

Sender prepare an agenda.

Figure 11: EMMY's summary report

Although Patel described her theoretical points in detail, the implemented system, EMMY has been tested by only a handful of messages and has remained unevaluated. In addition, the effect of considering all sentences in their active form needs more investigation. Finally, although EMMY reports the number of sentences in a message, there is no description of how sentence boundaries are recognised, especially in the Email domain where punctuation is not

fully respected.

6.2 A user-defined environment for handling conversations

The usability of Email-based communication can be increased by tools which help the user to organise the amount of interaction in which she/he is involved. Gasparotti and Simone [Gasparotti and Simone, 1990] reported a software module which pursues this idea. Gasparotti and Simone used the communication model defined within Speech Act theory and the semi-structured message approach reported by Malone [Malone et al., 1987]

The long term aim of the work is to construct a knowledge-based integration system supporting a user in the coordination of his activities and communication by Email within a group of cooperating agents. Gasparotti and Simone have used a combination of the Coordinator [Act87] and Information Lens [Malone et al., 1987] in their system to define a local environment by means of rules for filtering messages and for associating to them suitable actions. Three major actions involve the automatic handling of parts of the conversations, the updating of a knowledge base for the activities, and the organisation of the structure of the group of the people involved.

In the Coordinator, a message is essentially a Speech Act and a conversation is a sequence of Speech Acts. The focus is on how conversations and their related commitments can be handled by means of communication pragmatics, rather by means of the semantics of single Speech Acts.

The Information Lens's focus is on the representational role of language and attention is given to a message's propositional content (semantics) rather than its pragmatic meaning. The Information Lens is used to organise messages in semi-structured templates which are arranged in networks so that more specific messages types can inherit properties from the more general ones.

By combining these two orientations, they hope to achieve a theory of communication pragmatics with the flexibility of a knowledge-based approach to communication semantics. The approach they propose is based on the following concepts:

1. A conversation is a sequence of Speech Acts, each one defining the space of possi-

bilities for the subsequent Speech Acts.

2. A Speech Act has both a pragmatic and semantic content.

3. The content of a Speech Act is the basis for the rules for message handling.

In this system, a message can be divided into four sections: the context of the conversation, the Speech Act of the message, the content of the commitment expressed in the message, and temporal constraints. The context of the conversation includes 'From', 'To', 'Domain', and 'Title'. For each user, 'Domain' can be arranged in a hierarchy, as shown an example in

Figure 12:

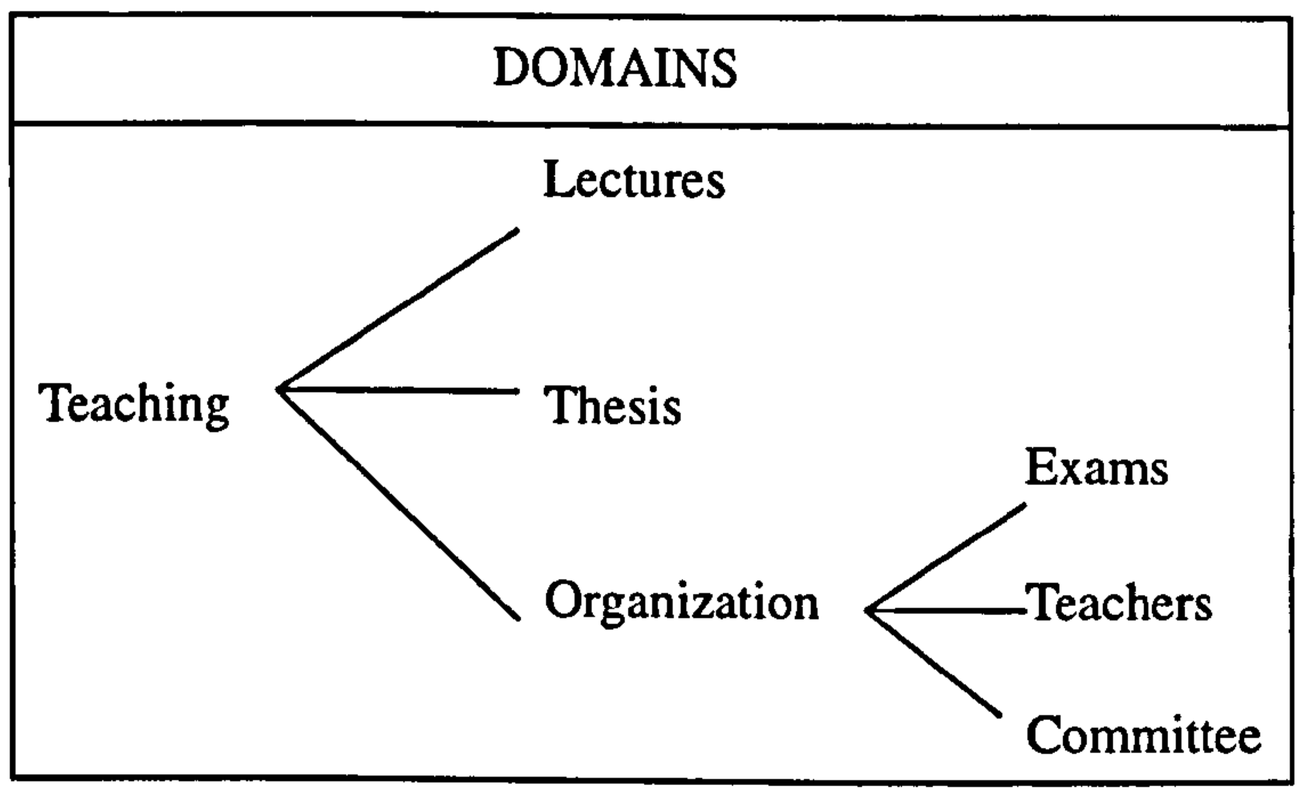


Figure 12: A hierarchical domain

The field 'Speech Act' is related to the Speech Act to which the message belongs; and the field 'Commit type' which itself can be organized in a hierarchy depends on individual user. They distinguish between conversation for action and conversation for information.

The paper illustrates how a user can select a Speech Act such as 'Request' from the main menu and, after filling the fields 'From', 'To', 'Domain', and 'Title', the user selects the 'Commit type', the user can fill in values for the fields presented in the selected template, as shown in Figure 13:

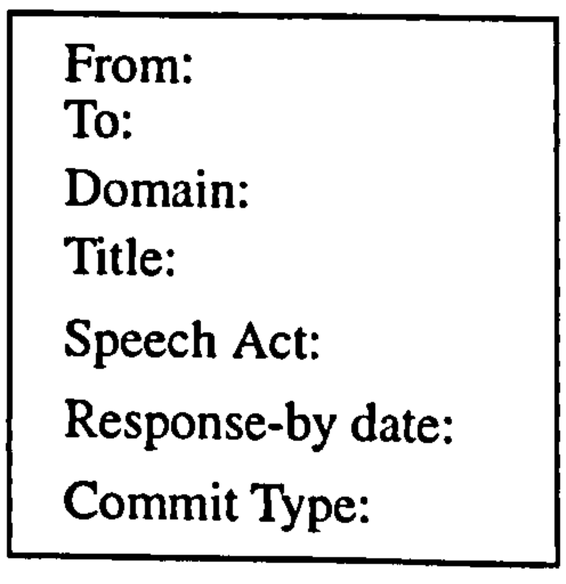


Figure 13: Fields in the main menu

The receiver of a 'Request' message can also select manually the 'Answer' option and choose required Speech Act and fill in the necessary information as an answer.

Message Filtering

Another ability of the system is defining rules for filtering messages. The paper introduces a template that each user can construct rules for filtering messages addressed to him or her. There are different ways that a user can filter his or her messages dependent on 'Domain' or 'Commit type'. Figure 14 shows how the structure of *IF condition THEN action* is available for filtering some messages which hold those *conditions* and react the *actions.* Four available actions are: *Set characteristic: <characteristic name>, Hide, Show,* and *Set priority = <priority value>.*

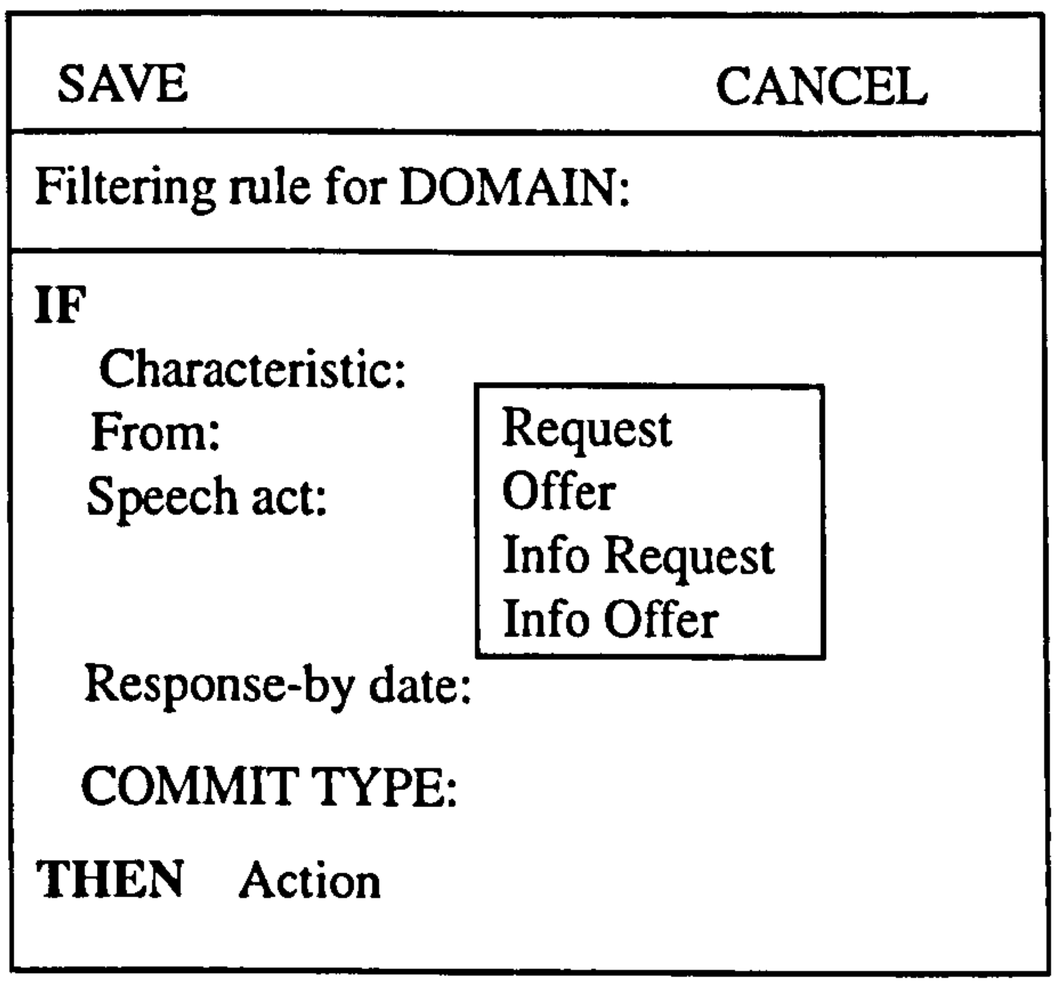


Figure 14: The structure of <IF condition THEN action>

Three action rules have been defined on the basis of the events and their effects. LL-rules, LK-rules, and KL-rules while L stands for Linguistic and K for Knowledge:

LL-rules have Linguistic events, both as a condition and action, such as: "open Conversation for action" or "Answer". A simplified example is:

IF

From: student-code

Speech Act: Request

THEN

Set characteristic: group thesis-open

Answer: Counter-offer

LK-rules and KL-rules can help the user to define the connections in a structured framework to improve the quality of the integrated system. Simplified examples of each rules are:

LK-rule

IF

Characteristic: group thesis-open

Speech Act: Accept

THEN

Set characteristic: group thesis OK

Do: store(student.ok)

KL-rule

IF

Characteristic: group thesis OK

THEN

Answer: Counter-offer

The paper concludes by noting that the implementation of the system is still under development and one of its future developments will be the integration of the customisation of the user environment in the more general framework of message interpretation, i.e., adapting the interpretation process to the presence of filters, automatic Speech Acts and rules managing them.

6.3 Summary

This chapter reviewed research work related to the analysis of Email content. Section 6.1 described work to that summarised and categorised Emails. One disadvantage of the work was the shortage of sufficient analysed Emails to make system evaluation possible. In the next chapter and chapter 8 we describe how systems such as the one we have implemented (see chapter 10) can benefit from corpus analysis.

Section 6.2 described an interface which has used speech act theory to define a communication model based on Email usage in group work. In chapter 9 we describe another approach to speech act recognition in Emails.

Chapter 7

Corpus Analysis

Corpus analysis is an experimental approach observing the way people communicate in the real world and in different situations using text. McEnery and Wilson [McEnery and Wilson, 1996] define it as follows: "corpus linguistics is perhaps best described for the moment in simple terms as the study of language based on examples of real life language use". Since on-line sources readable by computers have only recently become available, much of theoretical linguistics over the past fifty years has been based on the study of isolated sentences. This approach has some severe limitations, and is unable to glean what can be learned by studying patterns of language across texts and corpora. For example, there are certain aspects of language use which can be observed by considering their frequencies in corpus analysis.

In addition, although there are disagreements about the most important element in understanding a text, it is generally accepted that at least three elements are involved: the text itself, the speaker or writer, and the hearer or reader.

7.1 Properties of corpora

The first generation of computer readable corpora with about one million words each was set up in 1960s and 1970s. The individual corpus might be either spoken or written texts and drawn from many different genres: for example, scientific research paper, newspaper article, and conversation letters as examples of written texts, and radio broadcast, telephone conversation, and dialogue conversation as examples of spoken texts. According to McEnery and

Wilson [McEnery and Wilson, 1996] although any collection of more than one text can be called a corpus, the term corpus when used in the context of modern linguistics tends to have more specific connotations than this simple definition provides for. They defined four aspects for a corpus:

Sampling and representativeness

In linguistics we are interested in the whole variety of a language. Since it is impractical to collect all possible types and analyse them one by one, it is necessary to build a sample of the language variety in which we are interested. McEnery and Wilson refer to Chomsky's criticism [Chomsky, 1957],[Chomsky, 1965] concerning the possibility of skewedness in corpus analysis and they suggest that to avoid this problem the corpus should be representative as much as possible. In another words, the corpus should be drawn from samples of a broad range of different authors and genres.

Finite size

The term corpus usually implies a finite size of words and texts. A researcher planing to build a corpus will decide on how the language variety is to be sampled and how many words are to be collected.

Machine-readable form

Nowadays, all corpora are assumed to be machine-readable. Without this ability, it would be very difficult, if not impossible, to use any reasonably sized corpus. For example, counting the frequency occurrences of a specific word. Of course, a corpus might contain its data in different media as well e.g. spoken data recorded on a tape.

A standard reference

Being available to other researches is a positive feature of a corpus, although not an essential requirement. An advantage of a standard corpus is the possibility of comparing results with other published results.

No matter what corpus is used, the focus in analysing a corpus is usually on lexical and grammatical patterns in text, especially those patterns which express the speaker or writer's point of view.

7.2 Spoken and Written Language

Everyone is involved with language, in both spoken and written forms on a daily basis, but it is not easy to define the relation between written and spoken language. Written language has been described much better than spoken language. Not only historical, political, and social reasons lie behind this point; availability of written language sources is an important reason, too. In addition, written sources are more open to observation than spoken language. It is worth to mentioning that written language is not simply a spoken language written down and vice versa. Each form of language has its own specifications. For instance, while much written language is standard, formal, planned, edited, and non-interactive, spoken language is typically casual, spontaneous, and face to face.

Each form of language has endless sub-types which have some common features and some different, and corpus analysis has established that each language type has its own specific features and again the three aforementioned elements in text understanding, writer, reader, and text itself, are the most important elements. For example, a corpus analysis by Baker and Freebody [Baker and Freebody, 1989], of a corpus of 80,000 words of children's elementary reading books, shows some results which are different from a mixed written corpus. In term of frequency data, the word "boy" and "boys" appear more than "girl" and "girls". While "children" occurred in the corpus, the word "child" never occurred. Finally, in this corpus fathers paint and drive cars and mothers bake cakes and pick flowers. Baker and Freebody argue that such frequencies and distributions convey interpretations about the social world and how it is important as a way of identifying and talking about people and differences between people in different corpora. This analysis also emphasizes that findings should always be compared across different corpora to find out which patterns are normal and which are characteristics of a special corpus.

7.3 Lexical Density

As mentioned before, in spite of much involvement with language, both in spoken and written forms, there is no straightforward way to define these forms. It was also mentioned that each type has its own characteristics. This section investigates an approach to clarify how it might be possible to differentiate between spoken and written language by a corpus analysis method

by calculating its "lexical density".

Most written texts are more lexically elaborated than spoken texts. For example, in comparing a published academic article with a casual conversation text transcribed as: while the former is densely packed with information (more lexical words), the later is less informative, with less lexical words.

In general, the whole vocabulary of any language can be divided into two major categories: lexical words, which express content, and grammatical words, which relate lexical words to each other. Corpus studies by Halliday [Halliday, 1989] shows that spoken and written language have different lexical density. Some have referred to lexical words as major, full and content words and to grammatical words as minor, empty, and functional words. For example, in a simple sentence such as "the weather is cold", there are two lexical words, "weather" and "cold" and two grammatical words, "the" and "is". To apply the previous definition, the words "cold" and "weather" are independent sense-units, but "is" and "the", in spite of their independence in form, are not independent in meaning because they do not convey any idea by themselves. Generally, from a part of speech point of view, noun, adjective, adverb and main verb are considered lexical words and auxiliary verb, modal verb, pronoun, preposition, determiner and conjunction are considered grammatical words. In terms of frequencies, the number of lexical words is much higher than grammatical words.

Lexical density is one way to measure the frequency of lexical words in a text. If N is the number of words in a text and L is the number of lexical words in a text, then

lexical density = 100 x L/N.

Reports on corpus analysis [Ure, 1971] have shown that on average written texts have lexical densities in the range of 36 to 57, although a shopping list might have 100, and the lexical density in spoken texts is on average between 24 and 43; thus establishing one clear measure that distinguishes these two type of input to a corpus.

7.4 Summary

Corpus analysis as an experimental approach has become essential to many of most Natu-

ral Language Processing systems. This chapter reviewed some of the important aspects of corpus definition; it also reviewed an experimental test to distinguish between spoken and written genres, a distinction made use of when analysing the Email corpus prepared for this research.

Chapter 8

Email Corpus Analysis

Recently a number of researchers in Computational Linguistics have argued that any attempt to deal with language computationally needs to resort to statistical methods. These techniques require large amount of data. The availability of large amount of on-line text of different sizes and genres, like journal articles, Web pages, Emails and many other on-line corpora, has made it possible to have access to a huge amount of both tagged and natural data sources. The necessity of real data is more essential if the corpus consists of informal language like Email messages.

For the purpose of this research, a corpus of 1500 messages was prepared from a set of messages sent to the Computer Science Support Group at Sheffield University, Computer Science Department. Let us give a quick characterisation of this corpus against four criteria of a balance corpus set out by McEnery and Wilson [McEnery and Wilson, 1996]:

(1) Sampling and representativeness: the corpus contains messages from a variety undergraduate and graduate students, academics and research staff, visiting researchers and secretarial staff. The positive point about the variety of people is that the messages show different aspects of language both in the length of message and the characteristics of language.

(2) Finite size: the initial corpus contains 1500 messages sized from a single sentence to more than 15 sentences with about 100,000 words in all.

(3) Machine-readable form: the corpus is, of course, machine-readable.

(4) A standard reference: the corpus has not been accessible to any other researchers.

The main idea behind analysing this corpus is to find out how people use the Emails to communicate in this domain. The first major step in analysis is to decide on the gist of the messages and recognise the most important sentence(s), their "Focus Sentence(s)", in the text, and to assign Speech Acts to these focus sentences manually. A manual search for the gist of the messages and considering their Speech Acts proved that, because of the nature of this domain, more than 90% of these messages are requests but of different forms.

8.1 The structure of Email texts

8.1.1 Email usage

From the frequency of use point of view, it was almost impossible to find out the number of Emails used everyday in the domain I am investigating. To have some general idea, information from two sources was prepared. According to one of the support staff in the Department of Computer Science in Sheffield, during one week, the last week of July 1998, 13600 Emails were received and sent from the Department.

In a different attempt, the following procedure made it possible to get a rough idea of the frequency of use of Email in this Department:

1. Sending two test Emails to myself.
2. Calculating the amount of time between them in seconds.
3. Calculating the number of Emails received/sent by considering the difference between their id numbers.

Carrying out this procedure several times shows that the average time for an Email during work time hour is about 2.2 seconds for a message. Of course this average time during late evenings and weekends is much higher and is about 12.4 seconds.

8.1.2 Usage

It is obvious that by improving the capability of computers both in software and hardware, the functionality of Email has been increased. There are several versions of Mail Tool available

in computer systems with different facilities, although with very similar capabilities.

Today, compared to other available systems for communication like letter, telephone, and fax, Email has the best performance considering cost, response time, and archiving space.

8.1.3 The specification of Email text

A receiver would delete most of the messages as soon as he/she reads them. So most senders do not pay much attention to the way they type their messages and there are some misspelled words in Email messages. To investigate the number of misspelled words in the domain used for this research, 600 emails were checked using a spell check feature. The results show that 84 Emails out of 600 (14%) contain at least one misspelled word. The total number of misspelled words is 100 words, with a maximum of 5 misspelled words in a single message. Among these 84 Emails, the smallest one, one sentence with 7 words, contains 1 error and the largest one, 5 sentences with 91 words, contains 5 errors. The following original message from the corpus has the maximum 5 errors in a message. The misspelled words are shown in bold type.

> **Coould** someone take my P.C. away and give it a **thotough** check?
> It still crashes whilst I am using it on the network. I am only using windows, just win for **wordies** and excel.
> To be certain of the thing not crashing and losing my work I have to save after every entry in Excel, this makes it **unuseable**.
> When it does crash it is a total lock-up, the only way is to press reset and start again.
>
> give me a chance to finish paying the **invoicces** and you can have it

It is also a common habit for some people to use abbreviated words in their messages; using "U" instead of you, "pls" instead of please and "tnx" instead of thanks are some of these abbreviations. Finally, punctuation is not correct in many Emails (see figure 20). so this sentence from an original Email message:

wod u pls give me a box of blank floppies.

has all the above mentioned features. In general, the context of Emails are like spoken language written down (see section 8.3 below). Emails are neither a typically written form of

communication, nor entirely a spoken dialogue. It is a form of communication which potentially allows for a great deal of freedom, and which encourages the use of informal, unplanned language.

8.1.4 Requests in Emails

As mentioned in the introduction of this chapter, more than 90% of the messages in this corpus contain requests but of different forms. These requests can be divided into three Speech Acts, all sub-classes of Request Speech Act: Request-Information, Request-Action and Request-Permission while from another point of view they can be divided into two general sub-groups: direct requests and indirect requests. The terms "direct" and "indirect" refer to what was explained before in terms of "direct Speech Acts" and "indirect Speech Acts", (see section 5.2), so "Please tell me how to print postscript files" is a Direct Request-Information and "Can you please put my name in the c-users alias?" is an Indirect Request-Action.

To make it possible to deal with each Speech Act separately, the corpus was manually subdivided into smaller corpora depending on the Speech Acts of the messages. Request-Action and Request-Permission have the maximum and the minimum number of messages in the corpus respectively (600 Request-Action, 300 Request-Information, and 50 Request-Permission).

8.2 n-grams

As mentioned earlier, Becker [Becker, 1975] argues that "utterances are formed by repetition, modification and concatenation of previously-known phrases consisting of more than one word". Some of the more common used phrases can be recognised by counting their n-gram frequencies in a domain. The decision on "n" as the maximum number of the words in a phrase is highly depend on the size of the corpus. For example in the corpus analysed in this research, the maximum frequency of phrases with four words and more is not high enough to be used as the basis for a decision but in big corpus it might be useful to prepare the frequency of even ten words phrases.

Counting the frequency of 1-3 gram phrases occurring in the sentences of particular classes of Speech Acts in this corpus made it possible to pick up the clue words and phrases

which identify requests in the messages. Although some of these clue words are common in all request forms, e.g. "please", they can be separated into sub-classes with additional linguistic rules.

Tables 4 and 5 below show the frequencies of some of 1-3 gram phrases occurring as parts of Request-Information and Request-Action sentences of the corpus. The useless words and phrases like articles, proper names, etc. with higher frequencies are manually deleted from the tables.

1-gram		2-gram		3-gram	
Phrases	freq	Phrases	freq	Phrases	freq
have	91	tell me	41	you tell me	30
can	90	is there	30	Is it possible	15
go	90	could you	28	how do I	13
Is	58	can you	24	do we have	12
?	55	do I	23	could you tell	11
tell	44	how do	18	tell me what	10
please	38	Is it	18	tell me how	8

Table 4: The frequency of 1-3 grams in 300 Request-Information messages

1-gram		2-gram		3-gram	
phrases	freq	phrases	freq	phrases	freq
please	400	could you	180	could you please	60
you	370	please could	68	please could you	53
could	325	can you	68	have a look	25
can	182	can I	18	please can you	12
have	139	please can	17	could someone please	10
me	90	can someone	16	would you please	10
will	60	would you	15	can you please	9

Table 5: The frequency of 1-3 grams in 600 Request-Action messages

The comparison of the above tables shows that although the clue word "please" is a common word in requests, it is used in Request-Action forms much more than in Request-Information forms. In fact while 66.67% of the messages in Request-Action domain contains "please", only 12.67% of the messages in Request-Information include it. The main reason for this difference is that it seems the senders in this domain prefer to request their Request-Actions indirectly and asking their questions directly. In other words, questions like wh-questions or Yes / No questions, which are subsets of Request-Information Speech Act, are in direct form and usually without "please". This claim is proved by noticing that about 20% of the Request-Information messages include "?" which is a good clue for a direct question.

In fact clue phrases in a request search can be divided into two general categories. The first group are "common patterns" which occur in all request messages although with different frequencies; e.g. "please" and the second group is "distinguishable patterns" which are most likely related to a specific sub-class of requests. For example while "tell me" is a clue phrase for Request-Information, "May I please" is a clue phrase for Request-Permission.

8.3 Email text: Spoken or written language

Corpus types, especially in their general categories, spoken and written forms, were discussed in some detail in the previous chapter. It was also mentioned that lexical density is a factor that distinguishes between these two types. Obviously, it is true that statistical figures discussed here cannot be generalised to all types. Although Stubbs [Stubbs, 1996] reports of a lexical density for written texts of over 40 percent, in the range of 36 to 57, and for spoken texts under 40 percent in range of 24 to 43, he also mentions shopping lists with almost 100 percent lexical density.

To investigate the type of Email text based on its lexical density and compare the result with the lexical density of a natural occurring dialogue, dialogue d93-9.1 from the TRAINS corpus (see appendix 1) and a corpus containing 30 Emails were chosen. Both the dialogue and Emails were picked up randomly. The number of Emails in this corpus was chosen in a way so that total number of words in the corpus and the dialogue were close enough to be comparable. Also for more accuracy, all labels and text marks from the dialogue were removed before counting the total number of the words. The total number of words in the dia-

logue is 1136 words and in 30 Emails 1291 words.

Figure 15 below shows an Email from the corpus and a portion of the dialogue. The underlined words are the lexical words.

I have just spent most of the morning updating the phd list saving it as I went along. Then suddenly it would not save it and I had to close the document when I tried to re open it, it would not let me and it now appears to have disappeared. Can someone help me find it please?

The document is called phdlist.

Thanks

---

utt23 : u: um hm next uh <sil> hm <sil> okay I'd like to send <sil>
uh does it take any less time if it's just an engine <laughter>
utt24 : s: no it still takes three + hours +
utt25 : u: + okay + <sil> + um +
utt26 : s: + yep <sil> they have + to go the same speed
utt27 : u: okay
utt28 : could I move <sil>
one engine with two boxcars engine E two from Elmira to Corning
utt29 : s: yeah okay
utt30 : u: and <sil> then one hour later start the next engine from Elmira
utt31 : s: okay <sil> so um sure <sil> so then E three then at one a.m. <sil>
with um how many boxcars
utt32 : u: uh <sil> E three there's only <sil>
there are only + two + boxcars available

Figure 15: Lexical words in an Email and part of a dialogue conversation

We again used the formula: Lexical density = 100 x L / N, where L is the number of lexical words and N is the number of words in a text.

There are 356 lexical words in the dialogue and 508 lexical words in the Email corpus. Also the total number of words in the dialogue and in the corpus are 1136 and 1291 respectively. So:

Lexical density for the dialogue: $LD_d = 100 \times 356 / 1136 = 31\%$

Lexical density for the Emails: $LD_e = 100 \times 508 / 1291 = 39.35\%$

A comparing LD_d and LD_e with the results reported earlier shows the dialogue text is in the range of spoken texts, as expected, while Email texts are closer to spoken language than written form.

8.4 Summary

This chapter presents a detailed analysis of the corpus prepared for the current work. The specification of Email text and the frequencies of phrases found in n-gram tables suggest a pattern matching approach to recognise the requests appear in the texts. The lexical density of the Email text support the idea that the language used in Emails is closer to spoken than written language.

Chapter 9

An approach to the automation of Speech Act recognition

There are many possible approaches to automating the analysis of Speech Acts e.g. [Cohen, 1995], [Lee and Wilks, 1996]. Here we make use of an approach that could be called phrase matching, where that is to be taken, as we shall show, in a broader sense than a "clue-based" approach to Speech Act detection, such as Hinkelman's [Hinkelman, 1989].

From a different point of view, Guthrie, Walker and Guthrie [Guthrie et al., 1994] present a theory for determining, for a given document, into which of several categories it best fits. Although they claim that their theory is maximally effective for routing by features, it is not a suitable algorithm for Email text processing because:

i. The theory is based wholly on the frequency of words in text. Email texts, especially in this domain, are too short for significant counts.

ii. The best practical results they report are achieved for routing between two unrelated subjects: business and terrorism. In the Email task, the topics can be very close to each other.

In the corpus analysed, more than 90% of the messages contain some form of Request Speech Act. Therefore in this research, the main emphasis is on recognising the three forms: Request-Information, Request-Action and Request-Permission. In some of the residual messages, an Inform Speech Act is used to explain the request part of the message and, as mentioned in section 7.2, this could be used to solve some of the co-reference problems in the request sentences.

9.1 REQUESTS

Most requests, because of politeness, include either clue words like "please" [Hinkelman, 1989] or are in the form of an indirect Speech Act or appear as a combination of both.

9.1.1 Request-Information

Request-Information occurs both directly and indirectly in this domain.

a: Direct Request-Information: There are two general classes of request for information: questions and non-question statements in a form which seeks information. Three question types and two non-question types are considered as Requests-Information:

i. wh-questions. Sentences beginning with any wh-words like where, when, etc. are suspected of being wh-questions. To investigate in more detail, these sentences are supposed to be followed by an auxiliary verb such as "do"; e.g. "What do I have to do to set up a web page for local access only?" otherwise they are not wh-questions; e.g. "When I post an article in TIN, it stays only for 30 minutes and then disappears!". Dealing with wh-words like "what", and "how" needs more detailed consideration, because these wh-words can be followed by a noun and make a correct wh-question; e.g. "What Email address should I use for general enquiries about the software on the ACS?"

ii. yes/no questions without modal verbs. Sentences beginning with any word from the set {do, does, did, have, has, had, is, are was, were} are most probably yes/no questions, e.g. "Is there a reasonably fast 486 machine connected to the research network?" The only disagreement found in the corpus is sentences starting with "Have", which need more attention. For instance sentences like "Have a nice day", or "Have fun" are among those sentences which start with "have", but are not questions.

iii. yes/no questions with modal verbs and without "please", e.g. "Can you include me to forum alias?" This form causes most of the ambiguities. While this type is simply a yes/no question, it could be used indirectly as a Request-Action or Request-Permission. As described in the "Indirect Request for action" section, if yes/no questions with modal

verbs contain the word "please", regardless of its location in the sentence, the sentence is considered as an indirect request act. So while the above example is ambiguous between direct Request-Information and indirect Request-Action, "Can you include me to forum alias, please?" is certainly an indirect Request-Action.

iv. Non-question form sentences which include "please" and one of the inform-verbs followed by "me" or "us. Inform-verbs are "tell", "inform", "advise", etc. In this type since the sender does not express his request in indirect form, because of politeness, it is usually accomplished with "please".

"Please tell me how to print postscript files". Direct Request-Information [1].

"Please tell **him** how to print postscript files". Direct Request-Action.

"Please **show** me how to print postscript files". Direct Request-Action.

v. Non-question form sentences which include phrases like "let me know", or yes/no questions with highly informal expressions like "any idea". These phrases have been found in n-grams frequency counts as well as by manually analysing messages. "Any idea", "any suggestion", let me know", "how do I" or "I would like to know" are some of these phrases, for instance "Please let me know which server I am now on?"

b: Indirect Request-Information: This type is considered as an indirect Request- Information because senders require more than just a "yes" or a "no" reply, although the question is in yes/no question form. This type covers questions with modal verbs plus "please" plus any inform-verb with "you" as subject and "me" or "us" as objects. In general these sentences have the form: [please] {modal verb} you {inform-verb} {me, us} [please]. To see the role of the conditions mentioned above consider the following sentences. All of them, except the first one, differ from the first sentence in one of those conditions.

"Can you tell me about this list please?"

"Can you **put** my name on the list, please?" An indirect Request-Action, explained in the next section.

1.Not all the examples in this section and the next two following sections are original sentences from the corpus prepared for this research.

* "Can **he** tell me about this list, please?" Not a common form, but direct Request-Information.

"Can **he** tell me about this list?" A yes/no question and direct Request-Information.

"Can you tell **him** about this list please?" An indirect Request-Action.

"**Are you able** to tell me about this list?" A yes/no question and direct Request-Information.

9.1.2 Request-Action

Request-Actions occur both directly and indirectly in this domain.

a: Direct Request-Action: Most direct Request-actions are identified by non-question sentences with the clue word "please" plus a verb which is not an inform-verb, as defined above. Also requests with inform-verbs which are not followed by pronouns "me" or "us" are considered as Request-Action.

"Please restore my files in my directory". Direct Request-Action.

"Please tell everybody about this seminar". Direct Request-Action.

"Please **tell me** about this seminar". Direct Request-Information.

b: Indirect Request-Action: There are two forms of sentences which are considered as indirect Request-Action. Their main difference is depend on their verbs as being an Inform-verb or not.

Yes/no questions with modal verbs and subject "you" with "please" and without inform-verbs, are indirect Request-Action. Another class of indirect Request-Action is yes/no questions with modal verbs plus "please", plus inform-verbs followed by any noun or pronouns excluding "me" or "us".

"Can you please put my name in the c-users alias?" Indirect Request-Action.

"Can you please tell all students to be there?" Indirect Request-Action

"Can you put my name in the c-users alias?" Direct Request-Information.

"Can **they** please put my name in the c-users alias?" Indirect Request-Action but not from the hearer.

"Can you please tell **me** about the c-users alias?" Indirect Request-Information.

In general, inform-verbs can appear both in Request-Action and Request-Information. Consider these two requests: "Please **tell me** your name" and "Please **tell him** your name". The first is a direct Request-Information and the second a direct Request-Action. One reason for this claim is that the first one can be substituted with a wh-question like "What is your name?" which serves the same purpose, but there is no wh-question for the second one. To deal with this problem, the system should consider the nouns or pronouns that follow the verbs. The pronouns "me" and "us" when coming after inform-verbs indicate that the act is a Request-Information.

It should be noticed that most of the ambiguities in sentences like "Can you put my name on the list?" between Direct Request-Information and Indirect Request-Action is because of the lack of access to mutual beliefs between participants and their relative abilities to perform the action. In another words, in the above example, if the speaker knows that the hearer is able to put the speaker's name on the list then "Can you put my name on the list?" could be understood as an Indirect Request-Action, but if the speaker does not know whether the hearer can do it or not, the above example would be a simple yes/no question which is considered as a Direct Request-Information. In this research, we consider this form of yes/no question as a Direct Request-Information unless the sentence is accomplished with "please", which would change it to an indirect Request-Action

Another interesting point is the meaning of "Indirect Speech Acts level two". This type of indirect Speech Act is very context-dependent and is difficult to identify with a phrase approach. However, in the domain analysed, some of these indirect Speech Acts occurred frequently enough to make it possible to identify them as indirect Request-Actions. For example "rlab1 printer is out of toner." can be considered as a Request-Action as long as we remember that our domain is Emails to a Support Group and that the same message to somebody else would not be considered as a request.

9.1.3 Request-Permission:

Although this Speech Act has some similarity with Request-Information or Request-Action, it constitutes a different Speech Act. In this act, the sender seems uncertain about being allowed to have access to information or perform an action. For example in "Can I get a boot disk for

the PCs on the research network, please?" or in "May I have access to /share/nlp/directory?" the sender neither asks for information nor makes a request, but asks for permission, although in some situations the above examples could be considered as a polite Request-Action.This ambiguity is largely one of UK/US dialect difference, since in the US, 'may' is reserved for Request-Action e.g. 'Can/May I have the salt?' is a UK/US alternation for the Request-Action, and one on which Americans pick British speakers up by saying 'Do you mean "may you have it?"' The 'may' form is largely Request-Permission in the UK. Pyam therefore considers this request as indirect Request-Permission. Most Request-Permission messages appear as indirect Speech Acts in this domain.

By preparing a table ranked according to some priority of authority, the system is able to resolve some ambiguities between Request-Permission and Request-Action that depend on the sender. For instance, in this domain, while "May I have access to nlp directory" from an undergraduate student is more likely to be a Request-Permission; from a member of the staff, it is a Request-Action.

In this domain, most Request-Permission messages appear as indirect Speech Acts.

a: Direct Request-Permission Speech Acts are recognized as non-question sentences including "please" plus verbs such as "let", or "allow", with pronouns "me" or "us".

"Please let me use your terminal." Direct Request-Permission.

"Please let **them** use your terminal." Direct Request-Action.

"Please **give** me your terminal." Direct Request-Action.

b: Indirect Request-Permissions are yes/no questions starting with "May" and followed by "I" or "we" as a subject. Another type of Indirect Request-Permission is a yes/no question with any modal verb and "I" or "we" as the subject, plus the clue word "please". A general precondition for Request-Permission is that the rank code (see section 10.1.b) of the sender be lower than the rank of the user. In both cases, the rank code of the sender is compared with the rank code of the user. If the previous precondition, the rank code of sender being lower than the sender, does not hold, as in case of equality or undefined rank of the sender, the system considers the Speech Act as ambiguous between Request-Permission and any of the two other requests. If the rank of the sender is higher, the system considers the Speech Act as Request-

Information or Request-Action as defined before. In the following examples, the rank of the sender is supposed to be lower than the rank of the user:

"May I use your terminal, please?" Indirect Request-Permission.

"Can I know your name, please?" Indirect Request-Permission.

The same messages from some one with a higher rank code are as:

"May I use your terminal, please?" Indirect Request-Action.

"Can I know your name, please?" Indirect Request-Information.

and from an unknown ranked sender as:

"May I use your terminal, please?"

Indirect Request-Permission or Indirect Request-Action.

"Can I know your name, please?"

Indirect Request-Permission or Indirect Request-Information.

9.2 INFORMS

In the analysis of the corpus, most of the Inform Speech Acts function as either a pre-explanation or post-explanation of requests. "Will you have a look at it? There appears to be a fault on the CD drive." or "The printer queue of rlab2 seems to be stuck. Could you please have a look at it?" are examples where, (in the former) the inform part of it serves as a post-explanation, and, (in the latter) as a pre-explanation. Although these parts of messages do not generally contain the gist of the message, they can be used to substitute for any possible co-references in the request act. It gets more important if you are able to browse only the request part of the messages. It would be more informative to see, in the first example, "Will you have a look at [the CD drive]" instead of "Will you have a look at it". The next chapter explains how this could be achieved.

9.3 Summary

As mentioned before, Searle argues that the area of directives (which includes Requests) is the most useful to study. This chapter has explained "Request" Speech Acts which occur very frequently in the corpus prepared for this research. Three types of Requests: Request-Infor-

mation, Request-Action, and Request-Permission were distinguished. This chapter also considered all these three types in direct and indirect forms. In addition, although the "inform" Speech Act was not examine in detail, part of the text including "inform" Speech Act was found useful as pre-defined or post-defined part of a "Request" Speech Act.

Comparing the rank code of a sender against that of a receiver was introduced to deal with the ambiguity between Request-Permission and other forms of request.

The next chapter describes an implemented system based on the approach which was explained in this chapter.

Chapter 10

Implementation

There are a variety of options for Natural Language Processing systems to deal with Information Extraction. The most common way is using some sort of syntactic and semantic parser with a part of speech tagger. In the domain of Email messages, this is not a good alternative because, as mentioned before, Email messages usually contain many misspelled words, fillers and abbreviations which make it difficult to parse such a message properly section 8.1.3). However the lack of correct punctuation is the main problem for considering such an approach. Based on Jones [Jones, 1994] definition, punctuation are those non-lexical marks found in written texts: commas, colons, semi-colons, full-stops, question marks, exclamation marks, open and close brackets and parentheses, quotation marks, speech marks and hyphens. In the system implemented for this work, those punctuation which could be used at the end of sentences are most considered as sentence separators, such as full-stops or question marks.

Another alternative, as used in the MUC framework, is defining different templates to extract the relevant information in a message. This approach is recommended only if the inquired information is pre-defined; i.e. proper names, dates, or organisations. A quick overview of the subjects of the Emails in the mentioned domain shows that too many templates would have to be defined to extract required information; i.e. file management, tools, aliaes, etc.

The approach on which this research is based, is a pattern matching approach. Corpus analysis explained in chapter 8 and Speech Act definitions in the previous chapter lead us to implement a system which be able to recognise and differentiate Request Speech Acts.

10.1 Pyam

The implemented system Pyam, which is written in PERL consists of four sections: Email header-lines, preprocessing, focus sentences and report generation.

a. Email header lines

Pyam picks up the three most informative items from message header lines: Sender, Receiver, and Subject (and ignores the rest).

"Receiver" is used to find out whether the Email is a personal Email or sent to a group.

'Sender' is checked against prepared files, based on the rank of the sender compared to the receiver, to help disambiguate issues of Request-Permission: e.g. an undergraduate would normally be Requesting-Permission if the receiver were an academic but an academic would normally be requesting action if the receiver were a member of a support group. These ranking files are established in accordance with the individual rankings assigned by the end user of the system.

"Subject" is the most informative among the header lines. Although some users' habit of using the "Subject" line as part of the main text makes it difficult to give the Subject line a separate role in understanding a message, it is still an informative header line. Also from a historical point of view, the "Subject" header line shows whether the message is an original Email or a reply to a previous message. Usually a reply to a previous Email contains "Re:" at the beginning of the subject line.

b. Preprocessing

There are several jobs carried out by the preprocessing section. Abbreviated words, which have been hand listed, are replaced by their full words; e.g. "please" for "pls", and the sentences boundaries of the message are inserted.

Sentence recognition, in this sense, is one of the most difficult problems, especially in this domain. Sentence separators are not used in standard ways in Email: e.g. some people use uncommon separators like "-" and some use them repeatedly, like "???" and some do not use

separators at all (see figure 20). Some of the separators have multiple usages one of which is to act as a sentence separator; e.g. full stop. Other uses of the full stop are found in numbers (3.5), titles (Mr.), acronyms (N.L.P.) etc. To solve this problem and find out when a full stop is used as a sentence separator, a list of titles, such as (Dr.), which are hand extracted, are defined in the system. Acronyms are recognised as a sequence of one letters, in upper or lower case, each followed by a full stop. Pyam splits a message into sentences by the sentence separators after recognition of titles, numbers and acronyms and replacing multi separators by single one except for "..." which is replaced by etc.

Since it is necessary to recognise some of the focus sentences by their first word; e.g. yes/ no questions or wh-questions, the preprocessing module must remove what Schiffirin [Schiffrin, 1987] called "discourse markers", for example "also", "so" "and", etc. from the beginning of a sentence. Although it is not common in formal language that a sentence starts with these words, but it happens very often in Email messages. To recognise the negative form of yes/no questions properly, the preprocessor replaces abbreviated forms of negative modals with their full forms, e.g. "can not" for "can't".

The next job performed by the preprocessor is removing the greeting words from the beginning of the first sentence. Like titles, a list of these words and phrases, such as "hi", "good morning", etc. are hand extracted and supplied to the system. The problem with these words is that, since they are not separated from the first sentence by any separator, they appear as the first words of the sentence, which has an effect on those sentences whose first word would otherwise be a clue word such as wh-questions or yes/no questions.

At this stage, by comparing the user's rank with that of the sender, the rank code of the sender is determined. This rank code is used in the report generation section to assign the proper Speech Act to any ambiguous focus sentences, and differentiate Request-Permission from Request-Action or Request-Information. Any end user of the system is asked to prepare two data files in his/her directory. One file contains the names of all possible Email senders who have higher rank than the user and the second file contains the names of those with lower rank. If the name of a new received Email is not found in any of those files, Pyam considers him/her as of equal rank, although updating those files according to the status of the new name is always possible and recommended. At the end of this stage, rank codes -1, 0, 1

respectively are used to indicate lower, equal and higher rank.

c. Focus sentences

Sentences in messages that match any of the patterns mentioned in the previous chapter, are considered as focus sentences. There are two types of sentences which are recognised as focus sentences: these are based on either their syntactic structure or their linguistics patterns. All yes/no questions (either with modal verbs or without them, both in positive or negative forms) are examples of syntactically based focus sentences. These are recognised by finding any of the modal verbs as the first word of the sentence. The only exception are those sentences that start with "have", so as to avoid mismatching sentences like "Have a nice day". This particular auxiliary verb should be followed by a pronoun such as "you" or "they" to be considered as a focus sentence. All wh-questions are also picked up as focus sentences.

Another type of focus sentence is recognised by looking for a pre-defined phrase on the sentence; these phrases are established by high frequency in the n-gram analysis of the corpus. For example, phrases such as "let me know" or "any idea" are considered as a pattern matched for Request-Information, but phrases like "May I" are related to Request-Permission, Request-Action or Request-Information Speech Act depending on the rank code and the verb.

Table 6 below demonstrates how the rank code of the sender and the main verb of the focus sentence affect the assigned Speech Act:

Focus Sentence	Rank Code		
	Higher	Equal or unknown	Lower
May I **know** your address?	Indirect Request Information	Indirect Request Information or Indirect Request Permission	Indirect Reuest Permission
May I **have** access to your directory?	Indirect Request Action	Indirect Request Action or Indirect Request Permission	Indirect Request Permission

Table 6: Speech acts assigned to ranked Focus Sentences

Another pattern is "please" which is a general clue for all requests and any sentences which includes this word is considered as a focus sentence. More investigation, including checking verb and object, is necessary to assign the corresponding Speech Act to this sen-

tence.

Finally, any sentence with a "?" as its sentence separator is considered as a focus sentence. If all attempts fail to recognise a sentence with a "?" as a request, the system considers it so by default. Although it is very unlikely that Pyam defines a specific Speech Act to this type of focus sentence, it is still reasonable to report it to the user as an "UNKNOWN" request. An example below from an original Email demonstrates this point:

```
From Ted
To: support
From: Ted
Subject: CDROM
X-Lines: 6

So how do you load one of these CDROM things anyway?
And remove a CD from the drive?
```

Pyam recognises the first sentence, after deleting discourse marker "So", as a Request-Information. The question mark in the second sentence leads Pyam to pick up the sentence as a focus sentence and reports it as an UNKNOWN Speech Act. The Pyam's report for above message is:

```
SENDER:  Ted
RECEIVER: support
SUBJECT: CDROM
==========
The focus sentence(s) are:
==========
how do you load one of these CDROM things anyway?
remove a CD from the drive?

1. how do you load one of these CDROM things anyway?
    Wh-Question, Request-Information.

2. remove a CD from the drive?
    Not Known.
```

d. Report Generation

Pyam assigns a Speech Act to each of the focus sentences of the message recognised as described above. Since parts of patterns overlap in different Speech Acts, Pyam has a hierarchy order which ranks the most specific part of pattern first above the most general part at the end. For instance, in the sentence "Can you tell me about this list please?", considering "please" as a common pattern between all request Speech Acts at the first match will not be helpful. Instead, Pyam first recognises it as a yes/no question with modal "can"; then its verb, "tell", which is one of the "inform verbs"; then its direct object which is "me" and finally "please". Based on the above information, Pyam considers the sentence to be an Indirect Request-Information. Figure 16 below shows the decision chart which determines the Speech Act for each focus sentence.

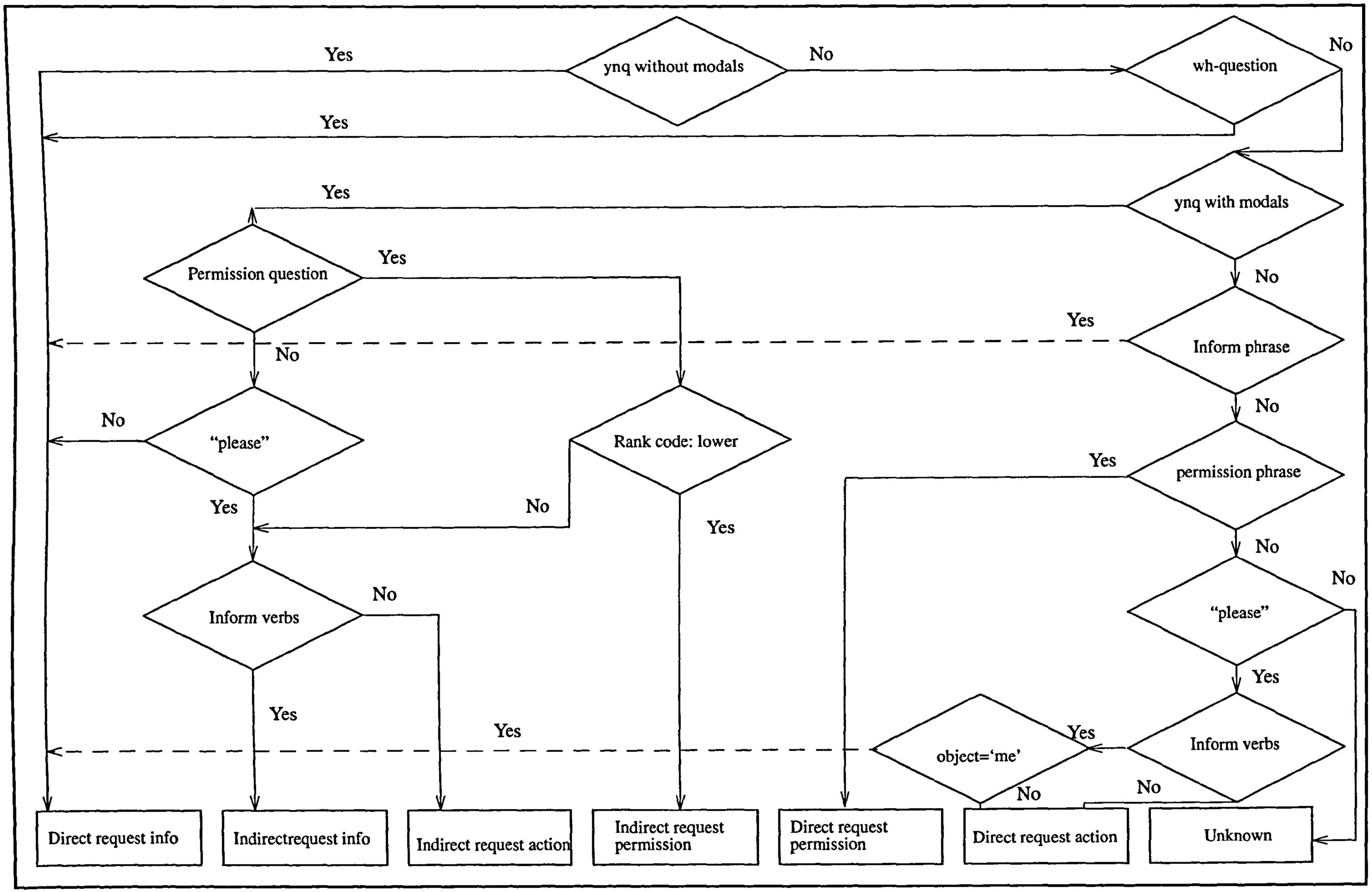


Figure 16: Flowchart of a Speech Act Decision

Pyam prints out partial messages containing the Speech Act of each focus sentence(s) as follows. The output for each Email message has three parts: Header-lines, Focus sentences, and Speech Act results. The Header-lines normally contain the Sender, Receiver, and Subject. For each focus sentence, Pyam prints out its corresponding Speech Act type. Figure 17 below shows an original Email message and the results prepared by Pyam.

From Ted

Date: Mon, 16 May 94 16:12:34 BST

From: Ted

To: support

Subject: question re. solaris 2

Content-Length: 109

Status: RO

Please tell me what issues would be involved in installing solaris 2.4 on a machines. there's some sun software that doesn't really get supported for s1 anymore. Could you please send me any related documents?

SENDER: Ted

RECEIVER: support

SUBJECT: question re. solaris 2

==========

The focus sentence(s) are:

==========

Please tell me what issues would be involved in installing solaris 2.4 on a machines.

Could you please send me any related documents?

1. Please tell me what issues would be involved in installing solaris 2.4 on a machines.

   Request-Information.

2. Could you please send me any related documents?

   Literally Yes / No Question. Indirect Request-Action.

Figure 17: Original Email and the result output by Pyam

Following the discussion above, we have ten different Speech Act message types:

1. Request-Information.

2. Yes/No Question, Request-Information.

3. Wh-Question, Request-Information.

4. Literally Yes/No Question. Indirect Request-Information.

5. Request-Action.

6. Literally Yes/No Question. Indirect Request-Action.

7. Request-Permission.

8. Literally Yes/No Question. Indirect Request-Permission.

9. Request-Permission or Request-Action.

10. Request-Permission or Request-Information.

These messages express more information than just their Speech Acts. For example, there are three different messages for Request-Information, depending on the type of sentence. This information could be used in future work if the structure of the sentences are important and a system distinguishes between yes/no questions and wh-questions, for example. Messages 9 and 10 are reserved for situations when the rank of the sender is unknown and the final decision is left to the receiver to make.

There are two other messages output by the system. The first one is: "This message has no REQUEST Speech Act." which appears when Pyam finds no request, and the second is "NOT KNOWN", which appears when a conventional phrase appears that allows it to be categorised as a request, but Pyam is unable to associate a conventional Speech Act. In a message like, "I used to print file from Word for Windows in the PC before but now I can't!!?"; although the sentence is really a Request-Information, Pyam is unable to recognise this Speech Act. In fact the only available clue is "?", but there is no other pattern which makes it possible to recognise the correct Speech Act. The appearance of these diagnostics in the focus sentences should attract the attention of the end user.

In systems such as Pyam, which are based on phrase matching, an interesting question would be how the order of the words in the phrases might effect the results captured by the

system. To investigate this question it should be remembered that all focus sentences picked up by Pyam can be divided into two groups. In one of them the first word of the sentence is important and plays the main role in recognising the sentence. For example, yes/no questions with modal verbs or wh-questions fall into this group. The main differences in the results after a reordering of the words of a phrase would happen in this category. Table 7 below shows how changing the order of the words affects the Speech Acts recognised by Pyam.

Sentence	Speech Act
Can you pass the salt please.	Indirect Request-Action
Please can you pass the salt.	Direct Request-Action
Can you tell me your name please.	Indirect Request-Information
Please can you tell me your name.	Direct Request-Information

Table 7: The effects of changing word order on the Speech Acts.

This relationship is only true when questions with "please" are acceptable. In fact wh-questions or yes/no questions without modal verbs are not accomplished with "please".

The second group of sentences is found by matching pre-defined phrases. In this group, the order of the words has no effect on the results captured by Pyam. So "Please let me known its address.", "Let me know its address.", or "let me know its address please" are all recognised as Direct Request-Information.

10.2 Co-reference substitutions

In this domain most of the Inform Speech Acts accompany requests as pre- or post-explanations (see section 7.2). To substitute the co-references in the inform parts into the request part Pyam uses the co-reference module of LaSIE Information Extraction system [Gaizauskas et al., 1995], [Cunningham et al., 1995] which is part of a larger Information Extraction system. So, in the first example, the system is able to substitute "CD drive" for "it" and in the second example it substitutes "the printer" for "it". Figure 18 below shows an original message, the LaSIE's output and the substituted result.

The monitor on my machine is flashing like a faulty TV screen. Can some one look at it please?

<DOC>
<DOCNO> 1234 </DOCNO>

<s> <COREF ID="1">The monitor</COREF> on my machine is flashing like a faulty TV screen. </s>

<s> Can some one look at <COREF ID="2" TYPE="IDENT" REF="1">it</COREF> please?. </s>

</DOC>

The monitor on my machine is flashing like a faulty TV screen.

Can some one look at [The monitor] please?

Figure 18: Use of LaSIE to substitute co-references

In the above example, [The monitor] of the first sentence, which is an Inform Speech Act type, has been substituted for "it" in the second sentence. When Pyam picks up "Can some one look at [The monitor] please?" as an indirect Request-Action, it would be more informative than the original sentence. The brackets clarify the substituted part. Figure 19 below shows the original message and the results captured by Pyam after using LaSIE to substitute co-references:

From Ted

To: support

Subject: Monitor

Content-Length: 109

Status: RO

The monitor on my machine is flashing like a faulty TV screen.

Can some one look at it please?

SENDER: Ted

RECEIVER: support

SUBJECT: Monitor

==========
The focus sentence(s) are:
==========

can someone look at [The monitor] please?

1. can someone look at [The monitor] please?

   Literally Yes / No Question.

   Indirect Request-Action.

Figure 19: Pyam's result after using LaSIE to substitute co-references

10.3 Summary

Chapter 9 defined three types of request Speech Acts: Request-Action, Request-Information, and Request-permission. The definition was based on corpus analysis (chapter 8) plus some pragmatic rules. In addition, for each of these three request Speech Acts, the way to distinguish direct from indirect speech acts was explained. In this chapter, an implemented system for carrying out the recognition of these six types of speech acts was described (Pyam). The system produces appropriate Speech Act classification message for any focus sentence (see section 10.1.c) found in the Email texts.

The next chapter describes the evaluation of the performance of Pyam in different stages.

Chapter 11

Evaluation

Although no evaluation tool has been developed for Natural Language Processing systems in general, but only for sub areas, such as Machine Translators etc. [Sparck Jones and Galliers, 1996], it should be possible and useful to prepare an evaluation result for any single application in this area [Flickinger et al., 1987] if the work is to conform to normal standards of experimental acceptability.

There are two relevant qualities that can be evaluated. The first is the functionality of the system, i.e. whether or not it does what it is supposed to. This can be achieved by comparing its results with either another existing system or manually prepared results. The second is the usability of the system i.e. how easy it is to use the system. In fact, 'usability' has several aspects; for instance, the readability of the output is part of usability. A decoded or marked up text might not be a readable output, although it might serve as a good input for some further system. The system's response time to a query is another usability feature. If it takes a very long time for a system to respond to a question, the user may lose interest. For more detail on evaluating NLP systems see [King, 1996].

11.1 Functionality of the system

In terms of the functionality of this system, Pyam has been evaluated in three phases.

11.1.1 First phase of evaluation.

For the first stage of evaluation, 100 Email messages (different from the corpus messages) were marked by a volunteer; a native English speaker and PhD student in Natural Language

Processing. Each message had four possible Speech Acts attached. We avoid including directness and indirectness of speech acts to bypass any confusion and get the general reaction of the volunteer. Figure 20 below is an example from 100 messages with 4 possible answers:

> EG paper wont be ready till tomorrow, sorry. Please try to collect by 10.30 to take a look--lets get this one out and away and never discuss it again (Mark wd much apprecviate it if you cd ocme--some of yr words of wisdo on ear;lier versions I cdnt read my notes on!!)
>
> ALSO--I only have your mini 94 publocation lists for some of you--pls send if I havent had it already.
>
> 1- Request-Action ( )
>
> 2- Request-Information ( )
>
> 3- Request-Permission ( )
>
> 4- None ( )

Figure 20: An example for the first evaluation stage

The shortest Email had one sentence with 8 words and the longest message had 8 sentences and 75 words.

Comparing the marked results with the results prepared by Pyam showed that more than 80% of the Speech Acts assigned by Pyam were correct. The main disagreement between the subject and Pyam was about certain indirect Request-Action acts, which Pyam considered as Request-Information acts. For example in:

> "Could you have a look at the 486 on bench V1 in the Lewin Lab. It seems to have only 4mb of RAM available. This makes programs like Access run very slowly"

Pyam considered this request as Request-Information but the volunteer subject as Request-Action.

11.1.2 Second phase of evaluation.

In the second step, a new corpus was prepared from Email to the Computer Support Group, but different from the corpus that was used to formulate the criteria of the previous chapter.

The focus sentences (see section 10.1.c) of 24 messages were underlined and the Speech Acts for each of the focus sentences were assigned by three volunteers. This evaluation was done by a different group (from the first volunteer) to make sure that they did not have any prejudgment about the results prepared by the system. This was for the same reason as asking volunteers to prepare the manual results instead of the author. In addition, the subjects were unaware of the directness or indirectness of the Speech Acts.The most important point was their understanding of the messages without any confusion between a reply based on a direct or an indirect answer.

Although it was a small set for evaluation, considering three volunteers instead of one makes it more reliable. The shortest message had one sentence with 8 words and longest message 5 sentences had 78 words. The total number of text words (excluding header lines and signatures) were 770 words with an average of 32 words per message.

Here is an example of the corpus marked up by one of the volunteers.

From Ted

To: support

Subject: Windows does not run on Lab4 PC

We cannot run Windows on our PC. The title screen does not appear instead we get an error message involving, "PCNFS.386". Could you change the setup for us please?

Thanks,

R-ACTION	R-INFORMATION	R-PERMISSION
x		

Table 8: Speech Act marking in the test domain.

The distribution of the Speech Acts in this domain test had 11 Request-Actions, 9 Request-Informations and 4 Request-Permissions, either in direct or in indirect forms. This distribution between Speech Acts makes the results more reliable than in a situation where the domain contains a preponderance of a specific Speech Act.

In comparing the results captured by Pyam with the results prepared by the volunteers it

was noticed that, although there were 84 sentences in the messages with an average of 3.5 sentences per message, there was only one sentence underlined by two volunteers as a focus sentence which Pyam did not pick up. The underlined sentence in the following message is the one missed by Pyam but picked as a Focus Sentence by two subjects:

> From Ted
>
> To: support
>
> Subject: rlab1 printer
>
> Is the printer really out of order? If so, perhaps it would be a good idea to cancel the print queue and send us an email with information about when it is expected to be repaired.

Those subjects have considered this sentence as a Request-Action.

In Speech Act recognition, all three volunteers agreed with Pyam about 19 of the 24 Speech Acts (three different Request Speech Act types) in the focus sentences. In three of the messages two of them agreed, and in two Speech Acts only one of the volunteers agreed. Table 9 below shows that 90.28% of the results captured by Pyam are correct.

No. of volunteers agreeing with Pyam	No. of messages for which they agreed	Grades	Percentage
3	19	57	79.17%
2	3	6	8.33%
1	2	2	2.78%
Total	24	65	90.28%

Table 9: Pyam's evaluation result

In the calculation of this result, each agreement in the assigned Speech Act between any of the volunteers and the system has been considered as a grade. For instance when all volunteers agree on 19 message, Pyam gets 57 grades. The maximum possible grade is 72, when all three volunteers agree in all 24 available messages and Pyam achieves 65 grades which is 90.28%.

In the case of majority agreement, when at least two of three volunteers agree with Pyam, 91.67%, (100 x (19 + 3) / 24), of Pyam's results agree with at least two of the three volunteers.

11.1.3 Third phase of evaluation.

The last phase of the evaluation differs from the two previous ones about the nature of the message receiver. The system was originally trained on Emails to the Technical Support group and tested by different Emails to them. For this evaluation all Emails were taken from Emails to my supervisor. There were 100 random Emails to him and the author was unfamiliar with the content of the messages. They varied greatly in length and subject.

The longest message had 113 sentences and the shortest one only one sentence, but the average of sentences per message was ten sentences. The maximum number of Requests in a message was six while 33 of the messages had no Request and the rest had 116 Requests in their different forms.

Table 10 below shows the results prepared manually and by Pyam for these Emails. The first row shows the results for Focus Sentences (F.S.) and the second row shows the results for Speech Acts (S.A.)

	Manually	P Y A M				
	Recognised	Recognised	Correct	Wrong	Missed	Result
F.S.	116	116	108	8	8	95.73%
S.A.	116	91	82	9	21	84.58%

Table 10: The last evaluation results

From the 100 messages, the Requests in 79 of them have been recognised correctly both for F.S. and S.A., which demonstrates reasonable generality across Email domains by Pyam. The main reason that the number of recognised F.S.s is more than S.A.s is that some sentences in the messages are marked with "?" as a question without any further clue patterns, so the system picks them as a F.S. but the system is unable to decide on its S.A. For example in this

part of a message: "I'd like Sheffield to start with some comments about Aventinus. Agree? Comments?", questions like "Agree?" or "Comments?" are examples of this problem. Unrecognisable S.A.s might even happen in a full sentence like: "Regarding the dates, the 13th would be best for us if this is OK with you?".

The missed F.S.'s are mostly declarative, for example "I am writing to ask if you would be prepared to give an interesting talk or demonstration in a local school."

Calculation Methodology:

To calculate the accuracy of the results captured by Pyam against manually prepared results, for each message a maximum of one point for its F.S. and one point for its S.A. has been considered. In case of full recognition, the system gets one for each part. In the case of partial recognition, it gets a fraction of a point depending on the number of F.S.s and S.A.s found by manually searching. It should be mentioned that for any missed F.C., the system has been penalized twice, for both the F.S. point and for the S.A. point.

11.2 Usability of the system

Considering the usability of the system, Pyam's response time is very fast. In fact even for a long message and with any number of focus sentences, its response time is just a few seconds.

As mentioned in the previous chapter, the output messages of Pyam are in familiar English words. As an alternative it is possible to print the output in some sort of SGML [Goldfarb., 1990] marked-up text. The output result, if necessary, could be produced in the this way:

<FS TYPE= "*t*" SA= "*r*"> *sentence* </FS>

"t" can be either "D" for Direct Speech Act or "I" for Indirect Speech Act and "r" can be "RI" for "Request-Information", "RA" for "Request-Action or "RP" for "Request-Permission". For example <FS TYPE="I" SA="RA"> Can you please come to have a look at the monitor of warthog in rlab1? </FS> means the marked sentence has an Indirect Request-Action Speech Act. A marked-up corpus as described above has the potential to be used as the input text for different types of Natural Language Processing systems.

11.3 Evaluation of the system by other domains.

To evaluate the efficiency of Pyam further in different domains, two other corpora have been tested: The British National Corpus, and TRAINS [Allen and Schubert, 1993], which was collected as part of the TRAINS project [Allen et al., 1995]. Both these corpora are task-oriented spoken dialogues.

11.3.1: The British National Corpus

The BNC is a very large corpus with over 6 million sentences and over 100 million words of modern English, both spoken and written. All dialogues in the BNC are SGML tagged.There are 268 question-patterns for n-grams ($1 < n < 9$) in the dialogue part of this corpus. For instance, "What V you V to do?" is a 6-grams patterns with a surface as "What are you going to do?".

To elaborate the way that Pyam has been evaluated for this corpus, the 7-gram question patterns are summarised in table 11 below. The third column of the table shows whether Pyam has been succeed in recognising each question pattern or not. As shown in the table, Pyam succeeded on only 5 of 13 patterns, while 6 of the 8 failures are because of the tag question patterns. Table 12 below summarises the results examined by Pyam for different n-gram patterns.

7-grams question pattern	Surface	Failed/ succeed
It is ADV ADJ is not it?	It 's too fragile is n't it	Failed
It is a ADJ N is not it ADV	It's a cultural thing is n't it	Failed
Any N on N N N the N	Any advance on seventy five pounds the lot	Succeed
Any N on N N the N	Any advance on twenty pounds the lot	Succeed
N N N N N N N N N N N N N N ADV many ADV we V	One two three one two three four five six seven eight nine ten how many more do we need	Succeed
That is a ADJ N is not it	That 's a better bit is n't it	Failed
he V that does not he	Aye oh he loves that does n't he	Failed
Any N on N N and N N the N	Any advance on one hundred and fifteen pounds the lot	Succeed
and he V and he V what	and he goes and he goes what	Failed
It is ADJ ADV is not it	It 's easy really is n't it	Failed
That is ADV ADJ is not it	That 's very convenient is n't it	Failed
What is the N of the N	What 's the cost of the Havanas	Failed
It may V a N of N for some of the some of you but what you V	It may take a bit of imagination for some of the some of you but what do you think	Succeed

Table 11: 7-gram question patterns and results examined by Pyam

54.1% are succeed patterns while 17.9% of the failed patterns belong to tag-question patterns which do not occur very frequently in the domain of Emails. Some examples of the unrecognisable tag-questions in the BNC corpus are: *"We've already got that haven't we?"* or *"It is the twenty fourth of September isn't it?"*. It seems that tag-questions are good patterns in dialogues and cannot be used in a one-way communication.

n-grams	# of patterns	Succeed		Fail		tag questions	
8-grams	5	4	80%	1		1	
7-grams	13	5	38.5%	8		6	
6-grams	50	26	52%	24		18	
5-grams	50	35	70%	15		8	
4-grams	50	32	64%	18		3	
3-grams	50	25	50%	25		4	
2-grams	50	18	36%	32		8	
Total	268	145	54.1%	123	45.9%	48	17.9%

Table 12: The results of evaluation of Pyam over the BNC

11.3.2: TRAINS

This corpus contains task-oriented spoken dialogues: 98 dialogues, about 5900 speaker turns, and 55000 transcribed words. The main problem dealing with the TRAINS corpus is the way that dialogues have been segmented: utterances can have more than one sentence without any end of sentence indicator. Pyam was able to recognise all the requests which appeared as defined patterns described in the previous chapters, but unable to capture requests embedded in another sentence. For example in dialogue d93-26.3

utt1: s: hello can I help you

utt2: u: yes um <sil> to take two boxcars <sil> with the <sil>

two engines from Elmira <sil> to Corning <sil> is <sil>

would be how many <sil> hours

Pyam is able to recognise "*can I help you*" as a request but is unable to recognise "*is would be how many hours*" as a Request-Information. The following example is part of a dialogue in which all bold sentences are requests and those which are underlined are recognisable by Pyam.

utt9 : u: okay <sil> um <sil> **how long does it take <sil> to get from Corning to Dansville**

utt10 : s: Corning to Dansville takes <sil> one hour

utt11 : u: okay uh <sil> **only one train can be on the track at one time**

utt12 : s: right exactly

utt13 : u: **even if they're heading both in the same direction**

utt14 : s: um <sil> just as long as there's like <sil> some like <sil> time in <sil> between them

utt15 : u: okay <sil> um <sil> hm

utt16 : uh <sil> **can more than one boxcar be put on <sil> an engine**

utt17 : s: yeah

utt18 : u: + okay +

utt19 : s: + so there + can be at most three <sil> boxcars <sil> on an engine <sil> + or like + tanker cars

utt20 : u: + okay +

utt21 : well <sil> first of all I'd like to s- <sil> start off <sil> by sending <sil> uh the engine E one from Avon to Dansville

utt22 : s: okay E one <sil> from <sil> Avon to Dansville okay

utt23 : u: um hm next uh <sil> hm <sil> okay I'd like to send <sil> uh **does it take any less time if it's just an engine** <laughter>

The requests recognised by Pyam are all in direct Request-Information form. Unrecognisable questions are those either embedded in a sentence; e.g. utt23, or are in question form because of their intonations; e.g. utt11. Appendix 1 contains the complete dialogue with 134 turns. There are twenty one requests for information of which Pyam is able to recognise fifteen, which is more than 71%.

Obviously, the best performance of the system is observed dealing with Emails with 90% correct Speech Act recognition

11.4 Summary

This chapter reviewed different stages of Pyam's evaluation. The lack of availability of human subjects to mark up large scale messages, forced us to consider a small set of tests at this stage. However, the results obtain by Pyam show that the system is able to recognise three types of request Speech Acts, in both direct and indirect forms, especially in the domain

of Emails as explained in chapter 9. My own regular use of Pyam, especially when reading long Emails, shows the system has the potential for daily use for end users.

The lower performance of Pyam when dealing with dialogues, because of the difference between genres, suggests that further investigation is required for this application.

Chapter 12

Conclusion and Future work

12.1 Introduction

The main objective of this research has been to design and implement an approach to locate the gist of Email messages by recognising all the Speech Acts that occur in the text. It could be described as Speech-Act routing over a reasonably large sample corpus. The approach is a combination of pattern matching and pragmatic rules. Although recognising all possible Speech Acts is a long term aim, this research suggests a practical approach to identify forms of the "Request" Speech Act, especially in a pre-defined domain.

The major contributions are:

Identifying text language type.

The lexical density test, implies that in the corpus analysed in this work, the language used by Email senders, is closer to spoken language than to written.

Defining "Request" Speech Act types.

The Request Speech Act is one of the most important aspects of an utterance to be recognised, in order to find out the gist of a message. The present work has concentrated on three sub-divisions of Requests as Request for: Information, Action, and Permission. Other types of Requests such as "Contra check" are ignored, because they occur more often in conversational situations rather than long distance messages.

An algorithm to recognise Speech Acts

Patterns found more frequently in a domain, taken together with linguistic rules make it possible to recognise most of the Request acts in the corpus. The result of the evaluation of the system is encouraging and suggests this approach is a good option to consider to implement a fast and friendly system for routing Email by function.

Evaluation

The results of three independent evaluation stages of the system indicate that while the best performance of the system is captured from Emails to the Support group, results for other corpora could be improved, by adding additional rules and specifying the domain by investigating its n-gram frequencies.

The results captured by Pyam show that most Requests-Information, Action, and Permission, both directly and indirectly, can be recognised by this approach.

12.2 Future directions

There are several options to follow up this work.

1. More Speech Acts

As mentioned before, the long term aim of this research is recognising all standard Speech Acts. A preliminary investigation shows that some Speech Acts are easier to capture by this approach than others. "Promise", "Offer", and "Suggestion" Speech Acts may well be easier to capture than "Inform".

2. LaSIE

In the field of Information Extraction, using LaSIE as an option to solve the co-reference problem and get more informative results, is an alternative direction to follow, although it might have the side-effect of a longer response time.

3- Same techniques in different domains

This approach could be used in other domains as well, such as standard "Call for papers" messages, or "Seminar announcements", where it would be possible to prepare structures for the most important phrases including "date", "place", etc. and extract the relevant informa-

tion, thus determining the function of the message and also capturing the differences between conference calls for papers, announcements, results of paper submissions and so on, so as to provide a service to users that captured more content, as well as the broad routing of speech act types we offer here. Standard information extraction modules for dates and places, which are known to work well over a wide variety of surface forms in English, could augment this fairly straightforwardly.

12.3 Summary

This chapter reviewed the major contributions of this research and suggested some future directions. From the evaluation point of view, since this research tackled only Request Speech Acts as the main speech act to work on, there was no reason to think about lower or higher success rates for other Speech Acts.

Applying the approach described in this research to other domains such as "Call for papers" messages, or "Seminar announcements" was one of the suggested future directions. Comparing these domains with Email text suggests that, since these new possibilities are plainly highly stereotyped texts, it might be reasonable to expect better results.

REFERENCES

Allen, J. (1987). *Natural Language Understanding*. Benjamin/Cummings Publishing Company, CA.

Allen, J., Ferguson, G., Miller, B., and Ringger, E. (1995). Spoken Dialogue and Interactive Planning. In *Proc. of the ARPA Spoken Language Technology Workshop*, Austin, TX.

Allen, J. and Schubert, L. (1993). Language and discourse in the TRAINS project. In Ortony, A., Slack, J., and Stock, O., (editors), *Communication from an Artificial Intelligence Perspective*. Springer-Verlag, Heidelberg.

Allen, J. F. and Perrault, C. R. (1979). Analysing intention in utterances. In Grosz, B., Sparck Jones, K., and Webber, B., (editors), *Readings in Natural Language Processing*. Morgan Kaufmann, Los Altos, CA.

Aretoulaki, M. (1996). COSY-MATS: A Hybrid Connectionist-Symbolic Approach To The Pragmatic Analysis Of Texts For Their Automatic Summarisation. PhD thesis, The University of Manchester Institute Of Science and Technology.

ARPA (1991). *Proceedings of the Third Message Understanding Conference (MUC-3)*, San Mateo, CA. Morgan Kaufmann, Los Altos, CA.

ARPA (1992). *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, San Mateo. Morgan Kaufmann, Los Altos, CA.

Austin, J. (1962). *How to do things with words*. Oxford University Press, Oxford.

Baker, C. and Freebody, P. (1989). *Children's First School Books*. Blackwell, Oxford.

Becker, J. D. (1975). The Phrasal Lexicon. In R.Schank and B.L.Nash-Webber, (editors), *Theoretical Issues in Natural Language Processing*. Cambridge, MA.

Brill, E. (1994). Some Advances in Transformation-Based Part of Speech Tagging. In *The Twelfth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, WA.

Califf, M. and Mooney, R. (1997). Relational Learning of Pattern-Match Rules for Information

Extraction. *In Proc. ACL-97 Workshop in Natural Language Learning.*

Chomsky, N. (1957). *Syntactic Structures.* Mouton: The Hague.

Chomsky, N. (1965). *Aspects of the Theory of Syntax.* MIT Press, Cambridge, MA.

Cohen, P. and Perrault, C. R. (1979). Element of a Plan-based Theory of Speech Acts. *Cognitive Science*, 3(3).

Cohen, W. (1995). Learning to Classify English Text with ILP Methods. In Raedt, L. D., (editor), *Advances in Inductive Logic Programming.* IOS Press.

Cowie, J. (1983). Automatic analysis of descriptive texts. In *ACL Proceedings, Conference on Applied Natural Language Processing*, Santa Monica, CA.

Cowie, J. and Lehnert, W. (1996). Information Extraction. *CACM*, 39(1).

Cunningham, H. (1997). Information Extraction - a User Guide. Research memo CS - 97 - 2, Institute for Language, Speech and Hearing (ILASH), and Department of Computer Science, University of Sheffield.

Cunningham, H., Gaizauskas, R., and Wilks, Y. (1995). A General Architecture for Text Engineering (GATE) – a new approach to Language Engineering Research and Development. Research memo CS-95-21, Department of Computer Science, University of Sheffield.

DaSilva, G. and Dwiggins, D. (1980). Towards a Prolog text grammar. *SIGART 72.*

Davison, A. (1975). Speech Acts and what to do with them. In P.Cole and Morgan, J., (editors), *Syntax and semantics*, Volume 3: Speech Acts. Academic Press, NY.

DeJong, G. (1979). Prediction and substantiation: A new approach to Natural Language Processing. *Cognitive Science*, 3.

DeJong, G. (1982). An overview of the FRUMP system. In Lehnert, W. and Ringle, M., (editors), *Strategies for Natural Language Processing.*, Erlbaum, Hillsdale, NJ.

Evan, D. and Zhai, C. (1996). Noun-Phrase Analysis in Unrestricted Text for Information Retrieval. In *Proceedings of the 34th Annual Meeting of Association for Computational Linguistics*, Santa Cruz, CA.

Flickinger, D., Nerbonne, J., Sag, I., and Wasow, T. (1987). *Toward Evaluation of NLP systems.* Hewlett Packard Laboratories, Palo Alto, CA.

Gaizauskas, R., Humphreys, K., Wakao, T., Cunningham, H., and Wilks, Y. (1995). LaSIE - Description of the Sheffield System Used for MUC-6. In *Proceedings of the 6th Message Un-*

derstanding Conference. Morgan Kaufman, Los Altos, CA.

Gaizauskas, R. and Robertson, A. (1997). Coupling Information Retrieval and Information Extraction: a new text technology for gathering information from the Web. In *Proceedings of RIAO'97 - Computer-Assisted Information.*

Gaizauskas, R. and Wilks, Y. (1998). Information Extraction: Beyond Document Retrieval. *Journal of Documentation*, 54(1).

Gasparotti, P. and Simone, C. (1990). A user defined environment for handling conversations. In Gibbs, S. and Verrijn-Stuart, A. A., (editors), *Multi-User Interfaces and Applications*. Elsevier Science Publishers, North-Holland, Amsterdam.

Gazdar, G. (1979). *Pragmatics: Implicature, Presupposition and Logical form.* Academic Press, New York, NY.

Gazdar, G. and Mellish, C. (1989). *Natural Language Processing in Prolog*. Addison-Wesley, New York, NY.

Goldfarb., C. F. (1990). *The SGML Handbook*. Oxford University Press, Oxford.

Grice, H. (1975). Logic and Conversation. In Cole, P. and Morgan, J., (editors), *Syntax and Semantics*, Volume 3: Speech Acts. Academic Press, New York, NY.

Grosz, B. and Sidner, C. (1986). Attentions, intentions, and the structure of discourse. *Computational Linguistics*, Volume 12

Guthrie, L., Walker, E., and Guthrie, J. (1994). Document Classification by Machine:Theory and Practice. In *Proc. COLING 94: The 15th International Conference on Computational Linguistics*, Volume 2.

Halliday, M. (1989). *Spoken and Written Language, Second Edition.* Oxford University Press, Oxford.

Halliday, M. and Hasan, R. (1976). *Cohesion in English*. Longman, Harlow.

Heeman, P. and Allen, J. (1995). The TRAINS 93 Dialogues. Technical Note 94-2, The University of Rochester, Rochester, NY.

Hinkelman, E. (1989). Linguistic and pragmatic constraints on utterance interpretation. PhD thesis, Computer Science Department, University of Rochester, Rochester, NY.

Hinkelman, E. and Allen, J. (1989). Two constraints of speech act interpretation. *In Proceedings of the Association for Computational Linguistics.*

Huffman, S. (1996). Learning information extraction patterns from examples. In Wermter, S., Riloff, E., and Scheler, G., (editors), *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*. Springer, Berlin.

Jones, B. E. (1994). Can punctuation help parsing? In *Proc. 15th International Conference on Computational Linguistics COLING 94*, Kyoto, Japan.

King, M. (1996). Evaluating Natural Language Processing Systems. *Communications of the ACM*, 39(1).

Krotov, A., Hepple, M., Gaizauskas.R., and Wilks, Y. (1998). Compacting the Penn Treebank grammar. In Proc. *COLING-ACL'98*, Montreal.

Lee, M. and Wilks, Y. (1996). An ascription-based approach to speech acts. In *Proc. COLING-96: The 16th International Conference on Computational Linguistics*, Copenhagen.

Lehnert, W. G. (1982). Plot Units: A Narrative summarisation Strategy. In Lehnert, W. G. and Ringle, M. H., (editors), *Strategies for Natural Language Processing*. Erlbaum, Hillsdale, NJ.

Lehnert, W. G. (1987). Automating the Acquisition of Semantic Preferences. *In Proc. AAAI-87.*

Lehnert, W. (1991). Symbolic/Subsymbolic Sentence Analysis: Exploiting the Best of Two Worlds. In Barnden, J. and Pollack, J., editors, *Advances in Connectionist and Neural Computation Theory*, volume 1. Ablex Publishers, Norwood, NJ.

Levinson, S. (1983). *Pragmatics*. Cambridge University Press, Cambridge.

Lewis, D. and Sparck Jones, K. (1996). Natural Language Processing for Information Retrieval. *CACM*, 39.

Lytinen, S. (1992). Conceptual Dependency and its descendants. *Computers Math. Applic.*, 23(2-5).

Macaulay, M. (1996). Asking to ask: The strategic function of indirect requests for information in interviews. *Journal of Pragmatics*, 6(4).

Malone, T., Grant, K., Turbak, F., Brobst, S., and Cohen, M. (1987). Intelligent Information-Sharing Systems. *CACM*, 30.5.

Marcus, M., Santorini, B., and Marcinkiewicz, M. (1993). Building a large annotated corpus of English: The Penn TreeBank. *Computational Linguistics*, 19(2).

McEnery, T. and Wilson, A. (1996). *Corpus Linguistics*. Edinburgh University Press, Edinburgh.

Mey, J. (1993). *Pragmatics: An Introduction*. Blackwell, Oxford.

Morris, C. (1938). Foundations of the theory of signs. In Neurath, O., Carnap, R., and Morris, C., (editors), *International Encyclopedia of United Science*. University of Chicago Press, Chicago.

Patel, M. (1990). The Summarisation and Categorisation of Electronic Mail Messages. PhD thesis, University of Sheffield.

Poesio, M. and Traum, D. (1997). Representing Conversation Acts in a Unified semantic/Pragmatic Framework. *In Proc. AAAI Fall 1997 Symposium on Communication Action in Humans and Machines*. MIT, Cambridge, MA.

Riesbeck, C. and Schank, R. (1976). Comprehension by computer: Expectation-based analysis of sentences in context. Research Report 78, Yale University, Computer Science Department.

Riloff, E. (1993). Using Cases to Represent Context for Text Classification. In *Proc. Second International Conference of Information and Knowledge Management (CIKM-93)*, ACM Press, New York, NY.

Riloff, E. and Lehnert, W. (1992). Classifying Texts Using Relevancy Signatures. In *Proc. Tenth National Conference on Artificial Intelligence*. AAAI Press / The MIT Press, Cambridge, MA.

Riloff, E. and Lehnert, W. (1994). Information Extraction as Basis for High-Precision Text Classification. *In Proc. ACM Transaction on Information Systems*, 12(3).

Sager, N. (1981). *Natural Language Information Processing: A Computer Grammar of English and its Applications*. Addison-Wesley, Reading, MA.

Salton, G. and Buckley, C. (1988). Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24.

Salton, G. and Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41.

Schank, R. (1972). Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 3.

Schank, R. (1975). *Conceptual Information Processing*. North-Holland, Amsterdam.

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

Schiffrin, D. (1987). *Discourse Markers*. Cambridge University Press, Cambridge.

Searle, J. (1969). *Speech Acts*. Cambridge University Press, Cambridge.

Searle, J. (1975). Indirect Speech Acts. In Cole and Morgan, (editors), *Syntax and Semantics, Volume 3: Speech Acts*. Academic Press, New York, NY.

Searle, J. (1976). The Classification of Illocutionary Acts. *Language in Society*, 5.

Small, S. L. and Reiger, C. (1982). Parsing and comprehending with Word Experts (A Theory and its Realization). In Lehnert, W. G. and Ringle, M. H., (editors), *Strategies for Natural Language Processing*. Erlbaum, Hillsdale, NJ.

Soames, S. (1998). Presupposition. In Gabbay, D. and Guenthner, F., (editors), *Handbook of Philosophical Logic*, Volume IV. Reidel, Amsterdam.

Sparck Jones, K. (1993). What might be in summary? In Knorz, G., Krause, J., and Womser-Hacker, C., (editors), *Information Retrieval 93: Von der Modellierung zur Anwendung*. Universitatsverlag Konstanz.

Sparck Jones, K. (1994). Natural language processing: she needs something old and something new (maybe something borrowed and something blue, too). *Presidential Address, June 1994, Association for Computational Linguistics*.

Stalnaker, R. (1974). Pragmatic presuppositions. In Munitz, M. and Unger, P., (editors), *Semantics and Philosophy*. New York University Press, New York, NY.

Stubbs, M. (1996). *Text and Corpus Analysis*. Blackwell, Oxford.

Sundheim, B. and Chinchor, N. (1993). Survey of the Message Understanding Conferences. In *Proceedings of the DARPA Spoken and Written Language Workshop*.

Traum, D. (1993). Mental State in the TRAINS-92 Dialogue Manger. *Working Notes of the AAAI Spring Symposium Reasoning about Mental States: Formal Theories and Applications*.

Traum, D., Allen, J., Ferguson, G., Heeman, P., Hwang, C., Kato, T., Martin, N., Poesio, M., and Schubert, L. (1994). Integrating Natural Language Understanding and Plan Reasoning in the TRAINS-93 Conversation Systems. *In Proc. AAAI Spring Symposium on Active NLP*.

Ure, J. (1971). Lexical density and register differentiation. In Perren, G. and Trim, J., (editors), *Applications of Linguistics*. Cambridge University Press, London.

Wilks, Y. (1964). Text searching with templates. Technical Report ML 162, Cambridge Language Research Unit.

Wilks, Y. (1973). An artificial intelligence approach to machine translation. In Schank, R. and Colby, K., (editors), *Computer Models of Thought and Language*. W. H. Freeman and Com-

pany, San Francisco, CA.

Wilks, Y. (1975a). An intelligent analyzer and understander of English. In Grosz, B., Sparck-Jones, K., and Webber, B., (editors), *Reading in Natural Language Processing*. Morgan Kaufmann, Los Altos, CA.

Wilks, Y. (1975b). Preference semantics. In Keenan, E., editor, *Formal semantics of natural language*. Cambridge University Press, Cambridge.

Willett, P. and Ingwersen, P. (1994). An Introduction to Information Retrieval. In *SIGIR '94 17th International Conference on Research and Development in Information Retrieval*, Dublin City University, Dublin.

Wittgenstein, L. (1958). *Philosophical Investigations*. Blackwell, Oxford.

Zarri, G. (1983). Automatic representation of the semantic relationships corresponding to a French surface expression. In *Proc. ACL Conference on Applied Natural Language Processing*, Santa Monica, CA.

Appendix 1

Dialogue: d93-9.1

Number of utterances files: 159

Length of dialogue: 481.391344

Estimated number of turns: 134

utt1 : s: hello <sil> **can I help you**

utt2 : u: okay um

utt3 : **I want to know how long <sil> alright how long does it take to get <sil> one engine and one boxcar <sil> from Elmira to Corning <sil> and also the same <sil> for one engine from Avon to Dansville**

utt4 : s: okay <sil> from um Elmira to Corning it takes two hours

utt5 : u: + okay +

utt6 : s: + and + **from Avon to where did you say**

utt7 : u: to Dansville

utt8 : s: it takes three hours

utt9 : u: okay <sil> um <sil> **how long does it take <sil> to get from Corning to Dansville**

utt10 : s: Corning to Dansville takes <sil> one hour

utt11 : u: okay uh <sil> **only one train can be on the track at one time**

utt12 : s: right exactly

utt13 : u: **even if they're heading both in the same direction**

utt14 : s: um <sil> just as long as there's like <sil> some like <sil> time in <sil> between them

utt15 : u: okay <sil> um <sil> hm

utt16 : uh <sil> **can more than one boxcar be put on <sil> an engine**

utt17 : s: yeah

utt18 : u: + okay +

utt19 : s: + so there + can be at most three <sil> boxcars <sil> on an engine <sil>
+ or like + tanker cars

utt20 : u: + okay +

utt21 : well <sil> first of all I'd like to s- <sil> start off <sil>
by sending <sil> uh the engine E one from Avon to Dansville

utt22 : s: okay E one <sil> from <sil> Avon to Dansville okay

utt23 : u: um hm next uh <sil> hm <sil> okay I'd like to send <sil>
uh **does it take any less time if it's just an engine** <laughter>

utt24 : s: no it still takes three + hours +

utt25 : u: + okay + <sil> + um +

utt26 : s: + yep <sil> they have + to go the same speed

utt27 : u: okay

utt28 : **could I move <sil>**
one engine with two boxcars engine E two from Elmira to Corning

utt29 : s: yeah okay

utt30 : u: and <sil> then one hour later start the next engine from Elmira

utt31 : s: okay <sil> so um sure <sil> so then E three then at one a.m. <sil>
with um how many boxcars

utt32 : u: uh <sil> E three there's only <sil>
there are only + two + boxcars available

utt33 : s: + okay +

utt34 : okay <sil> so then E two <sil> will like take both boxcars + then +

utt35 : u: + right +

utt36 : s: okay

utt37 : u: to Corning

utt38 : s: + okay +

utt39 : u: + + and

utt40 : **how long is i- would i- <sil> how long does it take to fill up <sil>**
two boxcars

utt41 : s: it will take um <sil> one hour

utt42 : u: okay

utt43 : hm

utt44 : okay um **what time <sil> would it be once <sil> the <sil>**

I guess engine E two arrives at Corning with the two boxcars

utt45 : s: it would be + <sil> two +

utt46 : u: + two o'clock +

utt47 : s: yep <sil> it'll be two + o'clock +

utt48 : u: + and by + the time it loads it would be + three o'clock +

utt49 : s: + three o'clock + <sil> right

utt50 : u: okay <sil> um

utt51 : hm

utt52 : **and at three o'clock also the engine E one would be at Dansville correct**

utt53 : s: right

utt54 : u: okay <sil> um <sil> **how long does it take to get from Bath to Avon**

utt55 : s: from Bath to Avon takes <sil> four hours

utt56 : u: and then Avon to Dansville is <sil> two

utt57 : s: Avon to Dansville is three hours

utt58 : u: hm alright that's not gonna work <sil> um

utt59 : let's see here <sil> so the current state is at three o'clock <sil>

there is one <sil> uh engine with two boxcars at Corning

utt60 : s: right

utt61 : u: + and +

utt62 : s: + loaded up +

utt63 : u: loaded + and + <sil> one <sil> engine <sil>

and three boxcars at Dansville <sil> right

utt64 : s: + yep +

utt65 : right

utt66 : u: + okay +

utt67 : s: + with an + engine E one there

utt68 : u: okay <sil> and <sil> so <sil> and there's three boxcars in Dansville

utt69 : s: right

utt70 : u: uh so <brth> um <sil> hm <sil> **can we move the <sil>**

first of all I'd like to send the <sil> engine with thre- the <sil>

uh two boxcars full <sil> from the orange warehouse + to + Bath

utt71 : s: + yep +

utt72 : yep <sil> okay

utt73 : u: hm <sil> **how's this gonna work**

utt74 : **do I get time to think**

utt75 : s: sure

utt76 : u: okay

utt77 : **is it possible to start over**

utt78 : s: um no <sil> + <laughter> +

utt79 : u: + no + <sil> + okay +

utt80 : s: + so + um <sil> so <sil> **what are you trying t(o)- to um do exactly**

utt81 : u: okay <sil> well <sil> um

utt82 : I guess I started off with <sil> the <sil> uh engine <sil>

from E one that was at Avon

utt83 : s: yep

utt84 : u: moving to Dansville + which had + three boxcars + +

utt85 : s: + right +

utt86 : + yep +

utt87 : u: and then I was gonna move that to Corning

utt88 : s: yep <sil> okay

utt89 : u: + and +

utt90 : s: + so + we should be in Corning then with those three boxcars at <sil>

four a.m.

utt91 : u: right

utt92 : s: + okay +

utt93 : u: + the thing is + there's two boxcars available at Bath <sil>

that are sitting + unused +

utt94 : s: + right + <sil> right

utt95 : u: so

utt96 : s: so we so at like um three a.m. <brth>

we have engine E three in Corning <sil>

so it'll get to Bath at five a.m.

utt97 : u: mm-hm

utt98 : s: **so did you want to do that**

utt99 : u: okay <sil> yeah

utt100: s: okay so Corning will be there at three a.m. <sil> and on to <sil>

Bath <sil> so that'll be in Bath at five a.m.

utt101: u: + right +

utt102: s: + good +

utt103: + okay and then we're + gonna pick up those two boxcars

utt104: u: + and +

utt105: right

utt106: s: okay

utt107: u: and <sil> it takes an hour to get back to Corning

utt108: s: it takes an hour <sil> to get from <sil> Bath <sil> + from from + **where**

utt109: u: + right +

utt110: + from Bath + <sil> to Corning

utt111: s: + so f- +

utt112: okay that'll take two more hours <sil>

so we we'd get back into Bath at seven a.m.

utt113: u: right <sil> that doesn't do any good <sil> wait you'd g(et)- <sil>

you'd get <sil> back to Bath <sil> with one of + the +

utt114: s: + oh sorry + sorry <sil> we would get back <sil> to Corning

utt115: u: + right +

utt116: s: + at + seven a.m. wi- with the two boxcars <sil> loaded on <sil>

E three

utt117: u: hm

utt118: okay <sil> uh

utt119: s: so in as far as <sil> the other s- so as + far as engine E one +

utt120: u: + right <sil> there is there + should be one <sil>

there should be one engine <sil> uh <sil> at <sil> Corning right

utt121: s: um <sil> engine E two is there

utt122: u: right

utt123: s: um anyway it's gonna leave for Bath <sil> at <sil> three a.m. <brth>

so it'll get ba- to Bath at five a.m.

utt124: u: **what about the other engine from + <sil> Elmira +**

utt125: s: + the other engine + from <sil> <noise> <sil> the engine that a- <sil>

the other engine from A- <sil> um <sil>

Avon you mean or from + Elmira +

utt126: u: + from Elmira +

utt127: s: okay <sil> that engine wou(ld)- would have left at one a.m. right <brth>

would get to Corning at three a.m. <brth>

would get to Bath at five a.m. pick up the two boxcars and be back in

Corning at seven a.m.

utt128: u: and then it would take another two hours <sil> to get back to Bath <sil>

though

utt129: s: right exactly and y- and we would also ha(ve)- ha(ve)- have to load up

the oranges there + right +

utt130: u: + right + <sil> + so that's no good +

utt131: s: + so <sil> + we would get there at ten a.m.

utt132: u: right <sil> um <sil> well <sil> **at least could we send the uh <sil>**

three boxcars <sil> and the engine from Dansville <sil> to Corning

utt133: s: yeah <sil> okay so that'll get <sil> to <sil>

Dans- so that'll get to Corning at <sil> four a.m.

utt134: u: okay <sil> and <sil> I guess send that <sil>

load that up and then send it to Bath I assume

utt135: s: okay <sil> + so +

utt136: u: + that would be the best at + this point

utt137: s: okay so we would load it up at <sil> so it'll <sil>

take an hour to load it up so at <sil> five a.m.

utt138: we can go <sil> to <sil> Bath <sil> so we would get there at <sil>

+ seven + a.m.

utt139: u: + right +

utt140: okay so that would be a total <sil> five

utt141: s: yep

utt142: u: available <sil> at <sil> by eight a.m.

utt143: s: **right**

utt144: u: correct

utt145: s: yep

utt146: u: okay <sil> uh I would assume that <sil> that was <sil>
that would be the best of the current situation

utt147: s: so um <sil> what are you trying to <sil> do exactly

utt148: u: + okay +

utt149: s: + what's your um + <sil> goal

utt150: u: uh <sil>
okay determine the maximum number of boxcars of oranges that you could
get to Bath <sil> by seven a.m. + tomorrow morning +

utt151: s: + oh okay + <brth> see one other option is is that um <sil>
I'm wondering if we took engine E one <sil>
as oppo- and have it pick up the two boxcars at Bath

utt152: u: yeah <sil> + that's what I thought of afterwords <sil> + but <sil>
I said move + it <inc> Dansville +

utt153: s: + and then go to Corning +

utt154: + yeah but the thing is + is that <sil> that won't work <sil>
because it takes six hours to go from Avon to + Corning +

utt155: u: + right +

utt156: s: and then so yeah I think <sil> five is about the most that we'll get

utt157: u: right

utt158: s: okay good <sil> so we're done

utt159: u: okay