Automatic summarising and the CLASP system

Richard Tucker

PhD thesis, 1999

University of Cambridge Computer Laboratory
New Museums Site
Pembroke Street
Cambridge CB2 3QG
U.K.

Tel +44 1223 334600

ABSTRACT

This dissertation discusses summarisers and summarising in general, and presents a new summarising system, CLASP.

In chapters 1–3, I present a framework for thinking about summarisers in terms of *context factors* and the three stages of *analysis, condensation* and *synthesis*. I look at previous research in automatic summarising and identify four main directions that have been taken. I consider how summarising systems may be and have been evaluated.

CLASP, described in chapters 4–7, takes a new approach based on a shallow semantic representation of the source text as a *predication cohesion graph*. Nodes in the graph are *simple predications* corresponding to events, states and entities mentioned in the text; edges indicate related or similar nodes. Summary content is chosen by selecting some of these predications according to criteria of *importance*, *representativeness* and *cohesiveness*. These criteria are expressed as functions on the nodes of a weighted graph. Summary text is produced either by extracting whole sentences from the source text, or by generating short, indicative *summary phrases* from the selected predications.

CLASP uses linguistic processing but no domain knowledge, and therefore does not restrict the subject matter of the source text. It is intended to deal robustly with complex texts that it cannot analyse completely accurately or in full. Chapter 8 describes experiments in summarising stories from the Wall Street Journal. The results suggest that there may be a benefit in identifying important material in a semantic representation rather than a surface one, but that, despite the robustness of the source representation, inaccuracies in CLASP's linguistic analysis can dramatically affect the readability of its summaries. In chapter 9, I suggest ways in which CLASP could be modified to overcome this and other problems.

ACKNOWLEDGEMENTS

Any reader will notice the debt this dissertation owes to the work of Karen Sparck Jones. What will not be so apparent is the debt I owe her as my supervisor. It was thanks to her that I became interested in summarising and she deserves thanks too for her enthusiasm, encouragement, and her perseverance in asking difficult questions.

James Thomas and Nancy Chang, my erstwhile office-mates at the University of Cambridge Computer Laboratory, provided excellent company and many lively discussions on all areas of natural language processing, as did others in the natural language group there. At SRI Cambridge, Steve Pulman and David Carter gave invaluable advice on the Core Language Engine and how to use it.

Harlequin Limited gave me first a part-time and later a full-time job while I wrote up, and were always encouraging and keen to ensure that working for them didn't jeopardise my research.

Thanks also to the friends who read through drafts in various stages of preparation, and to those who helped by selecting target sentences for the experiments in chapter 8.

And finally thanks to Manish, who kept me sane with repeated assurances that I'd finish it in the end.

A NOTE ON COPYRIGHT

CONTENTS

PREFACE

Automatic summarising potentially involves an enormous variety of natural language processing and artificial intelligence techniques. To produce an appropriate summary of a text, one might suppose, we must not only understand it, but understand also a user or a group of users' particular needs and interests, apply this information to determine what to say in the summary, and then generate the summary text itself. The very great difficulty of defining what we want from a summary and of discovering how to get it, combined with the complexity and low accuracy of much existing linguistic processing, means that although many methods have been proposed and automated to varying degrees, there is no clear consensus on how automatic summarising should be done. The goal of a summarising machine that can do the job of a professional abstracter is still a long way off.

But there is much to be gained from research in automatic summarising, because it is a challenging testing ground for many kinds of NLP and AI technologies, because we may learn something about human language and thought, and because even if we cannot produce summaries comparable to those written by humans, they may still be useful. For example, if a summary, however ungrammatical in its expression or perverse in its content, indicates something of what a text is about, it is potentially useful as an aid to deciding whether to read the text in full.

This thesis divides into two parts. Chapters 1, 2 and 3 are about summaries and summarising in general; the remaining chapters are about my summariser, CLASP, in particular.

Chapter 1 presents a framework for thinking about summarisers. It discusses a number of *context factors* influencing the summarising task, and then presents a *three-stage model* of the summarising process (the stages being *analysis*, *condensation* and *synthesis*). I then describe a number of ways of categorising summarising strategies in terms of the processing carried out at each stage.

Chapter 2 is about previous summarising systems. It is not a comprehensive survey, but identifies the main approaches, and discusses particularly representative or important systems, in terms of the framework of chapter 1. It also asks to what extent 'general-purpose' summarising methods have been found, and whether such a concept is even valid considering the range of possible context factors.

Chapter 3 looks at evaluation. I discuss four broad approaches and their relative merits in terms of the applicability and validity of evaluation, and the cost of carrying it out. Within each of these broad categories there are many potential approaches: I illustrate some of the possibilities with examples from the literature.

The CLASP system applies robust linguistic processing in summarising without relying on any world or domain knowledge. In this sense it is an attempt to bridge a gap, evident in the systems discussed in chapter 2, between robust, generally-applicable methods which use no linguistic knowledge, and more linguistically sophisticated methods which are however much more restricted in their applicability. Chapter 4 gives an overview of the system's goals and design. The following chapters then discuss each stage of CLASP's processing in more detail.

Chapter 5 describes how source text is analysed to construct a *predication cohesion graph*, in which nodes correspond to events and ideas, and edges join similar or related nodes. How sets of these nodes are selected to form the summary representation is the subject of chapter 6. Here there are many possibilities, and many parameters which may be adjusted to modify CLASP's behaviour. Chapter 7 covers CLASP's two methods for producing summary text: extracting sentences of source text, and generating new *summary phrases*.

Chapter 8 describes experiments using CLASP to summarise stories from the Wall Street Journal. These experiments are not large enough in scale nor rigorous enough in methodology to constitute a thorough evaluation, but they do allow a simple comparison with human summaries and with less linguistically complex methods.

Finally, in chapter 9, I present some conclusions on the success or otherwise of CLASP, and on what directions could be pursued in future research.

I SUMMARIES AND SUMMARISERS

What is a summary? And what constitutes a good summary? The answers to these questions are complex, and depend not only on the text to be summarised, but on the reader of the summary and their intended use for it. To give three examples: an academic might want a summary of a research paper in order to decide whether it is of interest (and therefore whether to go and read it in full); a government minister might want a summary of a report as a substitute for the full text (because it is too long or too technical); a librarian might want a summary of a book in order to classify it for cataloguing.

This chapter introduces a framework for thinking about summarising, and many specific concepts that will be used later on. I begin by discussing (section 1.1) the main *context factors* which affect the summarising process (whether manual or automatic), considering three classes: *input factors*, *purpose factors,* and *output factors*. (n.b. I will use italics when introducing technical terms.) The analysis is developed from that of Sparck Jones (1989–1992, 1995, 1999): she introduces the three classes of factors and many of the individual factors discussed here, and I have followed her terminology except where otherwise stated.

In section 1.2, I present a three-stage model of the summarising process, consisting of *analysis*, *condensation* and *synthesis*. This is essentially the same model as that of Endres-Niggemeyer, Hobbs and Sparck Jones (1993); I use it to introduce a number of ways, many of them new, in which summarising strategies may be categorised, according to the kind of analysis and interpretation of source text they carry out, the methods used to decide what is important in the text, and the approaches used to generate actual summary text. The model and these categorisations provide a framework for thinking about summarising which will be useful when, in chapter 2, we look at previous work.

Given the wide range of possible context factors and summarising strategies, we must ask whether it makes sense to talk about summarising in general, whether there are such things as 'basic summaries' for general use, and what we can say about how to produce them. These questions are considered in section 1.3.

1.1 FACTORS AFFECTING SUMMARIES

The different factors affecting summarising we call *context factors*. Sparck Jones (1995, 1999) divides them into *input factors*, *purpose factors* and *output factors*. She does not give an explicit definition of these classes; here I will take input factors to be those concerning the *source text* only, output factors to be those concerning the *summary text* only, and purpose factors to be those concerning the relationship between the source and the summary text. (As noted in section 1.1.2, this definition results in some of Sparck Jones' output factors becoming

purpose factors.) Input and output factors are thus text properties, and not specific to the task of summarising – they could for example be applied to machine translation – whereas purpose factors are task-specific. All context factors, however, will have implications for summarising, whether manual or automatic.

1.1.1 *Input factors*

I assume throughout that we are concerned with summarising individual written texts in a natural language. There are other kinds of source material such as spoken text, descriptions in formal languages, data, and graphical material, but these are outside the scope of this discussion. Even within this restriction there is a great deal of possibility for variation in the source text.

Form: structure, scale, medium, genre, style

Sparck Jones (1999) identifies a class of factors which together characterise the *form* of the source text. They are *structure*, *scale, medium* and *genre*; I will add *style* to the list.

By *structure*, we mean such properties of the text as whether it is continuous or divided into chapters or sections, whether such sections are numbered or titled, and how they organise the text (for example, into 'Objectives', 'Method', 'Results' and 'Conclusions'). This is *large-scale structure*; there is also *small-scale structure* in the use of particular rhetorical or linguistic patterns. *Scale* is simply the length of the text.

Medium indicates the natural language (or sub-language) used, and *genre* the kind of writing the text consists of (for example, descriptive, narrative, critical, humorous) This use of 'genre' is more specific than its informal meaning. A text's genre will reflect its communicative function, which may be to inform, argue or entertain, and may influence its structure, particularly its small-scale structure. I use *style* specifically to mean the choice of linguistic constructions and vocabulary, and thus is distinct from *genre*.

Some combinations of these *form* factors are especially common. Novels, for example, are usually a few hundred pages in length, and divided into chapters; the genre is typically narrative. Scientific papers are considerably shorter but have a richer large-scale structure, being divided into named sections; the genre may be a mixture of descriptive, narrative and critical. Newspaper stories are shorter still, and usually consist of one continuous narrative or descriptive text; they are also often distinctive in style (for example, making particular use of reported speech, proper names, and complex noun phrases). Such 'newspaperese' is probably not sufficiently different from ordinary writing to constitute a different *medium*, but there is some overlap between these factors.

Although such combinations of factors are common, all *form factors* are independent, both theoretically and in practice: for example, there are novels without chapter divisions, humorous scientific papers, and newspaper articles

with numbered sections.

Intended readership

We can categorise texts according to whether they are intended for the general reader and assume no particular knowledge, for the specialist in a field who already possesses technical knowledge in that area, or for someone who, because they work or live in a particular place or for a particular organization, has knowledge about it that is local though not necessarily technical. (Sparck Jones calls this factor *subject type,* but it concerns the reader rather than the subject matter of the source text.) Obviously, a writer will choose a *style*, and hence a vocabulary, appropriate to their intended readership. Most newspaper stories, for example, have a general readership, so although they may use some terminology specific to their subject matter (whether it be football or finance) they avoid excessive use of jargon which would not be comprehensible to the average reader. A football supporters' newsletter, aimed specifically at knowledgeable fans, would include more slang and specialist terminology.

Subject matter

Since a summary must reflect at least some of the source text's subject matter (i.e. what it is about), this is in a sense the most important source factor. Many of the approaches to summarising discussed in chapter 2 are dependent on the subject matter of the source text, because they use knowledge specific to that subject area, both in analysing the text and in deciding what information should be included in the summary.

1.1.2 *Purpose factors and summary function*

Sparck Jones presents three types of purpose factors (*situation*, *audience* and *use*) that describe who will use the summary and what for. Here, I relate these to factors that more directly describe requirements for the summary itself. These factors are *summary type* and *coverage*. They are similar to Sparck Jones' output factors of *material* and *style*; however, because they are concerned with the relationship between the source and the summary, I classify them as purpose factors according to the definition in section 1.1.

Situation, audience and use

Situation describes in what context the summary will be used, *audience* describes who will use it, and *use* describes what they will use it for. The examples given in the first paragraph of this chapter illustrate three different possible situations and audiences for summaries, and also three different uses: these are *retrieving* (the researcher who uses a summary to spot a potentially important paper), *substituting* (the government minister who is too busy to read the original documents) and *categorizing* (the librarian who must catalogue a book). Sparck Jones suggests further possible summary uses such as *prompting*, *previewing* and *refreshing*.

Summary type

We can broadly categorise summaries according to the type of information they contain. An *indicative* summary identifies the main topics of the source text. An *informative* summary conveys the important information in the source text. A *descriptive* summary describes the source text as a textual object (and hence may describe its form as well as its content). An *evaluative* summary responds critically to the source text's argument, agreeing or disagreeing with it.

Looking again at the three examples: the researcher requires an *indicative* summary to retrieve relevant papers; the minister requires an *informative* summary to substitute for the original document; the librarian requires at least an *indicative* summary (for subject classification) and perhaps also one that is *descriptive* (for example, to classify the source as fiction or non-fiction). For most of the summary uses mentioned above, either an *indicative* or an *informative* summary will usually be appropriate: as discussed later in section 1.3, and as we will see in chapter 2, most work into automatic summarising has been aimed at producing these types of summaries.

Coverage

A second way to classify summary functions is by the *coverage* of the source text which they require. Here there are two dimensions: *scope* and *depth*. As far as scope is concerned, the summary may be focused on a particular topic (*narrow* scope), or it may be representative of the content of the source text as a whole (*broad* scope). In the narrow scope case, the topic to focus on may be determined by the audience and the situation as well as the source text. In terms of depth, the summary may need to give detailed information on the areas it covers, or just to provide a rough outline. When the need is for a highly focused summary which gives detailed information (i.e. *narrow* and *deep*), as in the MUC (Merchant 1993) tasks, for example, this is perhaps better described as 'information extraction' (or in some cases 'text extraction') than summarising.

One might object to the idea of a very detailed and broad summary, on the grounds that if it gives all the detail in the source text it can hardly be called a summary at all. But this is only to consider half of the idea of summarising, namely *shortening of content*. The other half is *shortening of expression*. A text can be summarised by reducing its content (i.e. selecting only certain material from the text for the summary), or shortening its expression (i.e. conveying all the information in the text, only more concisely), or, as is most likely in handwritten summaries, a combination of both.

1.1.3 *Output factors*

The purpose factors and summary types discussed in the previous section describe how a summary relates to its source text. There may be additional textual requirements for the summary itself, arising from the purpose factors: these are output factors. Since these factors are text properties, they are

essentially the same as the input factors discussed in section 1.1.1.

Form factors

There may be output requirements corresponding to the input factors of *scale* and *genre*: presumably the summary will be shorter than the source text, but it could still be from a single sentence to several pages in length. Since, as discussed earlier, the genre of a text reflects its communicative aims, the genre of a summary should be appropriate for its summary type, and, for an *informative* summary, it may depend on the genre of the source text. These factors, however, do not determine the *structure* of the summary; it could, for example, be presented under topic headings (e.g. for a scientific paper: 'significant results' and 'applications'), as continuous running text, or as a bulleted list of topics.

The *medium* of the summary will ordinarily be the same as the medium of the source text (unless we combine summarising with machine translation), except where the source or summary text are in a particular sub-language (telegraphese, for example). The *style* of the summary depends not just on the summary type and on the style of the source text, but on the intended readership of the summary, i.e. the *audience*. This need not be the same as the intended readership of the source text, and therefore the language used may have to be different. For example, a summary of a highly technical paper for the general reader may have to use less jargon than the original text.

Subject matter

The *subject matter* of the summary will depend on the subject matter of the source text, and on the summary type and coverage. In an informative summary, the subject matter is the subject matter of the source (or a part of it, if the summary has narrow scope).

In the case of a descriptive, evaluative or indicative summary, the subject matter of the summary is primarily the source text itself, and secondarily the subject matter of the source text (or a part of it) – these types of summary explicitly say something *about* the source text, whereas an informative summary does not.

It is important to note that subject matter is a broader notion than content, and that even when the subject matter of a summary is the same as the subject matter of the source text, the summary may still include information not explicitly stated in the source text, but rather deduced from it or from a world-model (as for example in summaries produced by the script-based summarising systems discussed in section 2.3).

1.2 A THREE-STAGE SUMMARISING MODEL

Considering the very wide range of possible context factors, and the variation in methods of computational analysis of natural language text, we must expect there to be a great many different possible methods for automatic summarising. They must however all have something in common: namely that they *analyse*

the source text in some way so as to represent it in the computer, perform some kind of *condensation* upon this representation to obtain a representation of a summary, and *synthesise* from this summary representation the summary text itself.

These three stages of *analysis*, *condensation* and *synthesis* (Endres-Niggemeyer, Hobbs and Sparck Jones 1993) form the model of summarising I will use here. Sparck Jones (1995) uses a two-stage model, in which the second stage, *generation*, contains both the second and third stages of the three-stage model. I prefer to keep condensation as a separate stage because functionally it serves a different purpose from synthesis, and because the techniques employed for these two stages may be quite different.

This model does not imply that every summarising method must involve three separate, sequential phases of processing. Instead the distinctions between stages are conceptual, concerned with the function of the processing and the representations it involves. The value of the model is that it allows us to compare and classify different summarising systems, and to see what approaches have yet to be explored. It is perfectly possible for a program to further divide these stages or to carry out two (or perhaps even all three) of them simultaneously. For example, methods which attempt to summarise text as they 'read' it (like humans) must necessarily begin condensation before analysis has been completed. This is the case, for example, in Kintsch and van Dijk's (1978) model of text understanding, which includes a kind of summarising as an essential part of reading and comprehension. However, my summarising system, CLASP (introduced in chapter 4), does indeed implement the three stages as three almost entirely separate steps of processing.

1.2.1 *Analysis and the source representation*

We can categorise different kinds of source representation (i.e. the representation produced by the analysis stage) by their *depth*, their *granularity* and their *large-scale structure*. The depth categorisation is obvious, while Sparck Jones (1995) discusses many aspects of large-scale structure.

Depth

Depth, i.e. the linguistic level of the representation, is important because it determines what level of information the condensation stage of the summariser will be able to consider, how far the representation is removed from the surface form of the text, and how near it is to representing the text's 'meaning' (whether we mean this in a logical or a communicative sense).

An extremely shallow source representation could consist simply of surface sentences, the analysis stage having done nothing except identify sentence boundaries. A little deeper, and the analysis might use shallow parsing to identify compound terms or phrases; deeper still, it may establish a semantic representation (e.g. using predicate logic or semantic nets) for each sentence of the source text. A yet deeper analysis might escape completely from the division

of the source text into sentences and represent its overall meaning, perhaps using a frame- or script-based format, as used in SAM (Schank and Abelson 1977, Cullingford 1981).

Usually, constructing a deep representation requires advanced processing, and constructing a shallow representation requires only simple processing; occasionally, however, this is not so. For example, FRUMP (DeJong 1982) uses a very deep script-based source representation, yet constructs it using relatively simple linguistic processing, because it knows exactly what kinds of information to look for in the text. On the other hand, one can imagine a system in which the source representation consisted of the source text with anaphors and ellipsis resolved: such a representation would be shallow, yet highly sophisticated processing would be needed to construct it.

Granularity

Largely independent of depth, we can consider the *granularity* of the source representation – that is, the size of its *basic units*. The importance of granularity lies not in how the representation is built, but in how it may later be used: thus, by basic units I mean the smallest units of representation which, in the condensation stage, we will be able to deem important or unimportant, to link, sever or rearrange to construct a summary representation. For example, we might represent a text as a set of sentences (or more accurately as a set of sentence representations) arranged in a tree-structure that is deemed to model the discourse structure of the text. In such a representation, the sentence representations could be logical forms, sets of stem words, or simple surface text. But if, in each case, the representation of a sentence cannot further be subdivided into smaller units, then the granularity of the representation is the same.

As we will see in section 1.2.2, condensation is bound to involve an element of *selection*, and may also involve *generalisation*. In *selection*, some content present in or derived or inferred from the source representation is chosen as important. If the source representation has no basic unit smaller than a sentence, then in condensation we must either decide that a whole sentence is important, or that it is unimportant. Granularity therefore determines with what precision the condensation stage will be able to identify important content in the source representation.

Four kinds of large-scale structure

Summarising a text is not just a matter of summarising each sentence or paragraph individually (although, as we will see in the next chapter, approaches which treat each sentence in isolation have been surprisingly successful), so we may want to represent the overall structure of the text as well as its basic units.

A text has several different kinds of large-scale structure: Grosz and Sidner (1986) distinguish *linguistic*, *intentional* and *attentional* structure as three components of discourse structure which are related but distinct. I will use their categories, but not just to refer to their particular theories. *Linguistic structure* is

concerned with relations between the meanings of sentences of a text, the way in which they are grouped into larger-scale segments and how these segments combine to make the whole discourse. Mann and Thompson's (1987) *Rhetorical Structure Theory* (RST) is primarily an example of a theory of linguistic structure. To determine such structure automatically we might investigate relations between the logical meanings of sentences, using some kind of inference engine and perhaps some world or domain knowledge, or look for cue words and phrases which signal particular rhetorical relations, as for example in the work of Marcu (1997).

Intentional structure is concerned with the intentions behind utterances or sentences, and how these intentions relate to one another (in Grosz and Sidner's theory, for example, there are relations such as 'dominance' and 'contribution'). A full treatment of intentional structure would presumably involve sophisticated modelling of speech acts and their consequences in terms of beliefs and intentions, perhaps coupled with planning and plan-recognition. Alternatively, we could posit a closer link between linguistic structure and intentional structure: we might then end up with something resembling RST's intentional component.

Attentional structure is to do with what the text is about; it is thus a property of the text, not the writer or reader. (This is a broader view than that of Grosz and Sidner). Sidner's (1983) theory of focus and Janos' (1979) theme-rheme analysis are examples of theories of attentional structure. Grosz and Sidner regard attentional structure as dynamic: as the text is read, salient entities and properties are pushed and popped from a focus stack. This dynamic theory reflects their opinion that attentional structure is not something to be represented itself, but is primarily an aid to sentence analysis (specifically, in resolving anaphors) and in building a representation of intentional structure. However, I see no reason not to use a static representation of attentional structure directly for summarising purposes; in a sense, to distinguish salient objects and information from less salient ones is already to have begun to summarise. In fact, I will argue in chapter 2 that several past summarising systems have operated on representations of attentional structure, as does the CLASP system introduced in chapter 4.

Discourse structure is a very general term: I will use it broadly to refer to all inter-sentential linguistic, intentional and attentional information. In addition to these, however, there is a fourth kind of structure that we can identify: the underlying structure of the information conveyed by the source text. Source representations that attempt to capture the meaning of the text – for example, by using conceptual dependency, frames or scripts that describe properties, events and objects in the world (or rather, the discourse world) – may not be concerned with the rhetorical, communicative or thematic devices by which information is conveyed, but only with the information itself. This fourth option I will call *informational structure*.

Of course the boundaries between these different kinds of structure are not always clear: communicative intentions are often closely paralleled in the

rhetorical structure of a text, and in a straightforward narrative the linguistic sequence corresponds to the underlying temporal sequence of events. Nevertheless, the logical distinctions between them will help in classifying the summarising methods described in chapter 2. The type of structure in the source representation of a summariser will inevitably affect the resulting summaries, as it determines what kinds of information (linguistic, intentional, attentional or informational) can be used in the condensation stage. An experiment comparing some of the possibilities (Sparck Jones 1995) is discussed in the next chapter.

Tree-like and graph-like structure

Independent of the substantive kind of structure used in the source representation, we can make a distinction between different *forms* of structure. There are two main possibilities: *tree-like* structures and *graph-like* structures. Tree-like structures are ones in which related units combine to form a hierarchy of larger units. This kind of structure is used, for example, in Rhetorical Structure Theory (Mann and Thompson 1987), in script-based analysis (e.g. Schank and Abelson 1977), and in Kintsch and van Dijk's (1978) theory of macropropositions.

Graph-like structures are ones in which there are relations between basic units, but no hierarchy of larger-scale units. Skorochod'ko's (1971) intersentential relations are an example of a graph-based text representation. Tree- and graph-based representations tend, as we will see in the next chapter, to lead to different kinds of condensation methods.

Hobbs' (1990) theory of coherence is an interesting case: he defines a number of possible relations between sentences of source text, together with conditions under which they will hold. Considering the very general nature of some of the relations, one would expect a full analysis of the coherence relations in a text to give a complex graph-like structure. However, in his examples, Hobbs identifies a tree-like structure of relations in the text, combining related sentences into higher-level nodes by applying a notion of headedness.

1.2.2 *Producing a summary representation*

Selection and generalisation

The condensation stage of a summariser must take the source representation and produce a summary representation from it. There is no requirement that these representations be of the same type, although this will often be the case. We can identify two main processes by which a summary representation can be formed: *selection* and *generalisation*. I define these as follows: *selection* is choosing parts of the source representation which are judged to be appropriate for inclusion in a summary; *generalisation* is combining information from more than one part of the source representation to form new information for the summary.

For example, suppose we are to summarise a text containing the sen-

tences: 'John baked a delicious fruit cake.' and 'John baked a lemon sponge.' Using selection only, we could say that John baked either or both of these cakes. However, to say 'John baked cakes' requires us to combine the information in these two sentences; this is generalisation. This example is very simple, but to make generalisations computationally is hard, requiring some kind of inference and world or domain knowledge (e.g. that 'lemon sponge' is a kind of 'cake'); therefore many summarising systems form their summary representations by selection alone. Now suppose that, in the above example, we said in the summary: 'John baked a fruit cake.' This is a more general statement than is given in the source text, as 'a fruit cake' is less specific than 'a delicious fruit cake'. However, we can regard this as an instance of selection operating on units smaller than an individual sentence: a single source sentence has been analysed to give two facts (that John baked a fruit cake, and that it was delicious) and only one of them has been selected for the summary.

Prescriptive and responsive summarising

Condensation methods can vary from being *prescriptive* to *responsive*. By a prescriptive summariser, I mean one which assumes that certain local features of the source representation indicate important or unimportant information directly. By a responsive summariser, I mean one which makes decisions about importance on the basis of the source representation as a whole. This distinction applies regardless of the type of information (linguistic, intentional, attentional or informational) captured in the source representation, as the following examples illustrate.

At one extreme, a prescriptive summariser using informational or attentional content might suppose that information about presidents, bombs and floods is intrinsically important, while information about bell-ringing, cheese and clothing is intrinsically unimportant. A prescriptive use of linguistic surface features would be to assume that parts of the source containing the word 'I' were important. A prescriptive use of linguistic structure would be to assume that, in general, a statement that justifies another statement is not important, but a statement that contrasts with another statement is important. Prescriptive condensation from a representation of intentional structure might assume that a discourse segment whose purpose is dominated by that of another is unimportant, whereas the remaining part of the dominating segment is important.

A responsive use of an informational representation would be not to insist that a particular topic was intrinsically important, but to consider the relations between topics in the source representation and look for topics which were in some sense central to the text. A responsive use of an attentional representation might look at which discourse entities and themes were salient at the most points in the text, and select these for a summary. Responsive summarising from an intentional or linguistic representation might start from a tree-structured analysis of text-structure and deem tree nodes nearer to the root to be more important than leaf nodes, regardless of what particular communicative functions they had or in what rhetorical relations they stood to the rest of

the text.

The difference between prescriptive and responsive summarising is closely related to whether the summariser looks at local or global information in the text. An extremely prescriptive system could select or reject whole sentences of source text independent of the rest of the text, using information which is entirely local within the text, (although of course it could use other information, such as a table of cue phrases). Such systems are discussed in section 2.2. In contrast, in a responsive system, whether a particular part of the source text is judged to be important can depend on other parts of the text, perhaps far removed from it.

1.2.3 *Generating summary text*

The synthesis stage may involve a number of distinct processes, from initial planning to eventual selection of surface text. Whatever the details of these steps are, we can make a broad distinction between two extreme kinds of synthesis: *rigid* and *flexible*.

Rigid synthesis consists of fitting information into a prescribed output template, whether at the level of surface text, or of deeper representations from which text can be generated. The output template need not be the same as any template (e.g. script or frame) that may have been used in producing the source or summary representation. Rigid synthesis might be appropriate for some summary functions (for example, if the reader of the summary wants it to contain certain information in particular) but for others it may be rather restrictive, producing summaries that reflect the internal template more than they reflect the source text.

In flexible synthesis the summariser decides how to order and present the content of the summary representation on the basis of what that content is, and perhaps also on how it is presented in the source text. For example, sentence extraction is a moderately flexible and easily achievable kind of synthesis; for true flexibility, however, we must turn to complex methods such as generating rhetorical structures or text planning, which are not so readily implemented.

1.3 BASIC OR ALL-PURPOSE SUMMARIES

The enormous possible variations in context factors and in the summarising strategies that we might employ to meet those factors (combined with the fact that we simply do not know how to do any kind of automatic summarising properly) make the summarising task a daunting one. A further difficulty is that, when a summary is produced, the precise context factors under which it will be used may not be known.

We must ask, then, if we can reasonably speak of a *basic summary* – a summary which is likely to be useful for a variety of purposes, which is not determined by *a priori* views, slanted towards particular information or aimed exclusively at a particular audience. Such summaries would be useful in

themselves, and we would hope that any techniques developed to produce them might be adapted to suit a wider range of summarising needs than a task-specific summariser could manage.

The answer to this question is a partial 'yes'. Sparck Jones (1989–1992) notes that while a true general-purpose summary is impossible, 'we can … think of a "basic" summary as implying a much closer, direct or straightforward, reflective relationship between source and summary text.' The key concept here is that the summary is *reflective.* By this, I mean that it has the same intended readership as the source text, it preserves something of the style of the source text, and it is representative of the source text as a whole, i.e. has wide scope. This kind of summary is *neutral* in that it is no more biased towards any particular subject or reader than the source text is, and does not impose an opinion on the text. It seems that what many people ordinarily think of as a 'summary' is close to a reflective summary that is also informative, and that shortens both content and expression – Sparck Jones notes that authors' abstracts of their own papers typically fit this description. In addition, many summarising systems have been designed, explicitly or implicitly, to produce such summaries. The CLASP summariser described in chapters 4–7 produces summaries which are reflective, but *indicative* rather than *informative*. This kind of summary is also likely to be useful for a variety of purposes, but is easier to produce automatically than an informative summary.

Basic texts

If we are to develop summarising techniques especially suited to a wide range of summary functions, then we must also make sure that they apply to a variety of source texts. Obviously some restriction is needed here too, as summarising a recipe is very different from summarising a novel. What we need therefore is an idea of a *basic text*. In some respects there is no such thing: of course there is no 'typical' length for a source text – it depends on the kind of text in question. Source texts will also vary in style, structure, intended readership and language, but here we can reasonably be more restrictive. I take a basic (English) text to be one intended for the general reader (hence requiring no specific prior or specialist knowledge), written in standard non-technical English, presented as continuous running text (perhaps in paragraphs but without headings or subheadings, bulleted lists or tables), and consisting primarily of factual material. The primary communicative purpose of a basic text is to convey information to the reader. This informal definition allows as basic texts many newspaper stories and magazine articles (perhaps excluding sports or business pages, and specialist magazines), and some books and papers (perhaps after removing headings). Within this definition, a great deal of variation is possible, both in the content and in the form of the text.

Much research in automatic summarising has already focused on making reflective summaries of basic texts such as these, and CLASP does likewise. As will be seen in the next chapter, however, some existing systems depend, to a

greater or lesser extent, on properties of the source text's style, form or subject matter to summarise it.

In making these definitions, we are supposing that, for certain kinds of variation in context factors, the intrinsic method for automatic summarising need not be changed (though of course parameters may need adjustment). The hope is that the task of producing a reflective summary of a basic text (which perhaps we can call *basic summarising*) is sufficiently specific that it can practically be attempted, and sufficiently general that it can usefully be applied. There is a wider question of just how much variation in factors a single method will allow: we must ask whether different tasks require of necessity different summarising strategies, and further, for which tasks different strategies, though not required, would be appropriate. We will see in the next chapter that, so far at least, whereas humans are able to cope with a wide variety of source texts and summarising tasks, automatic summarisers have been much more limited in their applicability.

2 PREVIOUS SUMMARISING SYSTEMS & METHODS

This chapter describes the main approaches taken in automatic summarising from the 1950s to the present (1998), using the theoretical framework of chapter 1. It is not an exhaustive survey, and where there are many similar methods I have not discussed them all. The systems described here were chosen either as typical instances of classes of systems or methods, or because they represent a notable advance on previous work, or because they take an interesting and individual approach. I have shown a preference for implemented systems over theoretical proposals (although it is not always easy to ascertain to what extent some of the summarisers were able to function without human assistance), and for methods which are, or might be made to be, applicable to a wide range of tasks over very task-specific methods. Where evaluation of systems has been carried out, I have given a brief description of the method and notable conclusions; chapter 3 gives a fuller description of evaluation in theory and practice. At the end of this chapter, I present some brief conclusions which motivate the new system, CLASP, presented in chapters 4–7.

As discussed in chapter 1, the word 'summarising' can cover a wide range of tasks; I have chosen not to include tasks more accurately described as text or information extraction (also known as 'message understanding'), where the problem is not to determine what is important in a text, but to find particular and pre-specified kinds of information in a text, whether they are important to it or not. Further, I am only considering summarising of individual (natural language) texts, rather than the production of ongoing reports.

Because summarising must ultimately require advanced natural language processing, many promising approaches cannot be automated with current technology. I distinguish between *systems*, which have been implemented, and *methods*, which may have been only partially implemented, or not at all. By *approaches*, I mean more general classes of methods, within which there are many possibilities.

Four main directions

I consider there to have been four main directions taken in previous work: summarising from *attentional networks* (section 2.1), summarising *sentence by sentence* (section 2.2), summarising from *informational content* (section 2.3), and summarising from *discourse structure* (section 2.4). The first two of these have tended to involve very shallow source representations (commonly the surface-text itself or something very close to it). The third and fourth approaches suggest the use of deeper representations, although, as will be seen, they may not necessarily require very sophisticated linguistic processing. Of course there is a continuum from shallow to deep, but it is striking that many systems use representations at one of the two extremes.

There are a number of useful review articles which cover many of the

techniques and systems mentioned here, among others. Paice (1990) in particular is a useful guide. There are also a few comparative studies (Edmundson 1969; Gladwin, Pulman and Sparck Jones 1991; Sparck Jones 1995): these I discuss in the appropriate sections below.

2.1 SUMMARISING FROM ATTENTIONAL NETWORKS

The central idea of all the approaches in this section is to ascertain what a text is 'about' by identifying concepts that are in some sense central to the text, on the basis of the occurrence of the same or related concepts in different parts of the source representation. By an *attentional network*, I mean a representation of the links in *aboutness* between different parts of the source. The idea of aboutness covers many possibilities: for example, we might take each content word in a sentence as representing something that sentence is about; alternatively, we could apply an attentional theory such as Sidner's (1983) theory of *focus*, to determine the main topic or topics of individual statements.

Although it is not always stated explicitly, the assumption behind this kind of summarising is that what is wanted is an indicative or informative summary, that reflects the subject matter of the source text. These methods are not highly specific to any particular intended readership or subject matter of source texts, but applicable to a wide variety of descriptive or narrative texts: they are attempts at producing *reflective summaries* of *basic texts*, as defined in section 1.3

In section 2.1.1, I discuss approaches in which summarising is performed on the basis of the frequency with which concepts occur in the representation. Such summarisers do not represent attentional networks directly, but can be thought of as doing so implicitly. This view leads naturally to the work described in section 2.1.2, where the graph of attentional links is explicitly represented.

2.1.1 *Frequency-based summarising*

Luhn

In 1958, Luhn published a paper on 'The automatic creation of literature abstracts'; his were the first experiments in anything which might be called automatic summarising. His program operated in two stages: first, it identified words (or more correctly, word stems) which it judged to be good discriminators between pertinent and non-pertinent information in the text; second, it computed for each sentence a score based on the number of these significant words it contained, and their proximity to each other. The 'auto-abstract' then consisted of all sentences with scores above a pre-defined cutoff point.

Luhn's central idea was that, up to a point, the more frequently a word occurs the more likely it is to indicate important material in the text. Beyond some point, the word's value begins to decrease – it occurs too frequently in the text to be of use in distinguishing one sentence from another; such words are

commonly function words which do not convey important meaning. Luhn proposed either to establish a cut-off frequency above which words would not be considered, or to have a list of function words (a *stop-list* in information retrieval terminology) which would be ignored.

Because Luhn's system used no knowledge about the type of text or its subject matter it was suitable for a wide variety of texts, and as it performed no linguistic analysis (except for a very basic kind of stemming) it was also very robust and practical. However, the summaries that resulted were sometimes simply not very informative or indicative, and they were often less than coherent. In addition, his system had a bias toward selecting long sentences, and so the extracts were sometimes wordy and convoluted. Luhn was well aware of these limitations, but suggested that with a knowledge of how the summaries were produced, users might overcome such difficulties: 'Once auto-abstracts are generally available, their users will learn how to interpret them and how to detect their implications'.

In terms of the 3-stage model, Luhn's analysis consisted simply of separating the text into sentences and counting word frequencies to compile a list of significant content words. The source representation was simply the original text plus the frequencies of significant words. The condensation stage then selected a number of sentences based on the occurrence of these words. The synthesis stage simply output the selected sentences.

Preston and Williams (NetSumm)

Since Luhn's early experiments, many researchers have tried similar methods, attempting to improve either on the relevance of the sentences extracted or the readability of the final summary text. A recent system operating on broadly similar lines is BT's NetSumm (Preston and Williams 1994), a sentence-extraction summarising system which uses word occurrence to select important sentences. Technical details of NetSumm's operation have apparently not been published (presumably for commercial reasons); my own experiments with the system suggest that it considers only the frequency with which words occur, ignoring sentence order and the order of words within sentences. Preston and Williams claim that 'an abridgement of typically only 5% of [an] original article will contain roughly 70% of the information in an author-written abstract, while a 25% abridgement contains essentially all of the information in the abstract' – without more information on how this evaluation was carried out, it is difficult to comment on the effectiveness of NetSumm's strategies. However, this system is available on the world wide web, and I report some comparisons between it and my own summariser in chapter 8.

NetSumm incorporates a number of interactive features designed to make the summaries more comprehensible to the user. If the user feels the summary has too much or too little detail, they can ask for a shorter or longer alternative; additionally, they may view the whole source text, the summary only, or the whole text with the selected sentences highlighted, which allows for quick skimming of the text while avoiding misleading implications in the extract.

Brandow, Mitze and Rau (ANES)

The ANES summariser (Brandow, Mitze and Rau 1995) is a recent system which applies a lesson from the field of information retrieval, namely the importance of corpus statistics, to Luhn's basic idea. ANES considers not only word frequencies within the document to be summarised, but word frequencies within a large collection of texts (for which Luhn's system had no equivalent). The intuition is that some words are typically more common than others, and the summariser should compensate for this when determining how significant frequent words are in the source text; this argument, however, seems much weaker than the corresponding one for information retrieval, where the task explicitly involves a corpus of documents.

In ANES, each sentence is assigned a weight which is the sum of the weights of *signature words* in that sentence. Signature words are words which score highly on a weighted frequency measure, which compares the frequency of the word or term in the document to be summarised to its frequency in the corpus, according to the following formula:

$$\frac{\log \frac{\textit{total words in corpus}}{\textit{occurrences of term in corpus}}}{\log \frac{\textit{total words in document}}{\textit{occurrences of term in document}}}$$

The more frequent the word in the document, and the less frequent in the entire corpus, the higher the weight given to it by this formula. There is no real justification for the use of this particular formula rather than many others with roughly similar behaviour, other than that this kind of weighting function has been very successful in the field of information retrieval, where it is known as a term frequency times inverse document frequency type formula, (*tf·idf* for short).

Which sentences are selected to form the summary then depends primarily on the sentence weights, and secondarily on their location (as suggested by Baxendale (1958)) and whether they appear to contain references to other sentences. Extra sentences may be added in an attempt to improve readability: for example, if two sentences have been selected which are separated in the text by a single sentence, this intervening sentence will be added irrespective of its weight. The system also performs *aggregation*, as described for the ADAM system in section 2.2.

Summaries produced by ANES were evaluated by readers who classified each summary as acceptable or unacceptable by comparing it to the source text. For comparison, Brandow, Mitze and Rau also evaluated summaries produced by taking an initial segment of the text up to the required length. Their remarkable conclusion was that the initial segment summaries outperformed the ANES summaries, being acceptable 92% of the time as opposed to 75% of the time (these are overall figures, considering summaries of 60, 150 and 200

words). They concluded that this was at least partly because of the particular kind of source texts used, namely news stories, which often begin with a concise statement of the main points of the text.

Aone et al (*DimSum*)

The DimSum summarising system (Aone et al 1997) combines *tf·idf*-type formulae with other kinds of corpus-based statistical natural language processing, specifically the identification of phrases and proper-names, and noun-noun collocations. *tf·idf* scores are computed not just for individual words, but for word senses (disambiguated using collocational information), synonym sets (produced with the lexical database WordNet (Miller et al 1990)), and short phrases. At present the various scores for a sentence are simply averages of some or all of these scores, but it is intended that eventually the system be trainable 'to different user and application needs'. However, whether the kinds of information available to the summariser are sufficiently varied to allow for summaries to be targeted to a very wide range of uses and users, is doubtful, considering that the system is entirely frequency-based, and synthesis proceeds entirely by sentence-extraction. A sophisticated summary browser allows users to view the source text with important names and keywords, as well as the selected summary text, highlighted. DimSum has so far been evaluated by comparing its summaries to manually-produced target extracts; despite a low-level of similarity overall, there were some interesting results. Notably, the identification of proper-names was found to be useful in that there was a beneficial effect when these names were *removed* from the terms to be considered. The authors suggest that this is because most proper names have very low occurrence frequencies in the corpus, and therefore high *idf* values, although the rare proper names in their source material were mostly the names of unimportant spokesmen.

Hovy and Lin (SUMMARIST)

The SUMMARIST system (Hovy and Lin 1997) is intended to include a variety of summarising modules, all based on the idea of first extracting individual salient concepts from a text, then generalising them to obtain higher-level unifying concepts. At present, the 'concepts' extracted are words, with weights assigned according to a *tf·idf*-type score, and according to their location in the text, and the system uses WordNet to supply a lexical hierarchy used in generalising concepts. SUMMARIST looks for 'concepts that each generalise a set of approximately equally strongly represented subconcepts'; when, instead, one particular subconcept dominates, it is retained, and the generalisation not made.

Future plans for SUMMARIST include the application of domain knowledge, in the form of *concept signatures*. These are sets of concepts pertaining to a particular overall topic, such as aerospace, banking, or the environment (the topic classification is one used by the Wall Street Journal); each signature will also have a *head concept* describing the topic. Rather than using only WordNet to perform concept generalisation, SUMMARIST will

identify concept signatures corresponding to sets of concepts, and generalise to the head concepts.

SUMMARIST at present produces summaries by sentence extraction; however, it is planned that eventually it will produce summaries by extracting simple phrases and clauses from the source text and concatenating them into sentences, and even by full-sentence planning and generation. But it is unclear how such advanced synthesis can fit into the primarily surface-based approach SUMMARIST has taken so far; a much more sophisticated analysis stage will presumably be needed.

Gladwin, Pulman and Sparck Jones

The systems so far described take every term (whether word, stem or phrase) in a sentence as something that that sentence is 'about'. Gladwin, Pulman and Sparck Jones (1991) suggest a simple way to use a deeper notion of aboutness. Their method involves applying Sidner's (1983) focus tracking algorithm to the source text, and counting the frequency with which each discourse entity is in focus. Although this analysis is fairly shallow, it is not easy to do automatically, and in this study was carried out by hand. The results, presented as lists of salient entities (a plausible first stage of summarising), look reasonable, but are not very different from those obtained with two other methods tried in the same study: simply counting the occurrence frequency of nouns, and counting the frequency with which entities occur as arguments of predications in the text.

Caldwell, Dersy

Caldwell (1994) and Dersy (1996) both implemented something similar to the predication-argument counting strategy just mentioned, at a shallow syntactic level only, by stemming and tagging source text, and using regular-expressions to try to find noun and verb groups. Caldwell's program identified salient entities (i.e. stems), whereas Dersy's selected both individual stems and pairs of stems that frequently were related in the text. The shallow processing used in these systems was inspired by the success of finite-state machines in MUC tasks.

Boguraev and Kennedy

The summarising system of Boguraev and Kennedy (1997) extracts phrasal expressions from the source text, and tracks their *local salience* on a sentence-by-sentence basis. Local salience of an expression is determined by weighted *salience factors*: primarily these are syntactic (e.g. whether the expression occurs as the subject of a sentence), but they also take account of how recently the expression has appeared, so that the local salience of an expression drops off once it is no longer used. Local salience is used to perform anaphor resolution, leading to the concept of *discourse referents*, represented as equivalence classes of expressions, and whose local salience depends on the salience factors of all the corresponding expressions. The notion of local salience is then extended to a non-decreasing *discourse salience* (a cumulative measure of salience), which allows the system to select the globally most important discourse referents.

Boguraev and Kennedy's work is based more on linguistics, and less on statistical methods, than the systems discussed so far. It also performs text segmentation, and thus incorporates a very basic representation of discourse structure. Because its measures of salience are designed to 'reflect the distributional properties of a referent as the text-story unfolds', I have categorised it with the other attentional-network methods; summarisers that operate on more sophisticated representations of discourse structure are described in section 2.4.

At present, this system does not generate full-text summaries, but rather 'capsule overviews', which consist of extracted phrasal sentence fragments. These might be useful as the basis for further processing, or provide an indicative summary in their own right.

2.1.2 *Summarising from cohesive links*

Skorochod'ko

Frequency-based summarisers count the number of times that individual terms occur or are salient in the source representation, but they do not consider the particular patterns of co-occurrence that appear. To take account of these patterns explicitly, we can put a graph-like structure on the source representation, in which an edge between two nodes represents the co-occurrence of a term in two places in the source representation. Skorochod'ko (1971) proposed such a representation, in which nodes corresponded to sentences and edges indicated the presence of the same word, or related words, in two sentences. His claim was that the shape of the graph would reveal something about the text structure: for example, some texts are very linear ('chained structure'), in others every sentence is related to many others ('monolith structure'), and in others the sentences can be arranged in a number of groups with many relations within groups and few between them ('piecewise structure'). Skorochod'ko suggested that, ideally, a method of automatic summarising should be *adaptive* – that is, it should be able to deal with different kinds of text structure in appropriate ways. He considered two facts about each node i: first, the number of related sentences N_i (that is, the number of arcs incident at i); and second, the degree to which this node holds the graph together, measured as the number of nodes in the whole (connected) graph M minus the maximum number of nodes that would be present in any connected component of the graph after removal of the node under consideration, M_i^*. He then defined the *functional weight* F_i of a node by: $F_i = N_i (M \; M_i^*)$.

As an example strategy for summarising, Skorochod'ko showed how an abstract might be produced (it is not clear whether his method was actually implemented) by successively eliminating sentences with low functional weights. Unfortunately, he did not give examples of summaries produced in this way, and, although his emphasis was on adaptive summarising, did not suggest any other strategies for summarising from these graph structures or indicate how to identify different kinds of text structure automatically.

Salton et al (SMART)

Salton et al (1994), and more recently Mitra, Singhal and Buckley (1997), have studied the use of similar graphs of textual cohesion for text passage retrieval and summarising in their SMART information retrieval system. Here, however, the nodes correspond not to sentences but to paragraphs; arcs between them are assigned numeric values to indicate the similarity in content of two paragraphs (measured using a *tf·idf*-type formula for each word or stem in common). In SMART, summarising consists not of sentence extraction but *paragraph extraction*; the presence of many, highly-weighted links to a node is taken as evidence of its overall importance. The corpus used is an encyclopedia, which makes the linking (hypertext-style) of related paragraphs a natural idea. The use of paragraphs rather than sentences as the basic units has important consequences. One is that, as they are longer and contain more words, the attentional network (or *text relationship map*) is much denser than in Skorochod'ko's system, and it becomes necessary to assign weights to edges. Another is that summaries tend to be more coherent. (Of course, paragraph extraction could be applied to any summarising system that worked by sentence extraction, if it improved the resulting summaries. A related but more sophisticated approach to producing coherent summaries is *aggregation*, used in the ADAM system discussed in section 2.2.) The system also aims to produce more comprehensible summaries by ensuring that consecutive paragraphs in the summary are ones that were strongly linked in the source representation. However, the large granularity of the representation makes it unsuitable for summarising short texts, or for producing concise summaries.

Benbrahim and Ahmad (TELE-PATTAN)

In the terminology of Halliday and Hasan (1976), the occurrence of the same word in two sentences is called *lexical cohesion by repetition* – other kinds of lexical cohesion occur when related words, such as synonyms, superordinate terms, or collocations, are present in two sentences. (Halliday and Hasan also discuss other types of cohesion, corresponding to reference, ellipsis, substitution and conjunction).

TELE-PATTAN (Benbrahim and Ahmad 1994) is a system which explicitly deals with different types of lexical cohesion. With the aid of a thesaurus, it classifies inter-sentence links into categories such as *simple lexical repetition*, *complex lexical repetition* (for inflected forms of the same word), *paraphrase* (for synonyms) and *superordinates*. The summarising part of TELE-PATTAN also categorises sentences according to their cohesive function; this it does by considering not only the number of cohesive links they have, but whether they are to preceding or to following sentences in the text. The source representation, then, is a graph like Skorochod'ko's, but with *directed* edges. If we choose (arbitrarily) to direct edges so that they start at the earlier of the two sentences they link and end at the later, then nodes from which many edges start but at which few end are classified as *topic opening*, nodes at which many edges end

but from which few start are *topic closing*; nodes which have many edges irrespective of direction are called *central* and those with few, *marginal*. Having performed this classification, a summary can be generated by selecting sentences of appropriate types, depending on the summary purpose and required length. Presumably, for a use such as retrieving (which primarily requires an indicative summary), topic-opening sentences could be chosen, as they tend to introduce topics rather than comment on them; whereas if the use is substituting (for which an informative summary is required), topic-central or topic-closing sentences might be preferable.

Barzilay and Elhadad

Although TELE-PATTAN considers lexical cohesion between related words, it makes no attempt to explicitly represent the general concepts underlying sets of related words. Barzilay and Elhadad (1997) present a system which summarises by identifying *lexical chains*, i.e. sequences of related words in the source text. The analysis stage of their summariser 'reads' the source text sequentially, performing very shallow parsing to identify noun-compounds, and building up a representation of cohesive links between words and compounds in the text that are deemed to be in the same lexical chain. As each item is encountered, it may be added to a pre-existing chain, or used to start a new chain. A number of candidate representations or hypotheses are maintained as analysis proceeds, and eventually one is chosen as the final source representation.

The condensation stage scores each chain, considering *length* (total number of chain members) and *homogeneity* (the number of distinct members divided by the length). A source sentence is then extracted for each of the highest-scoring chains: the system can select either (1) the first sentence to contain a chain member, (2) the first sentence to contain a representative chain member (i.e. one appearing with at least average frequency within the chain), or (3) a sentence from a part of the text which contains a high density of chain members. The approach is similar to TELE-PATTAN's topic-opening and topic-central sentences. An informal evaluation found that method (2) produced slightly better summaries than method (1), and than method (3) produced much worse summaries than either. Because this system selects a single source sentence per lexical chain, the summary will include material relating to many different chains; thus it is more likely to be a broad-coverage summary than one produced by TELE-PATTAN, in which a single lexical chain can dominate the graph of lexical cohesion.

Systems that summarise from attentional networks have much in common. They use source representations not far removed from the surface text, in which the same or related words or phrases are identified as being prominent in different segments (sentences or paragraphs) of source text. Which units are selected in the condensation stage depends either on occurrence-frequencies, or on a graph representing the pattern of lexical cohesion in the text; either way, the condensation stage is responsive rather than prescriptive, and is based on

attentional information, since the content words in a sentence indicate salient topics and objects in that sentence. Synthesis typically consists of extracting whole sentences or paragraphs from the source text, so the summary achieves no shortening of expression, only shortening of content.

2.2 SUMMARISING SENTENCE BY SENTENCE

Summarising is evidently an operation on a whole text, and whether any particular piece of information ought to be included in a summary surely depends on its context within the rest of the source text. The idea that a summary could be produced by considering each sentence in isolation therefore might seem ridiculous. Yet there have been a number of summarising systems that did something very close to this sentence-by-sentence summarising: they decided independently or almost independently for each sentence in the source text whether that sentence was important for summarising.

Edmundson

A comparative study by Edmundson (1969) looked at summaries produced by sentence extraction using four different methods. One, the *keyword-frequency* method, was an approach similar to Luhn's. The others were: a *title-keyword* method, in which sentences were selected if they contained many words which appeared in the text's title or section headings; a *location* method, in which sentences were selected according to their position in the text, whether they began a paragraph, etc; and a *cue word* method, in which sentences were selected or rejected depending on whether they contained any of a set list of cue words. Edmundson's way of comparing the resulting summaries was to see how many sentences they had in common with a 'target extract' which he prepared himself; despite the shortcomings of this approach and the fact that each of the methods he investigated was capable of extensive variation, his results are still interesting. Edmundson found that the keyword-frequency method was the least effective of the four, followed by the title-keyword method, the cue word method, and the location method, in order.

The location method is implicitly concerned with discourse structure: the division of a text into paragraphs, and the choice of the order of information within it, are aspects of linguistic structure. Therefore the location method, which selected initial and paragraph-initial sentences, was a prescriptive use of (extremely shallow) linguistic structure.

In the cue word method there is even less explicit idea of discourse structure: the idea is instead that some words or phrases indicate that the sentence they occur in is particularly salient or non-salient. Some cue words such as 'however' and 'furthermore' indicate rhetorical relations, and to use them as direct indicators of salience or non-salience is therefore to summarise prescriptively from linguistic structure, even though that structure is not explicitly represented. With other cue words such as 'noteworthy' and 'possibly', and phrases such as 'this paper' the situation is less clear. These do

not indicate rhetorical relations, but rather say something about the intentions of the writer or about the information conveyed in the sentence. The cue word method of summarising is therefore a prescriptive one that uses several kinds of information: linguistic, intentional, and (if there are domain-specific cue words) informational and attentional too.

Rush, Salvador and Zamora (ADAM)

The cue method was used, in combination with some location criteria, by Rush, Salvador and Zamora (1971) (who give, as the abstract for their paper, one produced by their system). This work, and the resulting ADAM system (Pollock and Zamora 1975), used a *word control list* (WCL) of words and phrases with associated semantic weights. Interestingly, it was found that the most effective use of cue words was not to indicate particularly important sentences but instead to identify particularly unimportant ones; therefore, most of the entries in the WCL are terms such as 'obviously' and 'perhaps' with negative semantic weights.

In addition to this sentence-by-sentence selection process, ADAM was able to detect many instances of anaphoric reference by having a list of words and phrases that require antecedents, and to improve the coherence of the eventual summary by a technique known as *aggregation* (Paice 1990): rather than including sentences containing such phrases on their own, the system either adds preceding sentences or rejects the sentence in question if the preceding sentences are judged unsuitable for inclusion.

At least on the evidence provided by sample summaries, the results obtained with this system were quite impressive, but there were some problems: one was that, since the summariser worked largely by excluding unsuitable sentences, it tended to produce rather long summaries.

Paice

Paice (1981) sought to rectify this problem by concentrating on positive indicators of sentence suitability. He focused not on single words or short word strings, but on longer *indicator phrases* – phrases which 'explicitly state that the sentences containing them have something important to say about the subject matter or the "message" of the document.' Indicator phrases are often introductory statements such as 'Our purpose here is to discuss …' or 'Our studies indicate that …'. Rather than attempting to list such phrases individually, Paice constructed a number of templates with alternative sections which between them stood for hundreds of possible phrases. As in ADAM, Paice proposed an analysis of anaphors (and also of exophors, i.e. expressions which refer to known objects in the world outside the discourse) to decide when to apply aggregation to the summary text. Whereas ADAM's cue words and phrases were intended to indicate in general that a sentence might or might not contain important material, Paice's indicator phrases are much more closely tied to an implicit categorisation of the kinds of information found in the documents to be summarised, and specifically to particular categories (such as goals and

conclusions) which may be suitable for a summary. That good summaries consistently contain certain kinds of information and not others, is a view supported by studies into the structure of professionally-produced summaries (Liddy 1991), at least in the case of scientific abstracts in certain journals. Paice does not explore the idea, but presumably summaries tailored to different summary purposes and users could be produced by using different selections of indicator phrases.

Sentence-by-sentence summarising suffers from two problems. First, it is by definition very prescriptive, because it takes sentence-level features as being direct indicators of importance. If and when such a strategy works, it is not because the summariser grasps the meaning of the source text and identifies what is important in it, but because the author of the text has helpfully left clues that tell us without much analysis which parts are important. More recent work has taken the idea of cue words and phrases, but seen them as indicators of an underlying discourse structure which should be expressed in the source representation. This is the approach taken in the more responsive summarising systems of Miike et al (1994) and Marcu (1997), discussed in section 2.4.

The second problem with this approach to summarising is that it tends not to work on all kinds of source texts. In developing the ANES system discussed above (Brandow, Mitze and Rau 1995), the use of indicator phrases was rejected because 'phrases of this type were heavily source document-dependent and could not generalise across the entire range of material being summarised. … This approach can work for narrow kinds of texts, especially scientific articles … but is inappropriate for domain-independent summarisation.'

Indicator phrases may be common in scientific papers, but they might be scarcer in texts whose form is less controlled, whose subject matter more diverse, or whose expression of their content is more compact. Some cue words and phrases are inherently specific to a particular form ('in this paper, we show that') or subject matter ('bomb' and 'war' as cue words for newspaper stories). To achieve better and more general-purpose summarising, it seems we must make some attempt to go beyond the surface of the source text; indeed, Edmundson concluded that '… future automatic abstracting methods must take into account syntactic and semantic characteristics of the text' (Edmundson 1969).

2.3 SUMMARISING FROM INFORMATIONAL CONTENT

This category comprises methods and systems that attempt to 'understand' the text – i.e. to represent and reason about its meaning. This kind of processing implies more sophisticated linguistic analysis and deeper source representations, but allows the production of a much more detailed summary representation, and the generation of fresh summary text rather than the extraction methods so far described.

Schank, Abelson and Cullingford (SAM)

The SAM system (Schank and Abelson 1977, Cullingford 1981) was designed not primarily for summarising, but for general-purpose text analysis and synthesis. It represents events and states described in a text using the theory of Conceptual Dependency (CD), and integrates them into a global text representation by means of *scripts*. A separate summarising component takes instantiated scripts, and from them produces summary representations, which then can be processed by SAM's synthesiser to give summary text. The summarising component itself is simple; for each script it knows which parts are more and which less interesting from a summarising point of view, and produces a summary representation corresponding to the interesting ones, favouring those which have been instantiated from the source text rather than deduced in the process of analysis. The relative simplicity of the summarising component arises from the fact that a script-based analysis of a text already goes some way towards summarising it; to say 'the events described in this text are an instance of such and such a script' is itself to give a summary of the text, albeit a very brief one. SAM's approach was prescriptive in as much as the scripts were marked to indicate important information, but responsive in that there was flexibility within the script structure (there could be alternative tracks or subscripts) and in that what was chosen for the summary also depended on what was given most attention in the source text.

DeJong (FRUMP)

FRUMP (DeJong 1982) was a quite successful example of the practical application of scripts specifically for summarising. The analysis performed by FRUMP was considerably simpler than SAM's, because it used *sketchy scripts* (scripts containing only essential information), and rather than analysing the whole text in detail, it would ignore phrases which did not substantiate anything in the chosen script. This strongly directed processing operated by making predictions at the script and event level of what kind of information might be found in the text, then analysing the source with the specific intention of finding that, and only that, information. This meant that FRUMP could process real-world texts, specifically stories from UPI newswires, at reasonable speed. However, it also meant that the system was very *prescriptive*: where stories deviated from the script, contained additional information, corresponded to none of the available scripts, or combined material from topics corresponding to two or more scripts, the summaries did not reflect this variation. Indeed they could not, because not just the condensation stage but the analysis of the text was directed at extracting only the kinds of information which had been specified in advance as being important.

FRUMP had a fixed opinion, in this sense, of what would interest a typical reader in each category of story, a problem that was made worse by the fact that only one script could be selected for each story. For example, FRUMP's sketchy-script for the breaking of diplomatic relations had only three instantiable slots:

the two countries involved and the level at which relations were broken. If this script was selected, and a story said that relations had been broken off because of a terrorist incident, the incident would not be mentioned in the summary. Conversely, if a script about terrorist incidents was selected, the information about diplomatic relations being broken off would not be present in the summary.

FRUMP had the rare distinction, for a summarising system, of being evaluated in a quasi-realistic situation: in experiments it was connected to a news wire service, and it attempted to summarise those incoming stories to which it was able to match one of its sketchy scripts. (This evaluation is discussed further in chapter 3.) The results suggest that FRUMP often produced concise, relevant summaries when an appropriate script was selected, although sometimes failures in script selection could lead to confusing or misleading summaries.

Tait (SCRABBLE)

Although it attempted to analyse the whole text in some detail, and had a more sophisticated script system, SAM's summaries, like FRUMP's, were likely to suffer from being overly prescriptive. Tait's SCRABBLE summariser (1983) attempted to overcome this problem: a flexible representation made it a much more responsive summariser than other script-based systems; compared to FRUMP its generation was also more flexible.

Like SAM, SCRABBLE analysed sentences to give CD representations of meaning, and used these to instantiate scripts. But its scripts were simpler than SAM's, more than one could apply (perhaps concurrently) to a single story, and the representation could also incorporate information which did not fit into any appropriate script.

SCRABBLE's condensation stage operated on a fundamentally different principle from that of SAM or FRUMP, taking the view that it is precisely the information that is unexpected and does not fit into a script that is most important in a text. The summary representation was formed by preserving such unexpected material and instantiating CD templates for the scripts used. Unexpected material was anchored to a point in the script corresponding to its position in the source text, so that it could be inserted at an appropriate point in the summary.

More domain-specific systems

SAM, FRUMP and SCRABBLE were all intended to be extensible in the sense that more scripts could be added to reflect more possible situations or text topics. A number of other summarising systems have been developed which operate only in a specific domain. For example, SCISOR (Rau, Jacobs and Zernik 1989), a script-based summariser and information extraction system, only deals with texts about corporate takeovers. It is described as being 'built with domain-independence in mind,' but this appears to mean that the same technology can be used to produce a similar system operating in a different domain, not that

SCISOR itself can be extended to deal with texts from a mixture of domains. Some other frame-based and scripty systems are even more specifically designed to process particular kinds of source text in particular contexts: for example, TESS (Young and Hayes 1985), and TICC (Allport 1988) deal with very short, telegraphic sources (banking telexes and traffic incident reports), and process source texts in sequence, producing summaries of the messages received so far.

Extensibility of script-based systems

Although SAM, SCRABBLE and FRUMP (and systems such as that of Marsh (1984), where the source representation consists of surface words and phrases arranged under domain headings) can in theory be extended to deal with wider and wider varieties of subject matter, there may be difficulties here too: the number of different situations which require scripts is enormous, and aside from the human effort involved in writing them, this may pose problems in itself. As the number of scripts increases, we would expect the difficulty of selecting the (or even a) correct script for a text to increase too – often there will be several candidate scripts, each partially applicable. DeJong and Tait were certainly aware of these issues, but to deal with this complexity may well require more advanced and flexible concepts of scripts and sub-scripts than those used in these systems. Without the experience of attempting to apply script-based processing to a really wide range of subject matters, it is hard to say how well these methods would scale up.

A second, more fundamental problem is with the *genre* of the source text (as defined in section 1.1.1). Script-based systems are quite well equipped to deal with texts which consist of factual description of an event or events, particularly when presented as a narrative, but these systems are not good at dealing with texts which consist of opinion and argument instead of or as well as simple description. This problem is a consequence of the event-based nature of script representations, which take little account of linguistic or communicative content and structure.

Lehnert

Lehnert (1982) suggests a very different, bottom-up approach to summarising from informational content. Her theory of *plot units* centres on emotions and 'states of affect' of characters in stories: events are linked causally by concepts of motivation, and have positive or negative outcomes for the characters concerned. Combinations of events give primitive plot units such as 'change of mind', 'loss' and 'hidden blessing', which in turn combine to form higher-level plot units with relations between them, and an overall story structure. Lehnert's claim is that a good summary of a story is one which conveys, explicitly or implicitly, its high-level plot units. She presents a method for producing summaries, although it has not been implemented. In the analysis stage events and plot units are identified in a source text. In condensation the top-level plot units are considered, and from them one *pivotal* unit is chosen (i.e. a unit related

to as many other units as possible). A number of additional plot units, related to the pivotal unit, are also selected. In the synthesis stage, summary text corresponding to the pivotal unit is generated, and is augmented with information from the related plot units. This method is innovative and interesting, but very difficult to implement, as analysing the text in terms of the emotions of characters involved requires an (unspecified) inference mechanism. The bottom-up approach makes it (in contrast to script-based systems) a very responsive summarising strategy, but it is one with limited application, as the whole approach is designed specifically to work with a subclass of narrative texts only, namely ones in which it is characters' emotions that are important.

Taylor

Taylor (1975, also Taylor and Krulee 1977) presents a method of summarising based on semantic networks: graph structures that give a deep semantic representation of a text. His method was very different from that of the script-based systems, in that no domain or world knowledge was used in condensation.

In Taylor's source representation, nodes corresponded to discourse entities and events, and edges to various kinds of relationship that can exist between them. The relationships used were deep case-relationships such as *agent* and *instrument*, intended to be universal (and presumed to be innate), and not dependent on the style or content of the source text. Each edge was given a weight, depending on the underlying relationship, producing a weighted graph. The condensation methods applied to this graph were related to those used on the graphs of lexical cohesion discussed in section 2.1.2 (Skorochod'ko 1971, Benbrahim and Ahmad 1994), in that they selected nodes without considering the nature of the underlying relations, and in that the central intuition in all these methods was that nodes to which many other nodes are connected are in some sense more central to the text than nodes with fewer connections. The crucial difference between Taylor's method and these others was that here the units being selected were objects in a *semantic* representation, reflecting the informational content of the text, rather than a superficial representation of attentional content.

Taylor's condensation algorithm was also more complex than that of Skorochod'ko or Benbrahim and Ahmad. It operated in two stages. In the first stage, the graph was split up into connected components. In the second stage, the largest component was considered, and *signal flow analysis* (Mason and Zimmerman 1960) was applied to assign numerical significance values to each of its nodes.

Signal flow analysis is a technique developed for analysing electronic circuits, in which a signal is considered to be flowing through the graph, with weights indicating the strength with which the signal is transmitted along each edge. When two nodes are connected by an edge or a chain of edges with high weights, the signal will be strongly propagated from one to the other.

In Taylor's system, the *significance* of each node was the strength with

which a signal entering at that node would propagate through the graph and arrive back at that same node. Nodes with many highly-weighted edges and loops of edges incident to them were therefore given high significance. Taylor's algorithm selected the node with the highest significance for the summary, and the procedure was then repeated (from the first stage) on the smaller graph obtained by removing this node. After a number of iterations this process was terminated, and the largest component of the remaining graph was combined with all the nodes removed in previous stages to give the summary representation.

This condensation algorithm seems quite *ad hoc* – it is difficult to tell exactly why signal flow analysis was chosen – but it does address issues of importance and representativeness of the summary. The measure of numerical significance was intended to reflect the global importance of nodes in the graph, capturing not just the degree to which each node was directly connected to others, but also considering indirect connections involving several edges. The algorithm will select nodes not just from the largest component in the graph, but subsequently from other large components, producing a more representative summary than we would obtain simply by taking all the nodes with highest significance (these would tend all to come from the same component). The criteria of importance and representativeness are ones I will return to in chapter 6.

Taylor's condensation method, being concerned with the shape of the whole graph, can be categorised as a very responsive one based on the informational content of the text. We might hope that such a method, not being dependent on the subject matter or narrative genre of the source text, could be used to summarise documents to which, for example, script-based systems could not be so successfully applied. However, Taylor only implemented his graph-condensation algorithm – he assumed that the analysis and generation would be performed separately; this meant that evaluation was limited to considering texts for which a deep semantic representation could be constructed automatically, or to representations constructed by hand. This restriction makes it difficult to say how effective Taylor's condensation methods might have been.

2.4 SUMMARISING FROM DISCOURSE STRUCTURE

Some of the summarising systems discussed so far use discourse structure, but only in a limited way. On the one hand, in section 2.1 (summarising from attentional networks), we saw a number of systems that apply very generic notions of 'aboutness' to the source text; on the other hand, in section 2.2 (summarising sentence by sentence), we saw methods which used surface indicators of linguistic and intentional structure, but did not attempt to represent that structure. The methods discussed in this section, however, apply discourse theories in their analysis of the source text, and produce an explicit representation of source discourse structure.

There are a variety of theories of discourse structure to choose from, ranging from low-level theories of focus or centre to larger-scale models such as rhetorical structure theory (RST) (Mann and Thompson 1987) or Hobbs' (1990) coherence relations. The whole area of discourse structure is a complicated one, and although many theories have much in common, there are important differences in general approach as well as in detail (Sparck Jones 1991) and often there are no clear reasons to prefer one theory to another.

Some work has been done on using discourse structure for summarising, but much of it is largely theoretical, as the identification of discourse structure in many cases presupposes an accurate syntactic and semantic analysis of text, and further, demands a combination of pragmatics, general-purpose inference and world knowledge – requirements which put automatic analysis of discourse structure beyond the current state of the art in natural language processing. Where specific domain knowledge is available, however, something can be achieved.

Hahn (TOPIC)

The TOPIC system (Hahn 1990) produces structured index descriptions (a kind of summary) by identifying dominant themes in paragraphs and considering thematic progression in the source text. The system represents attentional structure as captured in theme–rheme relationships (Janos 1979), but because TOPIC operates in a restricted domain (the source texts are computer manuals) it can do so without performing a complete linguistic analysis (the processing is more like that of FRUMP, in that it is geared towards finding instances of a predefined set of useful linguistic items and structures). Because of this, the representation of discourse structure is rather limited; TOPIC does not attempt to produce an overall analysis of discourse structure in the source text as such.

Sparck Jones

Sparck Jones (1995) describes some experiments into summarising from large-scale discourse structures: what she presents is not a performance comparison but an analysis (by hand) of some possible methods using some current theories of discourse as examples. These are categorised according to the type of content they represent (linguistic, intentional or informational, in the terminology of chapter 1) and whether they are top-down or bottom-up in approach.

The strategies based on informational content are similar to the script and frame-based methods discussed in section 2.3. RST (Mann and Thompson 1987) is taken as a bottom-up linguistic representation, and story grammars in the style of Rumelhart (1977) as a top-down one. For a representation of intentional content, Grosz and Sidner's theory of intentional structure (1986) is used. These three theories all give tree-structured representations of discourse structure, and in each case there is a natural summarising strategy. For intentional structure, select the discourse purpose associated with the topmost segment; for RST, start at the topmost node and recursively take the nucleus until a ground clause is reached; for story grammars, each rule has an associated summarising rule

which produces a summary for the parent constituent (usually by selecting the summary of one or more of its children).

These strategies give interesting results, but again it is not feasible (nor was it the intention) to attempt to build an automatic summarising system based on them: the analysis required is too advanced. This is not to say that all methods of summarising from text structure must be equally impractical, but it does mean that to take such an approach we must look for a theory of discourse structure which does not necessarily require deep semantic interpretation of the source text.

Although the representations are very different, the three summarising strategies described above have an important similarity: each uses the tree-structure of the representation by supposing that the topmost node either represents directly or can be used to determine indirectly what is most important in the text. It seems that, given any representation of discourse in a hierarchical structure, the strategy of selecting nodes from the top of the hierarchy is in some sense a natural method for condensation (although there is still the potentially complex issue of what content should be chosen to correspond to these nodes): indeed, just such a strategy is employed by the next two systems to be discussed.

Miike et al

The summarising system of Miike et al (1994) is an example of a practical summariser which relies on discourse structure. It uses the analyser of Sumita et al (1992) to assign a tree structure to the source text, based on its use of connectives such as 'for example' and 'in addition' (or rather, their Japanese counterparts). The presence of such connectives is taken to indicate a rhetorical relation between two sentences (a default relation is assumed when no connectives appear), and precedence rules are applied to determine which of the possible combinations of these relations leads to the correct tree structure. The relations are also classified according to whether their left or right node is the nucleus, or both nodes have equal status. The result is a bottom-up representation that reflects the deep linguistic structure of the text, although it is inferred from evidence at a very shallow linguistic level.

Summaries are produced by a responsive method similar to Sparck Jones' RST-based summarising, by following the trail of nuclei from the top of the tree until a sentence is reached. For longer summaries, paths from the top of the tree which involve a small number of non-nucleus steps are also considered.

The results reported for this system are good: in a test on technical papers and newspaper editorials, an abstract of 24% of the length of the source text (admittedly quite a long abstract) included 51% of sentences previously identified as 'key sentences' by hand. However, we might expect other styles of writing to give different results since, as was the case with cue-words in section 2.2, the frequency of discourse connectives varies considerably between genres. A test on newspaper stories, for example, might prove less effective.

Marcu

The summarising system of Marcu (1997) has many similarities with the system of Miike et al. It too performs relatively shallow text processing to obtain a very deep source representation, namely a rhetorical structure tree (Mann and Thompson 1997) for the source text. There are also some important differences between these systems: Marcu's summariser splits the source text up into clauses rather than whole sentences, and uses an analysed corpus to hypothesise what rhetorical relations may hold between units on the basis of discourse connectives and markers. A set of constraints is used to rule out invalid structures, and a metric is applied to those remaining to determine the best structure.

The most salient units are then determined by much the same method as in Sparck Jones' and Miike's work: starting at the top of the tree and following the trail of nucleii.

Marcu evaluated his system by asking independent judges to rate textual units according to their importance, and then computed *precision* and *recall* of the most important units, both for his summariser, for a random extract, and for a summariser included in Microsoft Office 97. The Microsoft summariser was found to be only slightly better than random, having precision of 28% and recall of 26%, whereas the RST-based system had precision of 53% and recall of 50%. The evaluation, however, was very small: only five texts were involved (from 161 to 725 words in length). In addition, texts were chosen (from *Scientific American*) that 'were considered to be well-written' (the example text given contains a large number of discourse markers). As with the previous system, therefore, it is unclear how well Marcu's summariser would perform on other kinds of text.

2.5 SOME BRIEF CONCLUSIONS

In this section the focus is on issues that are important in designing automated summarising systems rather than theoretical methods, and in what in chapter 1 was tentatively called *basic summarising* – i.e. producing reflective summaries of basic texts. Specifically I want to consider *robustness*, what we can say about *general-purpose methods*, and issues of *granularity* in the source representation and *precision* in the condensation stage.

2.5.1 *Robustness*

Robustness, i.e. the capability to deteriorate gradually rather than fail all at once when errors occur, is a desirable property for all NLP systems, as current methods of syntactic and semantic analysis are very far from 100% accurate or reliable. Robustness is particularly important for applications such as summarising which involve complex, real-world text.

We can distinguish *robustness in processing*, which is dependent on

implementation, from *robustness of the representations* we use. Robustness in processing, for example during the analysis stage, means being able to analyse the rest of a text at least partially-correctly even when analysis fails on one part. Robustness of a source representation means that when some part of the text cannot be represented accurately or at all, it is still possible to give a reasonable representation for the rest of it. The two are closely related: obviously if a representation is not robust, then any processing which produces it cannot be robust either. So to produce a robust summarising system, we must begin by looking for robust representations of the source text.

Obviously there is likely to be a trade-off between robustness of source representations and their depth, or distance from the surface text. A representation consisting simply of the surface text, divided into sentences, is very robust. Deep semantic representations may not be, in the sense that if one part of a sentence is not understood, the whole sentence may have to be ignored, and if the sentence introduces entities and events which will be referred to anaphorically, then anaphor resolution may fail and later information about them may also not be represented properly.

Local, graph-like and tree-like representations

Robustness of a representation is crucially affected by the form of its structure. In this respect the summarising systems and strategies discussed in this chapter almost all fall into one of three categories: those whose representations are essentially local, consisting of individual unit representations with no relations connecting them (e.g. Pollock and Zamora (1975)); those which represent relations between units in a graph structure (such as Skorochod'ko (1971), Benbrahim and Ahmad (1994), and Taylor (1975); and those which use relations to combine units in a tree structure (many of the methods of Sparck Jones (1995), Miike et al (1994), Marcu (1997) and all the frame-based systems – although some of them may not in fact allow nested frames).

The entirely local kind of representation is very robust, since an error in the representation of one part of the text does not affect the rest of it (we are not considering here the robustness of the processing used to produce such a representation, only the robustness of the representation itself). Graph representations achieve a degree of robustness in that an extra edge here or a missing node there need not affect the correctness of the rest of the graph. Top-down tree-structured representations (such as scripts) are very robust, because they impose order on the content of the representation. Bottom-up tree-structured representations (such as RST) are not very robust, because there is the risk that a single error, even near the bottom of the tree, may have consequences that propagate upwards to the top of the tree, or even make the construction of a single tree impossible.

If we are aiming for robustness, then, and require something more sophisticated than a local representation, we should consider choosing a top-down tree-structured representation, or a graph-structured representation. FRUMP, for example, takes the former option, and is able to robustly ignore

sections of text that it cannot analyse. But although FRUMP is very robust once an appropriate script has been selected, if it selects an inappropriate script it may produce a representation which, though well-formed, is completely wrong for the source text.

It is perhaps surprising that graph structured representations have not been pursued much except in very shallow approaches. (Taylor's (1975) method uses a graph-structured representation, but one sufficiently deep that the source analysis involved would be impractical for most texts). The use of graph representations at a shallow but nevertheless semantic level might well help us to build more advanced summarisers than those based on surface text representations, while retaining some of their applicability to complex text. This is the approach taken by the CLASP summariser, introduced in chapter 4.

2.5.2 *More and less general-purpose methods*

There are two different ways in which summarisers might try to be general-purpose. Firstly, by being able to summarise a reasonably wide variety of source texts of different genres and about different topics. Secondly, by producing reflective summaries that do not make strong assumptions about who the user is or what they are interested in. Of course, these requirements are subjective: people will disagree on what constitutes a 'reasonably' wide range texts or a 'strong' assumption. As noted in chapter 1, there can never be truly general-purpose summaries when we consider all possible context factors.

Much of the work in automatic summarising has attempted to find methods that are general-purpose in the second sense. However, although some systems, such as TELE-PATTAN (Benbrahim and Ahmad 1994), can produce several different types of summary, there is no explicit modelling of what the user's needs are, either in terms of the condensation stage (i.e. what the content of the summary should be) or the synthesis stage (i.e. how it should be presented). While not ideal, this is understandable both on the grounds that reflective summaries are genuinely useful for a range of tasks, and that we have no readily automatable way of expressing context factors anyway.

Whether summarising methods are general-purpose in the first sense depends largely on the kind of representation they use. Those systems with script- or frame-based representations are not general-purpose because for some genres of source text this kind of representation, which is suited primarily to descriptive or narrative texts (i.e. ones describing properties of objects and sequences of events), is simply not applicable, and because such systems can only function in domains where the appropriate scripts or frames are available. Other methods which use the informational content of the text are less specialised – for example, Taylor's reduction of semantic networks is a condensation method which is concerned with the text's meaning without requiring domain knowledge.

Some other methods fail to be general-purpose because they rely on particular aspects of the form, genre and style of the source text. These include

many of the techniques discussed in section 2.2, such as the use of cue words and indicator phrases, discourse connectives, or headings, paragraphing and other presentational features.

This leaves relatively few summarising methods that can justifiably be called general-purpose: the attentional-network methods, some of the discourse-structure methods (if we assume that theories such as RST and Grosz and Sidner's intentional structure are generally applicable), and perhaps Taylor's (1975) graph reduction. Of these, however, only the attentional-network methods can be readily implemented. As noted in section 2.4, the automatic analysis of discourse structure is difficult, and those systems that do perform it rely on the presence of discourse markers that may not be sufficiently frequent in some kinds of text. The complex analysis required to obtain the source representation for Taylor's method could currently only be achieved for simple texts in a limited domain, as it relies on a complete semantic analysis of the source.

2.5.3 *Granularity and precision in selection*

Most of the systems discussed here operate by *selection*; that is, the summary representation consists of fragments of the source representation, rather than new material deduced from it. This is explicitly the case for methods which work by selecting or rejecting sentences or other units independently of the rest of the source. Most of the summarisers that use occurrence-frequencies or attentional networks also produce output summary text by sentence or clause extraction, as do Miike et al (1994) and Marcu's (1997) summarisers that use discourse-structure. The major category of summarisers for which summarising does not operate only by selection is the script- or frame-based systems. By instantiating a script or frame, these systems are able to generalise the material in the source text, and give information in the summary which is deduced from the text as a whole rather than from a specific part of it.

Where selection is the means of forming a summary representation, the kinds of summaries produced will depend greatly on the granularity of that representation, i.e. the size of the units that can be selected. Large unit size can be a problem for sentence-extraction systems: when a single sentence contains a mixture of important and unimportant information, the system cannot separate the two. Selecting whole paragraphs at a time makes this problem worse, although it helps presentation and therefore may allow the user to better assess the relative importance of information. Clearly, a fine-grained source representation will allow the process of selection to pinpoint the content suitable for summary purposes more precisely; Marcu's system, for example, selects not whole sentences but individual phrases of source text. However, selecting very small units may be detrimental to summary comprehensibility and coherence. For example, the methods tested by Gladwin et al select individual entities, or corresponding nouns; the result is a list of separate words. (Of course, these were not intended to be used as a summary, but as the basis for a summary,

although it is not clear how a summary should be constructed from them.) This is not just a problem for surface-text approaches: Taylor reported than his original condensation algorithm would sometimes select a verb and its object, but not its subject, for the summary representation; special rules had to be written to force the summariser to select enough information to form a meaningful summary. The problem is that the nodes in Taylor's representation do not correspond to facts, but to entities: by selecting a node, the summariser can choose to say something about an object or person mentioned in the text, but it cannot explicitly choose *what* to say about that object or person, as that kind of content is captured only in the graph structure.

It seems, then, that in choosing an appropriate granularity of source representation we must consider what are the kinds of things we can consider suitable or unsuitable for inclusion in a summary. In chapter 4, I take the view that what is important or unimportant in a text is not primarily the entities (people, object, places) that figure in it, but the ideas it expresses about those entities; a suitable representation would therefore have basic units larger than a word or entity, but smaller than a sentence or complex proposition. The CLASP summariser, described in chapters 4–7, uses just such a representation, in which the basic units are simple predications (complex propositions being broken down into their constituent predications). CLASP's condensation stage is then able to select individual predications for the summary representation.

3 EVALUATING SUMMARISERS

Evaluating summarising systems and the summaries they produce is difficult. We want to establish whether a summary meets the requirements imposed by its context factors (specifically, use and output factors) but it is typically hard to express these factors precisely, and having expressed them it is usually hard to measure objectively to what extent a given summary meets them.

There are several features we might expect of a good summary: it should include important information from the source text, and exclude unimportant information; it should be clear and readable; it should be concise and avoid repetition. However, although these are desirable properties of many summaries (and form the basis of many of the evaluation criteria described later in this chapter), what makes a good summary ultimately depends on the use to which it will be put.

Ordinarily, summaries are designed to be used by more than one person and on more than one occasion; however, each individual use will take place in a slightly different context. Even the same person reading a summary for the second time will interpret it in a different context – their needs may have changed in the meantime, and their knowledge *has* changed, even if only because they have already read the summary. Similarly, someone reading a summary who has already read another summary of the same document may have quite a different experience of the summary as a result. In a sense, then, each use of a summary is unique. These issues, as pointed out by Sparck Jones, are not specific to summarising but apply to other NLP and AI tasks. She argues that an appropriate evaluation of summaries must take context factors into account (Sparck Jones 1999).

In this chapter, I discuss several approaches to evaluating summarisers, both in theory and with reference to what has actually been done. I have identified four main categories of approach (*direct evaluation, target-based evaluation, task-based evaluation*, and *automatic evaluation*). This is a general-purpose categorisation, intended to include future evaluation methods as well as those that have been tried; the background for it is the work of Sparck Jones and Galliers (1996), who give a more comprehensive view of NLP evaluation in general. I end with some brief conclusions, and a note on the (rather limited) evaluation of my system, presented in chapter 8.

3.1 DIFFERENT WAYS OF EVALUATING SUMMARISERS

General framework

Sparck Jones and Galliers (1996) present a framework for thinking about NLP evaluation in general. They describe evaluation in terms of *criteria* expressing desired system behaviour, which are translated into specific *measures* of performance, which are applied according to a test *method*, within an experi-

mental design. In the case of a summarising system, evaluation criteria might be general requirements such as that the summaries contain important material, and that they are readable. Measures for these criteria could include, for example, the number of key points (as previously determined by the experimenters) included in a summary, and readers' individual judgements of readability. Test methods must be much more specific: e.g. take one hundred documents from a particular source, give summaries to ten (adult, educated, non-specialist) readers, ask them to read each for 2 minutes then report readability on a scale of 1 (incomprehensible) to 10 (perfectly clear and readable).

Before producing criteria, measures and methods for evaluation, we must consider what it is we are to evaluate, and from what perspective. Sparck Jones and Galliers make a fundamental distinction between a *system* consisting of computer hardware and software, and a *setup*, which consists of a system operating in a particular context. Either system or setup (of course there will be many possible setups for a given system) may be the subject of evaluation, and may be evaluated *intrinsically* according to its internal objectives, or *extrinsically* according to its external function. Evaluation criteria which are extrinsic to the system may well be intrinsic to a surrounding setup, but a setup may in turn have extrinsic criteria of its own. For an automatic summarising system, making the system the subject of evaluation, we might have intrinsic criteria such as whether the summary contains important information from the source text, its accuracy, its representativeness of the source text and its readability. Extrinsic criteria include, for example, whether people find the summaries useful, or how easily they can be touched up to make professional quality abstracts. If instead we want to evaluate a setup involving this system, for example by attaching it to a document retrieval system to aid users in selecting relevant documents, we have different criteria. Intrinsic criteria include the extent to which users find more relevant documents and fewer irrelevant documents than if they did not have the summariser, and whether they spend less time in searching for the information they need. Extrinsic criteria involve wider issues such as whether people would choose to use such a document retrieval service, or whether it would be cost-effective for an organisation to buy it.

Four kinds of evaluation

We can categorise evaluations by how the chosen criteria are measured. In evaluation by direct judgement (*direct evaluation*), readers make individual judgements on whether each summary meets its criteria. In evaluation by target-comparison (*target-based evaluation*), a human reader constructs a target from each source text, and a measure of how closely each summary 'hits' the target is taken. In *task-based evaluation*, rather than judging summaries directly, readers use summaries to assist them in performing a task, and we measure the success with which the task is performed. Lastly, in *automatic evaluation*, the summaries are processed automatically to measure to what extent the criteria are met.

Because this categorisation is based on the *measures* used, it says nothing about the choice of criteria and little about the test *methods* used. Therefore within each approach, evaluations may differ greatly in detail. I will give examples of actual and potential evaluations, and discuss relevant similarities and differences.

3.2 EVALUATION BY DIRECT JUDGEMENT

This seems the most obvious way to evaluate a summariser: look at the summaries and decide whether they are any good. It means evaluating the system without using any particular setup, typically using intrinsic criteria such as readability and representativeness, or perhaps vaguer criteria such as 'acceptability'. Whatever criteria are chosen, the *measure* of them is to ask readers whether, or to what extent, they are met.

3.2.1 *Examples of direct evaluation*

There have been some evaluations of summarising systems of this type; the most thorough perhaps being that of the ANES system (Brandow, Mitze and Rau 1995), whose performance was compared with that of a simple algorithm which simply extracted sentences from the start of the source text until the required summary length was reached. The comparison was made by producing summaries of 250 documents chosen to be representative of the intended type of source text (newspaper stories). For each text, summaries of 60, 150 and 250 words were produced by ANES and by the naive initial-sentence(s) summariser. The summaries were then judged by readers who were experienced news analysts, in relation to specific guidelines on criteria of content and readability. The readers compared the summaries to the source texts, and decided whether each summary was acceptable or not. The measure of content was therefore related to what news analysts thought was important in news stories, and so perhaps has some relation to an imagined application of the summaries, although this is not made explicit.

A more limited kind of evaluation by human judgement was carried out by Earl (1970), whose program, like ANES, produced summaries by sentence extraction. Earl did not evaluate the acceptability of extracts in their entirety; rather she classified each sentence in each extract as acceptable or unacceptable and thus gave each summary a percentage score: the proportion of its sentences that were acceptable. Necessarily, Earl's measures were different from those of Brandow, Mitze and Rau's evaluations: she rejected sentences 'if they seemed trivial in content, if they contained unclear antecedents, or if they were redundant with another better sentence.' Whether Earl's evaluation, which involved only four texts, was large enough to provide any useful information is questionable, but her evaluation measure is certainly interesting. It reflects criteria of relevance and comprehensibility, but in a limited, sentence by sentence way. Presumably 'trivial in content' is not intended to be an

application-specific judgement, but it is not clear whether it is defined absolutely or contextually (i.e. depending on the rest of the text). To achieve consistency in such evaluations, we would need to find a way to quantify just how trivial a sentence must be to be rejected.

Minel, Nugier and Piat (1997) describe two protocols, FAN and MLUCE, for direct evaluation. The FAN protocol is concerned with legibility, and considers the summary on its own. It includes a count of dangling anaphors, a count of tautological sentences, and a judgement of overall legibility (as being very bad, mediocre, good or very good). The authors report that, despite the subjective elements in the FAN protocol, there was very little variation in judgement between different assessors. The MLUCE protocol is concerned with summary content, and is based on a statement of summary use. Minel et al give evaluation criteria for two particular uses: allowing the reader to decide whether to read the source text (*retrieving*, in the terminology of section 1.1.2) and helping the reader write a synthesis of the source text (*prompting/previewing*). For both uses, the criteria of 'identifying the field or nature of the source text' and 'presenting the essential ideas' are given. For the *retrieving* task, the third criterion is that sentences in the summary should not be irrelevant or cut off from context. For the *prompting/previewing* task, the third criterion is that the summary should 'highlight the logical linking of ideas' in the source text. All these criteria are subjective, but quite specific measures and methods are introduced for them, involving readers filling in multiple-choice grids for source and summary texts. How to apply the FAN and MLUCE protocols is therefore defined in some detail; however, they are specific to one particular summarising system in a number of respects.

3.2.2 *How professionals summarise*

We might expect that research into how professional abstracters produce their summaries would suggest more specific (and more easily measurable) criteria than the general notions employed in the evaluations just discussed, and that such criteria could usefully be applied in direct evaluation. For example, Liddy (1991) compared the opinions of expert abstracters with the observed discourse structure of actual abstracts. This was not an evaluation of summaries; nor was it an attempt to discover criteria for evaluating summaries or methods for producing them. Instead, Liddy's goal was to produce a discourse structure model (in the form of a list of information categories) for empirical scientific abstracts, and then test it against further examples. In one part of the study, she asked professional abstracters to list information categories (e.g. motivation, methods, results) contained in scientific papers and to select from these types those which they would include in a summary. A comparison with actual abstracts was performed to see how well the information in them matched the categories selected by the experts. Liddy concluded: 'There does appear to be a predictable discourse-level structure in empirical abstracts,' and further that 'it is very similar to the internalised structure that abstracters offered.' The most

typical topics of an empirical abstract, according to this study, are those classified as subjects, results, purpose, conclusions, methodology, references and hypothesis.

This suggests that, if we want to evaluate a summarising system that is designed to produce abstracts of scientific documents (and specifically, documents describing scientific experiments), we might usefully judge each summary according to whether it contains information on those topics considered most appropriate or typical for such abstracts. In the case of a system which operated by specifically attempting to categorise material from the source text according to such a discourse model, this approach has an obvious benefit as a white box (or glass box) method of evaluation: we can see which kinds of material the system is most and least effective at identifying, and therefore how it needs to be improved. Such criteria are also potentially useful for black box evaluation, as presumably they reflect the opinions of the intended readership (the scientific community) as to what kinds of content are valuable in an abstract. Of course, the disadvantage of these criteria is that they are quite specific to a particular form and (though less so) subject matter of source text, and to particular applications (i.e. summary functions and intended readerships.) An additional difficulty is that it may be difficult to identify information categories of the right *granularity* – i.e. specific enough to pinpoint important information, but general enough to apply across a range of source text. A category such as 'results', for example, may be too coarse to be helpful in evaluating a summary of a scientific paper, if the paper reports a number of results only some of which are particularly relevant.

3.2.3 *Problems with direct evaluation*

Even when the criteria used are application-specific, evaluation by direct human judgement may be criticised for not actually evaluating summaries in the context of a realistic situation or use. Certainly the readers have no need for the summaries; instead we are relying on their ability to say, in advance, whether summaries would be useful if they (or an imagined user) did have a need for them in some (unspecified) situation. It might well be the case, therefore, that summaries which are rated highly in such an evaluation could turn out to be less useful in a real application, or vice versa. It may be particularly difficult for the readers to judge summaries impartially if, as is likely, they are aware that the summaries have been produced automatically. To overcome these problems, we must evaluate summaries by actually using them; this is discussed below in *task-based evaluation*.

There is also a practical problem: to carry out an extensive evaluation of this sort is expensive, requiring readers (who may have to be experts) to make many time-consuming judgements of individual summaries. To obtain meaningful results, it may be necessary for many readers to evaluate summaries for a large number of documents, which will further increase the cost. Perhaps for these reasons, no large-scale evaluations of this kind have been carried out.

3.3 EVALUATION BY TARGET-COMPARISON

Direct evaluation, as discussed above, relies on each summary being read and judged by at least one reader. In a situation where we want to compare the performance of a summarising system using many different settings, or where we want to compare many different systems, such an evaluation may be impractical. It would be much easier if the number of human judgements required per source text did not increase with the number of different systems to be evaluated. Target-based evaluation is the obvious solution to this problem. Instead of judging summaries directly, a person reads the source text and from it produces a *target*. The target itself is not necessarily a summary to which the systems being evaluated should approximate. It need not satisfy the same use and output factors, and although a target could be a summary in itself, it could equally be simply a set of key concepts, words or sentences which a good summary should include or refer to, or, for example, a set of questions one might want to be able to answer given a good summary. What is important is that the summaries to be evaluated can be compared automatically or at least quickly and relatively objectively to the target – i.e. we can measure whether a summary hits the target. In some cases, we may be able to produce a numerical measure of summary quality.

Target-comparison is not just a way to reduce the number of human judgements required to evaluate summaries. It is also a way to make evaluation criteria more precise, by instantiating them for each individual text. Instead of saying that a good summary must contain important concepts, we specify, for each document, exactly which concepts a good summary should contain. We may also be able to establish baselines from which to measure performance: for example, the percentage coverage of key sentences obtained by extracting random sentences from the text.

Target-based evaluation is evaluation of a system, not a setup (although the target could be based on a hypothetical setup), and, as in direct evaluation, the criteria will usually be intrinsic ones. But target-based approaches involve two assumptions not required for direct evaluation: firstly, that good summaries of a particular text are all in some sense similar, and secondly that the similarity can be captured in a target. The first assumption is itself questionable: Rath, Resnick and Savage (1961), for example, found that when asked to produce summaries by sentence selection, four human judges chose sets of sentences with only 25 percent overlap, and Furnas et al (1987) have shown that people disagree greatly when choosing index terms for even simple texts (recipes, in this case).

Target-comparison has been used extensively in the MUC (Message Understanding Conference) evaluations (ARPA 1993), where the task is not summarising but information extraction. This task is well-suited to this kind of evaluation, because there is a very precise statement of exactly what kinds of information should be extracted from the text, and therefore it is relatively clear

for each document what the target should be. In summarising, however, it may be not at all clear just what should be included in the target.

The second assumption above is also problematic: not all desirable properties of a summary can be captured in a target. Whereas criteria of importance or representativeness can be approximated as, say, coverage of key sentences or concepts, it is difficult to see how one could do something similar for criteria of cohesiveness or clarity. So typically, target-comparison evaluation is restricted to issues of summary content, not its expression.

3.3.1 *Examples of evaluation by target-comparison*

One of the evaluations carried out by Miike et al (1994) of their Japanese-language summarising system is a good example of the target-comparison approach. The experimenters went through a collection of test documents (newspaper editorials and papers from Toshiba Review) choosing key sentences which they thought should be included in any summary produced by sentence extraction. The system then produced summaries of the documents of various lengths, and the percentage of the key sentences that were covered was computed. Each story could have several key sentences, so for each a 'most important' key sentence was also chosen, and coverage of these computed too.

Miike et al attempted to compensate for the potential variation in key sentences picked out by human judges, by having such sentences chosen by three people for each document. What they did not avoid, however, was the possibility that for some texts, there may be many sentences each of which is equally suitable for inclusion in a summary, and all of which convey the same crucial information. Their method did not allow for such 'alternative' key sentences.

These problems were addressed to a limited extent in a target-based evaluation by Salton et al (1997). Their system was designed to process longer documents, and produce summaries by extracting whole paragraphs of source text. For each document, two readers chose which paragraphs were 'most important for summarising the article'. The performance of the summariser was then measured by considering the amount of overlap between the automatic and the manual extracts. Salton et al proposed four different ways of evaluation: *optimistic* (select as the target the manual extract which has the most overlap with the automatic one), *pessimistic* (select as the target the manual extract which has the least overlap with the automatic one), *intersection* (construct a target by taking those paragraphs contained in both manual extracts) and *union* (construct a target by taking all paragraphs contained in either of the manual extracts).

Kupiec, Pedersen and Chen (1995) used a kind of target-based evaluation in which the target was not a manual extract, but a professionally-produced abstract, albeit one in which summary sentences were 'inspired by particular sentences in the original documents'. A combination of automatic and manual techniques was used to classify sentences in the manual summaries, according to

whether their content was derived from one source sentence (a *match*) or several (a *join*), and whether they expressed exactly that content (*direct*) or added or removed information (*incomplete*). Some summary sentences did not correspond to any combination of source sentences, and were labelled *unmatchable*. Two measures were introduced: the first was 'the fraction of manual summary sentences that were faithfully reproduced by the summarizer'. Since only manual summary sentences classified as direct matches or direct joins correspond exactly to source sentences, a score of 100% was not always possible with this measure. Further, since a direct join was only considered to be faithfully reproduced if the summariser selected all the corresponding source sentences, there was no benefit in selecting some but not all of the sentences involved in a direct join. Thus this was quite a demanding measure. The second measure was the fraction of all the matchable source sentences that were selected, where matchable sentences are those sentences involved in a direct match or direct join. This measure is more lenient, as it does not require all the sentences involved in a direct join to be selected to contribute to the score, and it ignores incomplete and unmatchable summary sentences. Kupiec, Pedersen and Chen used these measures to train their summariser as well as to evaluate it: the results suggest – in agreement with Edmundson (1969) – that a combination of location information, cue phrases and sentence-length scores can give better results that any one of these methods. However, these experiments used only one target summary per source text, and thus did not address the issue of alternative summaries that might be equally acceptable.

A different kind of target-comparison evaluation was carried out by Paice and Jones (1993). Instead of key sentences, they identified relevant concepts in the text, classifying them as *focal* or *non-focal*. For each summary, percentages were calculated of the focal and non-focal concepts of the source text that were included in the summary. To say whether a summary includes a concept or not is clearly not a simple algorithmic process, so this kind of target-comparison evaluation requires human decisions to be made not just in drawing up the targets (the lists of focal and non-focal concepts) but in saying how nearly they have been hit. Nevertheless, the work required is substantially less than would be involved in judging the quality of each summary directly, and the results may be more reproducible.

Johnson et al (1993) propose a similar but more sophisticated approach to evaluation. A template, reflecting the discourse structure of the source text, and divided into sections such as aim, methods and findings, is used to identify concepts in the text, which are then assigned numerical scores indicating their relative importance. The measure of summary quality used is simply the sum of the scores of concepts included in the summary. This type of evaluation is interesting, not least because, as suggested by Johnson et al, it may be applied to professionally produced abstracts as well as automatic ones (evaluation by measuring coverage of key-sentences is, in contrast, only applicable to summaries produced by sentence extraction).

These examples of target-comparison evaluation have all been of the

black box kind. Although measuring coverage of key sentences assumes that summaries are produced by sentence extraction, such evaluations use no knowledge of how relevant sentences are to be chosen. In contrast, the evaluations of DeJong's FRUMP summariser (1979, 1982), also an example of target-comparison, were of the white box type. That is, they used internal criteria: rather than measure whether the system produced good summaries, they measured to what extent the system did what it was supposed to.

DeJong describes two evaluations of FRUMP, in each of which the system was run in real time on UPI news stories. In one the system ran for a day, using 48 sketchy scripts. The second test involved running FRUMP for six days, but using just seven of the scripts. In each case, the stories were classified according to whether FRUMP had processed them correctly or not, i.e. firstly, whether it had selected an appropriate script, and secondly, whether it had instantiated script variables with appropriate values. The target with which the story representation (not the eventual summary) was compared was thus a script instantiated as fully as possible by a human reader.

These evaluations reveal much about the extent to which natural language processing of real-life text could (at the time) be made to work, in the sense of obtaining parses and semantic representations of selected parts of the source text, and generating new surface text from them. They do not, however, tell us directly whether FRUMP produced appropriate summaries. For this, we have to rely on the (in this case, quite plausible) assumption that the scripts really are good recipes for summaries, and also on the assumption that FRUMP was able to adequately express the content of the summary representation in the summary text.

3.4 TASK-BASED EVALUATION

The direct and target-based approaches to evaluation discussed so far have been of summarising systems alone, not of systems operating in a particular setup. Because of this, they all suffer from a central problem: even if summaries score well in acceptability ratings or by comparison with summary targets, does this mean they will actually be helpful in any given situation? To gain this kind of assurance, we need to evaluate not just a summarising system, but its operation in a particular setup. Rather than attempting to measure qualities of the summaries directly, we instead measure the ability of users to perform some task in which the summaries should help them. What the setup should be depends of course on what use we are intending to put the summarising system to. One particular type of use, however, is especially appropriate, because it represents a real use of summaries and because it is an area in which there is already much experience in how to carry out performance evaluations. That area is information retrieval, and the use of summaries by users to determine which retrieved documents are likely to be of relevance to their queries.

3.4.1 *Examples of task-based evaluation*

Miike et al (1994) carried out a (very limited) evaluation of this kind, in addition to the target-comparison evaluation mentioned above. They took a set of queries, and for each one retrieved ten documents from a database of newspaper editorials. Five users then had the task of selecting those documents from among the retrieved documents that were relevant to the query; each user was either given the full text of the documents, or given the summarising system, which they could ask to produce longer or shorter summaries as required. Of course the users did not have a real need for these documents: the evaluation therefore involved not a realistic task, but a simulation of one.

Miike et al reported that accuracy at the task, measured in terms of precision and recall, was much the same for the users with the summariser and for those with the full documents, but the time taken to identify the relevant documents was twenty percent less when using the summariser.

At first glance, this evaluation may seem rather on the small side (the paper does not state how many queries were processed, but if the corpus of documents consisted only of those summarised in the target-comparison evaluation then there could have been at most thirty texts). However, this evaluation is in fact remarkable for having been carried out at all, as there are many difficulties with carrying out this kind of experiment.

Task-based evaluation in TIPSTER

Phase III of TIPSTER (which includes the Message Understanding Conference (MUC) and the Text Retrieval Conference (TREC)) will include an extensive task-based evaluation of summarising systems (Hand 1997). Initially, two tasks are to be investigated: subject categorisation, and judgement of relevance to a given topic. In each case the experimental method simulates a real information retrieval situation. An IR system is used to retrieve the documents to be summarised. For the categorisation task, participants are given the documents and produce indicative, neutral summaries from them. For the relevance judgement task, participants are given the documents and the queries used; they thus have the opportunity to produce indicative summaries focused on the query topic.

For each task, a number of 'professional information analysts' read the summaries or documents (no one is given more than one summary of the same document) and must categorise or judge the relevance of the document, as appropriate. Their decisions are compared to the 'correct' answers, as determined by a further two assessors who read the documents in full.

The *measures* for these evaluations are: the accuracy of the assessments, the time required, the length of the summaries, and the assessors' expressed opinions as to the suitability of the summaries.

Once again, these are not complete evaluations of a setup, because although the tasks are realistic, the operational context is not: users perform the

task for the purpose of evaluating the summaries, rather than because they really want to find relevant documents. Nevertheless, the evaluation method is carefully designed to simulate real tasks, so the results should tell us a great deal about the potential utility of current summarising technology.

3.4.2 *Difficulties with task-based evaluation*

Although other experimenters have certainly been aware of the potential value of evaluating summarising systems in a realistic setup (for example, Paice and Jones (1993) discuss the possibility among several others), they have not often carried out such evaluations, for several reasons.

The first problem is one of scale: to obtain meaningful and realistic measures of performance in an information retrieval experiment we would require a very large number of documents and many users, with many queries, to achieve sample sizes large enough to overcome random variation. We would then need either a supply of alternative summaries for each document, or a supply of similar documents with different summaries. It may simply not be feasible to prepare summaries for all the documents in the collection, or to produce them as documents are retrieved, perhaps because the summariser may be too slow, or because it is restricted to processing texts of a particular length, form or style, or concerning a particular topic.

A second, more fundamental problem with using an information retrieval task to evaluate summarisers is that it imposes very few constraints on the summaries, and so very different summarisers might perform equally well. The results of such an evaluation say very little about how well summaries meet other requirements such as coherence and readability – in this sense information retrieval is an undemanding task. To measure the extent to which summaries meet these criteria, we would need to carry out additional evaluation, using other tasks (for example, a comprehension test: measure users' ability to answer simple questions about the text having read the summary).

In addition to these problems, task-based evaluation may not be very helpful in determining what features of a summary are of value or are not of value, and so it may be difficult to use the results of such an evaluation to improve summary quality, presumably a secondary goal of many evaluations.

3.5 AUTOMATIC EVALUATION

The cost of task-based evaluations would be reduced if they did not involve so many users. Perhaps there are potential tasks for which a summary is useful, but rather than being read by anyone, is used by another program or system in carrying out a task. Given such a setup, and automated measures of how well the 'user' programs complete their tasks, we could completely automate the process of summary evaluation, making it cheaper and faster while retaining some of the advantages of a task-based evaluation. This approach was taken by Brandow, Mitze and Rau (1995) who, in addition to an evaluation by direct

judgement of summary acceptability, used the summaries produced by their system as the collection of *documents* to be searched by an information retrieval engine. They took twelve queries which they considered to be typical and used them to retrieve documents from a database of around 20 000 documents of various types, both by matching against the full text and against summaries produced by ANES and the initial-sentence(s) method. Users classified all the retrieved documents as relevant or not relevant, and precision and recall were used as measures of performance. The experimenters found that both kinds of summary improved precision significantly, 'but at the cost of a dramatic loss in recall.'

As large collections of texts, queries and relevance judgements are now available (e.g. from the TREC conferences (Harman 1993)), this kind of test can be carried out quite cheaply on a large scale. However, it is not clear that there is any reason to expect the summaries which are most appropriate for such tasks to be most suitable for human use. Indeed, we might well expect a list of key words or phrases to perform well in such an evaluation, but to be less helpful for human users. In this respect, automatic evaluation suffers from some of the disadvantages of target-based evaluation: as the summaries are not read by human users, evaluation will be poor at measuring readability (typically deemed to be important); and unless the task involves complex language processing (which in the case of information retrieval it usually will not) automated evaluation will not take the content of the summary into account except insofar as it is reflected in the content words used and their locations in the summary.

3.6 CONCLUSIONS

Rather than being evaluated using one of the four sorts of approach discussed above, summarising systems have often been given no evaluation beyond that of the experimenters looking at some sample output summaries and commenting on them. In a sense, this is evaluation by direct human judgement, but often no formal evaluation was performed, only a few texts considered, and no numerical measure of summary quality produced. Even the evaluations described above were mostly quite small-scale: evaluations involving fifty texts or more have been something of a rarity.

There are several possible reasons for the lack of emphasis on evaluation. Evaluating summaries is likely to be time-consuming and may be expensive, and design constraints and practical constraints on the operation of summarisers (length of source text, subject matter, vocabulary used, need for human intervention during processing etc.) make evaluation, and especially comparative evaluation of different systems, particularly difficult. In some cases, the objectives of summarising systems, and hence the criteria by which the summaries should be judged, are not clearly defined. It is also possible that some researchers judged the performance of their summarising systems too poor to warrant large-scale evaluation, and instead focused their efforts on improving the systems.

Combining the approaches

When evaluation is to be carried out, it may be worthwhile to combine two or more of the four kinds of evaluation. If, for example, a task-based or target-comparison evaluation is to be carried out to assess summary content, a direct evaluation could be added to investigate presentational criteria which are inadequately measured by the first method. Such a direct evaluation need not require readers to consult original source texts, and therefore could be quicker and cheaper than a direct evaluation aimed at measuring content criteria. Alternatively, if a mixture of white- and black-box evaluation is required, then a target-comparison white-box evaluation of system behaviour (as in FRUMP) could be combined with a direct or task-based black-box evaluation of summary quality.

Evaluating CLASP

For my summarising system, CLASP, introduced in the next chapter, practical constraints ruled out large-scale evaluation in the time available: the linguistic analysis of the source text is very slow, and can take up to several hours to process a single sentence. So a small evaluation was carried out involving twenty source texts (stories and articles from the *Wall Street Journal*). Because I wanted to measure the performance of the system using several different parameter settings, I chose to evaluate it by a target-comparison method, comparing summaries produced by sentence extraction with sets of important sentences chosen by human readers. By using simple sentence-extraction, we can also compare CLASP's summaries to those produced by other, less linguistically sophisticated, summarisers. CLASP, in addition to sentence-extraction summaries, can also produce lists of short summary phrases for each source text. These I have not evaluated at all formally; rather I give a discussion of various issues of content and expression in these summaries.

The experiments and evaluation carried out with CLASP are described in full in chapter 8, after the system itself is described in chapters 4–7.

4 A NEW SUMMARISING SYSTEM

CLASP is a new summariser that uses a shallow semantic representation of the source text. In the broad categorisation of chapter 2, it is an attentional-network summariser, but one that uses a deeper source representation than those presented there. In this chapter, I explain and justify the main design decisions made in developing this summariser, in terms of the 3-stage model of chapter 1 and with reference to the discussion of previous systems and methods in chapter 2. A more detailed account of the processing involved in the three stages of analysis, condensation and generation is given in chapters 5, 6 and 7.

Section 4.1 describes the goals of the system, the kind of summaries to be produced and the natural language processing technologies available. Then, in section 4.2, I discuss my choice of source representation, its interpretation, and the implications of its use for summarising. The source representation consists of *cohesive links and simple predications*, hence the name of the system, CLASP. Section 4.3 turns to condensation, and explains how graphical techniques can be applied to the source representation to select content suitable for summary purposes. Finally, section 4.4 outlines CLASP's approach to generating summary text, describing its two methods of generation and how they may be used to produce different kinds of summaries.

4.1 GOALS OF THE SYSTEM

My aim in creating CLASP was to address the question of how linguistic processing (and specifically, the use of a semantic source representation) might be of help in automatic summarising of real-world texts. In the terminology of section 2.5.2, CLASP attempts to be general-purpose both in the sense of not requiring source texts to be of a particular form or about a particular subject matter, and in the sense of producing reflective summaries, rather than ones targeted at a specific user or a specific area of interest.

There are two chief benefits which we might hope to gain from using a source representation deeper than surface text. Firstly, summarising is inherently concerned with the content of a text, not just its surface form. We should be better able to reflect the salient content of the source text in the summary if we represent at least some of this content explicitly: we can then select salient material directly, rather than selecting sentences or other surface text units which we hope contain salient content. In short, we hope to identify more reliably what is important and what is unimportant for summary purposes.

Secondly, as noted in section 2.5.3, an analysis of the source text which breaks sentences down into their constituent parts at the semantic level allows us to be more *precise* when selecting summary material. Specifically, individual sentences of source text may contain a mixture of important and unimportant

information, so we would like the basic units of the source representation to be smaller than the semantic representations of whole sentences.

We can describe the requirements for CLASP in terms of the context factors described in sections 1.1.1–1.1.3.

4.1.1 *Input factors*

We will only consider what I called *basic texts* in chapter 1. That is, written English prose, without graphical material or tables and with no headings, subheadings or other internal divisions except paragraphs. This decision is made because we are interested in fairly general-purpose summarising methods, and such presentational factors vary greatly from one form of text to another, so methods which rely on them for summarising will most likely be very dependent for their success on the form and genre of the source text. We place no restrictions on the *subject matter* of the text, but it must be intended for the general (adult, educated) reader and not use large amounts of specialist vocabulary or jargon (though such terms could be added to the lexicon). Finally, the primary *communicative aim* of the source text will be to convey information (facts, and in some cases opinions) to the reader. This last restriction is the most severe, as to convey information is not the primary purpose of some texts: fiction and humour, for example, have other communicative aims. Even among scientific papers this is not the only communicative aim: many texts seek not just to express information, but to argue for a particular viewpoint and persuade the reader of it.

In addition to the above theoretical restrictions, there will be another constraint, made for practical reasons. Because automatic semantic analysis is time-consuming, it will not be possible to process very long texts; therefore we will consider short texts of up to 1500 words only. This in itself has consequences for summarising: short texts require a lesser degree of shortening to make a summary, and are more likely to have only a single theme than longer texts. We will see in chapter 8, however, that even among texts of this length there are examples with two related themes, or a variety of interconnecting points but no clear central theme.

Newspaper stories are a good example of a basic text which fits our length requirements, and in particular a number of texts from the Wall Street Journal are used in evaluating CLASP in chapter 8, and as a source of examples in this and later chapters. The stories used are mainly factual descriptions of current events and business news, but also include arts reviews and longer 'feature' articles on specific topics. Because they are written to convey a large amount of information in a small space, these texts tend to be grammatically quite complex and to have many long sentences of thirty words or more.

4.1.2 *Purpose factors*

CLASP's summaries are intended to indicate to the general reader the topic and

the main concerns of the source text, without being focused on any particular topic or subject area within the text. That is, we would like the summaries to be reflective of the source text in terms of subject matter, but not in terms of communicative aims: it need not convey the salient information in the source text, merely say what it is about. We do not specify any particular use for such summaries, but envisage that they might for example be useful for *retrieving* or *previewing*. In short, CLASP's summaries should be indicative, neutral, broad (that is, wide in scope) but not deep, and will involve shortening both of content and of expression. This statement of purpose factors reflects an intuition that producing indicative summaries may be easier than producing informative summaries, and a belief that indicative summaries are of use for many applications.

4.1.3 *Output factors*

CLASP's summaries will reflect the source text in readership and in the language and vocabulary they use. However, as the summaries are to be indicative rather than informative, the other output *form* factors (as defined in section 1.1.3) need not reflect those of the source text. In particular, there is no requirement that the summary consist of full English sentences integrated into a cohesive text.

This decision was made for two reasons: firstly, the main emphasis in CLASP's synthesis stage is on producing a summary that captures the content of the summary representation, without inadvertently misleading the reader into believing that the source text says something that it does not. Although some consideration has been given to making summaries concise, I have concentrated less on presentational aspects of the synthesis stage, such as syntactic cohesion. Secondly, CLASP is intended to be robust, and thus we would like to produce a summary even if we are only partially able to analyse the source text. In such a situation, it is very unlikely that we could generate a well-integrated, cohesive text of a sophisticated kind without the use of world-knowledge in the representation or as an aid in generation.

In practice, CLASP is able to produce two different kinds of summary: a running text consisting of whole sentences extracted from the source text, and a list of short 'summary phrases' which indicate the main topics of the text. These are both compromises, and the choice between the two will depend on exactly what we intend to do with the summary. Generating summary phrases allows the system to focus more precisely on individual topics in the text, whereas summaries produced by sentence extraction, as well as being more readable, will prove to be useful in comparing CLASP with other summarising systems.

4.1.4 *Previous use of linguistic processing*

We have seen, in chapter 2, some examples of summarising systems that involve linguistic processing. In considering the extent to which such systems meet the goals set for CLASP, we must ask to what extent they are specific to a particular

application, to a particular form of source text, or to a particular domain. If they make use of domain- or world-knowledge, we can also consider whether this is at the analysis, condensation or generation stages. We must also ask how linguistically advanced is the processing performed by each system, and how deep the source representation is (as suggested in section 1.2.1, these two are potentially independent).

A number of summarising systems have been designed for a particular application – that is, a particular form and subject matter of source text and a particular set of summary requirements. For example, TOPIC (Hahn 1990) is able to construct quite deep source representations by relying on extensive domain knowledge, including a specialised lexicon; domain-knowledge is used again in synthesising summary text. However, the central ideas behind TOPIC's condensation stage are not specific to the domain, although they are particularly appropriate to descriptive texts. Other application- and domain- specific systems that rely on domain-knowledge to perform linguistic processing include SCISOR (Rau, Jacobs and Zernik 1989), TESS (Young and Hayes 1985), and TICC (Allport 1988).

The summarising systems SAM (Schank and Abelson 1977, Cullingford 1981), SCRABBLE (Tait 1983) and FRUMP (DeJong 1982) all used scripts (a representation of some kinds of world-knowledge) in the analysis, condensation and synthesis stages. In SAM and SCRABBLE, a quite sophisticated linguistic analysis was performed to build a deep source representation; this meant that in practice only sufficiently simple source texts could be processed. FRUMP's representation was constructed with much simpler processing; this was made possible by using the script to predict the kinds of information that would be found in the text, and focusing the analysis on looking only for those kinds of information. FRUMP was thus able to process much more difficult and complex texts, in particular stories from the Wall Street Journal.

In contrast to TOPIC, a truly domain-specific system, FRUMP and the other script-based systems were not domain-specific because they were able to choose an appropriate script for the source text. (Although as discussed in section 2.3, there may be some kinds of text which are intrinsically 'non-scripty'.) In practice, however, scripts are not available to deal with every potential subject matter, so script-based systems still rely on the subject matter of the source text falling within a particular range of topics.

Taylor's algorithm (Taylor 1975) did not use world- or domain-knowledge *in the condensation stage,* instead applying general graph-based techniques to a deep source representation. However, to build a summariser using this algorithm would require an analysis stage capable of delivering such a source representation; as this involves resolution of anaphors and ellipsis, and integration of individual sentence analysis into a detailed meaning representation, it is not currently feasible for any but the simplest of texts in restricted domains.

None of these systems were able to use even sentence-level linguistic processing for summarising without imposing restrictions on the content or the

complexity of the source text, either because they were inherently domain-specific, or for practical reasons such as only having a small repertoire of scripts. An alternative approach is that taken by Miike et al (1994) and Marcu (1997), whose systems perform only very basic linguistic processing of individual sentences, but link them together in quite deep representations of (linguistic) discourse structure. These systems operate by looking for surface-text markers which indicate underlying discourse relations. This approach does not require domain knowledge, but does rely on there being sufficiently many of these markers in the source text, which makes it genre- and form-specific.

Considering the other systems described in chapter 2, it seems that the only kinds of linguistic processing which have so far been used in summarising without restricting the form or subject-matter of the source text, are very shallow processes such as the application of WordNet in SUMMARIST (Hovy and Lin 1997), the extraction of phrases and proper names performed in DimSum (Aone et al 1997) and the local salience analysis of Boguraev and Kennedy (1997). None of these techniques allows us to construct a semantic representation of the source text, one of the goals of CLASP.

4.1.5 *Available technology*

Sentence by sentence analysis

Systems are now available which are able, in conjunction with electronic dictionaries, to perform morphological analysis and parsing and to construct semantic representations of the logical-form type for many ordinary English sentences. With such resources, we should be able to process individual source-text sentences to obtain at least a partial semantic analysis. The system I have used is the Core Language Engine or CLE, developed by SRI International (Alshawi 1992), in the CLARE-3 version (Alshawi et al 1992). I chose this system because it was readily available and could be modified if required, because its grammar has a broad coverage of English, because it is able to cope fairly robustly with text that it cannot analyse in full, and because it performs both analysis and generation of surface text. This last feature means that we should be able to generate simple English phrases and sentences for an output summary, if we can construct appropriate logical forms from our summary representation.

Inter-sentence processing

The CLE is able to combine logical forms for individual sentences and resolve some anaphoric references, but these operations are not often successful when we have only partial semantic analyses. Unfortunately, the complexity of the source texts we want to process means that complete and accurate sentence analyses are likely to be rare: therefore, in CLASP, the CLE is not used for reference resolution or other inter-sentential processing.

For forming a representation of large-scale structure, there are few available technologies and it is also much less clear what form the source representation should take. Let us consider some candidate theories of large-

scale structure, and see whether they could be used in CLASP. As discussed in section 1.2.1, we can distinguish between linguistic, intentional, attentional and informational structure.

Linguistic and informational structure

Many theories of linguistic and informational structure (sometimes these two kinds of structure are combined in a single theory) involve various different kinds of relations between facts and events described in a text, or their expression in the text. For example, Kintsch and van Dijk's (1978) theory of *macropropositions* (primarily informational), rhetorical structure theory (Mann and Thompson 1987) (primarily linguistic), and Hobbs' (1990) theory of coherence relations (both informational and linguistic) all involve such relations. Some texts might contain sufficient discourse markers to indicate these relations without any semantic analysis (this is the approach used by Miike et al (1994) and Marcu (1997)), but the occurrence of such phrases is very dependent on the form and style of the text (Brandow, Mitze and Rau 1995) which makes it an unsatisfactory method for our purposes. Without such surface-text clues, to find relations (such as causal, figure and ground and contrastive relations) we would need more complete semantic analyses of individual sentences than we can expect to obtain, and a sophisticated inferencing method (perhaps involving extensive world-knowledge). Attempting this kind of analysis automatically is far from simple, as Sparck Jones (1995) notes.

Scripts, as we have seen in section 2.3, provide a way of representing informational structure that is more amenable to implementation than these theories. (A similar structure, albeit with a simpler underlying representation, is present in MUC templates.) As already noted, however, the use of such systems requires us to limit the subject matter of the source text to the range of scripts or templates we have available, which makes this approach unsuitable for CLASP, and there is a tendency for script-based representations to lead to prescriptive summaries.

Intentional structure

The intentional structure of text is also something we cannot expect to determine easily. To deduce intentions behind utterances in the way required to implement the intentional component of Grosz and Sidner's (1986) theory, for example, it seems we would have to apply background assumptions about the domain and the author, to represent their (and the reader's) beliefs about the world, and to reason about their plans and goals. None of this is a practical possibility for our summarising system.

RST has an intentional component (in its representation of the effects of linguistic structures) that is more closely tied to linguistic structure, so perhaps this kind of intentional structure could be deduced successfully from some texts. As noted above, however, RST-analysis is not feasible for CLASP.

Attentional structure

Attentional structure seems more susceptible to automatic analysis than informational, linguistic or intentional structure. Sidner (1983) has presented an algorithm for tracking the *focus* of a text as it changes from sentence to sentence. Grosz, Joshi and Weinstein (1995) give a broader (though much less algorithmic) analysis of what they call *centering*. In these theories, the emphasis is on a local analysis of attentional properties of discourse, aimed at assisting later processing rather than producing a static representation of attentional structure. Nevertheless, these theories both implicitly construct a representation in which the text is segmented according to changes in focus; by considering previous focuses (the *focus stack* in Sidner's algorithm), we can detect nested segments (at the end of which the focus is 'popped' off the stack) as well as segments following in sequence.

There are, however, substantial difficulties in implementing Sidner's algorithm computationally: in addition to the need to identify various role-fillers or thematic slots (such as AGENT, INSTRUMENT, VERB-COMPLEMENT) in the parsed sentences, to apply the algorithm we must be able to identify all the discourse entities in the text. These can be noun or verb phrases, and within some complex noun phrases there may be further discourse entities to be identified. Gladwin (Gladwin, Pulman and Sparck Jones 1991), considering texts considerably simpler than those (such as Wall Street Journal stories) we want to summarise, noted that the complexity of some of his texts 'effectively prohibited any machine-based implementation of the algorithm.' He also comments that Sidner's algorithm concentrates on the use of pronominal anaphors, and does not cope well with situations where focus is maintained by the use of nominal (or 'implicit') anaphors. In Grosz, Joshi and Weinstein's view, a full treatment of centering must involve 'syntactic, semantic, discourse and intentional factors.'

Halliday and Hasan's (1976) theory of *cohesion* describes a different, looser kind of attentional structure. They describe a range of linguistic devices which can link together the sentences of a text, including *reference*, *substitution*, *ellipsis* and *lexical cohesion*. The computational analysis of reference is, as just noted, a difficult task; resolving substitution and ellipsis seems a much harder one still. But analysis of lexical cohesion (the occurrence of the same or related words in different sentences) is much simpler, as it is a purely surface phenomenon; such an analysis leads naturally to a graph of cohesive relations between sentences, like those used by the attentional-network summarisers discussed in section 2.1.2. From CLASP's point of view such a representation is an attractive one: it does not require advanced and unspecified inference processes or world-knowledge to produce, and the graph-structured representation is a robust one (as discussed in section 2.5.1). However, the linguistic level of these representations is too shallow for CLASP, which aims to use a semantic source representation with a finer granularity.

In summary, although some theories of discourse structure have been automated, and some have been applied in summarising – e.g. Marcu's (1997) use of RST – the only theory that has been successfully applied to give a general-purpose, responsive and automatic analysis of discourse structure is Halliday and Hasan's (1976) theory of surface-level *lexical cohesion*, as used in TELE-PATTAN (Benbrahim and Ahmad 1994).

CLASP therefore requires a new kind of representation of large-scale structure, one that applies at the level of semantic representations, yet can be applied robustly to a wide range of source texts. The solution adopted, as described in the next section, is an attentional network that links predications mentioned in the source text.

4.2 THE SOURCE REPRESENTATION

CLASP's source representation is a *predication cohesion graph*, in which the nodes are *simple predications* (section 4.2.1) mentioned in the source text, and edges correspond to *cohesive links* (section 4.2.2) between them. The simple predications describe properties of and relations between *entities* (objects, events and states mentioned in the text), and many such predications will be obtained from the analysis of each source sentence; therefore this is a semantic representation with a fine granularity. The cohesive links are a semantic analogue of lexical cohesion (Halliday and Hasan 1976): they link predications which concern the same or similar entities. Therefore, a cohesive link between two predications is a link in aboutness, and the source representation as a whole is an attentional network, like those discussed in section 2.1 but at a deeper linguistic level.

4.2.1 *Simple predications*

Each of CLASP's simple predications is intended to correspond to one idea or piece of information in the text. For example, consider the following sentence (hereafter referred to as S-JAP): 'Japanese investment in Southeast Asia is propelling the region towards economic integration.' It conveys the following information:

1. investment is propelling Southeast Asia towards integration,
2. the investment is in Southeast Asia,
3. the source of the investment is Japan,
4. the nature of the integration is economic.

A conventional semantic representation for this sentence (of the logical form type) does not explicitly separate these four facts, but integrates them into a single proposition for the whole sentence. Nevertheless, a summarising system which could distinguish between them might decide that points 3 and 4 are relatively unimportant compared to points 1 and 2, and announce in its

summary: 'Investment in Southeast Asia is propelling the region towards integration.' If instead the system decided that it was items 2 and 3 that were important, it might instead write: 'Japan is investing in Southeast Asia.' Thus the division of the sentence into these four points allows a more precise identification of important material.

To rewrite points 1–4 above in a more formal way, we will express each one as an atomic predicate with one or more atomic arguments, which may be constant values or 'variables' representing discourse entities (things, states and events referred to in the text). I will use capital letters (Prolog-style) for these discourse entities.

In S-JAP, we have five discourse entities: the investment, **I**; Japan, **J**; Southeast Asia, **A**; integration **K**; and the event of propelling **P**. The four points listed above become the following four simple predications:

```
1. propel(P,I,A,K)
2. in(I,A)
3. source(I,J)
4. economic(K)
```

In addition to these, we will also have simple predications which express the identities of the discourse entities involved, i.e. that **I** is investment, **J** Japan etcetera:

```
5. name_of(J,Japan)
6. name_of(A,Southeast_Asia)
7. investment(I)
8. integration(K)
```

(That the entity **P** is a propelling event is already expressed by predication 1 above.)

Typically, as in this example, each content word – that is, each noun, proper noun, adjective, verb or adverb – will give rise to a simple predication, and so will each use of a prepositional phrase or other adverbial or adjectival construct. In more complex sentences such as those involving coordination, words may yield more than one predication each.

Simplification of examples

All the examples of simple predications in this chapter, including those just given, are slightly simplified from the actual predications used by CLASP. There are two main differences.

Firstly, I have omitted sense information from the predications. The CLE has sense readings for words in its core lexicon: for example, the predicate corresponding to 'economic' would be **economic_Financial**. For words with

more than one sense, the CLE performs disambiguation to select an appropriate predicate. However, multiple senses are the exception rather than the rule in the predications obtained with CLASP, since the external lexicon used has no sense information. In this chapter, therefore, it will be simplest to pretend that no sense disambiguation at all is performed, and that for each content word there is simply a predicate of the same name. Thus, I have reduced **economic_Financial** to **economic**.

Secondly, CLASP does not perform any anaphor resolution, whether of pronouns or definite noun phrases. In the examples given here, I have not assumed any inter-sentential anaphor resolution; however, in the analysis of S-JAP to obtain the four simple predications above, I have pretended (unrealistically) that CLASP knows that 'the region' refers to the entity introduced by the noun phrase 'Southeast Asia'. I have done so on the grounds that CLASP's inability to resolve anaphors is an unfortunate practical constraint, not a design decision, and the source representation certainly does not depend on it. (Although as we will see in section 4.2.2, CLASP's strategies for spotting cohesive links are designed to compensate for this limitation in some cases.) The lack of anaphor resolution makes my use of the term 'discourse entity' unconventional, since each entity appears only in predications coming from one sentence.

There are additional simplifications in the treatment of S-JAP. Firstly, I have assumed a 4-place predicate **propel** whose arguments are the event of propelling, the subject, the object, and whither it is propelled. In fact, CLASP's analysis would express this in two simple predications: **propel(P,I,A)** and **toward(P,K)**. Secondly, I have assumed the CLE analyses this sentence correctly,

which, as we will see in the next chapter, it does not. For now these differences are relatively unimportant. The predications actually produced by CLASP's analysis of the S-JAP sentence are shown, without these simplifications, in section 5.3.

Robustness

As well as allowing finer-grained selection of important content, there is a practical benefit in having a representation which allows us to split up sentences into individual predications. Because we do not represent a whole sentence in a single unit, we can obtain simple predications from sentences for which we can perform only a partial, fragmentary semantic analysis. Such sentences will be very common in complex real-world texts, so this robustness is a vital feature of CLASP's processing.

Interpretation of simple predications

It is tempting to interpret simple predications 1–8 above as a set of facts all of which are asserted by the source sentence, with each discourse entity existentially quantified: 'There exist **I** and **J** such that **J** is called "Japan" and **I** is some investment and **I** comes from **J** and …'. However, such an interpretation, though it happens to be true for this simple example sentence, will be invalid in

general. The reason is twofold: firstly, the representation does not specify the quantification of the entities involved; secondly, there is no representation of the higher-order structure into which the individual predications must fit. As an example, consider the sentence S-JAP, but with the addition of the preamble 'A recent report states that'. From the new sentence, CLASP would still extract predications 1–8 above (plus some more), yet the new sentence does not assert the truth of any of these predications, only that a 'recent report' does so. The reported speech construction corresponds to a logical structure which is not just a conjunction of simple predications. The same is true of sentences involving universal quantification, conditionals, modal constructions, negation, or any of several verbs which take sentential complements (e.g. 'doubt that', 'believe that').

This does not stop us from representing complex sentences as a number of simple predications in a manner sufficient for our purposes, but it does mean that we cannot expect the truth of a sentence to imply the truth of the simple predications extracted from it. *Therefore, instead of trying to interpret simple predications as facts asserted in the source text, we will interpret them only as ideas which are mentioned in the source text.* This is a vital fact about the source representation, and shows that it represents (some aspects of) attentional content, rather than informational content, i.e. what the text is about, not what it says about it.

Consequences for summarising

These considerations will limit the kind of summaries we are able to generate, if we are to generate surface text from simple predications. For example, if predications such as **`in(I,A)`**, **`name_of(A,Southeast Asia)`** and **`investment(I)`** are extracted from the source text, we can say in a summary that the text says something about investment in Southeast Asia; if in addition we have the predications **`integration(K)`** and **`propel(P,I,A,K)`** we can say that the text says something about investment in Southeast Asia propelling the region towards integration. But we cannot say, just from looking at these predications, that investment in Southeast Asia *is* propelling the region towards integration – the same predications could have been extracted from a story which denied or debated this proposition, rather than simply stated it. So when it comes to generating summary text from simple predications, we will only be able to say what the text is about, not what it says about it; i.e. we may be able to produce *indicative* summaries, but not *informative* ones.

4.2.2 *Relations between predications*

Having processed the sentences of the source text to obtain corresponding sets of simple predications, we need to integrate them into an overall source text representation. Since CLASP's simple predications represent attentional content, we will identify attentional relations between them, called *cohesive links*. These links are intended to reflect similarities in aboutness between two predications

– connections between topics in the text rather than links in meaning; they are unrelated to the order in which the predications appear in the source text, and do not keep track of local attentional focus, as, for example Sidner's (1983) theory does. Later, in the condensation stage, edges between nodes in the graph will be assigned weights on the basis of the number and type of these links. The attentional structure formed by the simple predications and their cohesive links, we call a *predication cohesion graph.*

CLASP's idea of aboutness is this: a simple predication is 'about' each of its arguments, and additionally 'about' the predicate itself. Two predications will be cohesively linked if they are 'about' the same or *similar* (as defined below) entities. The relations identified in this way are not like those of Hobbs (1990): there are no truth-conditional criteria for them to be present, and no meaning assigned to them. Instead, CLASP's cohesive links are more like an analogue of lexical cohesion (Halliday and Hasan 1976). Lexical cohesion connects sentences with the same or similar words in them; our relations connect predications with the same or similar arguments. Of course when two predications are related in this way, there may well be a stronger connection between them, just as when sentences are linked by lexical cohesion there is likely to be an underlying semantic connection. The point is that we are unable to analyse these deeper relations automatically, so we will have to make do with the shallower links.

To illustrate CLASP's cohesive links we need a slightly larger example. Consider the following two sentence fragments, taken from a Wall Street Journal film review (text LANEFILM in appendix A):

F-LANE example fragments:
1. 'a film about a sketch artist, a man of the streets'
2. 'Mr Lane has revived his artist in a full-length movie'

Analysis of these fragments yields the following simple predications:

F-LANE 1:
```
1. film(F)
2. artist(A)
3. about(F,A)
4. sketch(S)
5. nn(A,S)
6. man(M)
7. about(F,M)
8. streets(T)
9. genitive(M,T)
```

F-LANE 2:
```
10. name_of(L,Mr Lane)
11. artist(B)
12. revive(E,L,B)
13. movie(V)
14. full-length(V)
15. in(E,V)
```

(The **nn** predicate indicates a noun-noun compound; the meaning of the combination of two nouns – in this case, 'an artist who makes sketches' for 'a sketch artist' – is not something we can determine automatically.)

Identity links

We say that two predications are cohesively linked if they have identical predicates, or if one (or more) of their arguments are the same. These are *identity links*. Thus, in the predications from F-LANE 1 above, there are links between predications 3–7 (predicate **about** and first argument **F**); 1–3, 1–7 (**F**); 2–3, 2–5, 3–5 (**A**); 4–5 (**S**); and 6–7, 6–9, 7–9 (**M**). In the predications from F-LANE 2, there are links 10–12 (**L**); 11–12 (**B**); 12–15 (**E**) and 13–14, 13–15, 14–15 (**V**). In general, as in this example, there will be many such links between predications coming from a single sentence, but very few between predications from different sentences, because the entities involved will be different (as CLASP does not resolve anaphors). In this example, there is just one inter-sentential identity link, between predications 2–12 (predicate **artist**).

Similarity links

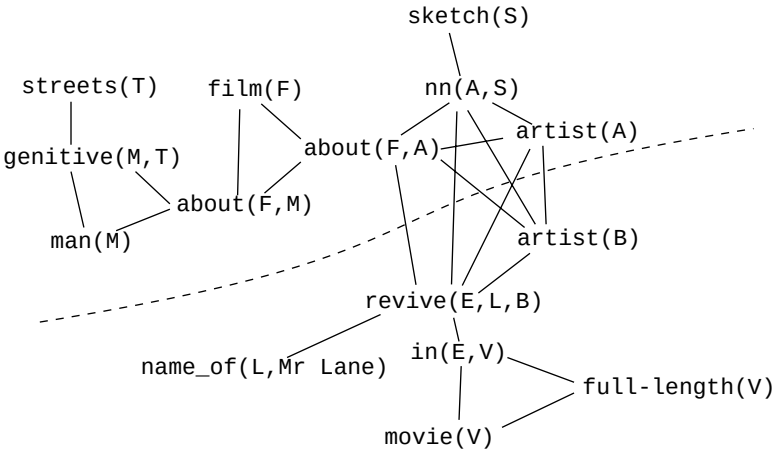
To find more inter-sentential links, we consider predications with arguments that are not identical, but only *similar*. To establish which entities are similar, CLASP defines a *semantic head* for each entity corresponding to its head noun or verb. For the F-LANE fragments, for example, we have the following semantic heads:

```
F  film              L  Mr Lane
A  artist            B  artist
S  sketch            V  movie
M  man               E  revive
T  streets
```

We then define two entities to be similar if they have the same semantic head. In this example, **A** and **B** are similar, and there are similarity links between predications involving them: 2–11, 2–12, 3–11, 3–12, 5–11, 5–12.

Just as the description of simple predications has been simplified in this chapter, so too has the description of cohesive links. As explained in full in chapter 5, CLASP has seven different types of cohesive links, and there can be multiple links between the same two predications. All the links, however, arise from the same or similar predicate or arguments occurring in predications, and they are all undirected (i.e. symmetrical) relations. In chapters 5 and 6 I describe in how the different kinds of links are dealt with; for the examples in the rest of this chapter we will treat them all the same.

We can show the final source representation as a graph, in which each node is a simple predication, and each edge indicates a cohesive relation between predications. The graph for the F-LANE example is given in figure 4.1.



Semantic heads:

A	**artist**	**L**	**Mr Lane**
B	**artist**	**V**	**movie**
E	**revive**	**S**	**sketch**
F	**film**	**T**	**streets**
M	**man**		

Figure 4.1: simplified predication cohesion graph for the F-LANE example. The analysed text consists of the two phrases 'a film about a sketch artist, a man of the streets' and 'Mr Lane has revived his artist in a full-length movie.' (The dotted line separates the predications from the two phrases.)

4.2.3 *Comparison with other representations*

CLASP's source representation has a strong formal and structural similarity to the attentional networks used by Skorochod'ko (1971), TELE-PATTAN (Benbrahim and Ahmad 1994) and Salton et al (1994). In those systems, nodes represented segments (sentences or paragraphs) of source text, and links between them indicated the appearance of the same or related words in two segments. CLASP's source representation is also an attentional network, but at a different linguistic level (semantics rather than surface text) and of a different granularity (there will be many more simple predications than surface sentences or paragraphs). Just as the surface-level attentional networks link sentences containing the same or related words, CLASP's predication cohesion graph links predications which involve the same or similar arguments, or the same predicate. In CLASP as in the other systems, there is an underlying intuition that the pattern of links reveals something about what is most salient in the source text.

Because it is a semantic representation of attentional structure, CLASP's source representation might seem similar to that of the TOPIC system (Hahn 1990) (section 2.4). In fact, however, they are very different. Firstly, TOPIC

attempts a much fuller analysis of the informational content of the text than CLASP's simple predications, which, as we have seen, do not have a truth-conditional interpretation. Secondly, TOPIC's analysis of attentional structure is concerned with local tracking of theme and focus, and therefore directly related to the sequential nature of the source text: it considers attentional change, not the overall weak attentional structure represented by CLASP.

Another method with a source representation very different from CLASP's is the condensation algorithm of Taylor (1975) (section 2.3). He used a deep semantic representation of text meaning: a graph in which nodes corresponded to discourse entities and events, and edges indicated semantic relations between them. These relations, unlike CLASP's, were informational (rather than attentional), had specific meanings, and were in many cases directed. Despite these differences, however, there are some similarities between Taylor's condensation method and CLASP's. Both begin by assigning numerical weights to edges in the graph, and then use numerical methods to find important or central nodes. And as we will see in chapter 6, CLASP, like Taylor's algorithm (but unlike Skorochod'ko's or TELE-PATTAN's), can consider not just at the edges directly incident at a node, but also longer paths through the graph, in selecting nodes for the summary representation.

4.3 GRAPH-BASED CONDENSATION

CLASP's condensation stage uses the predication cohesion graph to select predications to form the summary representation. The selection is done in three steps. First, from the predication-cohesion graph a *weighted graph* is built (section 4.3.1). Second, a *scoring function* is defined on sets of nodes that attempts to measure to what extent a set of selected nodes meets the three requirements of *importance*, *representativeness* and *cohesiveness* (section 4.3.2). Third, a *greedy algorithm* uses this scoring function to select as many nodes as are required (section 4.3.3). In the first and second steps in particular, the system is particularly flexible, with a wide range of parameters allowing us to experiment with different condensation strategies. Here I illustrate only some simple possibilities; more detail is given in chapter 6. Whatever weighting and scoring functions are chosen, however, condensation is based entirely on the cohesive links between predications: other information, such as the presentation order of the source text, is not used. The only exception to this is in *constrained selection*, (described in section 4.3.3), where predications are grouped according to the source sentences they came from, although the order of the sentences is not used.

4.3.1 *Building a weighted graph*

In this step, multiple links between the same pair of predications are combined into a single edge, and each edge is given a positive numerical weight. Like the cohesive links of the predication-cohesion graph, the resulting edges are not

directed.

The intention is that the weight of an edge corresponds in some sense to the degree of attentional similarity between to nodes. (A weight of zero, indicating no similarity, is equivalent to no edge at all.) In assigning weights there are several questions to be asked. Firstly, should different kinds of cohesive links between predications lead to edges with the same or different weights? Secondly, what weight should we give to an edge when there is more than one relation between two predications? Thirdly, do we perform any kind of 'normalization': adjusting weights to take into account the number of arguments of the predications involved, for example?

It is hard to see any a priori reasons for choosing any one set of answers to these questions rather than another, so I have tried a range of possibilities to see what differences (if any) these choices produce in the resulting summaries. The results of these experiments are reported in chapter 8.

For the purposes of presenting a simple example, let us now assume that all cohesive links are given a weight of one, and that where there is more than one link between two predications, the sum of their weights is used as the edge weight. Under these assumption, the F-LANE example will give us the weighted graph shown in figure 4.2.

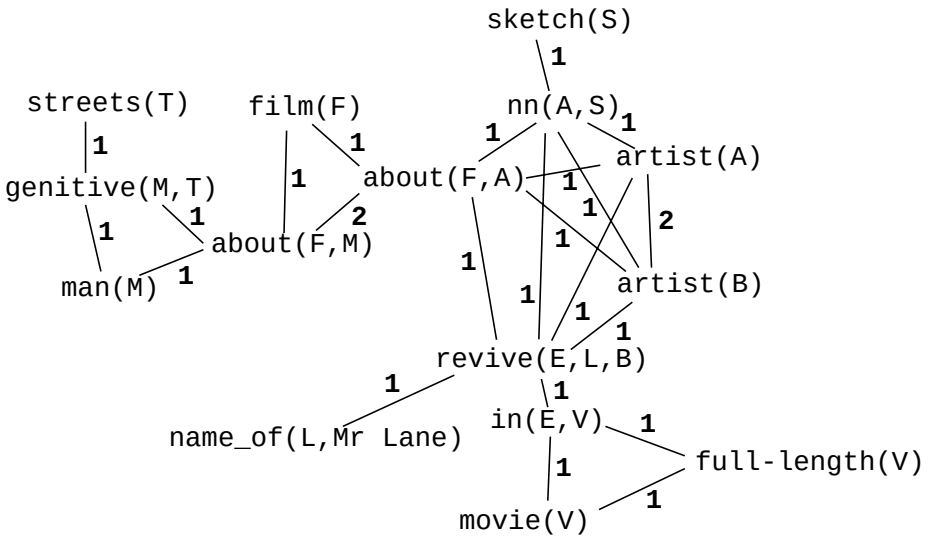


Figure 4.2. A weighted graph corresponding to the predication-cohesion graph for F-LANE (figure 4.1). All relations are given weight 1, and the sum of weights of relations between two predications is used as the edge weight.

4.3.2 *Scoring functions*

The second part of condensation is to define a scoring function on sets of nodes of the weighted graph. The scoring function does not take length requirements into account (this is done by the selection algorithm described in section 4.3.3), but attempts to reflect three criteria for good summary representations: that the set of selected nodes should contain *important* information, be *representative* of the whole text, and be internally *cohesive*. All CLASP's scoring functions are responsive, not prescriptive: they depend not on the predications themselves, but on the pattern and weight of edges between them. There are many possibilities in measuring each of the three criteria, and in addition there is the overall question of how much relative value to attach to each of them. The full range of choices available to CLASP is presented in chapter 6, which also discusses normalization issues not addressed here. In chapter 8 we look at summaries produced using a variety of different scoring functions. The following simplified examples illustrate the kind of scoring functions that are possible, using the graph in figure 4.2.

Importance

As noted in section 4.1.2, CLASP's summaries should indicate important topics in the text. In terms of the weighted graph, important predications are those that are highly relevant to the rest of the text, so important nodes should be those with many and strongly weighted connections to other nodes. The simplest way to score a node for importance in CLASP is to sum the weights of all edges incident at that node. With this measure (call it *imp-basic*), the top-scoring nodes from figure 4.2 are as follows:

about(F,A)	**7**	**(= 2+1+1+1+1+1)**
revive(E,L,B)	**6**	**(= 1+1+1+1+1+1)**
nn(A,S)	**5**	**(= 1+1+1+1+1)**
about(F,M)	**5**	**(= 2+1+1+1)**
artist(A)	**5**	**(= 2+1+1+1)**
artist(B)	**5**	**(= 2+1+1+1)**

For example, from the predication **about(F,A)** there is an edge with weight 2 to **about(F,M)** and there are edges with weight 1 to **film(F)**, **nn(A,S)**, **artist(A)**, **artist(B)** and **revive(E,L,B)**. Therefore the total score for **about(F,A)** is 7. The importance score for a set of nodes is simply the sum of the individual nodes' importance scores.

The *imp-basic* scoring function uses only very local information: it looks only at edges directly attached to the nodes being scored. More complex importance scores, as will be seen in chapter 6, consider not just the edges at each node, but also paths of two or more edges from the nodes being scored.

Representativeness

CLASP's summaries should be broad, rather than narrow. A set of simple predications which all have high importance scores might not constitute a suitable summary representation, if those predications all relate only to one topic or part of the source text. To ensure a broad summary, we need a way to measure the extent to which a set of nodes is related to the *whole* of the rest of the graph, and not just to part of it: this is *representativeness*. One of CLASP's simplest measures of representativeness is as follows: given a set of nodes, we count how many nodes are either in the set or connected to it by an edge (call this score *rep-basic*). Here are *rep-basic* scores for three sets of two nodes from figure 4.2:

```
{about(F,M), revive(E,L,B)}      12
{about(F,A), in(E,V)}            10
{about(F,A), nn(A,S)}             8
```

For example, there are 10 nodes directly connected to either **about(F,M)** or **revive(E,L,B)**, giving a *rep-basic* score of 12 when we add the two nodes in question. (The only nodes not connected to these two are **streets(T)**, **sketch(S)**, **movie(V)** and **full-length(V)**).

This example demonstrates that (unlike importance scoring) the representativeness score for a set of nodes is not just a sum of scores for each node in the set, since nodes which are connected to two or more predications in the set (in this case, **about(F,A)**) are still only counted once. This means that a set of nodes which have many neighbours in common will tend to achieve a lower representativeness score than a set of nodes with few common neighbours. More complex representativeness scoring in CLASP involves considering nodes at greater distances from the set under consideration, and applying importance scoring to the nodes thus found, rather than simply counting them.

Cohesiveness

This is a measure of the extent to which the selected nodes are connected to each other; this is desirable if we are aiming to produce a coherent summary (though of course it does not guarantee it), and it can be used to counteract the tendency of representativeness scoring to prefer sets of poorly connected nodes. Cohesiveness can be measured in a similar way to the *imp-basic* example above, except that we consider only edges between nodes that are both in the set under consideration. That is, for each node in the set, we sum the weights of edges from that node to other nodes in the set, and we add up the resulting sums (thus each edge is counted twice). Following this definition, we can compute example cohesiveness scores for sets of nodes from figure 4.2:

```
{about(F,A), about(F,M), film(F)}          8
{about(F,A), nn(A,S), revive(E,L,B)}       6
{about(F,A), revive(E,L,B), in(E,V)}       4
{genitive(M,T), nn(A,S), full-length(V)}   0
```

For example, in the set **{about(F,A), about(F,M), film(F)}**, there is an edge with weight 2 between **about(F,A)** and **about(F,M)**, and edges with weight 1 between **about(F,A)** and **film(F)**, and between **about(F,M)** and **film(F)**. Thus, considering only edges between nodes in the set, the total edge weight at **about(F,A)** is 3, the total edge weight at **about(F,M)** is also 3, and the total edge weight at **film(F)** is 2. Adding these together gives a cohesiveness score of 8 for this set.

4.3.3 *A greedy selection algorithm*

Given a scoring function, we want to choose, for the summary representation, a high-scoring set of nodes of the required size (this parameter allows control over the length of the summary). It is impractical to score all possible sets of nodes; therefore CLASP uses a greedy algorithm to produce progressively larger sets of nodes until enough have been selected.

Selecting individual predications

In its usual mode of operation, the greedy algorithm begins with the empty set of nodes. Then, at each stage, the scores of all the sets that could be formed by adding one more node to the set are computed, and the node which gives the highest-scoring set is added. If n predications are required for the summary representation, we simply stop after n iterations.

To illustrate, suppose we have chosen a scoring function by taking *rep-basic* + (*imp-basic* ÷ 10). (As defined in section 4.3.2, *imp-basic* is the sum of total edge weights at each node in the set, and *rep-basic* is the total number of nodes either in the set or directly connected to it by an edge). CLASP begins by considering the scores of all sets of one predication. Here are the highest-scoring such sets:

set of predications	*imp-basic*	*rep-basic*	score
{about(F,A)}	7	7	7.7
{revive(E,L,B)}	6	7	7.6
{nn(A,S)}	5	6	6.5

Thus **{about(F,A)}** is chosen. CLASP then considers sets consisting of **about(F,A)** and one other predication. The highest-scoring are as follows:

predication to add	*imp-basic*	*rep-basic*	score
in(E,V)	10	10	11
genitive(M,T)	10	10	11
full-length(V)	9	10	10.9
movie(V)	9	10	10.9
revive(E,L,B)	13	9	10.3
about(F,M)	12	9	10.2

(That is, the set **{about(F,A), in(E,V)}** has an *imp-basic* score of 10 and a *rep-basic* score of 10, etcetera.)

Thus, the predication **in(E,V)** or the predication **genitive(M,T)** will be chosen. (In a real example with a bigger graph and a more complex scoring function, it is rare for two candidate sets to have the same score.) As can be seen from the table above, if we were scoring using *imp-basic* only, we would instead have chosen the predication **revive(E,L,B)**; the reason adding this predication does not score so highly on *rep-basic* is that it is linked to the first predication selected. Roughly speaking, we can say that having first chosen to say something about films (**F**) and artists (**A**), the greedy algorithm, aiming for representativeness, has decided to say something about reviving (**E**) and movies (**V**), or men (**M**) and streets (**T**). (Of course, CLASP does not know that 'film' and 'movie' are near synonyms, or that the artist and the man are in fact the same person.)

Constrained selection

Depending on the kind of synthesis chosen, it may not be possible to generate summary text corresponding to an arbitrary set of predications. In particular, if summaries are produced by sentence extraction (section 4.4.1), we must effectively either select all the predications corresponding to a sentence of source text, or none of them.

This is exactly what is done in CLASP's *constrained selection*. The greedy algorithm is used as before, but at each step, instead of adding a single predication to the selected set, the algorithm adds all the predications from a chosen sentence to the set at once, choosing the sentence which gives the highest score for the resulting set.

4.4 SYNTHESIS

CLASP has two entirely independent methods for producing summary text. One, given a set of selected predications, generates short *summary phrases*. The other, after predications have been chosen by constrained selection, produces a summary by *sentence extraction*.

The reason for having two methods of synthesis is that there is a trade-off. Summary phrases are superior in terms of conciseness and precision, but sentence extracts are often clearer and more fluent. It is up to the user to choose whichever form of summary they prefer, or indeed to use both.

4.4.1 *The sentence extraction method*

In this method, the program outputs those sentences of the source text from which the selected predications were originally derived, in source order. The chief benefit of this kind of summary is that it presents full sentences of summary text, conveying something of the genre and style of the source, whilst making it unlikely that the reader will be misled into believing that the text states facts or expresses opinions which it actually does not. This can still happen if sentences are presented out of context: an opinion may be presented as a fact, or an anaphoric reference may be misinterpreted. Fortunately, however, such confusion is relatively uncommon: the result of taking material out of context is usually simply that it becomes obscure.

In evaluating CLASP (chapter 8), producing summaries by sentence extraction is additionally useful because it allows quantitative comparisons with summaries produced by other automatic methods, and with human sentence-extraction summaries too.

In the case of the text from which the F-LANE example was taken, for a very short (one sentence) summary we might extract the sentence from which the predication **`about(F,A)`** was obtained. Here is that sentence, in full:

> In 1976, as a film student at the Purchase campus of the State University of New York, Mr Lane shot *A Place in Time*, a 36 minute black-and-white film about a sketch artist, a man of the streets.

This example illustrates one of the problems with sentence extraction, namely that the summary is likely to include information which is not particularly important – the name of the campus where Mr Lane studied, for example. For this reason, summaries produced by sentence extraction do not go very far towards meeting our goals of identifying important content precisely or conveying it concisely.

4.4.2 *The summary phrase method*

This synthesis method exploits more fully the flexibility offered by the small granularity of the source (and hence the summary) representation, by generating new summary text from selected simple predications. However, two factors limit the extent to which we can generate running text. Firstly, the predications represent not facts stated in the source text, but ideas mentioned in it. Therefore, we must be careful not to give the impression in the summary that the source text makes assertions which in fact it may not. Secondly, because of the robust way in which we extract predications from source sentences even when they cannot be fully analysed, we may have predications that allow us to generate, say, a noun phrase, but no suitable verbal predication from which we could directly produce a whole sentence of summary text.

These considerations do not in themselves make it impossible to produce a summary consisting of whole sentences of running text. However, because there is no truth-conditional meaning or intentional context implied by the simple predications, such a summary would have to consist almost entirely of sentences of the form 'Something is said about X' or 'Y is mentioned'. This would not make very stimulating reading, being essentially a list of phrases (the Xs and Ys) wrapped up in a kind of standard disclaimer.

The solution I have adopted in the summary phrase method is not to attempt to present a continuous text summary or to insist on whole sentences, but to offer a list of short noun-phrases, verb-phrases and sentences, according to the kind of predications that have been selected in the condensation stage. These summary phrases achieve a greater shortening of expression than the summaries produced by sentence extraction, and they are more able to avoid the inclusion of unimportant material in the summary, so the important material is pinpointed more precisely. However, because these phrases convey less factual information to the reader, reflect less of the style of the source text, and do not constitute a continuous running text, sentence extraction summaries might be preferable in some circumstances.

Each summary phrase generated by CLASP does not necessarily correspond to a single predication, but rather may be produced by *clustering* – taking a group of related predications and generating a single phrase which combines all of them. This process has two benefits: it allows us to avoid repetition when several selected predications concern the same entity, and to increase readability by adding detail to produce longer, less 'bitty' phrases.

In the case of the F-LANE example, the following short summary might be generated if the two predications **`revive(E,L,B)`** and **`about(F,A)`** were selected:

> *This text says something about:*
> a film about a sketch artist,
> Mr Lane reviving his artist in a movie.

The first of these summary phrases is produced by clustering **about(F,A)** ('a film about an artist') with **nn(A,S)** ('sketch artist'). The semantic head information for the entities **F**, **A** and **S** is used to generate the head nouns for 'film', 'sketch' and 'artist'. In the second phrase, **revive(E,L,B)** ('Mr Lane reviving his artist') has been clustered with **in(E,V)** ('reviving … in a movie'), and again semantic-head information is used to generate the noun phrases 'Mr Lane', 'his artist' and 'a movie'. As shown in the second summary phrase, tense and aspect information from the source text is not preserved in the summary text; instead, the verb is presented in -ing form. For noun phrases, however, a similar approach is inappropriate, so the synthesis stage looks at the determiners used in the source text, and whether entities were singular or plural, and uses this information (extracted during the analysis of the source text, but not used for condensation) to choose determiners for use in the summary.

This example concludes our overview of the CLASP summariser, its goals, design, and implementation. The next three chapters address the three stages of analysis, condensation and synthesis in detail, considering at each stage the particular issues involved, design decisions made, and processing done.

5 ANALYSIS IN CLASP

CLASP's analysis stage takes a source text and builds a corresponding source representation. As described in section 4.2, this representation is a *predication cohesion graph* – a structure consisting of *simple predications* with *cohesive links* between them.

Section 5.1 describes the main issues to be addressed in the analysis stage, explains how I would ideally like to treat them, and outlines the approach taken in practice.

Sections 5.2–5.4 explain the three steps of processing carried out in the analysis stage, with particular emphasis on the difficulties encountered in dealing with real-life texts and the solutions adopted to overcome or ameliorate them. Section 5.2 describes the initial sentence-by-sentence analysis, which produces semantic representations from source text. Section 5.3 describes how simple predications are extracted from these representations. Section 5.4 describes how cohesive links are identified between these predications. The first of these steps is carried out by the Core Language Engine (CLE) (Alshawi 1992), a system for general-purpose analysis and synthesis of English text, which is also used in CLASP's synthesis stage. The processing in the other two steps is specific to CLASP.

5.1 ISSUES AND AIMS

Four issues which are important in each step of CLASP's analysis stage are *variation*, *ambiguity*, *practicality* and *robustness*. (By *variation*, I mean the question of to what extent we stop the source representation from reflecting differences of expression in the source text, whilst allowing it to reflect differences of content.) Variation and ambiguity are both theoretical issues, but ones where practical choices must be made. *Practicality* in a broad sense is about not relying on knowledge which may not realistically be available, or processes which we cannot yet automate. *Robustness* is about coping resiliently with failures in the different stages of analysis.

5.1.1 *Variation*

CLASP's simple predications are a semantic representation, intended to reflect concepts mentioned in the text, but not the particular way in which those concepts are expressed. Therefore when the same or similar material can be expressed in many ways, in extracting predications we would like to *suppress* such *variation of expression* and obtain the same predications in each case. When different surface texts convey different ideas, on the other hand, that is *variation of content*, and we would like to *reflect* it in the source representation. Of course the boundary between variation of content and variation of

expression is not clearly defined, and depends on the application.

Variation of expression

This can occur at all linguistic levels, from lexical ('pudding' / 'dessert') and syntactic ('She put her shoes on' / 'She put on her shoes') to the level of discourse: ('John put on his coat. He left.' / 'John put on his coat and left.') Ideally, we would like to suppress all these kinds of variation in the source representation; in practice, CLASP's treatment of variation is limited.

Many kinds of *syntactic variation*, such as choices of word order (both in overall sentence structure and on a smaller scale in the order of adjectival or adverbial modifiers) and the difference between active and passive voices, are reflected in the analyses produced by the CLE, but suppressed in extracting simple predications (section 5.3). It is not practical to suppress *lexical variation*, however, as to identify synonymous words in the source text requires us to do sense-disambiguation, which is not sufficiently reliable at this level. To deal properly with *variation at the discourse level* would require at least anaphor resolution and perhaps also some kind of discourse modelling; as explained in section 4.1.5 such processing is not feasible for CLASP. The only way in which CLASP suppresses discourse variation is that the predications and cohesive links extracted from the source text do not depend on the order in which the material was presented (although the order of material in the source text *is* considered in the synthesis stage).

Variation of content

The nature of the predication cohesion graph means that there are also some kinds of variation of content which are not reflected in the source representation. Most importantly, as explained in section 4.2.1, the logical meaning of the text and the way in which individual predications are combined in a higher-order propositional structure are not captured. In addition, CLASP's simple predications do not reflect the tense or voice of verbs, they conflate singular, plural and mass nouns, and they ignore determiners and quantifiers in the text. Thus none of these kinds of variation has any effect on what is selected in the condensation stage. However, some additional information, including the number and quantification or determination of noun entities, is recorded for use in the synthesis stage, to generate summary phrases.

5.1.2 *Ambiguity*

Ambiguity is the opposite of variation of expression: it occurs when more than one representation could correspond to a single section of source text. Thus we have lexical ambiguity ('pink', for example, has eight separate entries in Chambers' (1993) dictionary) syntactic ambiguity ('She saw a man with a telescope'), semantic ambiguity ('Two chefs cooked a lobster') and discourse ambiguity ('Bill saw Fred. He was wearing his hat').

Since CLASP's simple predications are intended to reflect text content

rather than its expression, we would like to resolve as much ambiguity as possible. At the level of syntax, choosing correct parses would allow us to accurately decompose a sentence into constituent predications, while at the discourse level, resolving anaphors would allow us to form cohesive links between predications from different parts of the text which concern the same discourse entity.

CLASP does attempt to resolve some kinds of sentence-level ambiguity: the CLE chooses between rival syntactic parses and quasi-logical semantic analyses. As already noted, however, CLASP is unable to resolve anaphors (whether intra- or inter-sentential) or analyse discourse ambiguity.

Lexical ambiguity

Whether we should expect any significant benefit from distinguishing between word senses (apart, that is, from any consequent improvement in syntactic and semantic analysis) is not clear. Because CLASP does not attempt to relate the source representation to a database of world or domain knowledge, it can know nothing about the meaning of different senses of a word, only that they *are* different. Correctly distinguishing between word senses would allow us to avoid finding spurious cohesive links when two words have the same surface form, but it may be that such spurious links are rare enough not to have a significant effect on summarising.

CLASP's approach to lexical ambiguity is a compromise. The CLE attempts sense disambiguation, but only on words in its core lexicon. Even this limited sense-disambiguation is often wrong, so CLASP is able to ignore it when finding cohesive links. This means that, even if the sentence analysis incorrectly supposes 'pink' to mean a sailing ship in one sentence and a socialist in another (when in fact both were referring to carnations), links between predications concerning these entities will be formed in the source representation. How much weight to attach to such links depends not only on how accurate the analysis is, but on how they affect the rest of the graph: as we will see in chapter 6, this is one of the many parameters in the condensation stage of the summariser.

5.1.3 *Practicality and robustness*

CLASP is not intended for a particular operational setup, but it is intended to process real texts and produce summaries of them. CLASP's processing is in most respects very practical, since it is automatic and does not involve the use of world- or domain-knowledge. The one respect in which CLASP's analysis is not very practical is that, as implemented, it is very slow.

A requirement for *robustness* is a consequence of practicality. CLASP's analysis is robust because the CLE can find partial parses and partial semantic analyses of sentences it cannot process fully, and the predication cohesion graphs are a robust source representation (in the sense of section 2.5.1), because an error in constructing one part of the graph need not propagate to other parts,

and because the decomposition of sentences into many simple predications allows us to represent even sentences which we can only partially analyse.

5.2 SENTENCE-BASED PROCESSING WITH THE CORE LANGUAGE ENGINE

As described in section 4.1.4, CLASP uses a pre-existing system, the Core Language Engine (Alshawi 1992), to perform syntactic and semantic analysis of source sentences.

The CLE is a general-purpose tool for morphological, syntactic and semantic analysis and generation of text, implemented in the Prolog language. The particular version used in CLASP is the 'CLARE-3 version' (Alshawi et al 1992). The CLE is equipped with a grammar of written English with about 200 syntactic rules, a similar number of semantic rules, and a *core lexicon* of over 1000 words with detailed subcategorisation information and a simple *sortal hierarchy* that enables a limited amount of disambiguation. An additional external lexicon can be attached: CLASP uses the MRC lexicon, which has about 100000 entries but is much less detailed than the core CLE lexicon; for each word it gives only the basic part-of-speech label – noun, adjective, verb or adverb – with some additional annotation to indicate irregular formations and words with initial capital letters. There are no subcategorisation details or any sense information.

CLASP's use of the CLE is perhaps a slightly profligate use of a sophisticated system: there are many features that CLASP does not use, and our simple predications are much less sophisticated than the CLE's semantic representations. But the fact that the CLE is available and can process a wide range of English text fairly robustly makes it quite suitable for our purposes. Some alternative approaches to the initial step of analysis are discussed in chapter 9.

Section 5.2.1 describes the semantic representation used by the CLE, called *quasi-logical form* (QLF). Section 5.2.2 outlines the processing done to produce QLFs from source sentences. Section 5.2.3 considers the performance of the CLE within CLASP, and section 5.2.4 gives an example analysis. The process of extracting simple predications from QLF is described in section 5.3.

5.2.1 *Quasi-logical form*

In general, QLF is intended to represent the meaning of sentences, insofar as it can be determined from each sentence in isolation. The true logical meaning, if required, must subsequently be determined by a process of QLF *resolution*, which would involve resolving anaphoric references, choosing quantifier scoping and locating the events described in time. However, despite the fact that lexical and syntactic ambiguity also require resolution in context, each quasi-logical form corresponds to a particular choice of word senses (for words for which the CLE has sense information) and a particular syntactic parse.

A full definition of the QLF formalism as used in CLASP is given by Alshawi et al (1992). It is not necessary to explain all the details of QLF here, so I

In quasi-logical form, or QLF:
• Capital letters **A**, **B**, … are variables.
• A list **[**⟨*predicate*⟩**,** ⟨*arg*⟩**,** ⟨*arg*⟩**,** …**]** represents a predication.
• ⟨*variable*⟩**^**⟨*body*⟩ indicates a function which takes an argument

⟨*variable*⟩ and returns ⟨*body*⟩ (a predication) as the result. (In the lambda calculus, this would be the function λ⟨*variable*⟩.⟨*body*⟩.)
• **term(**⟨*string*⟩**,** ⟨*category*⟩**,** ⟨*index*⟩**,** ⟨*restriction*⟩**)** represents a discourse entity. ⟨*category*⟩ gives information about type and quantifiers, ⟨*index*⟩ is a variable which can be used elsewhere in the QLF to refer to this same entity, and ⟨*restriction*⟩ is a function which yields a true predication when applied to the entity in question.
• **form(**⟨*string*⟩**,** ⟨*category*⟩**,** ⟨*index*⟩**,** ⟨*restriction*⟩**)** represents a predication which may require interpretation in context. Such predications can arise from verbal forms, prepositional phrases, noun-noun compounds, conjunction, ellipsis and other linguistic structures – the ⟨*category*⟩ indicates which of these kinds of **form** it is. ⟨*index*⟩ is a variable representing the event or state (as appropriate) which the form describes. ⟨*restriction*⟩ is a function which returns the meaning of the form in a particular context, when it is applied to an unspecified context-dependent resolving function.

Figure 5.1. Main points of the QLF formalism.

have summarised some of its essential features in figure 5.1, and will discuss them with reference to the example sentence S-BALLOON: 'John has a red balloon'. As is discussed later, all the QLFs shown here have also been slightly simplified by removing a few details not used by CLASP. Here is the QLF for S BALLOON:

```
[dcl,
 form(l([John,has,a,red,balloon]),verb,A,
      B^
      [B,
       [have_3p,A,
        term(l([John]),proper_name,C,
             D^[name_of,D,John]),
        term(l([a,red,balloon]),q(a,sing),H,
             I^[and,[balloon_NounMRC,I],
                [red_Coloured,I]])]])]
```

Main features of QLF

As illustrated, QLF involves *variables* (capital letters **A**, **B**, etc.) and *predications* (lists in square brackets, e.g. **[red_Coloured,I]** for **red_Coloured(I)**). In the case of a predicate such as **red_Coloured**, the part before the underscore (**red**) indicates the corresponding surface word, and the part after the underscore (**Coloured**) is a mnemonic for the sense reading. For words from the external lexicon, for which we have no sense information, the part after the underscore indicates only the part of speech (e.g. **NounMRC**). QLF also includes a simple *functional* notation: **D^[name_of,D,John]**, for example, means the function which takes an argument **D** and returns the predication **[name_of,D,John]**.

QLFs also contain **term** and **form** expressions. A **term** expression represents a discourse entity, with arguments that provide: the corresponding list of surface words; information about number, determiners and quantifiers used; a variable to represent the discourse entity; and a function describing the entity. Thus for example a single **term** expression tells us that the discourse entity referred to by the phrase 'a red balloon' (**l([a,red,balloon])**) is singular and quantified with 'a' (**q(a,sing)**), may be represented by the variable **H**, and satisfies the predicate of being both a balloon and red in colour (**I^[and,[balloon_NounMRC,I], [red_Coloured,I]]**).

A **form** expression represents a predication which requires interpretation or resolution in context. It may be a verbal predication describing an event or state, or it may correspond to a prepositional phrase, a noun–noun compound, a possessive (**X**'s **Y**) or genitive (**Y** of **X**) construction, or other phenomena such as ellipsis. A **form** expression's arguments provide: a list of the corresponding surface words (**l([John,has,a,red,balloon])**); information about the kind of predication represented (**verb** for verbal predications); a variable to represent the event or state that the **form** describes; and a function describing the predication before resolution (**B^[B,[have_3p,...]]**).

As each QLF **term** or **form** introduces a variable corresponding to the entities, events or states mentioned (e.g. **C** for John, **H** for the balloon, **A** for the state of having), these variables can be used to refer to the entity or state elsewhere in the QLF, so that the entire **term** or **form** need not be repeated.

Representation of ambiguity

QLF **term** expressions represent ambiguity in that, although they contain information on the quantifiers and determiners used to describe entities in the source text, they do not specify logical quantification of those entities, or whether an entity is or is not actually the same as another entity from a different (or indeed, the same) sentence. As they require resolution in context, **form** expressions also represent ambiguity. In the case of noun–noun compounds, for example, 'car park', 'country cottage' and 'film producer' all involve different relationships between the nouns, but give similar QLF **form**s.

None of these kinds of ambiguity will be resolved in CLASP's later processing, when simple predications are extracted from QLF. Quantification

and scoping of discourse entities is irrelevant to CLASP because its simple predications have no large-scale propositional structure. For noun–noun compounds, we will use a catch-all predicate (**nn**) to stand for all the possible relations; a similar approach is taken for prepositional relations, possessives and genitives.

Higher-level structure

As seen in the S-BALLOON example, QLF predications may have variables, predications, **term**s, **form**s, or functions as their arguments. Some predicates, such as **red_Coloured**, express straightforward properties of entities (such as could be modelled in basic set-theory), whereas other predicates have more complex interpretations. For example, the outermost predication in the QLF above, beginning **[dcl,...]**, says that the sentence is declarative – a statement rather than a question or an order – and it takes the whole of the rest of the QLF as its argument. There is no syntactic distinction between simple predicates such as **red_Coloured**, logical connectives such as **and**, and higher-order modal predicates such as **dcl** or those expressing beliefs and intentions. CLASP's simple predications are not intended to reflect higher-level propositional structure; therefore when simple predications are extracted from QLF (section 5.3), some predications (such as **[dcl,...]**) that are deemed not to be useful for summarising will be discarded.

Simplification of examples

QLF as produced by the CLE is slightly more complicated than the examples given here. In reality, **term** and **form** expressions have extra arguments that are initially unbound. These arguments exist to allow for subsequent processes of resolution and scoping, which would instantiate them. The *categories* of **term** and **form** expressions (i.e. their second arguments) are also more elaborate than those shown here: **term** categories contain additional information about sentence roles and possible referents; **form** categories for verbal predications contain additional information about tense and aspect. As none of this information is relevant to CLASP's analysis, I have omitted it from all the QLFs shown here.

5.2.2 *Outline of* CLE *processing*

This section describes only the processing carried out by the CLE in CLASP; there are many other features and capabilities of the CLE which are not used by CLASP and therefore not described here. Fuller descriptions are provided by the CLE book (Alshawi 1992) and the CLARE manuals (Alshawi et al 1992).

The CLE grammar formalism uses unification of feature values to enforce syntactic agreement; lexical items and categories have features indicating number, verb form, subcategorisation type and so on. Syntactic parsing is performed by a bottom-up chart parser with a left-corner restriction to speed up processing. First the parser attempts to understand each sentence given to it

as a whole; if this fails, it tries *partial parsing*, i.e. it looks for sequences of well-formed phrases within each sentence.

For each syntactic rule in the grammar, there are one or more corresponding semantic rules, and so for each parse found (either of a whole sentence or of a phrase) there may be one or more semantic analyses. Again, unification of feature values is used to construct semantic representations compositionally, giving quasi-logical forms for each sentence or sentence fragment parsed.

Though the CLE has a limited capability for inter-sentential resolution of QLF into *resolved logical form* (RLF), it is not used in CLASP for two reasons. Firstly, much of the resolution process, such as quantification and scoping, is irrelevant to the extraction of simple predications. Secondly, although anaphor resolution is very desirable, in practice it is not sufficiently accurate to be useful, because we have no world or domain model and because the complexity of real texts means that often only fragmentary parses, and hence fragmentary QLFs, can be obtained. In newspaper text especially, a great many anaphoric references are in the form of definite noun-phrases (sometimes involving a different head noun from that used previously to refer to the same entity), which may require a great deal of world knowledge to be resolved.

Even before these later stages of processing, analysis of an English sentence is very likely to yield many alternative QLFs. There may be several syntactic parses, and one or more semantic analyses for each parse. The CLE uses its *sortal hierarchy* to discard unacceptable semantic analyses (e.g. by requiring that certain arguments of certain predicates must be animate objects, etc). A *preference system,* derived from analysis of a large corpus of text and human judgements of which analyses are correct, then allows it to select from among the remaining candidate QLFs one which it considers most likely to be appropriate. The result is either a single QLF corresponding to an analysis of the whole sentence, or a set of QLF *fragments* corresponding to a sequence of phrases identified in partial parsing.

5.2.3 *Performance of the* CLE

In using the CLE to process real-life text (such as those in appendix A), we must consider how well the grammar and lexicon cover the range of linguistic phenomena to be analysed, and how efficiently and accurately the CLE's algorithms perform this analysis (and choose between alternative analyses).

Coverage

The CLE's grammar has a very broad coverage of English syntax: in addition to complex verbal and nominal phrases, it covers a wide variety of movement phenomena such as topicalisation, cleft sentences, and questions. The main area in which it is deficient for CLASP's purposes is in dealing with direct and indirect speech: given such input, the CLE will usually parse the speech only as a separate fragment. The core lexicon is quite thorough but it is small (less than 2000 words). The external MRC lexicon is much less detailed, with the result that no

distinction is made between transitive and intransitive verbs, and no allowance made for verbs with more complex complements. Thus if a verb such as 'suspect' is not in the core lexicon, its use with a sentential complement cannot be correctly analysed. Unknown words are assumed to be proper or common nouns, on the basis of capitalization. The CLE is also able to recognise multi-word proper nouns, and numbers written out or with numerals, but its morphological analysis does not include rules for expressions such as '£100' or '20lbs'.

Efficiency and accuracy

The CLE's processing is robust but not particularly accurate, and very slow. Robustness is achieved by being able to deal with unknown words, and by the use of partial parsing whenever full parsing fails. The treatment of unknown words is mostly correct, but when a proper noun appears at the start of a sentence it may be mistaken for a common noun. Another common mistake is to interpret a capitalised adjective as a proper noun (in figure 5.9, for example, 'Japanese' is understood in one sentence as an adjective, and in another as a noun).

Full parsing can fail if the grammar fails to describe a linguistic construction used in the sentence, if a word is not present in the lexicon with the correct category, if practical limitations (described in the next paragraph) do not allow a complete parse to be found, or if the sentence itself contains errors of grammar or spelling. In practice, with *Wall Street Journal* texts, the long sentences and the frequent use of reported speech and ellipsis mean that partial parsing is the norm, rather than the exception, in CLASP's processing.

The CLE's parsing algorithm, although not inefficient, is not fast, and the system, running on a HP 9000-series workstation, will happily spend hours parsing a single sentence. (In comparison with parsing, all the other stages of processing performed by the CLE and by CLASP are much quicker.) As this is impractical when we are trying to analyse a whole story, CLASP uses an alternative version of the parser (provided with the CLE) which allows a limit to be placed on the 'cost' of parses to be found. The cost of a parse is the sum of the costs for the words and rules used in constructing it. In CLASP, all rule costs are set to 1 and all word costs to zero, so the parse cost limit becomes the maximum number of rule applications that can be used in analysing a constituent. A setting of, say, 15 allows the parser to analyse short sentences as before, but a sentence of 30 words is almost certain not to receive a full analysis (since most rules are unary or binary): when this happens, the CLE will divide the sentences into two or more fragments and produce a QLF for each.

When it comes to choosing between alternative syntactic and semantic analyses, the sortal hierarchy is successful in ruling out some incorrect analyses, but it can only operate successfully on QLF predications resulting from words in the core lexicon, as the external lexicon has no sortal information. In CLASP, the CLE's preference system is not very effective, because the complexity of the text and the time taken to analyse it made processing a large corpus of data

impractical. Thus, the CLE makes many errors in choosing between QLFs: a common example is in the choice of attachment of prepositional phrases to noun or verb phrases, as illustrated in the next section.

5.2.4 *Example analysis*

As an illustration of the QLF resulting from the CLE's analysis, we will consider sentence S-JAP: 'Japanese investment in Southeast Asia is propelling the region toward economic integration.' This is a sentence (albeit a rather shorter than average one) from a story in the *Wall Street Journal*, an example of the kind of material we are aiming to summarise. The story, JAPINV, is given in full in appendix A. The (simplified) QLF produced by the CLE for S-JAP is shown in figure 5.2. Note that the prepositional phrase 'toward economic integration' has been incorrectly attached to 'the region' rather than to the verb phrase 'propelling the region'. This will later result in an incorrect predication being extracted and included in the predication-cohesion graph.

Inspecting the QLF, we can see that it contains two kinds of information. There is the quasi-logical structure of predications and entities, and, in the second arguments of the **`term`** and **`form`** structures, there are descriptions of the form the expression of these structures takes. For example, the information that 'Southeast Asia' is a proper name, that 'the region' is a definite singular noun phrase, and that 'economic integration' is a mass noun phrase, is contained in the second arguments of the corresponding **`term`**s: **`proper_name`** for 'Southeast Asia', **`ref(def,the,sing)`** for 'the region' and **`q(exists,mass)`** for 'economic integration'. This syntactic information is not part of the logical structure, but would be used in the process of resolving QLF to RLF. Since CLASP does not attempt such resolution, and does not represent tense, number, quantification, etc. in its simple predications, this information is largely unneeded – in extracting simple predications from QLF, it is primarily the coarse quasi-logical structure that we will consider.

```
[dcl,
 form(l([japanese,investment,in,Southeast Asia,is,propelling,the,
         region,toward,economic,integration]),
      verb,A,
      B^
      [B,
       [propel_TransitiveVerbMRC,A,
        term(l([japanese,investment,in,Southeast Asia]),
             q(exists,mass),C,
             D^
             [and,
              [and,[investment_NounMRC,D],[japanese_AdjectiveMRC,D]],
              form(l([in,Southeast Asia]),prep(in),E,
                   F^
                   [F,C,
                    term(l([Southeast Asia]),proper_name,H,
                         I^[name_of,I,Southeast Asia])])]),
        term(l([the,region,toward,economic,integration]),
             ref(def,the,sing),P,
             Q^
             [and,[region_NounMRC,Q],
              form(l([toward,economic,integration]),prep(toward),R,
                   S^
                   [S,P,
                    term(l([economic,integration]),q(exists,mass),U,
                         V^[and,[integration_NounMRC,V],
                            [economic_Financial,V]])])])])]]
```

Figure 5.2: QLF produced by CLE analysis of sentence S-JAP: 'Japanese investment in Southeast Asia is propelling the region towards economic integration'.

5.3 EXTRACTING SIMPLE PREDICATIONS FROM QLF

Having analysed individual sentences of the source text, we now extract *simple predications* and *semantic heads* from their QLFs. From the QLF for S-JAP (in figure 5.2), we will obtain the following:

Simple predications:

```
propel_TransitiveVerbMRC(A,C,P)
investment_NounMRC(C)
japanese_AdjectiveMRC(C)
in(C,H)
name_of(H,Southeast Asia)
region_NounMRC(P)
toward(P,U)
economic_Financial(U)
integration_NounMRC(U)
```

Semantic heads:

```
A: propel_TransitiveVerbMRC
C: investment_NounMRC
H: Southeast Asia
P: region_NounMRC
U: integration_NounMRC
```

To get from the complexity of QLF to the relative simplicity of these predications requires not further interpretation, but rather a change of representation. There are several differences between QLF and simple predications, and corresponding requirements to be met by the extraction process:

1. Predications in QLF are combined into a large scale structure: we will therefore need to traverse the structure extracting the individual predications encountered.

2. Predications in QLF are of two kinds. Some are presented explicitly, as lists **[**⟨*predicate*⟩**,** ⟨*arg*⟩**,** ⟨*arg*⟩**,** …**]**. Others are presented as **form**s. In the absence of any mechanism for performing context-dependent resolution, we will need to do some kind of default resolution on these **form**s to produce predications from them.

3. Predications in QLF do not have simple variables or atoms as their arguments. Frequently, a **term** or **form** is itself the argument of a predication, and in the case of modal predicates or logical connectives (**and**, **or**, etc) the arguments may themselves be predications. We need to select, for each argument of a predication in the QLF, a variable or atom to be the corresponding argument of the corresponding simple predication.

4. In QLF, there can be many variables which ultimately correspond to the same discourse entity, event or state. We need to replace variables as appropriate to ensure that there is one variable per entity in the eventual predications.

As noted in section 5.1.1, simple predications are intended to suppress many kinds of variation; thus the extraction process will ignore some kinds of information in QLF, such as number and tense information.

Requirements 1 and 2 above are met in stage 1 of extraction, *extracting raw predications* (section 5.3.1). Stage 2, *selecting atomic arguments* (section 5.3.2) deals with requirement 3. Stage 3, *resolving variables* (section 5.3.3) deals with requirement 4. Extraction is completed by stage 4, *filtering out unwanted predications* (section 5.3.4), and stage 5, *assigning semantic heads* (section 5.3.5). Section 5.3.6 describes how CLASP also extracts additional information about entities for use in the synthesis stage only.

In section 5.3.7, I present a slightly larger example of the results of this sentence-by-sentence processing, which will be used in the discussion of graph construction in section 5.4.

5.3.1 *Extraction stage 1 – extracting raw predications*

Predications in QLF occur either explicitly as lists, or implicitly as **`form`** structures. So CLASP recurses through the QLF, extracting a predication each time a list or a **`form`** is encountered.

As we saw in section 5.2.1, a **`form`** stands for a predication requiring interpretation in context. Since CLASP has no context model and our source representation is simple enough not to need such interpretation, we simply apply one of a set of default predication-extraction rules to the *form restriction* (the fourth argument of the **`form`** structure), the choice of rule depending on the *form category* (the second argument of the **`form`** structure).

For example, for *verbal forms* (such as the outermost form in the QLF for S-JAP in figure 5.2), the *form restriction* will be of the form **`[X^[X, [`***predicate*`,`** ***arg*`1,`** ***arg*`2,`** `...]]`, where the *arg*s are the arguments (event variable, subject, and object if present) of the verbal predicate. We therefore extract the predication ***predicate*`(`*`arg`*`1,`** ***arg*`2,`** `...)`. From the **`form`** in figure 5.2, we would obtain the following raw predication:

```
propel_TransitiveVerbMRC(A, term(...), term(...))
```

in which the two '**`term(...)`**'s correspond to 'Japanese Investment in Southeast Asia' and 'the region toward economic integration' respectively.

Similarly, each of the other kinds of form has its particular format, and we apply a corresponding extraction rule to obtain a predication. For prepositional phrases, the preposition itself is the predicate. For noun-noun-compounds, the

```
dcl(form(l([japanese,investment,...,is,propelling,...]),...))
propel_TransitiveVerbMRC(A,term(l([japanese,investment,...]),...),
                         term(l([the,region,...]),...)))
and([investment_NounMRC,D],[japanese_AdjectiveMRC,D])
investment_NounMRC(D)
japanese_AdjectiveMRC(D)
and([and,...],form(l([in,Southeast Asia]),...))
in(C,term(l([Southeast Asia]),...))
name_of(I,Southeast Asia)
and([region_NounMRC,Q],form(l([toward,economic,integration]),...))
region_NounMRC(Q)
toward(P,term(l([economic,integration]),...))
and([integration_NounMRC,V],[economic_Financial,V])
economic_Financial(V)
integration_NounMRC(V)
```

Figure 5.3: raw predications (abbreviated) from s-JAP, after extraction stage 1.

---

predicate is **nn**; for genitives, **genitive**; and for possessives, **possessive**.

For the other predications (those that appear in the QLF as lists), the procedure is very simple: from a list **[***predicate***,** *arg*1**,** *arg*2**,** ...**]** we just extract the raw predication *predicate***(***arg*1, *arg*2, ...**)**.

Figure 5.3 shows the raw predications extracted from the s-JAP example QLF in figure 5.2.

5.3.2 *Extraction stage 2 – selecting atomic arguments*

Each extracted raw predication already has an atomic predicate, but its arguments may be variables, atoms, predications, QLF **term**s, **form**s, or other structures. In this step, an atomic value, usually a variable, is chosen for each argument, according to the following rules (examples from figure 5.3):

- *Arguments which are already atomic.* These are left unchanged. Examples: **economic_Financial(V)** and **integration_NounMRC(V).**

- *Arguments which are* **term***s or* **form***s.* These introduce variables that correspond to discourse entities, events and states. When a **term** or **form** occurs as an argument of a predication, we therefore replace it by the corresponding entity, event or state variable. This is the *term index* or *form index* variable, i.e. the third argument of the **term** or **form** structure (see figure 5.1). For example,
  in(C,term(l([Southeast Asia]),proper_name,H,...))
  becomes **in(C, H)**.

```
dcl(A)
propel_TransitiveVerbMRC(A,C,P)
and(investment_NounMRC,japanese_AdjectiveMRC)
investment_NounMRC(D)
japanese_AdjectiveMRC(D)
and(and,E)
in(C,H)
name_of(I,Southeast Asia)
and(region_NounMRC,R)
region_NounMRC(Q)
toward(P,U)
and(integration_NounMRC,economic_Financial)
economic_Financial(V)
integration_NounMRC(V)
```

Figure 5.4: predications from S-JAP, after extraction stage 2.

• *Arguments which are predications expressed as lists*. Typically these occur as arguments of modal predications or logical connectives. As such predications cannot properly be represented as simple predications, when we choose atomic arguments we will inevitably simplify them. CLASP replaces list arguments by the first element of the list, which is the predicate and hence itself atomic. For example, the predication
`and([investment_NounMRC,D],[japanese_AdjectiveMRC,D])`
becomes `and(investment_NounMRC, japanese_AdjectiveMRC)`.

Applying these rules to the raw predications extracted from S-JAP (figure 5.3), we obtain the predications in figure 5.4.

5.3.3 *Extraction stage 3 – resolving variables*

After stage 2, the predications are beginning to look like the simple predications we are aiming for, but still have superfluous variables – for example, in the S-JAP predications (figure 5.4), variables **C** and **D** both refer to the same discourse entity ('Japanese investment in Southeast Asia'). In stage 3, we resolve these variables, replacing them so that eventually there is only one variable per discourse entity.

There are two ways in which different QLF variables can refer to the same entity. One arises because although each **term** structure supplies a variable (⟨*index*⟩) that corresponds to the discourse entity introduced, the *term restriction* is expressed as a function ⟨*arg*⟩^⟨*body*⟩ (see figure 5.1). In predications extracted from the restriction, ⟨*arg*⟩ will therefore appear instead of ⟨*index*⟩. CLASP simulates applying the term restriction to the entity, by replacing ⟨*arg*⟩ by

```
dcl(A)
propel_TransitiveVerbMRC(A,C,P)
and(investment_NounMRC,japanese_AdjectiveMRC)
investment_NounMRC(C)
japanese_AdjectiveMRC(C)
and(and,E)
in(C,H)
name_of(H,Southeast Asia)
and(region_NounMRC,toward)
region_NounMRC(P)
toward(P,U)
and(integration_NounMRC,economic_Financial)
economic_Financial(U)
integration_NounMRC(U)
```

Figure 5.5: predications from S-JAP, after extraction stage 3.

⟨*index*⟩ wherever it occurs in an extracted predication.

In the QLF for S-JAP (figure 5.2), there are four **term**s, each leading to a replacement: we replace **D** by **C** ('investment'), **I** by **H** ('Southeast Asia'), **Q** by **P** ('the region') and **V** by **U** ('integration'). Applying these replacements to the predications obtained in stage 2, we get the simple predications shown in figure 5.5.

The other case where variable replacement is necessary (though not in the S-JAP example) is in complex noun phrases, such as conjunctions or noun-noun compounds, where there may be separate QLF **term**s for component nouns, and for the whole phrase.

When a predication has as an argument a variable corresponding to a conjunction of entities, we want to replace it by a set of predications, each involving only one of the conjoined entities (that is, a sentence such as 'John ate bread and jam' should give *two* eating predicates, one for bread and one for jam). In the QLF, the **term** for a conjunction will contain two or more other **term**s for the conjoined noun phrases. CLASP replaces the outer **term**'s index variable by *each* of the inner **term**s' index variables, in each predication where it appears, making as many copies of the predication as there are conjoined items.

For other complex noun phrases, such as noun-noun compounds and possessives, the QLF **term** includes a nested **term** for the head noun, each having its own index variable. Predications involving these variables in fact concern the same discourse entity, so CLASP replaces the outer **term**'s index variable by the inner **term**'s index variable in each predication where it appears.

```
propel_TransitiveVerbMRC(A,C,P)
investment_NounMRC(C)
japanese_AdjectiveMRC(C)
in(C,H)
name_of(H,Southeast Asia)
region_NounMRC(P)
toward(P,U)
economic_Financial(U)
integration_NounMRC(U)
```

Figure 5.6: simple predications from S-JAP, after extraction stage 4.

5.3.4 *Extraction stage 4 – filtering out uninformative predications*

The output from stage 3 is substantially what we are aiming for. However, it includes predications which do not express any of the content of the text. Predications like **`and(and,E)`** in figure 5.5 reflect syntactic features of the QLF, or its high-level logical structure, which cannot accurately be expressed in simple predications. Stages 1–3 have simplified these predications to the point of being uninformative, and hence useless for summarising. It might seem careless to have extracted such predications in the first place, but it is simpler for CLASP to extract all the predications and then eliminate unwanted ones. This is done with a short *predicate stop-list*, which consists of **`and`**, **`or`**, **`apply`**, **`dcl`**, and a few other predicates. All predications whose predicates are on this list are discarded.

It is possible (though rare) for two identical raw predications to be extracted in stage 1, or for two predications which started off different to become identical after stages 2 and 3 of extraction. For CLASP's purposes, such duplicate predications are uninformative (since they do not necessarily reflect reiteration in the surface text) and are removed in this stage.

Figure 5.6 shows the predications from S-JAP remaining after this stage. These are the simple predications we will use in constructing the predication cohesion graph.

There is a parallel between this filtering and the use of stop-lists in processing surface text. For example, TELE-PATTAN (Benbrahim and Ahmad 1994) uses a stop-list to distinguish function words from content words, as repetition of function words does not constitute lexical cohesion. CLASP's predicate stop-list is used analogously to divide the space of all possible predications into *content predications* and *function predications*; the latter are ignored because they have no useful attentional content. However, the situation with predications is more complicated than with surface words, and (as described later in section 5.4.2) CLASP also uses a second stop-list to avoid finding spurious cohesive links between predications that are not attentionally related.

5.3.5 *Extraction stage 5 – assigning semantic heads*

Section 4.2.2 introduced the concept of *semantic heads* for entities appearing in simple predications; these are used in finding cohesive links between predications, and also in generating summary phrases.

After stage 4 of extraction, most variables appearing in the simple predications are either the index variables of simple nominal **term**s (either common noun phrases or proper nouns), or the index variables of **form**s. For each such variable, CLASP defines one simple predication to be a *head predication*, and identifies an atomic *semantic head*, as follows:

- *Index variables of common noun* **term***s*: the head predication is the predication corresponding to the head noun; the semantic head is its predicate.

- *Index variables of proper noun* **term***s* (*including the names of months, seasons, etc*): the head predication is the one giving the name of the entity; the semantic head is the proper noun itself.

- I*ndex variables of verbal* **form***s*: the head predication is the verbal predication; the semantic head is the verbal predicate.

- *Index variables of non-verbal* **form***s*: the head predication is the predication corresponding to the **form**, and the semantic head is its predicate (e.g. **genitive**, **possessive** or a preposition).

Figure 5.7 shows the head predications and semantic heads for the entities in the S-JAP sentence.

Sometimes there may be variables in simple predications which do not correspond to a nominal or verbal entity mentioned in the source text. This can happen in two ways: firstly, if an entity is not mentioned in the source text, but is implied by it (for example, in a passive sentence such as 'John was bitten', the entity that bit John is not mentioned), the QLF will contain a **term** for that entity that says only that it is an entity. Secondly, if full parsing fails, and the CLE produces QLF corresponding to sentence fragments, there may be variables for which there is no corresponding **term.** (For example, the QLF for the fragment 'in the region' would involve a variable, but no **term**, for whatever was in the region). In both these cases, CLASP does not assign head predications or semantic heads.

5.3.6 *Recording additional information about entities*

In addition to the simple predications and semantic heads used to construct the predication cohesion graph, CLASP's analysis records some further information

entity	*head predication*	*semantic head*
C	investment_NounMRC(C)	investment_NounMRC
H	name_of(H,Southeast Asia)	Southeast Asia
P	region_NounMRC(P)	region_NounMRC
U	integration_NounMRC(U)	integration_NounMRC
A	propel_TransitiveVerbMRC(A,C,P)	propel_TransitiveVerbMRC
E	in(C,H)	in
R	toward(P,U)	toward

Figure 5.7: head predications and semantic heads for the S-JAP sentence.

about entities, which will be used in the synthesis stage to generate summary phrases (chapter 7), but is not used in condensation.

Specifically, CLASP records whether entities are *nominal*, *verbal* or neither (e.g. for prepositional phrases), and for nominal entities, whether they are common nouns or proper nouns. In the case of common nouns, we also record whether they are *singular*, *plural* or *mass*, and what *determiner* (for definite noun phrases) or *quantifier* (for indefinite noun phrases) was used.

All this information is obtained from the **term** or **form** *categories* (the second argument of the **term** or **form**); these also contain details about tense, voice and aspectual information for verbal **form**s, which are not relevant to CLASP.

S-JAP has four nominal entities: **C**, **H**, **P** and **U**. **H**'s term category is **proper_name**, indicating that it is a proper noun. **C** ('investment') and **U** ('integration') both have category **q(exists,mass)**, indicating that they are existentially quantified mass nouns. **P** ('the region') has the category **ref(def,the,sing)**, indicating that it is a definite singular common noun entity, with determiner 'the'.

5.3.7 *A slightly larger example*

Unfortunately a whole text is too long to give a useful example of CLASP's analysis, as typically any text of more than a few sentences will yield hundreds or thousands of simple predications and semantic heads. However, a larger example than S-JAP is required, to further illustrate CLASP's sentence analysis and to show how the predication cohesion graph is constructed. Therefore I will take as an example the short text T-JAP (figure 5.8), which consists of S-JAP and two more sentences from the JAPINV story.

After analysis with the CLE and extraction of simple predications and semantic heads from the resulting QLFs, this text yields the simple predications, head predications, and semantic heads shown in figure 5.9.

Predications 1–9 and entities **A**–**E** are those from S-JAP. Sentence S-JAP2

S-JAP:	Japanese investment in Southeast Asia is propelling the region toward economic integration.
S-JAP2:	In Thailand, for example, the government's Board of Investment approved $705 million of Japanese investment in 1988, 10 times the US investment figure for the year.
S-JAP3:	Japan's swelling investment in Southeast Asia is part of its economic evolution.

Figure 5.8. T-JAP, a simple three-sentence source text.

could not be fully analysed by the CLE, which failed to interpret the number of dollars of Japanese investment, or the relationship between this and the U.S. investment figure. Instead, it produced analyses of three fragments of the sentence: 'In Thailand … approved $' (yielding predications 10–20); 'Japanese investment' (predications 21–23); and 'times the … year' (predications 24–32). The analysis was unable to produce QLF for '705 million' or 'in 1988, 10', so there are no predications involving any of these numbers. Note that in the phrase 'Board of Investment', 'Investment' was interpreted as a proper name, whereas elsewhere in the text it was interpreted as a common noun, with the result that entity **K** has a different semantic head from entities **B**, **M**, **P** and **T**. In addition, the wrong meaning of 'times' has been chosen (with the predicate **time_Occasion**) leading to the incorrect predications 29 and 30.

Analysis of S-JAP3 produced a single QLF, but with two errors. Firstly, the possessive 'Japan's' was incorrectly parsed, and instead of a **possessive** predication, 'Japan' and 'investment' have been combined with a **nn** (noun-noun compound) predication (number 36). Secondly, 'swelling investment' has been misinterpreted not as an investment that swells, but an investment in swelling (whatever that may be). Therefore instead of entity **T** (investment) being the second argument of the verbal predication (number 35), we have an unspecified entity **W** as the subject of 'swelling', and the swelling itself (**V**) is related to the investment by another **nn** predication (number 37).

These errors are typical of the mistakes made in CLASP's CLE-based analysis; such mistakes are inevitable, and CLASP is designed to function despite them. The consequences for building the predication-cohesion graph are not severe, as the majority of simple predications are correct, and there is no means by which errors are propagated to other parts of the representation. Nor are incorrect predications a specific problem in condensation, since CLASP's condensation stage considers only the pattern of cohesive links, not the predications themselves. However, if incorrect predications are selected for the summary representation, there is a possibility that CLASP may generate incorrect or nonsensical summary phrases in the synthesis stage.

SIMPLE PREDICATIONS

(S-JAP)

```
 1 propel_TransitiveVerbMRC(A, B, D)      2 investment_NounMRC(B)
 3 japanese_AdjectiveMRC(B)               4 name_of(C, Southeast Asia)
 5 in(B, C)                               6 region_NounMRC(D)
 7 integration_NounMRC(E)                 8 economic_Financial(E)
 9 toward(D, E)
```

(S-JAP2)

```
10 approve_TransitiveVerbMRC(F, G, H)    11 name_of(I, Thailand)
12 in(F, I)                              13 example_NounMRC(J)
14 for(F, J)                             15 board_Predicate(G)
16 name_of(K, Investment)                17 genitive(G, K)
18 government_PeopleInCharge(L)          19 possessive(G, L)
20 dollar_AmountOfMoney(H)               21 investment_NounMRC(M)
22 japanese_Predicate(N)                 23 nn(M, N)
24 figure_Number(O)                      25 investment_NounMRC(P)
26 nn(O, P)                              27 name_of(Q, United States of America)
28 nn(O, Q)                              29 time_Occasion(R)
30 nn(O, R)                              31 year_TimeMeasure(S)
32 for(O, S)
```

(S-JAP3)

```
33 investment_NounMRC(T)                 34 japan_Predicate(U)
35 swell_IntransitiveVerbMRC(V, W)       36 nn(T, U)
37 nn(T, V)                              38 name_of(X, Southeast Asia)
39 in(T, X)                              40 part_Portion(Y)
41 be(T, Y)                              42 evolution_NounMRC(Z)
43 economic_Financial(Z)                 44 genitive(Y, Z)
```

HEAD PREDICATIONS AND SEMANTIC HEADS

```
A:  1, propel_TransitiveVerbMRC          B:  2, investment_NounMRC
C:  4, Southeast Asia                    D:  6, region_NounMRC
E:  7, integration_NounMRC               F: 10, approve_TransitiveVerbMRC
G: 15, board_Predicate                   H: 20, dollar_AmountOfMoney
I: 11, Thailand                          J: 13, example_NounMRC
K: 16, Investment                        L: 18, government_PeopleInCharge
M: 21, investment_NounMRC                N: 22, japanese_Predicate
O: 24, figure_Number                     P: 25, investment_NounMRC
Q: 27, United States of America          R: 29, time_Occasion
S: 31, year_TimeMeasure                  T: 33, investment_NounMRC
U: 34, japan_Predicate                   V: 35, swell_IntransitiveVerbMRC
W: (no semantic head)                    X: 38, Southeast Asia
Y: 40, part_Portion                      Z: 42, evolution_NounMRC
```

Figure 5.9. Simple predications and semantic heads extracted from text T JAP (figure 5.8).

Identity links:	
argument link	10–17 (**G**)
predicate link	8–43 (**economic_Financial**).
Similarity links:	
similar argument link	1–26 (**B**, **P**: **investment_NounMRC**)
similar head-predication link	21–25 (**M**, **P**: **investment_NounMRC**)
Semantic stem links:	
stem-similar argument link	17–23 (**K**, **M**: **investment**)
stemmed predicate link	3–22 (**japanese**)
stem-similar head-predication link	16–21 (**K**, **M**: **investment**)

Figure 5.10: types of cohesive links, with examples from the predictions for T-JAP (figure 5.9).

5.4 CONSTRUCTING THE PREDICATION COHESION GRAPH

After extracting simple predications from all the sentences in a source text, we have a (potentially very large) set of predications with no overall structure. These will be the nodes of the *predication cohesion graph*, which we now have to construct, by identifying the *cohesive links* that will be its edges. As we saw in section 4.2.2, these links indicate attentional connections between predications, arising from similarity of predicates and/or arguments.

Figure 5.10 shows the types of cohesive links used by CLASP, in three categories: *identity*, *similarity* and *semantic stem links*. Section 5.4.1 defines the notions of similarity and semantic stemming, and section 5.4.2 defines the types of links themselves. Section 5.4.3 presents the predication cohesion graph for the T-JAP example text.

5.4.1 *Identity, similarity and semantic stemming*

Consider the T-JAP text (figure 5.8), and the simple predications extracted from it (figure 5.9). Predications 1 and 3 both express ideas related to entity **B** (semantic head **investment_NounMRC**): the idea of it being Japanese and the idea of it propelling something. This connection is a *cohesive link*, arising because two predications have a common argument; similarly, links arise when two predications have distinct arguments but share the same *predicate*. For example, predications 8 and 43 have distinct arguments but both express the idea of something being connected with the economy.

Less clearly related are predications 1 and 41. They both express ideas about Japanese investment in Southeast Asia (that it is propelling something, and that it is part of something), but they use different arguments (**B** and **T**) to represent this single concept. If CLASP could resolve anaphors, it would replace **T** by **B** and find a cohesive link like that between predications 1 and 3; since CLASP cannot resolve anaphors, however, we need to identify a link between these

predicate or proper noun	semantic-stem
`propel_TransitiveVerbMRC`	`propel`
`propel_IntransitiveVerbMRC`	`propel`
`economic_Financial`	`economic`
`Southeast Asia`	`asia`
`United States of America`	`america`
`america_Nation`	`america`
`japan_Predicate`	`japan`
`Japan`	`japan`

Figure 5.11: some examples of semantic-stems.

predications even though their arguments are different.

Now consider predications 25 and 26. These also say something about investment, but not about Japanese investment: rather they involve entity **P**, which is in fact U.S. investment. Nevertheless, they are clearly related to the predications about Japanese investment we have just been discussing, and we should identify a cohesive link between, for example, predications 1 and 25. That is, we want to construct cohesive links based on *similarity* of arguments, without requiring *identity*.

CLASP defines two entities to be *similar* if they have the same semantic head (but are not identical). This definition allows CLASP to find links in both the above examples, since entities **B**, **T** and **P** are all similar.

However, entities **B** and **K** are not similar, even though they both correspond to the word 'investment'. This is because for entity **K**, investment was interpreted as a proper name, giving the semantic head **Investment** (rather than **investment_NounMRC**).

To deal with such cases, CLASP has a notion of *semantic stemming*, applicable to predicates and proper nouns. There are two good reasons for introducing this idea: one is to allow CLASP to compensate for errors made in the CLE's analysis, such as confusion between proper and common nouns (as with entities **B** and **K**), or an incorrect choice between word senses. The other is to reflect the fact that genuinely different predicates and proper nouns may in fact be related: a word used as both a transitive and an intransitive verb, for example, or 'America' and 'North America'. The semantic stem of a predicate is defined to be the predicate with the underscore and everything after it (i.e. the sense or part-of-speech information) removed. The semantic stem of a proper noun is the last word in the proper noun, converted to lower case. Figure 5.11 gives some examples of predicates and proper nouns, and their semantic stems.

Since semantic heads are either predicates or proper names (see section 5.3.5), we can apply semantic stemming to them; thus we define two entities to be *stem-similar* if they have semantic heads with the same semantic stems (but are not *similar*). Under this definition, **B** and **K** are stem-similar (with semantic stem **investment**).

5.4.2 *Types of cohesive links*

Broadly speaking, cohesive links are identified *whenever predicates, arguments or head-predications are identical, similar or stem-similar.* The exception is that some predicates, rather than reflecting content-words in the source text, arise from particular syntactic structures: such predicates include prepositions, **nn**, **possessive** and **genitive**. We do not want to identify links between predications simply because of the occurrence of these common predicates, so CLASP has a list of such predicates, the *link stop-list*, and disregards them when searching for cohesive links.

We now define the seven types of cohesive links, with reference to the predications obtained from T-JAP (figure 5.9) and the examples in figure 5.10. Except where a *similar* or *stem-similar head-predication link* is involved (see below), different types of link are identified independently, and there can be any number of links between the same two predications.

Argument links

If the same atom or entity variable appears as an argument of two predications, there is an *argument link* between those predications. Note that there is no requirement that the common variable or atom appear in the same argument position in the two predicates: for example, **G** is the second argument in predication 10 and the first in predication 17. No argument link exists when the common argument is the name of a predicate on the link stop-list, or when it is a variable whose semantic head is on the link stop-list. Because predications from different sentences involve different variables, there will not be many inter-sentence argument links.

Predicate links

A *predicate link* exists between any two predications which have the same predicate, unless that predicate is on the link stop-list. There is a predicate link between predications 8 and 43, because they both have the predicate **economic_Financial**. Predications 21 and 33 both have the predicate **investment_NounMRC**, but these predications have a *similar head-predication link* which replaces the predicate link (see below). Predications 5 and 39 both have the predicate **in**, but there is no predicate link between them because **in** is on the link stop-list.

Similar argument links

A *similar argument link* exists between any two predications which have two *similar* but not identical entities occurring as arguments, unless the semantic-head of the entities is on the link stop-list. (Intuitively, this is like an *argument link*, but weaker.) Entities **B** and **P** are similar, so there is a similar argument link between predications 1 and 26. There is no such link between predications 2 and 25, however, because a *similar head-predication link* replaces it.

Similar head-predication links

A *similar head-predication link* exists between any two head-predications of similar entities. **M** and **P** are similar entities, so there is such a link between predications 21 and 25.

Predications 21 and 25 would seem to be linked in two ways even without this definition: they have the same predicate and similar arguments. However, these links are hardly independent, as the predicate and the semantic head of the arguments are the same: **investment_NounMRC**. Similarly, two head predications of proper nouns (such as predications 4 and 38) would seem to have both a similar argument link (entities **C** and **X**) and an argument link, but in fact the argument they have in common (the atom **Southeast Asia**) is the semantic head of the similar entities.

The reason for introducing similar head-predication links is that we do not want to restrict CLASP's condensation stage to considering these multiple links as independent relations: it is better to represent this common case in the predication cohesion graph as a different kind of link. Therefore, when a similar head-predication link is identified, it replaces the predicate and similar argument links that would otherwise be found.

Stem-similar argument links

These are identical to similar argument links, except that they join predications having as arguments variables or atoms which are only *stem-similar* (i.e. which have semantic heads with the same semantic stem). Entities **K** and **M** are stem-similar, so there is such as link between predications 17 and 23. There is no such link between predications 16 and 21, because a *stem-similar head-predication link* (see below) replaces it.

Stemmed predicate links

These are identical to predicate links, except that they join predications whose predicates are not identical, but have the same *semantic stem*. There is such a link between predications 3 and 22 of figure 5.3, because both have predicates with semantic-stem **japanese**.

Stem-similar head-predication links.

These are identical to similar head-predication links, except that they join head predications of entities that are only *stem-similar*. As with similar head-predication links, such a link replaces the stem-similar argument links and stemmed-predicate links that would otherwise be found. Note that whereas a similar head-predication link can only exist between head predications of two common nouns, two proper nouns or two verbs, a stem-similar head-predication link can exists between predications of different types. For example, entities **K** (a proper noun) and **M** (a common noun) are stem-similar, so there is a stem-similar head-predication link between predications 16 and 21.

5.4.3 *The predication-cohesion graph for* T-JAP

Identifying the cohesive links between simple predications forms the predication cohesion graph, and completes CLASP's analysis stage. Typically, some parts of the graph have fairly few cohesive links, while in other parts, corresponding to concepts frequently mentioned in the source text, the graph is very dense. In particular, when a large number of entities are *similar* or *stem-similar*, and when there are many predications about each one, the graph will contain large cliques – clusters of nodes each of which is connected to all the others. For this reason, it is difficult to draw the predication-cohesion graph for even a fairly small example without there being a dense tangle of edges.

Figure 5.12 shows the overall structure of the predication-cohesion graph for T-JAP. Note that, for brevity, *only the semantic-stems of the predicates are shown, and not every cohesive link is drawn explicitly*. The solid black edges indicate the *argument links* that occur within sentences. The four shaded regions enclose predications related to the predicate stems **`japanese`**, **`investment`**, **`Asia`** and **`economic`**. Within each of these regions every predication is connected to every other (though by various kinds of link); there are no other inter-sentential links.

Of course the T-JAP text was chosen to illustrate the different kinds of links, and therefore has a very high number of cohesive links. Nevertheless, for texts with one main topic, we should expect quite dense predication cohesion graphs – for example, the majority of the sentences in the JAPINV text refer to one or more of 'investment', 'Asia' or 'Japan'.

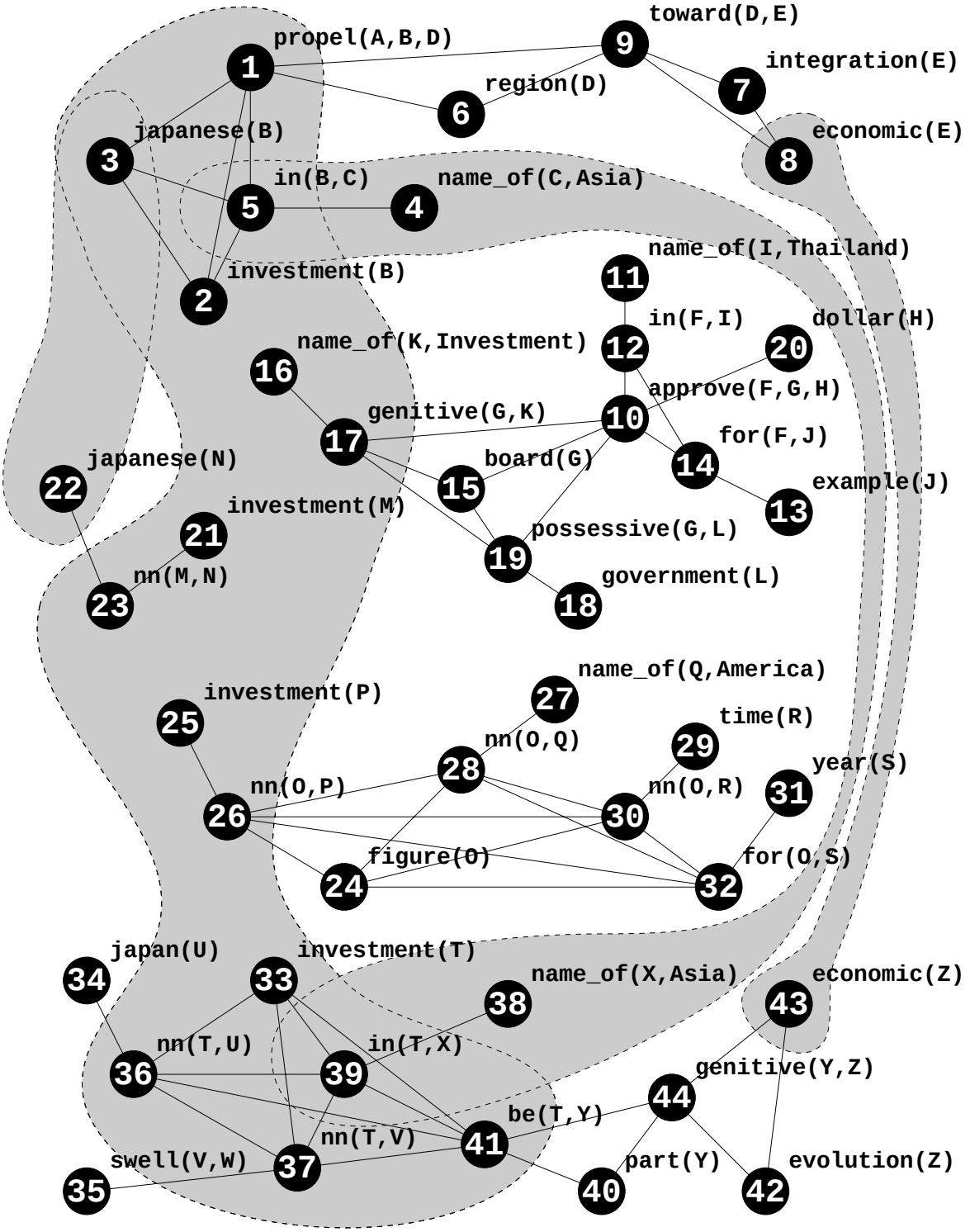


Figure 5.12: predication cohesion graph for the T-JAP example text (simplified).

6 CONDENSATION IN CLASP

The previous chapter described how CLASP processes source text to construct the source representation – a predication cohesion graph. This chapter describes the next stage of processing, *condensation*, in which a summary representation, i.e. a set of selected simple predications, is produced. The intuition on which this stage depends is that the predication-cohesion graph tells us something about which predications are related to which others, and how strongly, and that by considering the pattern of these relations we can discover which predications are central to the text (with many, strong relations) and which are peripheral (with few or weak relations).

The details of the processing described here are of course specific to CLASP: however, the central ideas – the three criteria of *importance*, *representativeness* and *cohesiveness,* and the graph-based scoring functions that reflect them – are more widely applicable. In fact, CLASP's graph-based selection algorithms could be applied to other kinds of source representation consisting of weighted graphs, such as TELE-PATTAN's graphs of lexical cohesion (Benbrahim and Ahmad 1994) or Taylor's (1975) weighted semantic structures. All that is required is a graph structure from which a summary representation can be formed by *selection*, and for which the intuition described above holds.

CLASP's condensation stage involves three steps. Section 6.1 describes how the predication-cohesion graph is turned into a *weighted graph*. Section 6.2 defines the *scoring functions* that CLASP uses to measure how good individual sets of nodes (i.e. predications) would be as summary representations. Section 6.3 describes how selection of high-scoring sets of predications is done using a *greedy algorithm*. A difficulty in applying these techniques is that there are many parameters to be set for which the context factors and the nature of the source representation give us no obvious values. We have therefore to try a range of plausible possibilities and see what happen. Here I present a number of alternatives; they are investigated in chapter 8.

6.1 PRODUCING A WEIGHTED GRAPH

To recap, the source representation consists of simple predications joined by cohesive links. These links are of various kinds (described in section 5.4.3) depending on whether they are between predications with the same or similar arguments or predicates. Although we expect predications may well be linked when there is *some* semantic relation between them, the links themselves have no 'meaning' – they do not tell us in what way one predication is related to another.

An example fragment of a source representation is shown in figure 6.1. This graph consists of some of the predications and links obtained from the S-JAP example sentence (whose analysis was considered in chapter 5) and another

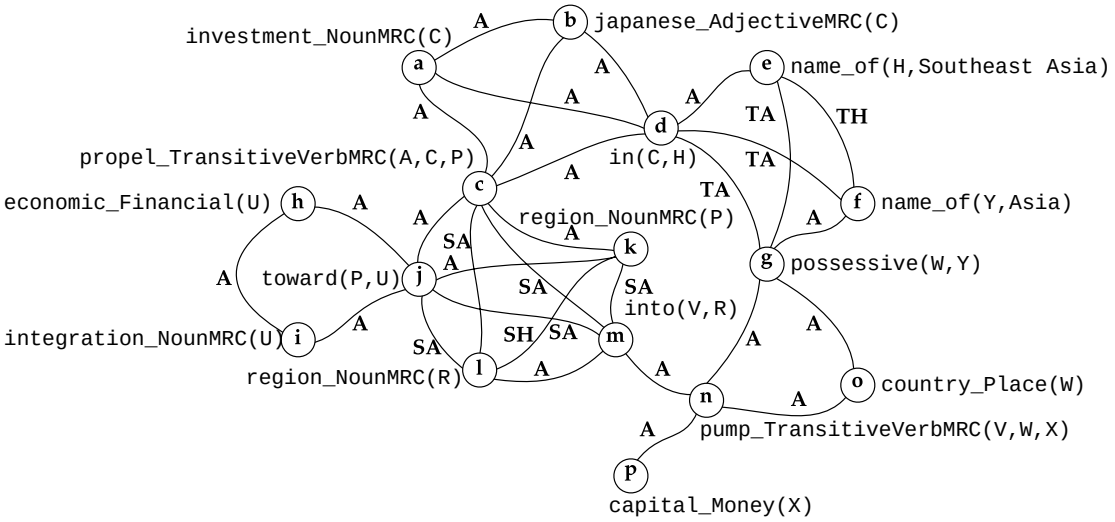


Figure 6.1: part of the predication-cohesion graph for S-JAP ('Japanese investment in Southeast Asia is propelling the region toward economic integration.') and S-JAP4 ('Asia's other cash-rich countries are following Japan's lead and pumping capital into the region.') Links are labelled as follows:

A	argument link
SA	similar argument link
TA	stem-similar argument link
SH	similar head-predication link
TH	stem-similar head-predication link

sentence (S-JAP4) from the JAPINV text. (Recall that *similar argument links* join predications involving arguments with the same semantic head, such as **P** and **R**, whereas *stem-similar argument links* join predications involving arguments whose semantic heads have the same stem, such as **H** and **Y**.)

It seems reasonable to suppose that we can determine something about the *strength* of the relation between two predications from the links between them: we would expect that, for example, predications with many arguments in common might be 'more strongly related' than those with only one argument in common. Without any idea of what the relations mean, however, this notion of more and less strongly related predications, though it is natural and intuitive, must remain informal and non-rigorous. Nevertheless, it is crucial to CLASP's condensation stage, which begins by constructing a *weighted graph*. This is identical to the predication-cohesion graph, except that between each pair of nodes, instead of multiple links of various types, there will be at most one *edge*, which is given a positive numerical weight. The weight is intended to indicate the strength of the relation between the two predications. Edge weights are

greater than zero because any relation, however weak, is stronger than none at all. Once it has constructed such a graph, CLASP's condensation stage uses the edge weights only, and pays no further attention to the individual cohesive links from which they were derived.

Types of links and multiplicity

There are two questions to be answered in assigning edge weights. Firstly how to treat the different kinds of links, and secondly what to do when there are multiple links between predications.

As a starting point, consider the 'null hypotheses' that there is no difference in strength between any of the different kinds of links, and that multiple links between predications indicate no stronger a relation than single links. Following these hypotheses, we would simply place an edge of equal weight (1, say) between every two cohesively-linked predications.

On the other hand, it is plausible that the presence of multiple links between two predications indicates a stronger relation than a single link (just as a single link indicates a stronger relation than no link at all). It is also plausible that, even where only one link is involved, some types of cohesive links indicate a stronger relation than others – for example, perhaps argument *identity* links indicate stronger relations than argument *similarity* links.

Following these intuitions, CLASP can assign different edge weights depending on the type of link. For example: 1 for argument and predicate links, 0.9 for similar argument and similar head links, and 0.8 for stem-similar argument, predicate and head links. (The size of the numbers chosen is not important; what will be considered in selecting predications is the *relative* strength of links.) Figure 6.2 shows the result of applying these weights to the example graph in figure 6.1. When there is more than one link between predications, CLASP can either choose the edge weight corresponding to the strongest link (the null hypothesis) or add up the weights from the individual links to get a larger total edge weight.

The best choices to be made here are a matter for experimentation, and may be affected by our choice of scoring function later on. In chapter 8, I describe experiments with the possibilities just mentioned, and in addition using graphs in which certain kinds of links are disregarded entirely.

6.2 SCORING SETS OF NODES

CLASP's scoring functions are intended to reflect three specific criteria for a good summary: *importance*, *representativeness* and *cohesiveness*. By importance I mean that the summary should include content that is important in the context of the source text. By representativeness I mean that the summary should reflect the content of the whole source text, not just a part of it. By cohesiveness I mean that the summary is a whole, rather than consisting of unrelated parts.

Importance and representativeness are two of the properties of reflective summaries, as defined in section 1.3. Cohesiveness is additionally desirable

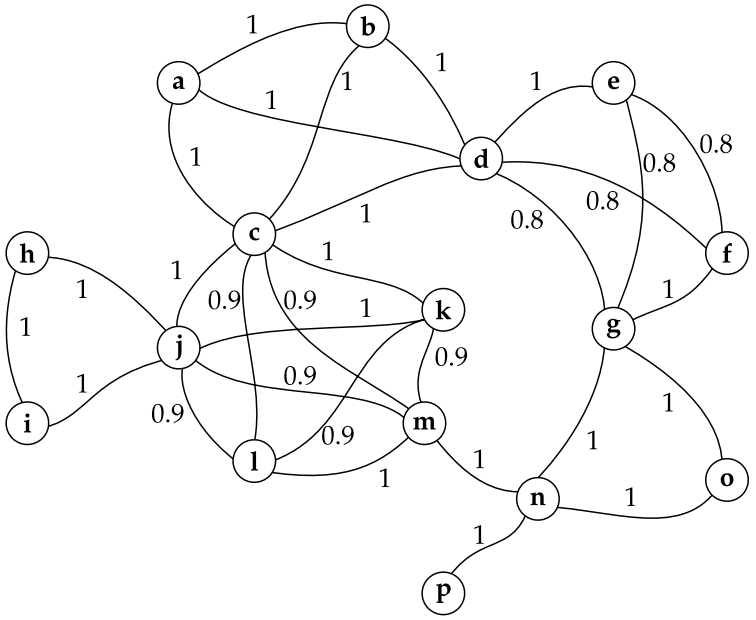


Figure 6.2: a simple weighted graph, obtained from figure 6.1 by giving stem-similar argument, predicate and head-predication links a weight of 0.8, similar argument and head links a weight of 0.9, and argument links a weight of 1. See figure 6.1 for the actual predications involved.

---

because it will tend to lead to clearer and more readable summaries.

Interpreting these criteria in terms of a weighted graph, we can say that information is important if it is highly relevant to the rest of the text, so important predications should be those with many and strongly weighted connections to the rest of the graph. For a set of predications to produce a representative summary, they should be, between them, connected to as much of the graph as possible. And to produce a cohesive summary, we should choose predications which are related to each other.

In this section, I present scoring functions which attempt to quantify the extent to which sets of nodes satisfy these three criteria, independent of constraints imposed by our ability to generate summary text, or additional output requirements (length in particular) that may be imposed later. To illustrate, I will refer to the example graph in figure 6.2. This is of course only a small fragment of an actual weighted graph for a whole source text, but for the purposes of explanation we will consider it as a whole graph on its own. A few concepts from graph theory (neighbourhoods, paths and path weights) will be introduced as they are required.

Because our requirements are intuitive and not precise, and because our representation captures so little about the underlying relations in the source

text, there is no hope of elegantly translating the criteria into mathematical formulae or an algorithm. As a result, all the scoring functions I present are very *ad hoc*, and justified only on the grounds that they seem reasonable or intuitive. In this sense, they are not a 'solution' to the summarising problem, but more like heuristics which we hope but cannot guarantee will guide us towards better summaries. In this respect, CLASP's condensation stage follows in the tradition of the attentional-network summarisers described in section 2.1. I claim, however, that CLASP's scoring functions are less *ad hoc* than most of those systems because they are *explicitly* intended to reflect particular desirable properties of a good summary.

6.2.1 *Importance scoring*

Measuring local importance

We intuit that the more and the stronger edges incident at a node, the more relevant the corresponding predication will be to the source text. This suggests scoring individual nodes for importance by adding up the weights of all the edges from them. Because this considers only edges immediately adjacent to a node, it is a measure of *local importance*. We can score sets of nodes for importance by adding up the importance scores of all the nodes in the set, and dividing by the sum of the importance scores of all the nodes in the graph. The resulting score varies from 0 (the empty set of nodes, or a set of nodes with no edges) to 1 (all the nodes), and reflects the proportion of predications, weighted by importance, included in the set.

Computing the score for a set of nodes by adding the importance scores of all the nodes in the set presupposes some kind of independence of content between nodes. If two nodes carry the same information, for example, then clearly selecting both of them does not give us a summary with twice as much important content. In CLASP's case, the analysis stage guarantees that the same predication will not occur twice in the graph; beyond this, however, there is no easy way to tell to what extent different predications may overlap in content (and certainly no way to tell from the predication-cohesion graph alone). Therefore we make the assumption of independence simply because at this stage we don't know any better. (In chapter 7, we will see how in some cases CLASP's synthesis stage is able to avoid generating summary phrases that give no new information.)

Writing G for the set of nodes in our weighted graph, we can represent the edges by a function $w : G \times G \to \mathbf{R}$, where $w(g, g) = 0$, and $w(g, h) = w(h, g)$ is the weight of the edge between nodes **g** and **h**, or zero if there is no such edge. The sum of edge weights at a node g, which we will call $\sigma_1(g)$, is then defined:

$$\sigma_1(g) = \sum_{h \in G} w(h, g).$$

For subsets $H \subseteq G$, we define $\sigma_1(H) = \sum_{h \in H} \sigma_1(h)$. Then our first measure of importance, $\mathbf{imp}_1$, is defined as follows:

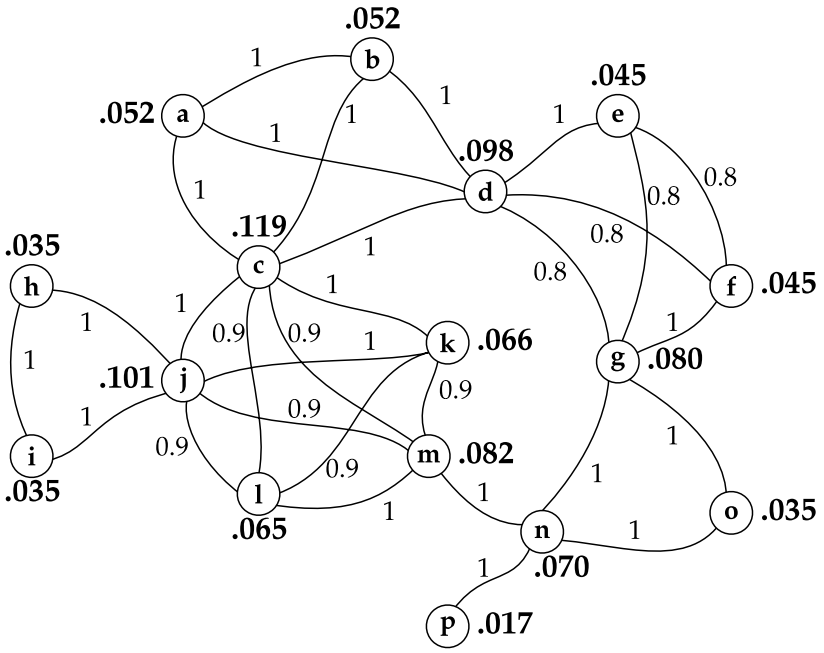


Figure 6.3: importance scoring with $\mathbf{imp}_1$.

$$\mathbf{imp}_1(H) = \sigma_1(H) / \sigma_1(G).$$

The value of $\mathbf{imp}_1$ on our example graph is shown in figure 6.3, where each node g has been labelled with $\mathbf{imp}_1(g)$.

This measure of importance is related to measures used by Skorochod'ko (1971) and Benbrahim and Ahmad (1994) in their summarising systems. Skorochod'ko's measure of *local significance* is exactly the function $\sigma_1(g)$ above, and Benbrahim and Ahmad use the same function to find *central sentences*. In both these approaches, however, the edges of the graph all have weight 1. An important addition in Benbrahim and Ahmad's work, however, is that their graphs are *directed* – edges are oriented according to the order of sentences in the source text. As well as finding central sentences, they also count separately the number of bonds to earlier and later sentences, in order to find 'topic opening' and 'topic closing' sentences in the source text. CLASP does not consider this kind of information (presentation order) in its condensation.

Less local importance measures

A limitation of $\mathbf{imp}_1$ is that it considers only the edges immediately incident at a node, and does not look at the nodes to which they lead. For example, in figure 6.3, predications **h** (`economic_Financial(U)`) and **o** (`country_Place(W)`) both have two edges of weight 1 going from them, so $\sigma_1(\mathbf{h}) = \sigma_1(\mathbf{o})$. But we might

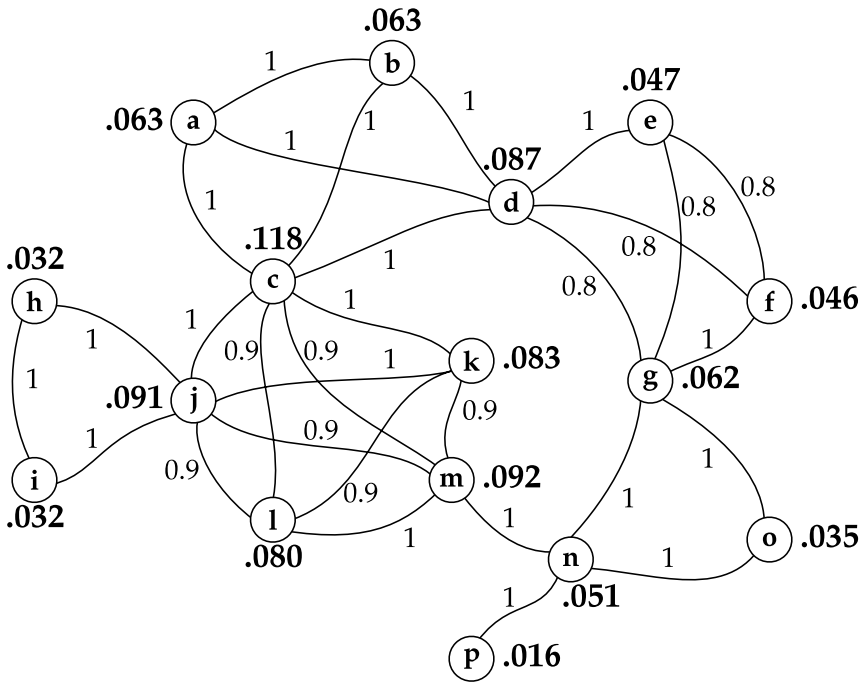


Figure 6.4: importance scoring with $\mathbf{imp}_2$.

---

argue that because the one of the nodes connected to **h** is itself not very important, we should give **h** a lower importance score than **o**. In general, being connected to nodes which themselves have many strong connections is more likely to indicate importance than being connected to nodes which have relatively few or weak connections. Following this intuition, we can establish a second measure of importance for sets of nodes. First, we define:

$$\sigma_2(g) = \Sigma_{h \in G}\, \sigma_1(h)\, w(h, g).$$

That is, we multiply the weight of each edge incident at g by the sum of edge weights at the node it leads to (i.e. $\sigma_1(h)$), and add up the resulting products. As before, we define $\sigma_2(H) = \Sigma_{h \in H}\, \sigma_2(h)$, and normalise to obtain a new measure of importance:

$$\mathbf{imp}_2(H) = \sigma_2(H) \,/\, \sigma_2(G).$$

Figure 6.4 shows the result of applying $\mathbf{imp}_2$ to the graph in figure 6.2; many of the relative importance scores of predications have not changed, but predication **m** now scores higher than **j**, whilst **k** and **l** have overtaken **g**, and predication **o** now has a higher score than **h** or **i**.

There is no need to stop at $\mathbf{imp}_2$. We can easily define a whole family of

measures of importance:

$$\sigma_0(g) = 1,$$

$$\sigma_n(g) = \sum_{h \in G} \sigma_{n-1}(h)\, w(h, g),$$

$$\sigma_n(H) = \sum_{h \in H} \sigma_n(h), \quad \text{for } H \subseteq G,$$

$$\mathbf{imp}_n(H) = \sigma_n(H) \,/\, \sigma_n(G).$$

Another way to think of the functions σ_n is as a sum of *path weights*. Let a *path* of length l be a sequence of $l+1$ nodes connected by edges (for our purposes it is acceptable, unlike in some definitions of path, for a node to occur more than once in the sequence), and let the *weight* of such a path be the product of the weights of the edges involved. Then $\sigma_1(g)$ is the sum of the weights of all paths of length 1 ending at g, $\sigma_2(g)$ the sum of the weights of all paths of length 2 ending at g, and $\sigma_n(g)$ the sum of the weights of all paths of length n ending at g. These functions are progressively less local: computing σ_n involves considering all the edges within distance n of a node.

The effect of cliques

As we increase the value of n, $\mathbf{imp}_n$ becomes less dependent on the edges immediately incident at a node, and more dependent on the structure of the rest of the graph. One effect in particular is noticeable: if we have a large *clique* of nodes (i.e. a set of nodes all of which are connected to each other) then they reinforce each other's importance scores. If we set n too high, $\mathbf{imp}_n$ will give a high score to all the nodes in the clique, and a low score to other nodes. In figure 6.2, there are two cliques of four nodes {**a**, **b**, **c**, **d**} and {**d**, **e**, **f**, **g**}, and a clique of five nodes: {**c**, **j**, **k**, **l** and **m**}. As we increase n, the five-node clique will start to dominate. In fact, a complete clique is not required for this to happen: all that is needed is a cluster of nodes with many, strong connections between them.

In some natural language processing tasks, such as classification, algorithms are created specifically to look for such strongly-connected clusters in graphs. For example, in a graph of index-terms with edges corresponding to co-occurrence in a corpus of documents, one would look for clusters of terms corresponding to broad document topics. In searching for important nodes, however, we are not looking for clusters with strong internal connections, but rather for information that is important in the context of the whole graph. However, the measures of cohesiveness presented in section 6.2.3 do give high scores to such clusters.

Combining importance measures

In scoring for importance, we do not want to restrict ourselves to considering just one of the $\mathbf{imp}_n$ measures (that is, to looking at paths of just one length), so

as a scoring function for importance we will take a weighted combination of these measures:

$$\mathbf{imp}(H) = \sum_i a_i \, \mathbf{imp}_i(H).$$

The longer the paths we consider ending at g, the less we can reasonably say they indicate directly that node g is important; therefore we should give greater weight to the lower $\mathbf{imp}_i$, by ensuring that $a_1 \geq a_2 \geq a_3 \geq \ldots$. And to make sure that importance is measured on a scale from 0 to 1, we also choose coefficients such that $\sum_i a_i = 1$. To avoid allowing large cliques to dominate the importance scoring, we will need to set a cut-off point after which all the a_i are zero. In chapter 8, I report on experiments using various values of a_1, a_2 and a_3.

6.2.2 *Representativeness scoring*

The second of the three criteria for selecting nodes was that we want the summary to be representative of the whole text. That is, we want a summary with broad rather than narrow scope. When scoring sets of nodes for importance, we simply added up scores for individual nodes. Such an approach cannot be taken for representativeness, as it is a property of the summary as a whole. What we want to quantify is the amount of the graph to which a set of nodes is connected. We can do this by looking at *neighbours* and *neighbourhoods*.

By the *set of neighbours* of a set of nodes H, written $N(H)$, we mean the set of all nodes which are joined by an edge to some node in H. That is,

$$N(H) = \{ g \in G \mid \exists h \in H, w(g,h) \neq 0 \}.$$

Taking sets of neighbours n times, we obtain the set of nodes connected to H by a path of length exactly n, written $N^n(H)$ (i.e., $N(N(\ldots(H)\ldots))$).

The *n-neighbourhood* of a set of nodes H, written $B^n(H)$, is then defined as the set of all nodes reachable from H in n steps *or fewer*. Formally,

$$B^n(H) = H \cup N^1(H) \cup N^2(H) \cup \ldots \cup N^n(H).$$

A simple measure of representativeness would be to count the proportion of nodes in the 1-neighbourhood of the set H; that is, to measure $|B(H)| \,/\, |G|$. However, the criterion of representativeness is not simply that the summary be equally representative of all the material in the text; we would expect a good summary to be more representative of the important content than of less important material. So, once we have chosen a scoring function **imp** to measure the importance of sets of nodes (as in section 6.2.1), we can apply this function to $B(H)$ instead of simply counting its elements:

$$\mathbf{rep}_1(H) = \mathbf{imp}(B(H)).$$

(In fact, counting the elements of $B(H)$ is a simple instance of this approach, since $|B(H)| \,/\, |G| = \sigma_0(B(H))$.)

Figure 6.5 shows the results of applying $\mathbf{rep}_1$ to two sets of nodes from figure 6.2, taking $\mathbf{imp} = 3/4\ \mathbf{imp}_1 + 1/4\ \mathbf{imp}_2$. The sets in question are nodes **c**, **j** and **d** (the three predications with highest importance scores) and nodes **c** and **g**. Note that although the second subset contains fewer nodes and scores lower in importance than the first, it scores higher in representativeness: the two predications **c** and **g**, coming from different parts of the graph, have more important nodes in their neighbourhood than the three others, which are closer to each other and none of which is related to the high-scoring predication **n**.

Just as we extended $\mathbf{imp}_1$ to obtain less local importance scoring functions, we can extend $\mathbf{rep}_1$ to consider not just the immediate neighbourhood of the subset in question, but also more distantly related nodes. The obvious generalisation is:

$$\mathbf{rep}_n(H) = \mathbf{imp}(B^n(H)).$$

As with importance, we combine the $\mathbf{rep}_n$ into a single representativeness score, giving different weights to different neighbourhoods of H:

$$\mathbf{rep}(H) = \sum_i b_i\, \mathbf{rep}_i(H).$$

A set of nodes H is (by common sense) less representative of nodes the more distantly they are connected to it; therefore, once again, we will choose coefficients so that $b_1 \geq b_2 \geq b_3 \geq \ldots$, and $\sum_i b_i = 1$. The latter restriction ensures that, like importance, representativeness is measured on a scale from 0 to 1, with the maximum score attained when the set of nodes is connected to the whole graph, i.e. $B(H) = G$.

Monotonicity of scoring functions

A notable property of both importance and representativeness, as measured by these scoring functions, is that adding nodes to a set can never cause its score to decrease. This reflects the way we have stated the criteria: the summary should contain important content (we do not require it to contain *only* important content), and should be representative of important content from the whole graph. It will be up to the greedy algorithm of section 6.3 to impose the additional requirement of the summary's length. Maximising importance and representativeness scores subject to a restriction on the size of the set of nodes will have the effect of disfavouring summaries that contain unimportant content or content which adds little to the summary's representativeness.

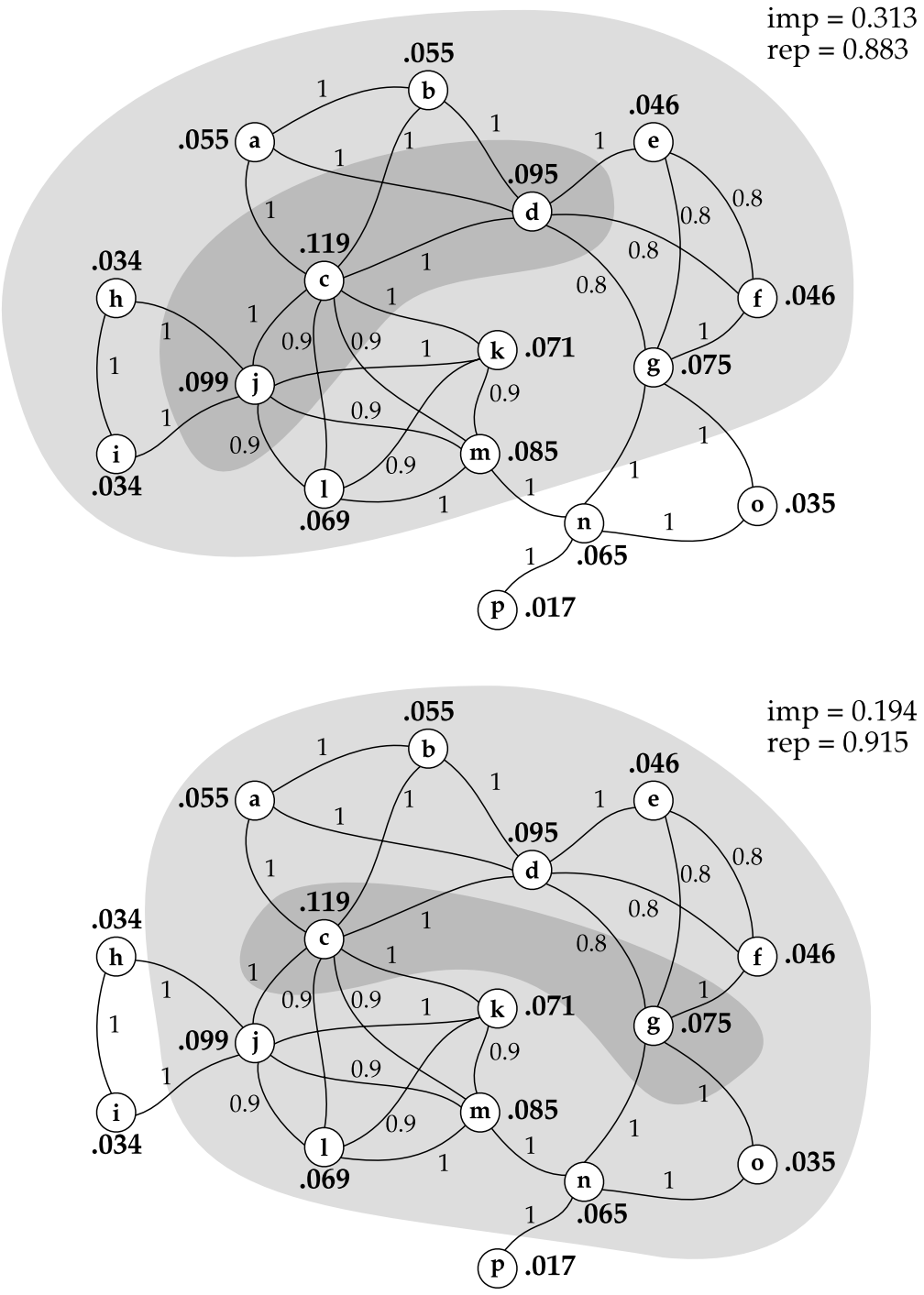


Figure 6.5: importance and representativeness scoring with **imp** and $\mathbf{rep}_1$. Two subsets H are shown, indicated by heavy shading; their neighbourhoods $B(H)$ are lightly shaded.

6.2.3 *Cohesiveness scoring*

In contrast to importance and representativeness, which are about a summary's relationship to the whole text, cohesiveness is about the summary's relationship to itself: it should seem to fit together as a whole rather than appearing disjointed. Like representativeness, cohesiveness is not something we can measure for each node, but rather a property of a set of nodes. However, we can measure for each node the extent to which it is connected to the selected set; we can then take this quantity, averaged over all the nodes in the set, as our measure of coherence.

In considering importance, we began by taking the sum of edge weights from a node. To measure cohesiveness of a node g with a set H, we take a similar approach, except that we *only* consider edges between g and nodes in H. By analogy with σ_n, we define, for a node g and a set H containing g:

$$\gamma_0(g, H) = 1,$$

$$\gamma_n(g, H) = \Sigma_{h \in H}\ \gamma_{n-1}(h, H)\ w(h, g),$$

$$\gamma_n(H) = \Sigma_{h \in H}\ \gamma_n(h, H) / |H|.$$

Note the important difference between these definitions and those for σ_n in section 6.2.1: whereas $\sigma_n(H)$ is a *sum* over elements of H, $\gamma_n(H)$ is an *average*. In measuring importance we are concerned with the total amount of importance content in the summary, but in measuring cohesiveness this approach would be wrong: adding unconnected information to a cohesive summary makes it less cohesive.

For each γ_n we obtain a measure of cohesiveness, just as each σ_n gave us a measure of importance, and as before we can combine the individual measures into an overall score:

$$\mathbf{coh}_n(H) = \gamma_n(H) / \gamma_n(G),$$

$$\mathbf{coh}(H) = \Sigma_i\ c_i\ \mathbf{coh}_i(H), \quad \text{for } c_1 \geq c_2 \geq c_3 \ldots, \Sigma\, c_i = 1.$$

Like our measures **imp** and **rep**, when applied to the empty set **coh** gives a value of 0, and when applied to the whole set, it gives a value of 1. Unlike the other scoring functions, however, $\mathbf{coh}(H)$ can be greater than 1, reflecting the fact that a summary can be more cohesive than the text it summarises, by omitting extraneous material which is included in the original text. Thus cohesiveness scores are not monotonic.

Figure 6.6 shows the value of $\mathbf{coh}_1$ on the same two subsets considered in figure 6.5. Individual nodes within the subsets are labelled with $\gamma_1(g, H)$. The cohesiveness score for the set {**c**, **d**, **j**} is quite high, considering that the subject

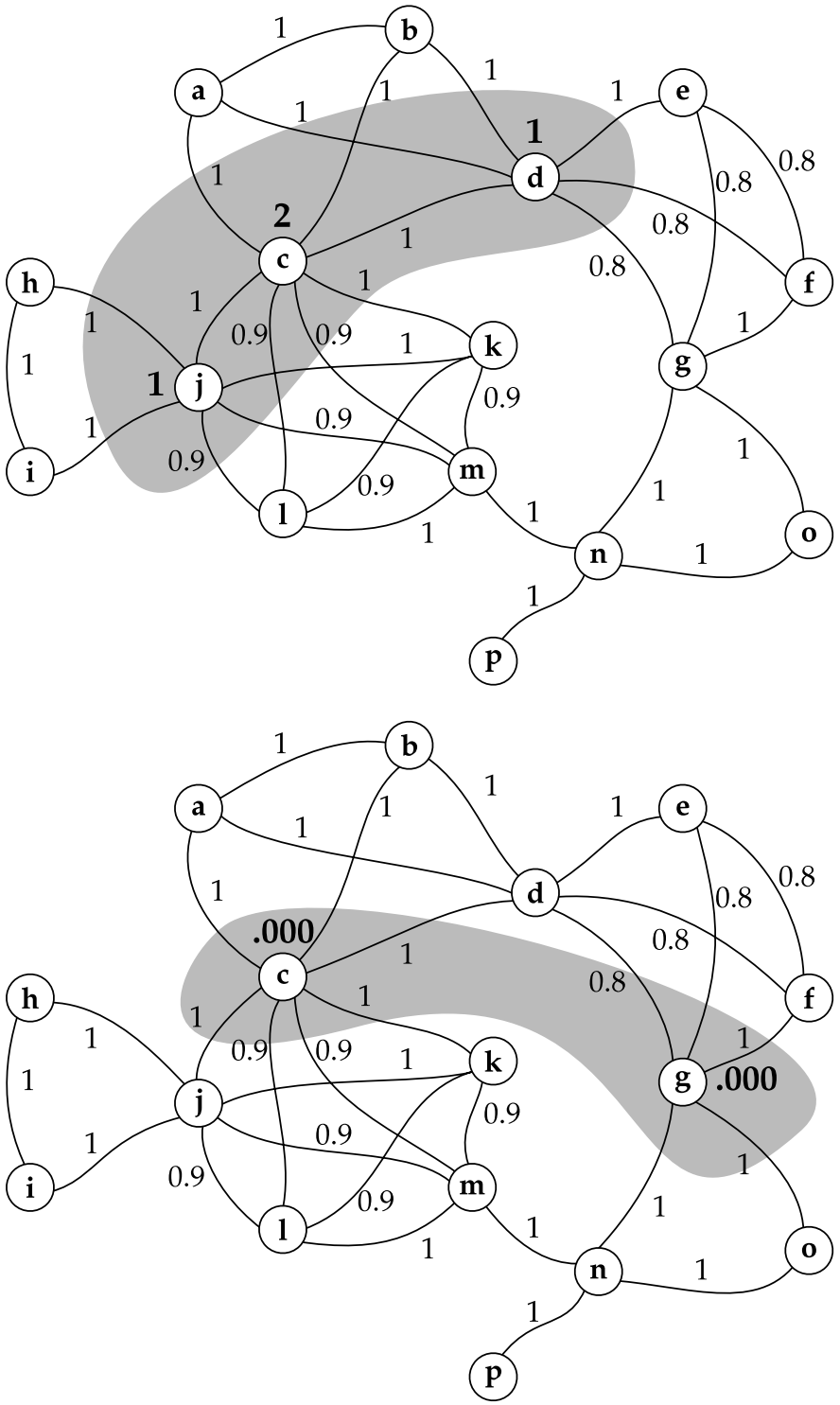


Figure 6.6: cohesiveness scoring with $\mathbf{coh}_1$. Two subsets H are shown, indicated by shading.

contains only three nodes. (A subset containing more nodes has the potential to achieve a higher cohesiveness score because the number of potential connections within the set increases. The clique **c**, **j**, **k**, **l**, and **m** for example, is very cohesive: $\mathbf{coh}_1 = 1.052$.) Because there is no edge between nodes **c** and **g**, the second example has a cohesiveness score of zero.

The balance between representativeness and cohesiveness

Figures 6.5 and 6.6 illustrate the trade-off between representativeness and cohesiveness as CLASP measures them. Measuring cohesiveness favours sets with many internal links, whereas measuring representativeness favours sets with many external links. This trade-off is not a problem with CLASP's graph-based summarising techniques, but rather reflects a choice between different kinds of summaries: in the terminology of section 1.1.2, a very cohesive summary will tend to have narrow scope, whereas a very representative summary will have broad scope.

6.2.4 *Combining the three criteria*

To decide on an overall scoring function to be used in condensation, we must choose a combination of the three criteria of importance, representativeness and cohesiveness that is appropriate for the summary's input, purpose and output factors. CLASP's general formula is simply a weighted sum of scores:

$$\mathbf{score}(H) = A\,\mathbf{imp}(H) + B\,\mathbf{rep}(H) + C\,\mathbf{coh}(H), \text{ where } A+B+C = 1.$$

Thus, while the choice of (a_i), (b_i) and (c_i) determines **imp**, **rep** and **coh**, the parameters A, B and C specify the overall balance between them. Although analysis of context factors may give us some idea of the relative importance of the criteria, in the end setting these parameters, and the others, must inevitably be done experimentally, by trial and error.

6.3 SELECTING SETS OF SIMPLE PREDICATIONS

The overall scoring function, **score,** is now used to select a high-scoring set of nodes of the required size, subject to whatever other constraints may be imposed.

As described in section 4.3.3, selection is performed by an algorithm, which, rather than considering every possible set of predications, proceeds by gradually adding predications to those it has selected so far. (For the example graphs in this chapter, we could indeed find the best set of predications simply by scoring all the possibilities, but when dealing with a whole story of even a hundred words, such an exhaustive search would become prohibitively slow.) It is a *greedy algorithm*, because it maximises the score at each step, and sticks to its earlier decisions even if changing them could lead to a higher scoring set later. There is no guarantee that it will select the 'best-possible' set of predications,

only that it will find a high-scoring set. An exception to this is that if only *importance* scoring is used (i.e. if A=1, B=0, C=0 in the formula of section 6.2.4), then the score for a set of nodes is simply a sum of individual node scores, in which case the algorithm will always select a maximally-scoring set of nodes.

Two kinds of selection

The 'other constraints' mentioned above depend on what kind of synthesis is being performed. In CLASP there are two possibilities in synthesis (described in section 4.4): either we generate summary phrases from predications, or we extract sentences of source text. In the first case, CLASP's condensation stage is free to select any combination of predications; the synthesis stage will do its best to generate corresponding surface text. In the second case, each source text sentence must either be included in the summary or not, so the condensation stage must either select or reject all the predications obtained from each original source sentence at once; this is *constrained selection*.

Thus the greedy algorithm either repeatedly adds individual predications to the selected set (for summary phrase synthesis); or it repeatedly adds the set of all predications corresponding to a source sentence to the selected set (for sentence extraction synthesis). In either case, the algorithm starts off with the empty set. At each stage, the algorithm compares the scores of the candidate sets of selected predications, and chooses the one with the highest score; a brief example is given in section 4.3.3.

Control of summary length

CLASP controls the length of its summaries by specifying how many steps of the greedy algorithm should be performed. Thus, the user must say how many predications should be selected, or, in the case of *constrained selection*, how many sentences should be extracted for the summary. Since the greedy algorithm proceeds by constructing a sequence of sets, each one a subset of the next, it is then very easy to produce longer or shorter summaries if required, simply by taking one of the smaller selected sets, or running the algorithm for another iteration. Although CLASP does not operate in such an environment, the ability to quickly expand on or reduce a summary could be useful for a summariser running interactively. In addition, the order in which the algorithm selected the predications may provide useful clues as to their relative value, which can be exploited in the synthesis stage.

7 SYNTHESIS IN CLASP

The synthesis stage of the CLASP summariser has the task of taking a set of simple predications selected by the graph-based methods presented in the previous chapter, and producing summary text for them. CLASP can produce two quite different kinds of output: either a summary consisting of whole sentences extracted from the source text (section 7.1), or a list of short summary phrases, generated from simple predications via QLF, that indicate the main topics of the text (section 7.2). This chapter describes the processing involved in producing these two kinds of summaries, and discusses different strategies for ordering and presenting the summary material.

7.1 GENERATION BY SENTENCE EXTRACTION

When summaries are to be produced by sentence extraction, the condensation stage is constrained to select either all or none of the predications extracted from each sentence of source text (see section 6.3). Given the resulting set of selected predications, the synthesis stage simply prints out the corresponding sentences of source text.

7.1.1 *An example of sentence extraction*

If we ask for a five-sentence summary of the JAPINV text (given in full in appendix A), CLASP (with appropriate settings for the condensation stage) selects all the predications from sentences 5, 7, 33, 15 and 0, in that order. Reproducing these sentences in the order they came in the source text, we obtain the following summary (JAP-SE1):

> Japanese investment in Southeast Asia is propelling the region toward economic integration. In Thailand, for example, the government's Board of Investment approved $705 million of Japanese investment in 1988, 10 times the US investment figure for the year. Asia's other cash-rich countries are following Japan's lead and pumping capital into the region. Japan's swelling investment in Southeast Asia is part of its economic evolution. Japan not only outstrips the US in investment flows but also outranks it in trade with most Southeast Asian countries (although the US remains the leading trade partner for all of Asia).

As a summary, this is not bad (in fact it is a better-than-average example of CLASP's output, as we will see in chapter 8). There are, however, some problems with this kind of summary: not all of the information they present is particularly relevant, and sometimes discourse effects make the summary hard to understand out of context.

7.1.2 *Importance and conciseness*

Summaries produced by sentence extraction are typically not very concise, and sentences often contain a mixture of important and unimportant information. CLASP can make judgements about the importance of individual predications during condensation, but this *precision* is not reflected when the summary is produced by sentence extraction. A second limitation is that in simply reproducing sections of source text, we have no control over the *expression* of the selected content: we cannot shorten or simplify it. To make individual sentences more concise would require going beyond simple sentence extraction.

We might suppose that the most important sentences could be made more prominent by placing them at the start of the summary. Since CLASP's condensation stage begins by selecting the predications from a single sentences, and adds sentences one by one, we could output the sentences in the order in which they were selected. Reordering JAP-SE1 in this way, we get the following summary (JAP-SE2):

> In Thailand, for example, the government's Board of Investment approved $705 million of Japanese investment in 1988, 10 times the US investment figure for the year. Asia's other cash-rich countries are following Japan's lead and pumping capital into the region. Japan not only outstrips the US in investment flows but also outranks it in trade with most Southeast Asian countries (although the US remains the leading trade partner for all of Asia). Japan's swelling investment in Southeast Asia is part of its economic evolution. Japanese investment in Southeast Asia is propelling the region toward economic integration.

In this example, however, it is not clear that the order of selection of these five sentences is at all related to their true importance. Experience with CLASP suggests that on average, more important information *is* selected earlier; however, it also suggests that the effect of reordering sentences on readability and discourse structure is far more significant.

7.1.3 *Discourse effects and readability*

JAP-SE2 reads much less well than JAP-SE1: it begins with an example and does not state what it is an example of until later; and it talks of other countries 'following Japan's lead' before it has been explained what lead it is that Japan is taking. The effect we see here is that in JAP-SE1, sentences 0 ('Japanese investment …'), 5 ('In Thailand, for example, …') and 7 ('Asia's other cash-rich countries …') stand in some discourse relation to each other. In terms of discourse structure, we can say that sentence 5 is an elaboration or example of sentence 0, and sentence 7 comes after sentence 0 in a temporal sequence (and

s0 *Japanese investment in Southeast Asia is propelling the region toward economic integration.*
s1 Interviews with analysts and business people in the U.S. suggest that Japanese capital may produce the economic cooperation that Southeast Asian politicians have pursued in fits and starts for decades.
s2 But Japan's power in the region also is sparking fears of domination and posing fresh policy questions.
s3 The flow of Japanese funds has set in motion 'a process whereby these economies will be knitted together by the great Japanese investment machine,' says Robert Hormats, vice chairman of Goldman Sachs International Corp.
s4 In the past five years, Japanese companies have tripled their commitments in Asia to $5.57 billion.
s5 *In Thailand, for example, the government's Board of Investment approved $705.6 million of Japanese investment in* 1988, 10 *times the U.S. investment figure for the year.*
s6 Japan's commitment in Southeast Asia also includes steep increases in foreign assistance and trade.
s7 *Asia's other cash-rich countries are following Japan's lead and pumping capital into the region.*

Figure 7.1: sentences 0–7 of the JAPINV text.

there may be a causal relation between them). In JAP-SE2, no such relations are understood, because sentence 0 has been moved to the end of the summary.

However, we must not assume that the same relations apparent in JAP-SE1 are present in the source text: the selected sentences in their original context text may have altogether different relations. Figure 7.1 gives the first eight sentences of the JAPINV text (with the extracted sentences italicised).

Reading this text, it seems that s5 is an example not of s0, but of s4, and that s7 is also related to s4 more directly than it is to s0. However, s4 is itself related to s0 – it is an elaboration of the point that Japan is investing in Southeast Asia. (The other central point of s0, that this investment is propelling Southeast Asia towards integration, is elaborated on later in the story.) So in the source text there is an *indirect* relation between s0, s5 and s7, which means that we in fact benefit from keeping them in order in the summary.

Experience with CLASP indeed shows that it is quite common for extracted sentences to stand in a direct or indirect relation to each other, and I have found that summaries using the original source order are generally much clearer and comprehensible. In particular, using this order reduces the number of summaries produced with apparently unresolvable anaphoric references in them, as if the sentence containing the necessary antecedent is selected, it will appear first in the summary.

However, it is still quite likely that CLASP's extracts will contain anaphoric references to objects not previously mentioned in the summary, or will, by

the use of cohesive conjunctions, refer to other material that has not been included. Some techniques for improving the readability and coherence of sentence-extract summaries were discussed in chapter 2. One approach is *aggregation* (Paice 1990, Rush, Salvador and Zamora 1971), in which, when a sentence is selected which requires an antecedent, immediately preceding sentences are added until the group of sentences does not require any further antecedents. The system of Miike et al (1994) can in some cases alter the surface text to remove dangling connectives such as 'however'. Johnson et al (1993) have developed techniques for distinguishing between anaphoric and non-anaphoric pronouns and noun phrases. NetSumm (Preston and Williams 1994) solves discourse problems at the expense of length, by presenting the whole of the source text, with the extracted sentences highlighted (rather like figure 7.1).

Certainly, CLASP's sentence extraction summaries could be made more readable by applying such techniques, but they would not help to solve the problems of precision and conciseness identified in section 7.1.2 (indeed, the addition of more sentences would tend to make summaries less concise). Therefore I have instead concentrated on an alternative form of synthesis, the generation of summary phrases.

7.2 GENERATING SUMMARY PHRASES

CLASP's summary phrases are generated directly from selected simple predications. These summaries are not what we would normally call 'text', however, consisting simply of a list of items which may be noun phrases or simple sentences.

Ideally we would like to be able to generate a continuous text summary, but for CLASP this would be extremely difficult. As discussed in section 4.2.1, the simple predications of the source representation do not convey facts asserted in the source text, but only ideas mentioned in it. In contrast, the QLF produced by the Core Language Engine is intended to represent meaning, but (at least as used in CLASP) it does not include any kind of anaphor resolution or discourse modelling, and the analysis is often incomplete, resulting in a fragmentary source representation (in figures 5.8 and 5.9, for example, analysis of the sentence S-JAP2 produced three separate fragments.) Therefore it too does not capture the full logical meaning of the text.

Some previous systems have successfully generated full-text summaries. FRUMP (DeJong 1982), for example, by combining prescriptive condensation and rigid synthesis, was able to produce very readable summaries, although occasionally they would make factual claims not supported by the source text. Such an approach is not appropriate for CLASP, as it is intended to function without domain- or world-knowledge. More flexible summary generation is also possible if we have a much more complete (and deeper) representation – this is the approach taken by Maybury (1995), for example – but again this is not possible for CLASP.

As noted in section 4.4.2, the lack of a representation of logical meaning

does not actually stop us from generating whole sentences of summary text, but it does mean that the summary must be *indicative* rather than *informative*. To avoid misrepresenting the content of the source text, each sentence would have to say something along the lines of 'The source text says something about …'. Such a summary would be no better (and would certainly be longer) than a list of the things that the text 'says something about', which is exactly what CLASP's summary phrases are.

CLASP's summary phrases are different from the 'capsule overviews' of Boguraev and Kennedy (1997). Their overviews also consist of a list of short phrases indicating important ideas in the text, but they are obtained by simply extracting short strings of words from the source text. It would be possible in many cases to find short segments of source text corresponding to CLASP's simple predications, but by generating them directly, we hope to achieve simplification of expression, greater control of presentation, and more precise identification of important material.

7.2.1 *Example summary phrases*

Here is a summary of the JAPINV text produced by the summary phrase method (JAP-SP1):

> *This text says something about:*
>
> Japanese investment in Southeast Asia propelling the region toward economic integration,
> Japan's commitment in Southeast Asia including steep increases in foreign trade,
> Asia's cash-rich countries,
> Asian nations,
> America encouraging Japan.

As can be seen in this example, CLASP's summary phrases can contain rather more information than a single selected predication. Producing a summary like JAP-SP1 involves not just generating summary phrases from individual predications (section 7.2.2), but also choosing determiners or quantifiers for nominal entities (section 7.2.3), combining multiple predications into a single summary phrase, or *clustering* (section 7.2.4), and deciding what order to present the summary phrases in (section 7.2.5).

7.2.2 *Summary phrases for individual predications*

It is quite straightforward to produce summary phrases for many types of simple predications, by considering the predicate and the semantic heads of its arguments. The examples in figure 7.2, all taken from CLASP's analysis of JAPINV, show how CLASP treats various kind of predications. Because the summary phrases are to be put in a list and introduced by 'Something is said about:', they

type	predication and semantic heads	summary phrase
nominal	`name_of(A, 'South Korea')` `A: South Korea`	South Korea
verbal	`produce_Make(A, B, C)` `B: capital_Money; C: cooperation_NounMRC`	capital producing cooperation
adjectival	`economic_Financial(C)` `C: cooperation_NounMRC`	economic cooperation
prepositional	`in(A, B)` `A: investment_NounMRC; B: Southeast Asia`	investment in Southeast Asia
genitive	`genitive(A, B)` `A: fear_NounMRC; B: domination_NounMRC`	fear of domination
possessive	`possessive(A, B)` `A: commitment_NounMRC; B: Japan`	Japan's commitment
noun-noun compound	`nn(A, B)` `A: machine_Device; B: investment_NounMRC`	an investment machine

Figure 7.2: summary phrases for individual simple predications.

are all presented as simple noun phrases (in the case of verbal predications, we use the *-ing* form), irrespective of how the predications were expressed in the original source text.

The summary phrases are produced by constructing appropriate QLF expressions, and then using the Core Language Engine to generate surface text fragments from the QLF. (QLF was described in section 5.2.1 and its main points summarised in figure 5.1.) For example, to generate 'economic cooperation' from the predication `economic_Financial(C)`, where `C` has semantic head `cooperation_NounMRC`, we construct the following QLF:

```
term(_, q(exists,mass), _,
 W^[and, [cooperation_NounMRC,W],[economic_Financial,W]])
```

Since the CLE has previously analysed the source text, it must have lexical entries corresponding to the predicates `cooperation_NounMRC` and `economic_Financial`, which it can use to produce the surface words.

CLASP can produce summary phrases corresponding to intransitive, transitive and ditransitive verbal predications, but it cannot produce summary phrases with more complex verbal forms, such as those involving sentential complements. In practice, this is not a severe limitation, as such predications are quite rare in the source representation for two reasons. Firstly, because the external lexicon used does not contain any information on verb categories, so

only verbs from the CLE's core lexicon can have such complex complements. Secondly, because many sentence analyses are fragmentary and incomplete, so when such complex constructions occur in the source text, even with verbs in the CLE's core lexicon, they are often not correctly analysed.

7.2.3 *Presenting noun entities*

In figure 7.2, simple predications were listed together with the semantic heads for their arguments. This information, however, is not quite sufficient to construct QLF for summary phrases: we also need to know whether the entities in question are nouns or verbs, and in the case of nouns whether they are common or proper nouns, and whether they are singular, plural, or mass. We may also want to know what quantifier or determiner was used with each noun. As described in section 5.3.6, this information was recorded during the analysis stage, precisely for the purpose of generating summary phrases.

For example, consider the predication **`propel_TransitiveVerbMRC(E, A,B)`**. The analysis stage has identified the following semantic heads and type information:

`E: propel_TransitiveVerbMRC`	verb
`A: investment_NounMRC`	mass noun, quantifier: exists
`B: region_NounMRC`	singular noun, determiner: the

The number and quantifier details correspond to the surface forms 'investment' (the existential quantifier is implicit) and 'the region'.

To generate a summary phrase to represent this predication, we must make choices of number and quantification or determination for the two nouns.

Following the source

CLASP's strategy is simply to use the same quantifiers or determiners, and the same choice of number, as in the source text. Thus in generating a summary phrase for **`propel_TransitiveVerbMRC(E, A, B)`**, we would make 'investment' mass and existentially quantified, and 'region' singular and introduced with the determiner 'the'. This gives us the summary phrase: 'investment propels the region'.

The obvious problem with this method is that the summary output may then refer to entities using a definite article or determiner when those entities have not previously been mentioned and the referent of the resulting anaphoric noun phrases are therefore not available to the reader. In an attempt to improve on the summary phrases obtained with this method, two other methods (*safe* and *mixed*) were implemented but eventually rejected in CLASP.

Playing it safe

To avoid the problem of anaphoric reference in summary phrases, we can preserve the *number* of each noun entity, but present all such entities (whether

they originally appeared with a determiner or a quantifier) with new, 'safe' quantifiers which do not imply any anaphoric reference. Thus 'the region' in the source text becomes 'a region' in the summary text, 'these options' becomes 'some options' or simply 'options', and 'that honey' becomes 'some honey' or 'honey'. This approach solves the problem of dangling anaphors, but creates another problem: *exophors* – phrases which refer to known objects external to the text ('the time', 'the world') – would be dealt with using the same rules as anaphors, producing nonsensical or incorrect phrases such as 'a time' and 'a world'. Although the number of single-word exophors is small, there are many longer exophoric noun phrases, and unfortunately the *safe* strategy often leads to much more confusing summary phrases than the problem of unresolved anaphors. The improvement in, for example 'defence lawyers' over 'the defence lawyers' is small, but if we were, for example, to change 'Baby boomers on both sides of the Pacific' (a summary phrase from the SHEEPCHASE text) to 'Baby boomers on sides of the Pacific' (as the *safe* strategy would do), the phrase becomes much more obscure.

A mixed strategy

As a compromise between *source* and *safe*, we can apply a simple test to try and guess whether entities that appear in the source text with a determiner are exophors or anaphors. The test is based on the assumption that if a noun phrase such as 'the investment' is an anaphoric reference, then we should expect to find elsewhere in the source text the same noun, without an anaphoric determiner (e.g. 'some investment' or 'Japanese investment').

Thus, to choose how to present a noun entity **E**, we consider all the entities in the whole of the source representation which have the same semantic head and the same number as **E**. If all these entities occur in the source text with referential determiners, then the reference is presumed to be exophoric, and we use the determiner 'the' in the summary phrase. Otherwise, we use an existential quantifier – specifically, if **E** is presented with an existential quantifier in the source text, it is used, otherwise 'a' is used for singular nouns and nothing for plural or mass nouns.

For example, if the source text introduces 'a film' and later refers to 'the film', there will be two entities (since CLASP's analysis performs no anaphor resolution), both of which would be presented as 'a film' in a summary phrase. If the text mentions 'the government' but never says 'a government', then the summary phrase will present the entity as 'the government' also. However, this simple strategy will go wrong when an entity is introduced using one noun ('a film') and subsequently referred to by another ('the movie'). To solve the problem in general, we need a proper algorithm for the identification of exophors, which would probably require some world-knowledge and have to be integrated with an anaphor resolution mechanism, neither of which is available to CLASP. Experimentation with Wall Street Journal stories suggests that the *mixed* strategy produces better summary phrases than the *safe* strategy, but the results are no better than those obtained with the simpler *source*

strategy. For simplicity, therefore, CLASP preserves quantifiers and determiners from the source text in its summary phrases.

Presentation of verbs

CLASP makes no attempt to provide any analogous strategies in the presentation of verbs. In the source text, verbs may occur in a variety of tenses, voices and aspects, and with or without auxiliaries. These distinctions are not reflected in CLASP's summary phrases, for two reasons. Firstly, the sequence of verb tenses in text depends on many factors, including the real order of events, and the perspective of presentation. Preserving a verb's original tense out of its original context will often be inappropriate. Secondly, the summary phrases are to be presented in a list, which may contain a mixture of noun phrases and verb phrases. For consistency and readability, it is better to present verbs and nouns as similarly as possible, by using the -ing form for all verbal predications.

7.2.4 *Clustering simple predications*

If we were to take the top few predications selected from the JAPINV text and generate summary phrases from them individually, we would get a summary something like the following (JAP-SP2):

> *This text says something about:*
>
> economic integration,
> Japanese investment,
> the region toward integration,
> increases in trade,
> Asia's countries,
> Asian nations,
> America encouraging Japan.

There are several problems with this summary. The phrases do identify some important concepts in the text, but the collection as a whole is rather incoherent. It is unclear how the summary phrases might relate to each other (is Japan, for example, one of the 'Asian nations' in the story?) and some of them are confusing on their own. 'America encouraging Japan' is less than informative if we do not know what Japan is being encouraged to do. The worst of these summary phrases is clearly 'the region toward integration'. It is doubly unsatisfactory: firstly because we have no antecedent for 'the region'; secondly because the meaning of the prepositional construction is unclear. (The second problem results in part from an incorrect analysis of the original source text, in which the phrase 'toward integration' has been attached to the object of the verb 'propel' instead of to the verb itself.)

To address these issues, we need to go beyond such simple phrases, and instead generate surface text from several predications at once. CLASP's method for doing this is called *clustering*.

By a *cluster* I mean a set of simple predications which are sufficiently related to allow a single summary phrase to be generated for the entire set. Of course, if we allowed summary phrases to include arbitrary conjunctions, then *any* set of simple predications would constitute a cluster, and we would get phrases such as 'Economic integration, increases in trade and Asian nations'. This, of course, is no improvement on having three individual summary phrases, and unless we have some reason to believe that predications are closely related, this kind of conjunction can be very misleading, implying, for example, causal relationships where none exist.

Therefore, CLASP's clusters are restricted as follows: each cluster is based on a primary predication (one selected in the condensation stage), and formed by the addition of related predications. Specifically, a predication may only be added to a cluster when it involves an entity (noun or verb) that already appears as an argument of a predication in the cluster – i.e. it has an *argument link* (section 5.4.2) to a predication in the cluster. The idea is that the added predication should embellish or clarify the summary phrase by adding information about one of the entities mentioned.

Because no anaphor resolution is performed in the analysis stage, each entity can occur only in predications from a single sentence; therefore clustering will not combine predications from different sentences. We could allow more clustering by combining predications joined only by *similar argument links*; effectively we would be assuming that two entities with the same semantic head corresponded to the same object. But this approach could lead to very misleading summary phrases: for example, if the source text mentions 'Japanese investment' and later mentions 'investment in Taiwan', it certainly does not follow that the text says anything about 'Japanese investment in Taiwan'.

An example of clustering

Suppose we start with the predication **`toward(A,B)`**, where the semantic head for **`A`** is **`region_NounMRC`** and for **`B`** is **`integration_NounMRC`**. From this predication we get the summary phrase 'the region toward integration', one of the example phrases in JAP-SE2. The predication **`economic_Financial(B)`** also has **`B`** as an argument, and therefore can be added to the cluster; from these two predications we get the summary phrase: 'the region toward economic integration'. This is generated by adding the new predication to the restriction of the QLF **`term`** for **`B`**, which becomes:

```
X^[and, [integration_NounMRC,X], [economic_Financial,X]]
```

Compared to the two individual summary phrases 'the region toward integration' and 'economic integration', this phrase is both more concise and more informative, because it makes it explicit that it is the same integration that is referred to in the two predications. The phrase is still unsatisfactory, however, because of the incorrect prepositional construction, and the unresolved anaphor.

Adding the related predication **`propel_TransitiveVerbMRC(E,I,A)`**

(semantic heads: **E**: **propel_TransitiveVerbMRC**, **I**: **investment_NounMRC**) to the cluster, we can generate the summary phrase: 'investment propelling the region toward economic integration'. (A QLF **form** is constructed for the verb, with the **term** we had for entity **B** as one of the arguments of the verbal predication.) In this phrase, the incorrect attachment of the prepositional phrase is no longer manifest in the surface text, as both the verb and the object are present. As a result the reader is no longer forced into an incorrect interpretation. This example shows that clustering can improve summary phrases by making up for errors made in the analysis of the source text. It may seem rather lucky that adding another predication to the cluster has improved the summary phrase in this way, but in fact the incorrect attachment of prepositional phrases is a quite common mistake in CLASP's source representation, and clustering can often overcome the problem by including in the summary the element to which the prepositional phrase should have been attached.

To the cluster consisting of these three predications, we can now add, in succession, another two: **nn(I, J)** and **in(I, S)**. (Semantic heads are **Japanese_Predicate** for **J** and **Southeast Asia** for **S**; the phrase 'Japanese investment' has been incorrectly analysed by the CLE as a noun–noun compound.) With the first of these, we get 'Japanese investment propelling the region toward economic integration'; with the second also, we get 'Japanese investment in Southeast Asia propelling the region toward economic integration.' Happily, the addition of this final predication provides an antecedent for 'the region', resulting in a summary phrase which combines three of the individual phrases in our earlier example summary with extra predications, and is fully comprehensible to the reader. Of course the resolution of 'the region' has come about by luck; we cannot expect to be so fortunate in general.

Constraining clustering

The above example shows several potential benefits of clustering. But in contrast to the improvements in comprehensibility which clustering can provide, adding extra predications to a cluster also has the potential to make the summary phrase less appropriate by adding irrelevant information to it. Indeed, if all clusters became sufficiently large, generating summary phrases by clustering could be much the same as extracting sentences of the source text. A decision must therefore be made as to how much clustering to allow.

Since CLASP's condensation stage can select sets of predications of whatever sizes are required, an obvious approach would be to allow clustering only by adding predications from a (perhaps quite large) set of selected predications. For example, if we are to base our summary phrases around the top-ten selected predications, we might allow any of the top-thirty selected predications to be added in clustering. This approach would allow us to combine selected predications in a summary phrase, reducing length, increasing cohesion and readability. It might also increase the comprehensibility and content of the summary phrases by identifying entities more exactly. However,

it would not usually have the benefit of adding predications to compensate for errors in the analysis of the source text, unless those predications happened to have been selected. In addition, sometimes two or more important predications could potentially be combined in a cluster, but only if a linking predication, perhaps not itself selected, is added too; clustering only with selected predications would not allow such combinations.

In contrast to such an approach, therefore, CLASP's clustering is governed not by the relevance of the predications, but only by some simple syntactic considerations. The synthesis stage will freely add verbal, adjectival, adverbial, prepositional, genitive, possessive and noun-noun predications to a cluster, subject to the constraints that conjunctions of adjectival and adverbial phrases only are permitted, and not of verbs or nouns, and that no relative or subordinate clauses are allowed. Within these limitations, CLASP continues to add predications to each cluster until no more can be added.

Since clustering can only involve predications from a single sentence, there is a limit to how much material might be combined in a single summary phrase. In addition, the CLE is often unable to achieve a full analysis of complex or long sentences; the resulting fragmentation of the QLF further reduces the number of predications that may be combined in a single cluster. If CLASP's analysis were improved so as to make it less fragmentary, or so that anaphors were resolved and argument links arose between sentences, it would become necessary to add more constraints to the clustering procedure. As it is, experimentation with CLASP suggests that these simple syntactic restrictions are usually enough to avoid very long and uninformative summary phrases.

Applying clustering to the simple predications that generated the JAP-SP2 summary phrases, we obtain the following summary phrases (the same as those in JAP-SP1):

> Japanese investment in Southeast Asia propelling the region toward economic integration,
> Japan's commitment in Southeast Asia including steep increases in foreign trade,
> Asia's cash-rich countries,
> Asian nations,
> America encouraging Japan.

The first of these phrases combines the first three summary phrases of JAP-SP2, while the second adds (a lot of) detail to the simple phrase 'increases in trade'. In the last two phrases, no clustering has occurred: in both cases, errors in analysis mean that there are no predications related by argument links which can be added to the cluster.

7.2.5 *Presenting summary phrases*

As the first stage of producing a summary, CLASP simply takes the top few selected predications and builds a cluster around each one. But before outputting summary phrases corresponding to these clusters, two decisions

must be made: which clusters to use, and in what order to present the resulting summary phrases.

Removing redundant phrases

The reason we may not want to use all the clusters based on top predications is that some of them may be redundant. Suppose, for example, that the phrase 'economic integration' occurs twice in a story. There will be two predications **economic_Financial(A)** and **economic_Financial(B)** and two entities **A** and **B**, both with semantic head **integration_NounMRC**. The two predications are distinct since they involve different entities, but because those entities have the same semantic head (and type: mass noun), they would generate identical summary phrases. Therefore if both these predications are selected, and if no additional predications are added in clustering, we should discard one of them to avoid repetition of 'economic integration' in the summary. If clustering adds additional predications to one of the selected predications but not to the other, then the smaller cluster should be discarded. On the other hand, if instead clustering adds additional predications to both selected predications, then the two clusters may convey different content, so neither of them is redundant.

More formally, we define a cluster C to be *made redundant by* another cluster D when the following three conditions hold. Firstly, for each predication in C, there is a corresponding predication in D with the same predicate and number of arguments. Secondly, two predications in C have an argument or arguments in common when and only when the corresponding predications in D do so. Thirdly, the arguments of predications in C and the arguments of the corresponding predications in D have the same semantic heads. (Effectively this is saying that C and D are equivalent as fragments of the predication cohesion graph.)

Having built a set of clusters around selected predications, CLASP checks whether any of them is made redundant by another cluster in the set. If so, redundant clusters are removed one at a time, until the set contains no redundant clusters.

Presentation order

It remains to choose an order in which to present the summary phrases corresponding to the remaining clusters. In section 7.1, we saw that, for summaries produced by sentence extraction, presenting selected sentences in their original order produced a better summary than presenting them in the order that they were selected.

Both these orders can be used with summary phrases. In *selection* order, we consider the predication that was selected first in each cluster, and output the phrases in in the order that these predications were selected. In *source* order, we output the phrases corresponding to clusters from earlier sentences first (recall that clustering can only combine predications from a single sentence); if two clusters come from the same source sentence, we revert to selection order. Summary phrases have fewer anaphors than source text sentences, and no

SELECTION ORDER

This text says something about:

Japanese investment in Southeast Asia propelling the region toward economic integration,
America encouraging Japan,
Asia's cash-rich countries,
Asian nations,
Japan's commitment in Southeast Asia including steep increases in foreign trade.

SOURCE ORDER

This text says something about:

Japanese investment in Southeast Asia propelling the region toward economic integration,
Japan's commitment in Southeast Asia including steep increases in foreign trade,
Asia's cash-rich countries,
Asian nations,
America encouraging Japan.

LENGTH ORDER

This text says something about:

Japan's commitment in Southeast Asia including steep increases in foreign trade,
Japanese investment in Southeast Asia propelling the region toward economic integration,
Asia's cash-rich countries,
Asian nations,
America encouraging Japan.

Figure 7.3: summary phrases for the JAPINV text, using three different presentation orders.

discourse connectives, and they are intended to constitute a list of ideas rather than a continuous text, so we would expect the benefit of using the source text order to be less for summary phrases.

We will also consider *length* order, in which we present the longest summary phrases (or more accurately, the ones corresponding to the largest clusters of predications) first. The motivation for this strategy is that it is the largest clusters that are likely to convey the most content, most comprehensibly, to the reader. When two clusters have the same size, we revert to selection order.

Figure 7.3 shows the results of applying these three orders to the summary phrases of JAP-SP1. The three orders all give different summaries, but it is hard to see any clear reason to prefer a particular one of them. All we can reasonably suppose from this example is that presenting the material in its original order is no longer as superior as it was in the case of sentence extracts. For the experiments in the next chapter, and the examples in appendix B, selection order was used.

8 EXPERIMENTS WITH CLASP

This chapter describes experiments with CLASP, both in producing summaries by sentence-extraction, and in generating summary phrases. CLASP's summaries are compared with those produced by less linguistically sophisticated methods, and to targets chosen by humans. They are also evaluated by direct judgement.

The experiments reported in this chapter are not a thorough evaluation of CLASP: the summariser is not operating in a realistic setup or a simulation of one, we do not explicitly consider users' requirements, and there are far too few source texts for us to be able to draw firm conclusions about summary quality. However, they do provide crude ways to assess to what extent CLASP's choice of source representation and linguistic processing are of benefit in summarising.

Where particular summaries are discussed, I have included them in this chapter; the source texts discussed are given in appendix A, and examples of CLASP's summaries of them are shown in appendix B.

8.1 SIZE OF EVALUATION

We saw in chapter 3 that large-scale evaluation of summarising systems has so far been rare. Notable exceptions were the ANES system (Brandow, Mitze and Rau 1995), which was evaluated by direct human judgement, using 250 source texts, and FRUMP (DeJong 1979, 1982), which ran for a week on a UPI newswire. An evaluation of comparable size has not been possible for CLASP, for two reasons. Firstly, and most importantly, CLASP's analysis stage is very slow, as described in section 5.2.3. Secondly, evaluation requires human input: to evaluate CLASP's sentence extraction summaries, for example, five sets of target sentences were chosen for each story by different readers. This is also time-consuming, if not as slow as parsing.

For these reasons, experiments with CLASP were limited to a set of twenty source texts of between 300 and 1500 words. CLASP could have processed more texts if we had chosen only very short texts of, say, 100–200 words, or if we had chosen only texts with simple syntax and short sentences. But neither of these options is acceptable: CLASP is explicitly intended to deal robustly with real-life input, so it is essential that we do not use artificially simple texts.

8.2 SOURCE TEXTS

The source texts used for all my experiments were a set of twenty stories from the Wall Street Journal. They were chosen at random subject to two simple criteria: they should be between a couple of hundred and around 1500 words long and should contain no pictures, graphics, tables or other non-textual elements. Titles and headings were also removed from the chosen stories, to give a continuous text.

12 news stories
subject matter: politics (3), financial (6), transport (1), media (2)
genre: descriptive, occasionally narrative
length: 300–900 words

6 feature articles
subject areas: wine, banking, bell-ringing, education, investment
genre: descriptive
length: 800–1400 words

2 review articles (1 novel, 1 film)
genre: descriptive, occasionally critical
length: 300–600 words

Figure 8.1: summary of the twenty Wall Street Journal stories used in experiments with CLASP.

Figure 8.1 summarises the 20 texts used. They all contain long and complex sentences (the average lengths vary from 18 to 20 words), with extensive use of coordination and relative clauses, and many proper names. The majority (twelve) of the texts are news stories, describing recent events. Most of these are about financial or political subjects. A further six texts are longer articles: these discuss subjects in greater depth, covering long-term trends and providing more background than news stories. There is greater variety in the subject matter of these texts, which could best be classified as 'feature' articles. Both the news stories and the feature articles contain quite large amounts of direct and indirect speech, with quotations from those involved and from commentators. The two remaining texts are reviews. They are similar in length to short news stories, but unlike news stories and feature articles, they contain a substantial amount of evaluation and opinion as well as descriptive writing.

8.3 EVALUATING SENTENCE EXTRACTS BY TARGET-COMPARISON

In these experiments, the aim was to investigate CLASP's ability to identify content that is appropriate for a summary. By restricting the condensation stage to select whole sentences, this can be measured by the correspondence between the sentences selected by CLASP, and those previously identified as target sentences by human readers. Since target sentences were to be chosen on the basis of the importance of their content, CLASP's condensation stage used *importance scoring* (section 6.2.1) only.

8.3.1 *Establishing sets of target sentences*

Five readers were given all the Wall Street Journal stories, and asked to read

them and to choose, from each one, three sentences which they felt contained information that should be included in a summary of the story. The readers (one of whom was me) were all aware that their responses would be used to judge automatically produced summaries, but they were not given any specific further indications as to what kind of summary was envisaged, and they had not seen CLASP's or any other system's summaries of the texts at the time. It is important to note that the readers were not asked to choose sentences which would fit together to make a coherent summary text; rather they were asked to choose sentences on the basis of content. Nevertheless, inspection of the responses suggests that, whether consciously or not, readers did indeed select sentences which, with only minor alterations, could combine to make a readable summary text.

Example target sentences

Figure 8.2 shows the target sentences chosen for the SHEEPCHASE story. This story is a review of a Japanese novel, *A Wild Sheep Chase*. We can distinguish several kinds of information in these target sentences: first, the title and the author of the novel (in s0); second, information about the characters and plot (in s11, s13, and to a lesser extent in s1, s9); third, public reaction (in s15, s17); fourth, the reviewer's opinions (in s0, s1, s6, s9).

Readers A, C and D, who have all selected s0 and s6, have chosen both title/author information and reviewer's opinion, and added either plot details or public reaction. Reader E, though having selected difference sentences, has chosen a similar combination, but concentrated more on plot. Reader B, on the other hand, has chosen only sentences giving plot information. It seems likely that these differences reflect genuine differences of opinion about what should go in a summary, and therefore that no summary would satisfy every reader.

8.3.2 *Measuring coverage of target sentences*

Because we want to look at automatic summaries of varying lengths, a measure of *coverage* is more useful than one of set-similarity: given a *target set* of sentences T and a *selected set* of sentences S, the coverage is $|T \cap S| \,/\, |T|$. That is, coverage is just the proportion of the sentences in the target set that are also in the selected set. (This is the measure used for target-comparison by Miike et al (1994).)

For each text we have five sets of target sentences T, and therefore five measures of coverage. Following Mitra, Singhal and Buckley (1997), we can be *optimistic* (and take the highest coverage) or *pessimistic* (and take the lowest coverage). On both these measures, a summary can score anything from 0 to 1.

We also look at *average coverage* over the five target sets: with this measure, target sentences selected by more than one reader will effectively be given greater weight. To attain a score of 1 on average coverage, a summary would need to include every target sentence selected by every reader, which may not be possible unless we are producing quite long summaries.

Reader A

s0: Judging from the Americana in Haruki Murakami's 'A Wild Sheep Chase' (Kodansha, 320 pages, $18.95), baby boomers on both sides of the Pacific have a lot in common.

s6: For an American reader, part of the charm of this engaging novel should come in recognizing that Japan isn't the buttoned-down society of contemporary American lore.

s11: A disaffected, hard-drinking, nearly-30 hero sets off for snow country in search of an elusive sheep with a star on its back at the behest of a sinister, erudite mobster with a Stanford degree.

Reader B

s1: Although set in Japan, the novel's texture is almost entirely Western, especially American.

s11: A disaffected, hard-drinking, nearly-30 hero sets off for snow country in search of an elusive sheep with a star on its back at the behest of a sinister, erudite mobster with a Stanford degree.

s13: Along the way, he meets a solicitous Christian chauffeur who offers the hero God's phone number.

Reader C

s0: Judging from the Americana in Haruki Murakami's 'A Wild Sheep Chase' (Kodansha, 320 pages, $18.95), baby boomers on both sides of the Pacific have a lot in common.

s6: For an American reader, part of the charm of this engaging novel should come in recognizing that Japan isn't the buttoned-down society of contemporary American lore.

s17: But he is just one of several youthful writers – Tokyo's brat pack – who are dominating the best-seller charts in Japan.

Reader D

s0: Judging from the Americana in Haruki Murakami's 'A Wild Sheep Chase' (Kodansha, 320 pages, $18.95), baby boomers on both sides of the Pacific have a lot in common.

s6: For an American reader, part of the charm of this engaging novel should come in recognizing that Japan isn't the buttoned-down society of contemporary American lore.

s15: The 40-year-old Mr. Murakami is a publishing sensation in Japan.

Reader E

s0: Judging from the Americana in Haruki Murakami's 'A Wild Sheep Chase' (Kodansha, 320 pages, $18.95), baby boomers on both sides of the Pacific have a lot in common.

s1: Although set in Japan, the novel's texture is almost entirely Western, especially American.

s9: That's not to say that the nutty plot of 'A Wild Sheep Chase' is rooted in reality.

Figure 8.2: target sentences for the SHEEPCHASE text.

GRAPH WEIGHTING

unweighted	every edge has weight 1
uniform	every link has weight 1
head	similar and stem-similar head links have weight 2, others have weight 1
mixed	argument links have weight 1.1, predicate, similar argument, and similar head links have weight 1, others have weight 0.9
argonly	as mixed but predicate and stemmed predicate links have weight 0 (i.e. they are ignored)

IMPORTANCE SCORING

General formula: $imp(H) = \Sigma_i\, a_i\, imp_i(H)$

simple	$a_1 = 1$
simplish	$a_1 = 0.9,\ a_2 = 0.1$
harder	$a_1 = 0.6,\ a_2 = 0.4$
hard	$a_1 = 0.5,\ a_2 = 0.3,\ a_3 = 0.2$

Figure 8.3: graph weightings and importance scores used in experiments with CLASP. (See section 6.2.1 for scoring formulae.)

Before applying these measures to CLASP's summaries, we should ask how similar the target sets themselves are. This we can do by computing the coverage of each target set with respect to each of the others. Averaging these coverages, and averaging again over all the stories, gives a value of 0.40. That is, if we pick a story and two sets of target sentences for it, the chance that any particular sentence in one target set is also in the other is 40%. This low figure partly reflects the fact that, in many of these stories, there are two or more sentences which convey much of the same basic information – two readers could therefore choose different sentences but be selecting essentially the same information as important. But it also reflects the fact that, as seen in figure 8.2, people simply do not always agree about what is most important in a text.

8.3.3 *Selecting important sentences with* CLASP

A variety of summaries were produced using CLASP's sentence-extraction. These were of various lengths, and used various different settings in the condensation stage, both in assigning weights to the edges of the predication-cohesion graph, and in establishing importance scoring functions on sets of nodes.

Figure 8.3 shows the choices of weights. Of these weighting strategies, *unweighted* and *uniform* were chosen as basic starting points. *Head* was based on the principle that since a head link is similar to an argument link and a

predicate link combined, it should have a higher weight. *Mixed* was based on the idea that the less direct the link between predications, the lower weight it should have. *Argonly* follows the intuition that links arising from similarity of entities (i.e. argument and head links) are much more important that links arising from similarity of adjectives and adverbs (i.e. predicate links).

Having assigned edge weights, we next choose an importance scoring function for sets of nodes. Section 6.2.1 gave the generic scoring function for importance: $imp(H) = \Sigma_i\ a_i\ imp_i(H)$. Figure 8.3 defines four specific importance-scoring functions: *simple*, *simplish*, *harder*, and *hard* (the names refer to the difficulty of computation only). In *simple* the score for a node is just the sum of its edge weights; in *hard*, paths of length up to 3 are considered in computing a node's importance score.

CLASP was used to produce summaries by sentence-extraction, using each combination of these four importance scores and five weighting strategies.

8.3.4 *Target-coverage results*

Figure 8.4 gives the optimistic, pessimistic and average coverage of target sentences, averaged over the 20 Wall Street Journal texts, for summaries of 3 sentences using each of the methods presented in figure 8.3.

Effect of weighting and scoring functions

The clearest result from this data is that, perhaps surprisingly, CLASP's performance is hardly affected by the choice of edge weights in the predication-cohesion graph. The scores of the naive *unweighted* and *uniform* weightings, which do not distinguish between the different link types, are not significantly different from any of the other weighting methods. The choice of importance scoring function also does not have a strong effect, although in general it seems *harder* and *hard* give lower coverage of target sentences that *simple* and *simplish*. Inspection of the summaries shows that indeed the selected sentences vary very little from one method to another: therefore they must be largely determined by the *shape* of the predication-cohesion graph, in terms of how well-connected different parts of it are. The fact that *simple* and *simplish* seem to perform better than *harder* and *hard* suggests that specifically the *local* structure of the graph (by which I mean the direct relations between predications, and not anything to do with proximity in the source text) is more useful than non-local structure in indicating important content. In the following comparisons, we take *uniform-simplish* as a representative strategy for CLASP.

Comparison with other summarising methods

For comparison, the same stories were given to BT's *NetSumm* (Preston and Williams 1994), which also summarises by sentence extraction. As with CLASP, summaries of 3 sentences were obtained. Two other methods were also considered: choosing 3 sentences at *random*, and choosing the first 3 sentences (the *initial* strategy). The coverages achieved by these methods and by *uniform-*

Optimistic target-coverage:

	IMPORTANCE SCORING FUNCTION			
	simple	*simplish*	*harder*	*hard*
WEIGHTING				
unweighted	0.39	0.39	0.38	0.40
uniform	0.41	0.40	0.37	0.33
head	0.40	0.40	0.41	0.34
mixed	0.39	0.40	0.37	0.34
argonly	0.38	0.38	0.37	0.34

Pessimistic target-coverage:

	IMPORTANCE SCORING FUNCTION			
	simple	*simplish*	*harder*	*hard*
WEIGHTING				
unweighted	0.08	0.08	0.08	0.08
uniform	0.08	0.08	0.08	0.07
head	0.08	0.08	0.07	0.07
mixed	0.08	0.08	0.08	0.07
argonly	0.08	0.08	0.08	0.05

Average target-coverage:

	IMPORTANCE SCORING FUNCTION			
	simple	*simplish*	*harder*	*hard*
WEIGHTING				
unweighted	0.23	0.23	0.22	0.23
uniform	0.24	0.24	0.22	0.19
head	0.24	0.24	0.23	0.20
mixed	0.23	0.24	0.22	0.19
argonly	0.23	0.23	0.22	0.18

Figure 8.4: target-coverage for 3-sentence extracts.

	optimistic	*pessimistic*	*average*
NetSumm	0.35	0.07	0.19
random	0.30	0.01	0.10
initial	0.53	0.14	0.30
CLASP *uniform-simplish*	0.40	0.08	0.24

Figure 8.5: comparison of CLASP's sentence selection with other methods. Target-coverage for 3-sentence extracts.

simplish with CLASP are shown in figure 8.5.

From this data, it appears that NetSumm, CLASP and the *initial* strategy are all better than random, but all achieve average coverage much less than the average coverage between two sets of target sentence (0.40). That is, the automatic summaries are more different from reader's sets of target sentences than those sets are from each other.

To avoid drawing unjustified conclusions from the data, however, we must consider whether these differences are likely to be significant. We can do this by interpreting the differences in terms of the minimum number of changes that could account for them. Each reader selected three target sentences, so changing one sentence in a summary can change the coverage score for that summary by up to 1/3. The data in figure 8.5 is averaged over 20 stories, so a single sentence change in one story could affect the overall score by up to 1/60, or 0.017. Looking at the *optimistic* scores (which show the greatest difference between the systems), CLASP and NetSumm differ by 0.05: the minimum change necessary to achieve this difference would be only three sentences in the whole twenty texts. This figure is small enough that we must conclude that although the data suggest CLASP may be more successful at selecting important sentences than NetSumm, there is insufficient data to conclusively show a difference.

8.3.5 *Initial sentences and the effect of summary length*

That selecting initial sentences achieves quite good target coverage should not be a surprise, given that Brandow, Mitze and Rau (1995) found that initial extracts of similar texts were judged to be acceptable summaries (for 60-word extracts, the acceptability rating was 87%). They note that when initial extracts were unacceptable, this was usually due to the presence of anecdotal material in the leading sentences. Of the twenty texts used in my experiments, five begin with anecdotal sentences, and the initial method scores lower than average on them. On two texts, an extract of the first 3 sentences achieves an optimistic coverage of zero: one (WINEMARKET in appendix A) begins with a five-sentence anecdote, and the other (PACKAGES) begins with an eight-sentence anecdote.

We might expect initial extracts to do well only when they are relatively short. Figure 8.6 shows how target-coverage of CLASP, *initial*, and *random*

3 SENTENCES EXTRACTED

	optimistic	*pessimistic*	*average*
random	0.30	0.01	0.10
initial	0.53	0.14	0.30
CLASP	0.40	0.08	0.24

5 SENTENCES EXTRACTED

	optimistic	*pessimistic*	*average*
random	0.41	0.01	0.18
initial	0.59	0.14	0.37
CLASP	0.55	0.08	0.35

7 SENTENCES EXTRACTED

	optimistic	*pessimistic*	*average*
random	0.51	0.03	0.24
initial	0.72	0.28	0.49
CLASP	0.73	0.27	0.51

9 SENTENCES EXTRACTED

	optimistic	*pessimistic*	*average*
random	0.60	0.07	0.31
initial	0.82	0.37	0.59
CLASP	0.83	0.36	0.62

Figure 8.6: target-coverage comparison. Here, 'CLASP' means *uniform* weighting and *simplish* importance scoring.

compare for summaries of 3, 5, 7 and 9 extracted sentences. As summary length increases, the difference between the coverage obtained with *initial* and CLASP decreases: at 7 sentences, the two methods achieve approximately equal coverage. Meanwhile, the difference in coverage between CLASP and *random* increases the more sentences are selected.

8.3.6 *Problem stories and success stories*

Figure 8.7 shows a 5-sentence summary of SHEEPCHASE (again using *uniform-simplish* settings) which achieved optimistic coverage of 1.

This summary scores so highly because it includes sentences 0, 6 and 11 (target sentences for the SHEEPCHASE text were shown in figure 8.1). In CLASP's predication cohesion graph for this story, there are many predications

[0] Judging from the Americana in Haruki Murakami's A Wild Sheep Chase (Kodansha, 320 pages, $19), baby boomers on both sides of the Pacific have a lot in common. [6] For an American reader, part of the charm of this engaging novel should come in recognizing that Japan isn't the buttoned-down society of contemporary American lore. [11] A disaffected, hard-drinking, nearly-30 hero sets off for snow country in search of an elusive sheep with a star on its back at the behest of a sinister, erudite mobster with a Stanford degree. [16] A more recent novel, Norwegian Wood (every Japanese under 40 seems to be fluent in Beatles lyrics), has sold more than four million copies since Kodansha published it in 1987. [17] But he is just one of several youthful writers – Tokyo's brat pack – who are dominating the best-seller charts in Japan.

Figure 8.7: CLASP's (*uniform-simplish*) summary of the SHEEPCHASE text. (Sentence numbers in square brackets.)

[7] These first magnitude wines ranged in price from $40 to $125 a bottle. [30] There are certain cult wines that can command these higher prices, says Larry Shapiro of Marty's, one of the largest wine shops in Dallas. [34] We've seen a dramatic decrease in demand for wines from the 40 and 50, which go for $300 to $400 a bottle. [57] Mr Martin has increased prices on some wines (like Grgich Hills Chardonnay, now $32) just to slow down movement, but he is beginning to see some resistance to high-priced red Burgundies and Cabernets and Chardonnays in the $30 to $40 range. [59] Wine merchants can't keep Roederer Cristal in stock, but they have to push Salon le Mesnil, even lowering the price from $115 to $90.

Figure 8.8: CLASP's (*uniform-simplish*) summary of the WINEMARKET text. (Sentence numbers in square brackets.)

[5] One of the fastest growing segments of the wine market is the category of superpremiums – wines limited in production, of exceptional quality (or so perceived, at any rate), and with exceedingly high prices.

[8] In the last year or so, however, this exclusive club has taken in a host of flashy new members.

[49] There may be sticker-shock reaction initially, said Mr Pratt, but as the wine is talked about and starts to sell, they eventually get excited and decide it's worth the astronomical price to add it to their collection.

Figure 8.9: Sentences chosen as target sentences for the WINEMARKET text by more than one reader.

---

concerning entities whose semantic heads correspond to 'Japan', 'novel', 'Americana', 'A Wild Sheep Chase', 'Kodansha', 'Murakami', 'sheep' and 'hero'. All these concepts are mentioned more than once in the source text, and each of sentences 0, 6 and 11 contains two or more of them. Because CLASP looks at predications not just surface words, it can in addition notice that sentence 11 says more about the 'hero' than does sentence 13, which also mentions him.

In contrast, the summary in figure 8.8 of the WINEMARKET test, does not

include a single target sentence chosen by any reader. Figure 8.9 shows the target sentences for this text that were chosen by more than one reader. The story is about a recent increase in the number of very expensive wines. CLASP's summary certainly suggests that the story is about expensive wines – in fact this would probably be clear from any sentence of it. But although in this sense it has some value as an indicative summary, the sentences it includes are all, with the exception of sentence 30, rather too specific: they give information about particular wines or particular prices. Part of the problem is that the text mainly consists of such specific information, and CLASP allows the large number of predications involving entities with semantic head **`dollar_ AmountOfMoney`**, and the even larger number for **`wine_NounMRC`**, to dominate the importance scoring. The other part is that there are no groups of related predications corresponding to the idea of the *category* of expensive wines, rather than particular wines. The text refers to 'the category of superpremiums' in sentence 5, and 'this exclusive club' in sentence 8. Elsewhere it mentions 'this group', 'these first magnitude wines', 'high-priced bottles', 'these wines', and 'expensive wines'. If CLASP could recognise that these phrases refer to the same or very similar concepts, there would be a high number of cohesive links between predications about them, and a chance that such predications would score highly in the condensation stage. Without this, however, sentences 5 and 8 will never be selected.

Interestingly, two readers reported difficulty in choosing target sentences for this text, so perhaps the large amount of information on specific wines and prices poses a problem for human readers too. They chose not to select such information in the target sentences, but arguably this has made the target unrepresentative of the text as a whole.

8.4 DIRECT JUDGEMENT OF SENTENCE EXTRACTION SUMMARIES

In addition to the target-comparison evaluation, I carried out a limited evaluation by direct judgement, in which five-sentence summaries were given a numerical *relevance score* from 0 to 5. This score was a count of how many sentences in the summary had indicative content that was both relevant (i.e. worthwhile in a summary) and new (not also present in a previous sentence of the summary). The aim of this score was similar to that of the target-comparison experiment: to see how well CLASP identified appropriate content for a summary. But it was intended to avoid a problem with target-comparison, namely that the summary could contain sentences with the same or very similar content to the target sentences, yet not achieve a high coverage. This is a definite possibility in newspaper stories, where often an important point is given once near the start of the text, and again later on, in more detail. The relevance score was also more specifically aimed at measuring the *indicative* value of summaries than the target-comparison evaluation.

By discounting duplicated information in the summary, the relevance score also attempts to disfavour non-representative summaries, which only contain information about a single topic. This experiment therefore involved

using CLASP's *representativeness* scoring as well as its *importance* scoring. Inspection of CLASP's output suggested that summaries did not suffer from low cohesion, however, so cohesiveness scoring was not used.

Experimental procedure

There was only one reader (me) for this experiment. I first read each story and made notes, writing down the main topics and some indication of their relative importance. Then I considered the summaries in a random order (different for each story), and gave each a relevance score following the definition above.

Scoring methods

The methods used to produce summaries for this experiment were combinations of some of the weighting and scoring functions in figure 8.3, plus a number of *representativeness scoring* methods. Summaries were also produced with the *initial* method, taking the first five sentences of the source text.

Recall the general formula for representativeness scoring from section 6.2.2: $\mathbf{rep}(H) = \sum_i b_i \, \mathbf{rep}_i(H)$. In this experiment all the b_i were set to 0 except for $b_1 = 1$, i.e. $\mathbf{rep}(H) = \mathbf{rep}_1(H) = \mathbf{imp}(B(H))$. Two overall scoring functions were tried:

$$\mathbf{repstrong}(H) = 0.75\,\mathbf{imp}(H) + 0.25\,\mathbf{rep}(H),$$
$$\mathbf{repweak}(H) = 0.9\,\mathbf{imp}(H) + 0.1\,\mathbf{rep}(H).$$

These functions can be combined with any choice of **imp**, to give, for example, an overall scoring function such as *uniform-simplish-repweak* or *head-simplish-repstrong*. As the choice of weighting and importance function had already been found to make little difference to the summary, only the combinations *uniform-simplish* and *head-simplish* were considered.

Results

Figure 8.10 shows, for each summary method, how many summaries were given each relevance score, and the average score over the twenty texts.

It seems that the *repweak* methods may produce slightly less relevant summaries than the importance scoring methods, but the difference is small enough that we cannot say anything with certainty. *repstrong* gives considerably lower scores still, and a much lower number of summaries with four or more relevant sentences. Inspecting summaries produced by importance scoring suggests that in fact duplication of content between sentences is not often a problem.

The comparison with *initial* is more revealing. The *initial* method gets a relevance score of 5 for a quarter of its summaries, whereas *uniform-simplish*, for example, only achieves this for one summary in ten. But *uniform-simplish* only once fails to select three relevant sentences (and always selects at least one), whereas *initial* scores less than three a quarter of the time. At least in this experiment, then, CLASP's summaries, though they do not get higher relevance

DISTRIBUTION OF SCORES						AVERAGE	
5:	4:	3:	2:	1:	0:		
2	9	8	0	1	0	3.55	*uniform-simplish*
2	9	8	0	1	0	3.55	*head-simplish*
1	8	9	2	0	0	3.4	*uniform-simplish-repweak*
1	10	7	2	0	0	3.5	*head-simplish-repweak*
1	5	10	4	0	0	3.15	*uniform-simplish-repstrong*
1	4	12	2	1	0	3.1	*head-simplish-repstrong*
5	7	3	1	2	2	3.4	*initial*

Figure 8.10: Distribution and average of relevance scores.

scores on average, are more consistent in the scores they achieve than the *initial* method. This is an encouraging result for CLASP, though not particularly surprising since, as already noted in section 8.1.5, *initial*'s performance is poor for those stories that begin with anecdotal material.

8.5 SUMMARY PHRASE EXPERIMENTS

This section looks at *summary phrases* (section 7.2) for the Wall Street Journal texts. First, in section 8.5.1, I consider the selection of simple predications; section 8.5.2 then presents a small evaluation of summary phrases using criteria of relevance and comprehensibility.

8.5.1 *Selection of simple predications*

When summary phases are to be produced, CLASP's condensation stage selects individual simple predications for the summary representation (rather than selecting all the predications for a whole sentence at once). For example, figure 8.11 shows the first few simple predications selected, for the JAPINV text, using *uniform-simplish* scoring and weighting. The predications are given in order of selection.

For individual predications, as for sentence-extraction, the choice of weighting and importance scoring has little effect on what is selected. In fact all the other combinations in figure 8.3 lead to the same top predications being selected for JAPINV, although the last three may be chosen in a different order. The effect is similar on the other source texts.

In figure 8.11, predications 1–3 all correspond to the idea of Japanese investment, while predications 5 and 6 are both about investment in Southeast

```
1. nn(A:investment_NounMRC, B:japanese_Predicate)
2. nn(C:investment_NounMRC, D:japanese_Predicate)
3. nn(E:investment_NounMRC, F:japanese_Predicate)
4. encourage_TransitiveVerbMRC(G, H:America, I:Japan)
5. in(A:investment_NounMRC, J:Southeast Asia)
6. in(K:investment_NounMRC, L:Southeast Asia)
7. than(M:region_NounMRC, N:United States of America)
8. nn(K:investment_NounMRC, O:japan_Predicate)
```

Figure 8.11: the first eight simple predications selected for JAPINV by *uniform-simplish*. Each variable is shown with its semantic head.

```
1. nn(A:investment_NounMRC, B:japanese_Predicate)
2. encourage_TransitiveVerbMRC(C, D:America, E:Japan)
3. possessive(F:country_Place, G:Asia)
4. nn(H:nation_Country, I:asian_Predicate)
5. toward(J:region_NounMRC, K:integration_NounMRC)
6. economic_Financial(K:integration_NounMRC)
7. in(L:increase_Growth, M:trade_NounMRC)
8. genitive(N:university_Predicate, O:School)
```

Figure 8.12: the first eight simple predications selected for JAPINV by *uniform-simplish-repstrong*.

Asia. There is an obvious redundancy in selecting many equivalent predications in this way, when there are other important ideas in the text that should be mentioned in the summary. The solution is to use CLASP's representativeness scoring to select predications which are related to more parts of the predication cohesion graph. Figure 8.12 shows the results of applying *uniform-simplish-repstrong* to the same text. There is now a much greater variety in the selected predications. Once again they are hardly affected by the particular weighting or importance scoring functions used. Ideally we would like to investigate a variety of representativeness scores from *repweak* to *repstrong* and beyond, but a large-scale evaluation by direct judgement would be prohibitively expensive, and we cannot easily evaluate selected predications or summary phrases by target-comparison. Therefore the experiments in the next section use only the strategies *uniform-simplish* and *uniform-simplish-repstrong*.

8.5.2 *Evaluation of summary phrases*

Summary phrases are intended to indicate clearly the main topics of a text. I therefore evaluated them by direct judgement on criteria of whether they made sense, and whether they were relevant to the text.

Strategy *uniform-simplish:*
165 summary phrases in total

	relevant	*not relevant*
clear	51 (31%)	40 (24%)
unclear	11 (7%)	63 (38%)

Strategy *uniform-simplish-repstrong:*
183 summary phrases in total

	relevant	*not relevant*
clear	60 (33%)	32 (17%)
unclear	13 (7%)	78 (43%)

Figure 8.13: categorisation of summary phrases.

For this evaluation, *uniform-simplish* and *uniform-simplish-repstrong* were used to select twenty simple predications from each text, and from these up to ten summary phrases were produced. Because of clustering and filtering, summary phrases can include more than one of the selected predications each, and in some cases less than ten summary phrases were produced from the twenty predications. On average eight summary phrases were produced per text with *uniform-simplish*, and nine with *uniform-simplish-repstrong*.

Two judgements were made about each summary phrase. Firstly, summary phrases that were grammatical and whose meaning was clear were classified as *clear*, others as *unclear*. Secondly, summary phrases which mentioned salient topics in the source text *not already mentioned by another summary phrase* were classified as *relevant*, others as *not relevant*. The two classifications were made independently, as there were both summary phrases that made perfect sense but did not mention any salient topic, and phrases that as a whole did not make sense, but which did have indicative value.

Figure 8.13 shows the resulting four-way categorisation of summary phrases. There were quite large differences in the summary phrases generated by the two strategies for many of the source texts but, in percentage terms, the number of phrases that were judged to be both *relevant* and *clear* is not very different.

There is a strong tendency in general for *clear* summary phrases to be *relevant*, and for *unclear* ones to be *not relevant*. There are several possible explanations for this. It may be that the underlying selected predications are equally relevant, but that unclear expression obscures this relevance from the reader. We would expect such a correlation if, for example, clustering tends to increase both relevance and clarity.

However, most *unclear* summary phrases are the direct result of inaccuracies in CLASP's analysis producing erroneous predications in the source representation. It may be the case that somehow CLASP's analysis of relevant

LANEFILM, strategy *uniform-simplish*

This text says something about:
Mr Lane reviving an Artist in a full-length,
Black-and-white film about an artist, about a man of the streets,
A sketch artist,
A sketch artist competing,
A romance for the Artist, with a young woman,
Mr Lane's final purpose to glamorize,
The Artist in charge,
The Artist having a routine,
Mr Lane's Artist,
The Artist hanging.

LANEFILM, strategy *uniform-simplish-repstrong*

This text says something about:
Mr Lane reviving an Artist in a full-length,
A man of the streets,
Sidewalk Stories about a modern-day tramp,
The film containing dialogue,
Movie director,
The double bass in classical music preparing an eclectic score,
Woman walk,
Charlie Chaplin's spirit,
Cute child turning out This,
Story line resonating This.

Figure 8.14: Sets of summary phrases for the LANEFILM text, (in *selection* order).

material tends to be more accurate than that of non-relevant material, but there is no obvious reason why this should be so. A more likely explanation is simply that the accuracy with which CLASP can determine the relevance of material depends on the accuracy with which that material is represented in the predication cohesion graph. This suggests that we might expect improvements in CLASP's analysis to increase the relevance of summary phrases as well as their clarity.

Comparing CLASP's summary phrases produced by importance and representativeness scoring, we can identify two common patterns of behaviour, illustrated in figures 8.14 and 8.15.

In figure 8.14, the upper summary (importance scoring) is dominated by phrases containing the word 'Artist'. There are sufficiently many entities in the predication cohesion graph with the semantic head **`Artist`** that the predications about them form a very large clique, and all get high importance scores. The result is a very narrow summary.

PACKAGES, strategy *uniform-simplish*

This text says something about:
Customers banking,
The banks promoting the packages,
Bank consulting firm in NC, in Charlotte,
A bank consulting firm in Atlanta,
Chemical Bank spending a dollar,
Some banks moving already in that direction,
Consumer banking these days,
First Atlanta National Bank introducing,
The banks getting a captive audience for a trouble,
More bank starting.

PACKAGES, strategy *uniform-simplish-repstrong*

This text says something about:
Customers banking,
A package designing Union,
Hefty fees on services,
President of Synergistics Research Corp consulting firm in Atlanta,
A popular Partner Senior Program,
A good deal on loans,
Well as say Ms Moore, as credit union,
Personal line of credit,
Strict price competition,

Figure 8.15: Sets of summary phrases for the PACKAGES text, (in *selection* order).

In the lower summary (representativeness scoring), CLASP begins by selecting a predication from the **`Artist`** clique, but thereafter the clique is entirely within the neighbourhood of the set of selected predications, and so choosing further predications about the Artist becomes less beneficial. The result is a much broader summary, which as well as introducing the Artist, refers to the title of the film (Sidewalk Stories) and the connection with Chaplin, which is an important theme in the text. This is a case where representativeness scoring has clearly improved the summary.

In figure 8.15, the upper summary is dominated by references to banks and banking, while the lower summary, produced with representativeness scoring, only mentions it in one phrase. In particular, 'The banks promoting the packages' is replaced by 'A package designing Union'. Since the source text is in fact all about banks promoting packages, in this case representativeness scoring seems to have gone too far; the narrow summary is better (although both contain many poor summary phrases).

Sad to say, the summaries in figure 8.14 and 8.15 are typical of CLASP's output. Only a little over half the summary phrases make grammatical sense; the rest (i.e. the 45–50% of phrases categorised as *unclear* in figure 8.13) almost all result from errors in CLASP's sentence-by-sentence analysis (usually in parsing). For example, 'Cute child turning out This' (in figure 8.14) arises from a misanalysis of a sentence beginning 'This cute child turns out to be…'. The CLE, having failed to parse the whole segment, found a possible parse for the fragment 'This cute child turns out', in which 'This' was interpreted as a proper noun and the word order was inverted.

8.5.3 *Comparison with other systems*

There are some other systems (described in chapter 2) which, like CLASP, produce summaries consisting not of running text but of short, indicative expressions. However, CLASP's summary phrases are sufficiently different from their output that it would be very difficult to carry out a fair comparative evaluation.

Boguraev and Kennedy's (1997) *capsule overviews* consist of fragments of sentences extracted from the source text. Rather than select such fragments explicitly, their system segments the text, selects entities which are deemed to be important in each section, and outputs a sentence fragment *for each occurrence of each of these entities*. There is little or no control over what the output says about the entities or how many fragments are output, but there is control over the length of the fragments. If the example Boguraev and Kennedy present is representative, capsule overviews are much more readable than CLASP's summaries. This is not to say that the fragments extracted are always complete grammatical phrases – indeed often they are not – but only that the reader has little difficulty in seeing how they could fit into a larger sentence.

Dersy's (1996) program outputs *summary content indicators* consisting either of single words or, more commonly, of word pairs. The word pairs correspond to stem classes that were frequently related in the text, but the pairs themselves are presented raw, with no indication of what the relation was, and there is no intention that they form grammatical phrases. Nevertheless, the output from Dersy's system is not as unpleasant to read as CLASP's summary phrases can sometimes be, perhaps because it is only the determiners, prepositions and other function words in CLASP's summary phrases that cause the reader to try to interpret them syntactically in the first place.

8.6 SENTENCE EXTRACTS VERSUS SUMMARY PHRASES

The intention of generating summary phrases was to allow important information to be expressed more *precisely* and more *concisely* than in a sentence extract. Clearly, summary phrases are indeed more concise than whole sentences, and equally clearly they do allow the condensation stage to select content more precisely, i.e. at a finer granularity, than sentence extraction does.

The question, then, is whether the condensation stage is able to take advantage of this by *accurately* selecting important material. The experiments reported in this chapter suggest that perhaps it is not.

Summaries consisting of five extracted sentences were judged to include on average about 3.5 relevant sentences (section 8.4). If summary phrases identified important content accurately, we might expect a similar or higher number of relevant phrases *in a five-phrase summary*. But even summaries consisting of eight or nine summary phrases could only achieve on average about 3 relevant phrases each (section 8.5.2). On the other hand, if we consider only the summary phrases judged to be *clear*, we see that for *uniform-simplish-repstrong* there were an average of 4.6 clear phrases per story, of which on average 3 were relevant. These figures are similar to those for sentence extraction.

Of course this is a very rough comparison – it does not take into account the expected relevance of randomly-selected sentences and predications, or the fact that both sentences and summary phrases can express more than one relevant concept. There is, however, a possible explanation for the failure to select important simple predications accurately. It is that, inevitably, CLASP's scoring functions for condensation do not reflect the true value of a set of selected nodes – they are merely an attempt to approximate it. Errors in this approximation may lead us to select inappropriate predications in condensation. However, when all the predications corresponding to a surface sentence are selected at once, the consequences of one or two predications scoring particularly high or low will be reduced. That is, by selecting many predications in a single step, we allow some of the errors in our approximation to average out.

This argument suggests that we should expect a trade-off between the precision with which we try to select important material, and the accuracy with which we manage to do so, and that such a trade-off is inevitable even if we have a perfectly accurate analysis of the source text (although doubtless errors in the analysis will increase the variation in the scoring functions). Some ways in which we might try to achieve a compromise between the coarseness of selecting whole sentences and the inaccuracy of selecting individual predications are suggested in the next chapter.

9 CONCLUSIONS

9.1 CLASP IN ITS PRESENT STATE

9.1.1 *Does* CLASP *meet our expectations?*

The source representation

Section 4.1 described the main goal for CLASP: to summarise real-world texts using *linguistic processing* and a *semantic representation*, without restricting the *subject matter* of the text, or depending on features that are specific to particular *form* factors. We hoped that such a representation would allow increased accuracy and precision in selecting important content.

My experiments comparing CLASP's summaries with those produced by NetSumm (section 8.3) suggest that CLASP may indeed be more accurate at selecting relevant sentences, but the difference in target coverage was sufficiently small that this cannot be more than a very tentative conclusion.

CLASP's predication cohesion graphs certainly allow greater *precision* in selecting summary content than, for example, sentence extraction – selection of individual predications demonstrates this. However, as section 8.5 suggests, there may be a loss in accuracy when selection operates at such a fine granularity.

Selection criteria

CLASP's scoring functions for condensation explicitly reflect the criteria of *importance, representativeness* and *cohesiveness* (section 4.3.2). Although representativeness scoring was not beneficial in my sentence extraction experiments (section 8.4), it had a substantial effect on the production of summary phrases (section 8.5.2), where the balance between importance and representativeness scoring controls whether the summary's scope is *narrow* or *broad*. In fact, importance scoring produces sufficiently narrow summaries that cohesiveness does not seem to be a problem. Thus the third criterion seems to be redundant, at least for stories such as those in appendix A.

Summary phrase generation

Summary phrases are intended to indicate selected content more concisely and more precisely than sentence extracts. Evidently, summary phrases are more concise than whole sentences, and they are more precise too, because they are generated from small clusters of individual predications. Where summary phrases are not so successful is in comprehensibility (almost half the summary phrases used in the experiment of section 8.5.2 were judged to be unclear). Some loss of clarity due to errors in analysis was expected, but such a high proportion of unclear phrases makes CLASP's output difficult to read.

9.1.2 *Problems and issues*

Errors in analysis

Inaccurate and fragmentary analysis is the single biggest problem with CLASP as it stands. When parsing fails or when incorrect parses or semantic analysis are produced, correct predications are omitted from the predication cohesion graph, and spurious ones may be added. The resulting errors can lead both to less accurate condensation and to inappropriate or nonsensical summary phrases.

Speed of processing

CLASP is too slow to be used for summarising in any realistic setup, even one in which summaries can be produced in advance. By far the slowest part of the system is syntactic parsing of the source text, which can take hours to process a single sentence. The problem is simply the vast number of possible parses for sentences or sub-strings of them. This suggests that if we could prune some incorrect analyses as soon as they are generated (or avoid generating them at all) we would improve both the speed and the accuracy of parsing.

Loss of cohesive links

In some cases, CLASP fails to identify cohesive links between predications about related entities. This can happen because no anaphor resolution is performed or because the system does not know that certain predicates are themselves related or similar (e.g. **`film`** and **`movie`**, or even **`japanese`** and **`Japan`**). In either case, the loss of cohesive links may stop the condensation stage from recognising the relevance of the predications involved.

Control of summary length

CLASP controls the length of summaries by limiting the number of sentences or predications selected at the condensation stage (section 6.3). There are two problems with this approach: firstly, it does not allow us to specify the actual length of the resulting summary in words; secondly (and more importantly) there is no attempt to choose a good cut-off point in selection. For example, if ten predications are requested, there is no way for CLASP to notice that there are actually nine or eleven particularly salient predications, and therefore to suggest a slightly shorter or longer summary.

Scalability and adaptiveness

Although my experiments have been limited to Wall Street Journal stories, CLASP is designed to have wider applicability, and it should be able to produce indicative summaries from a variety of source texts. However, the parameters for condensation, which currently are set manually, may need to be altered: for example, choosing the right balance between representativeness and importance in the summary may require different settings for different texts.

Ideally, as Skorochod'ko (1971) realised, a summariser should adapt itself to different kinds of input. Skorochod'ko suggested basing condensation on the text structure as revealed by a graph of lexical cohesion; we might also want to set parameters according to the length of the text, the number of simple predications produced, and the shape of the source representation.

Control of clustering in summary phrases

CLASP's algorithm for clustering when generating summary phrases (section 7.2.3) simply adds all the predications it can subject to some simple restrictions. That this does not produce extremely long summary phrases is entirely due to the lack of anaphor resolution: since each entity occurs in predications from a single sentence only, clustering cannot combine predications from different sentences. If CLASP's analysis were to be improved, and especially if anaphors were to be resolved, we would need more stringent ways to restrict clustering.

A second issue with clustering is that when predications are chosen to add to a selected predication, the system does not consider their relevance to the text or their value. A more consistent approach would be to use the scoring functions defined in condensation to choose between alternative summary phrases. This would require closer ties between CLASP's condensation and synthesis stages.

9.2 HOW MIGHT CLASP BE IMPROVED?

9.2.1 *The analysis stage*

Improving sentence-by-sentence analysis

CLASP relies on the CLE to produce semantic representations of individual source sentences. Some errors in this analysis could be avoided by the addition of more grammar rules and lexical entries. Rules for direct and indirect speech and improved treatment of proper names, numerical expressions and abbreviations would be particularly useful in dealing with newspaper texts. Adding rules, however, runs the risk of admitting new, incorrect analyses of sentences that were previously correctly analysed; providing more sortal information for lexical entries and training the CLE's preference system on real data should help avoid this problem and increase accuracy in general, although the slowness of parsing is an obstacle to training on large corpora.

To increase both accuracy and speed of CLE processing, the source text could first be given to a statistical parser or statistical bracketer. The CLE would then be restricted to only consider analyses consistent with the resulting bracketing or parse. Statistical parsers, such as those of Magerman (1995) and Collins (1996), are fast, robust, trainable, and as accurate as parsers using manually-written grammars (Magerman 1995).

Identifying more cohesive links

CLASP's idea of what entities are *similar* (section 5.4.1) is limited to a simple

stemming operation on predicate names. We could give CLASP the ability to recognise more kinds of cohesive links corresponding to various types of lexical cohesion (Halliday and Hasan 1976), by applying a thesaurus or lexical database such as WordNet (Miller et al 1990). WordNet (used in several systems described in section 2.1) represents links between word senses; to apply this in CLASP we would establish what words or word senses corresponded to entities by looking at their semantic heads, and WordNet would then supply relationships such as coordinate terms, synonyms, hyponyms, etc.

To find further cohesive links, some kind of anaphor processing is required. As noted in chapter 4, the very fragmentary nature of CLASP's sentence analysis makes full anaphor resolution unrealistic. It may be, however, that some simple heuristics can suggest what are the likely referents for some anaphors. Even if we are not sufficiently confident to resolve an anaphor in the source representation (i.e. replace two entities by one), we can still use the suggested resolution to identify a similarity between entities, just as we do when they share the same semantic head.

9.2.2 *The condensation stage*

Choice of better scoring functions

What scoring function is appropriate in condensation is likely to depend on the source text as well as the summary requirements. This suggests two directions in which CLASP's scoring functions might be developed. Firstly, the system could have a range of scoring functions appropriate for different kinds of text, and use the shape of the predication cohesion graph, or other information from the source text, to choose between them. (The choice does not have to be rigid, as we can easily take a weighted average of two scoring functions.) Secondly, appropriate scoring functions might be found by some kind of training process. This would of course require training data, either in the form of human selections of simple predications, or sets of target sentences.

Extension of constrained selection

When summaries are to be produced by sentence extraction, CLASP's *constrained selection* (section 6.3) selects not one simple predication at a time, but all the simple predications corresponding to a sentence of source text. When summary phrases are to be produced, simple predications are selected singly, yet each summary phrase can express information from several simple predications, because of clustering. A natural development would be for the synthesis stage to determine for what sets of predications summary phrases could be generated, and for the condensation stage to choose between these sets, rather than between individual predications. All that is required is for constrained selection to be able to consider arbitrary sets of predications, not just those corresponding to whole source sentences. This change would allow CLASP to directly compare the scores of candidate summary phrases, and because several predications would be selected at each step, it might also reduce

the effect of inaccuracies in scoring functions (as described in section 8.6).

Summary length and cost–benefit comparison

Although in some cases we may want to specify an upper limit on summary length, the requirement is unlikely to be for a summary of exactly a certain number of sentences or summary phrases, which is what CLASP produces at present. Rather than selecting a set number of sentences or summary phrases, it would be better if the greedy selection algorithm could compare at each selection step the *benefit* that would be gained from having the extra content in the summary, with the *cost* of increasing the summary length. By specifying a cost function, the user could either impose a strict requirement on length, or suggest an approximate length and allow the selection algorithm to find a cut-off point at which adding further content would have a higher cost than benefit.

This approach would also allow CLASP to avoid selecting extremely long sentences unless they are especially high-scoring. Combined with the extension of constrained selection described in the previous section, the cost-benefit comparison would similarly help solve the problem of controlling clustering in summary phrases, described in section 9.1.2, simply by attaching a high cost to very long summary phrases.

Estimation of summary quality

CLASP's scoring functions provide simple measures of summary quality – for example, importance scores measure what proportion of relevant content is included in the summary. Providing a simple estimation of summary quality as part of the output might benefit users: if the quality is low, they will know not to rely too heavily on the summary. Of course the value of the estimate depends on its accuracy, which would have to be evaluated separately.

9.2.3 *The synthesis stage*

Readability of sentence extraction summaries

A variety of surface-level techniques exist to increase the readability of summaries produced by sentence extraction, centered on identifying anaphors that have no antecedent in the summary (see section 7.1.3). These techniques were not applied in CLASP as they are well-known and unrelated to CLASP's semantic representations or condensation algorithm. To use CLASP's sentence extraction in an operational system, we would need to consider methods such as aggregation, pruning conjunctions, and discarding sentences containing unresolvable anaphors.

Summary phrases

As suggested in section 9.2.2, CLASP could be made to select summary phrases on the basis of all the simple predications they express, not just a single selected predication. This would provide a mechanism for controlling clustering, and choosing between candidate summary phrases that expressed some of the same

important content. Improvements could certainly be made in CLASP's treatment of determiners and quantifiers (section 7.2.3), but the best way to reduce the number of unclear or nonsensical summary phrases would be to reduce the number of erroneous simple predications in the source representation, i.e. to improve the analysis stage.

Surface summary phrases

If we accept that the analysis *is* inaccurate, it may be better to produce indicative summary phrases by extracting fragments of source text. The idea behind generating fresh summary phrases was to allow precise identification of important material, more control over expression, and more conciseness than could be obtained by simply taking surface phrases; however, as noted in section 8.5.3, surface phrases may well be much clearer, even when they are not grammatically complete constituents, than CLASP's summary phrases.

Firstly, we would need to identify which words in a source text sentence correspond to selected simple predications (predications are always derived from individual sentences). Since the majority of selected predications correspond to content words in the text, and we can find head words for the entities they involve, this should not be too difficult. Secondly, we would need to choose appropriate start and end points for a fragment containing these words, for example by applying some very shallow parsing or phrasal parsing to the text, to identify boundaries between constituents. The resulting phrases would be similar to the those in Boguraev and Kennedy's (1997) *capsule overviews*.

9.3 A BROADER SUGGESTION

CLASP shows that, even with a representation designed to be robust, errors in analysis can lead to poor summaries. A reasonable approach in building a new summariser along similar lines might be not to aim for as deep a representation as CLASP's, and certainly not to produce quasi-logical forms for surface sentences. In this section, I sketch how such a system could be built, and what technologies it could use.

9.3.1 *Statistical analysis*

Statistical parsers, trained on suitable corpus data, are as accurate at processing complex source text as systems (such as the CLE) that use manually-written grammars. Magerman's SPATTER (1995) and the head-dependency parser of Collins (1996) establish a model that defines the relative probabilities of individual parses of a surface sentence. These models take into account not only part-of-speech information (as determined by a statistical tagger), but the specific lexical items involved. Compared to earlier probabilistic parsing methods such as Briscoe and Carroll (1993), the use of lexical information gives increased accuracy. With suitable dynamic programming techniques, these

methods allow for very fast processing, as there is no need to consider all possible parses to find the most likely one. For example, Collins' parser could process 200 sentences of Wall Street Journal text per minute (in 1996), making it several orders of magnitude faster than the CLE.

What these parsers do not do, however, is produce any kind of semantic representation. Thus, extracting predications from a quasi-logical form representing sentence meaning, as CLASP does, is not a possibility. However, we can extract a simpler kind of 'predication' directly from the parse tree.

9.3.2 *Head dependencies*

The parsers of both Magerman (1995) and Collins (1996) define a head-child for each constituent in the parse tree, and thus a head-word is associated with each constituent in the conventional way. (For noun phrases, the head-word is the head noun; for verb phrases, clauses and sentences it is the verb; for prepositional phrases it is the preposition.) Collins' parser in particular is concerned with 'dependencies' between these head-words: specifically word X is said to modify word Y ($X \rightarrow Y$) if there are constituents A and B in the parse tree such that X is the head-word of A, Y is the head-word of B, A and B have the same parent P, and B is the head-child of P. For example, in the sentence 'John has a red balloon', there are three dependencies:

1. **John → has**
2. **balloon → has**
3. **red → balloon**

The statistical model used by Collins also involves considering the syntactic categories of A, B and P. For this example, they are as follows:

	A,	*B*,	*P*
1. John → has	**NP,**	**VP,**	**S**
2. balloon → has	**NP,**	**V,**	**VP**
3. red → balloon	**ADJ,**	**N,**	**NP**

We can use this information to distinguish certain kinds of dependencies between words: in the example above, we can infer that 'John' is the head-word of the subject of the verb 'has', and 'balloon' the head-word of the object.

Clearly these dependencies, like CLASP's simple predications, describe relations between events, states and entities. Unlike CLASP's simple predications, each involves only two elements, and entities are represented by surface words rather than by logical variables. It would nevertheless be relatively easy to process these dependencies to obtain simple predications very similar to CLASP's, by creating logical variables to correspond to each noun or verb in a sentence, and combining some groups of dependencies (such as subject-verb and object-verb) into a single predication.

Alternatively, the dependencies themselves could be used as the units of the source representation, by defining links between them along similar lines to CLASP's cohesive links between predications. In the example above, dependencies 1 and 2 would be linked by the common word 'has', and dependencies 2 and 3 would be linked by the common word 'balloon'. Applying a stemming algorithm such as that of Porter (1980) would allow us to identify links between dependencies involving similar, but not identical, heads. The same condensation techniques used in CLASP could then be applied to the resulting graph, although we might expect different parameter settings to be appropriate.

9.3.3 *Syntactic summary phrases*

In section 9.2.3, I outlined how one might go about extracting surface summary phrases corresponding to selected simple predications in CLASP. If we were to have head dependencies as the basic units of the summary representation, extracting corresponding summary phrases would be straightforward, since for each head dependency there is a corresponding constituent *P* in the parse tree. Simply outputting the word sequence for *P* might give an appropriate summary phrase in some cases, but sometimes it would yield unattached prepositional phrases, or verb phrases with no subject. To avoid such output, we could have a list of appropriate syntactic categories for summary phrases (such as NP and S), and search for the smallest constituent above *P* in the parse tree with an appropriate category.

9.3.4 *Comparison with* CLASP

A summarising system built along these lines would in some respects be less ambitious than CLASP. Its analysis would not capture the full range of phenomena that the CLE can deal with – for example, NP movement in a sentence such as 'The man who John thought I saw' would mean that no relation is found between 'saw' and its object, 'man'. In generating summary phrases there would be less scope to differ from the source text, and therefore less flexibility, than there is in CLASP. But there could be many advantages. The analysis would be robust and most likely more accurate than CLASP's, and the summary phrases produced might well be more readable. Above all, such a summariser would be fast, making it of more practical benefit, and allowing more thorough evaluation, than the present system.

APPENDIX A — EXAMPLE SOURCE TEXTS

This appendix contains the following five source texts from the Wall Street Journal: JAPINV, LANEFILM, PACKAGES, SHEEPCHASE, WINEMARKET.

A.1 JAPINV

Japanese investment in Southeast Asia is propelling the region toward economic integration.

Interviews with analysts and business people in the U.S. suggest that Japanese capital may produce the economic cooperation that Southeast Asian politicians have pursued in fits and starts for decades. But Japan's power in the region also is sparking fears of domination and posing fresh policy questions.

The flow of Japanese funds has set in motion "a process whereby these economies will be knitted together by the great Japanese investment machine," says Robert Hormats, vice chairman of Goldman Sachs International Corp.

In the past five years, Japanese companies have tripled their commitments in Asia to $5.57 billion. In Thailand, for example, the government's Board of Investment approved $705.6 million of Japanese investment in 1988, 10 times the U.S. investment figure for the year. Japan's commitment in Southeast Asia also includes steep increases in foreign assistance and trade.

Asia's other cash-rich countries are following Japan's lead and pumping capital into the region. In Taiwan and South Korea, rising wages are forcing manufacturers to seek other overseas sites for labor-intensive production. These nations, known as Asia's "little tigers," also are contributing to Southeast Asia's integration, but their influence will remain subordinate to Japan's.

For recipient countries such as Thailand and Malaysia, the investment will provide needed jobs and spur growth. But Asian nations' harsh memories of their military domination by Japan in the early part of this century make them fearful of falling under Japanese economic hegemony now.

Because of budget constraints in Washington, the U.S. encourages Japan to share economic burdens in the region. But it resists yielding political ground. In the coming decade, analysts say, U.S.–Japanese relations will be tested as Tokyo comes to terms with its new status as the region's economic behemoth.

Japan's swelling investment in Southeast Asia is part of its economic evolution. In the past decade, Japanese manufacturers concentrated on domestic production for export. In the 1990s, spurred by rising labor costs and the strong yen, these companies will increasingly turn themselves into multinationals with plants around the world. To capture the investment, Southeast Asian nations will move to accommodate Japanese business.

These nations' internal decisions "will be made in a way not to offend their largest aid donor, largest private investor and largest lender," says Richard Drobnick, director of the international business and research program at the University of Southern California's Graduate School of Business.

Japanese money will help turn Southeast Asia into a more cohesive economic region. But, analysts say, Asian cooperation isn't likely to parallel the European Common Market approach. Rather, Japanese investment will spur integration of certain sectors, says Kent Calder, a specialist in East Asian economies at the Woodrow Wilson School for Public and Internatonal Affairs at Princeton University. In electronics, for example, a Japanese company might make television picture tubes in Japan, assemble the sets in Malaysia and export them to Indonesia.

"The effect will be to pull Asia together not as a common market but as an integrated production zone," says Goldman Sachs's Mr Hormats.

Countries in the region also are beginning to consider a framework for closer economic and political ties. The economic and foreign ministers of 12 Asian and Pacific nations will meet in Australia next week to discuss global trade issues as well as regional matters such as transportation and telecommunications. Participants will include the U.S., Australia, Canada, Japan, South Korea and New Zealand as well as the six members of the Association of Southeast Asian Nations – Thailand, Malaysia, Singapore, Indonesia, the Philippines and Brunei.

In addition, the U.S. this year offered its own plan for cooperation around the Pacific rim in a major speech by Secretary of State James Baker, following up a proposal made in January by Australian Prime Minister Bob Hawke.

The Baker proposal reasserts Washington's intention to continue playing a leading political role in the region. "In Asia, as in Europe, a new order is taking shape," Mr Baker said. "The U.S., with its regional friends, must play a crucial role in designing its architecture."

But maintaining U.S. influence will be difficult in the face of Japanese dominance in the region. Japan not only outstrips the U.S. in investment flows but also outranks it in trade with most Southeast Asian countries (although the U.S. remains the leading trade partner for all of Asia). Moreover, the Japanese government, now the world's largest aid donor, is pumping far more assistance into the region than the U.S. is. While U.S. officials voice optimism about Japan's enlarged role in Asia, they also convey an undertone of caution. "There's an understanding on the part of the U.S. that Japan has to expand its functions" in Asia, says J Michael Farren, undersecretary of commerce for trade. "If they approach it with a benevolent, altruistic attitude, there will be a net gain for everyone."

Some Asian nations are apprehensive about Washington's demand that Tokyo step up its military spending to ease the U.S. security burden in the region. The issue is further complicated by uncertainty over the future of the U.S.'s leases on military bases in the Philippines and by a possible U.S. troop reduction in South Korea. Many Asians regard a U.S. presence as a desirable counterweight to Japanese influence.

"No one wants the U.S. to pick up its marbles and go home," Mr Hormats says.

For their part, Taiwan and South Korea are expected to step up their own investments in the next decade to try to slow the Japanese juggernaut.

"They don't want Japan to monopolize the region and sew it up," says Chong-sik Lee, professor of East Asian politics at the University of Pennsylvania.

A.2 LANEFILM

As an actor, Charles Lane isn't the inheritor of Charlie Chaplin's spirit. Steve Martin has already laid his claim to that.

But it is Mr Lane, as movie director, producer and writer, who has been obsessed with refitting Chaplin's Little Tramp in a contemporary way. In 1976, as a film student at the Purchase campus of the State University of New York, Mr Lane shot "A Place in Time," a 36-minute black-and-white film about a sketch artist, a man of the streets. Now, 13 years later, Mr Lane has revived his Artist in a full-length movie called "Sidewalk Stories," a poignant piece of work about a modern-day tramp. Of course, if the film contained dialogue, Mr Lane's Artist would be called a homeless person. So would the Little Tramp, for that matter.

I say "contained dialogue" because "Sidewalk Stories" isn't really silent at all. Composer Marc Marder, a college friend of Mr Lane's who earns his living playing the double bass in classical music ensembles, has prepared an exciting, eclectic score that tells you what the characters are thinking and feeling far more precisely than intertitles, or even words, would.

Much of Mr Lane's film takes a highly romanticized view of life on the streets (though probably no more romanticized than Mr Chaplin's notion of the Tramp as the good-hearted free spirit). Filmed in lovely black and white by Bill Dill, the New York streets of "Sidewalk Stories" seem benign. On Wall Street men and women walk with great purpose, noticing one another only when they jostle for cabs. The Artist hangs out in Greenwich Village, on a strip of Sixth Avenue populated by jugglers, magicians and other good-natured hustlers. (This clearly is not real life: no crack dealers, no dead-eyed men selling four-year-old copies of Cosmopolitan, no one curled up in a cardboard box.

The Artist has his routine. He spends his days sketching passers-by, or trying to. At night he returns to the condemned building he calls home. His life, including his skirmishes with a competing sketch artist, seems carefree. He is his own man.

Then, just as the Tramp is given a blind girl to cure in "City Lights," the Artist is put in charge of returning a two-year-old waif (Nicole Alysia), whose father has been murdered by thugs, to her mother. This cute child turns out to be a blessing and a curse. She gives the Artist a sense of purpose, but also alerts him to the serious inadequacy of his vagrant life. The beds at the Bowery Mission seem far drearier when he has to tuck a little girl into one of them at night.

To further load the stakes, Mr Lane dreamed up a highly improbable romance for the Artist, with a young woman who owns her own children's shop and who lives in an expensive

high-rise apartment building. This story line might resonate more strongly if Mr Lane had as strong a presence in front of the camera as he does behind it.

Mr Lane's final purpose isn't to glamorize the Artist's vagabond existence. He has a point he wants to make, and he makes it, with a great deal of force. The movie ends with sound, the sound of street people talking, and there isn't anything whimsical or enviable in those rough, beaten voices.

A.3 PACKAGES

For 10 years, Genie Driskill went to her neighborhood bank because it was convenient. A high-balance customer that banks pine for, she didn't give muchthought to the rates she was receiving, nor to the fees she was paying.

But in August, First Atlanta National Bank introduced its Crown Account, a package designed to lure customers such as Ms Driskill. Among other things, it included checking, safe deposit box and credit card – all for free – plus a good deal on installment loans. All she had to do was put $15,000 in a certificate of deposit, or qualify for a $10,000 personal line of credit.

"I deserve something for my loyalty," she says. She took her business to First Atlanta.

So it goes in the competitive world of consumer banking these days. For nearly a decade, banks have competed for customers primarily with the interest rates they pay on their deposits and charge on their loans. The competitive rates were generally offset by hefty fees on various services.

But many banks are turning away from strict price competition. Instead, they are trying to build customer loyalty by bundling their services into packages and targeting them to small segments of the population.

"You're dead in the water if you aren't segmenting the market," says Anne Moore, president of Synergistics Research Corp., a bank consulting firm in Atlanta.

NCNB Corp. of Charlotte, N.C., recently introduced its Financial Connections Program aimed at young adults just starting careers. The program not only offers a pre-approved car loan up to $18,000, but throws in a special cash-flow statement to help in saving money.

In September, Union Planters Corp. of Memphis, Tenn., launched The Edge account, a package designed for the "thirtysomething" crowd with services that include a credit card and line of credit with no annual fees, and a full percentage point off on installment loans. The theory: Such individuals, many with young children, are in their prime borrowing years – and, having borrowed from the bank, they may continue to use it for other services in later years.

For some time, banks have been aiming packages at the elderly, the demographic segment with the highest savings. Those efforts are being stepped up. Judie MacDonald, vice president of retail sales at Barnett Banks Inc. of Jacksonville, Fla., says the company now targets sub-segments within the market by tailoring its popular Seniors Partners Program to various life styles. "Varying age, geography and life-style differences create numerous sub-markets," Ms MacDonald says. She says individual Barnett branches can add different benefits to their Seniors Partners package – such as athletic activities or travel clubs – to appeal to local market interests. "An active 55-year-old in Boca Raton may care more about Senior Olympic games, while a 75-year-old in Panama City may care more about a seminar on health," she says.

Banks have tried packaging before. In 1973, Wells Fargo & Co. of San Francisco launched the Gold Account, which included free checking, a credit card, safe-deposit box and travelers checks for a $3 monthly fee.

The concept begot a slew of copycats, but the banks stopped promoting the packages. One big reason: thin margins. Many banks, particularly smaller ones, were slow to computerize and couldn't target market niches that would have made the programs more profitable. As banks' earnings were squeezed in the mid-1970s, the emphasis switched to finding ways to cut costs.

But now computers are enabling more banks to analyze their customers by age, income and geography. They are better able to get to those segments in the wake of the deregulation that began in the late 1970s. Deregulation has effectively removed all restrictions on what banks can pay for deposits, as well as opened up the field for new products such as high-rate CDs. Where a bank once offered a standard passbook savings account, it began offering money-market accounts, certificates of deposit and interest-bearing checking, and staggering rates based on the size of deposits.

The competition has grown more intense as bigger banks such as Norwest Corp. of

Minneapolis and Chemical Banking Corp. of New York extend their market-share battles into small towns across the nation. "Today, a banker is worrying about local, regional and money-center [banks], as well as thrifts and credit unions," says Ms Moore at Synergistics Research. "So people who weren't even thinking about targeting 10 years ago are scrambling to define their customer base."

The competition has cultivated a much savvier consumer. "The average household will spread 19 accounts over a dozen financial institutions," says Michael P Sullivan, who runs his own bank consulting firm in Charlotte, N.C. "This much fragmentation makes attracting and keeping today's rate-sensitive customers costly."

Packages encourage loyalty by rewarding customers for doing the bulk of their banking in one place. For their troubles, the banks get a larger captive audience that is less likely to move at the drop of a rate. The more accounts customers have, Mr Sullivan says, the more likely they are to be attracted to a package – and to be loyal to the bank that offers it. That can pay off down the road as customers, especially the younger ones, change from borrowers to savers/investors.

Packaging has some drawbacks. The additional technology, personnel training and promotional effort can be expensive. Chemical Bank spent more than $50 million to introduce its ChemPlus line, several packages aimed at different segments, in 1986, according to Thomas Jacob, senior vice president of marketing. "It's not easy to roll out something that comprehensive, and make it pay," Mr Jacob says.

Still, bankers expect packaging to flourish, primarily because more customers are demanding that financial services be tailored to their needs. "These days, banking customers walk in the door expecting you to have a package especially for them," Ms Moore says. Some banks are already moving in that direction, according to Alvin T Sale, marketing director at First Union Corp. in Charlotte. First Union, he says, now has packages for seven customer groups. Soon, it will split those into 30.

Says Mr Sale: "I think more banks are starting to realize that we have to be more like the department store, not the boutique."

A.4 SHEEPCHASE

Judging from the Americana in Haruki Murakami's "A Wild Sheep Chase" (Kodansha, 320 pages, $18.95), baby boomers on both sides of the Pacific have a lot in common. Although set in Japan, the novel's texture is almost entirely Western, especially American. Characters drink Salty Dogs, whistle "Johnny B Goode" and watch Bugs Bunny reruns. They read Mickey Spillane and talk about Groucho and Harpo. They worry about their careers, drink too much and suffer through broken marriages and desultory affairs. This is Japan?

For an American reader, part of the charm of this engaging novel should come in recognizing that Japan isn't the buttoned-down society of contemporary American lore. It's also refreshing to read a Japanese author who clearly doesn't belong to the self-aggrandizing "we-Japanese" school of writers who perpetuate the notion of the unique Japanese, unfathomable by outsiders. If "A Wild Sheep Chase" carries an implicit message for international relations, it's that the Japanese are more like us than most of us think.

That's not to say that the nutty plot of "A Wild Sheep Chase" is rooted in reality. It's imaginative and often funny. A disaffected, hard-drinking, nearly-30 hero sets off for snow country in search of an elusive sheep with a star on its back at the behest of a sinister, erudite mobster with a Stanford degree. He has in tow his prescient girlfriend, whose sassy retorts mark her as anything but a docile butterfly. Along the way, he meets a solicitous Christian chauffeur who offers the hero God's phone number; and the Sheep Man, a sweet, roughhewn figure who wears – what else – a sheepskin.

The 40-year-old Mr Murakami is a publishing sensation in Japan. A more recent novel, "Norwegian Wood" (every Japanese under 40 seems to be fluent in Beatles lyrics), has sold more than four million copies since Kodansha published it in 1987. But he is just one of several youthful writers – Tokyo's brat pack – who are dominating the best-seller charts in Japan. Their books are written in idiomatic, contemporary language and usually carry hefty dashes of Americana.

A.5 WINEMARKET

When Warren Winiarski, proprietor of Stag's Leap Wine Cellars in Napa Valley, announced a $75 price tag for his 1985 Cask 23 Cabernet this fall, few wine shops and restaurants around the country balked. "This is the peak of my wine-making experience," Mr Winiarski declared when he introduced the wine at a dinner in New York, "and I wanted to single it out as such."

It is in my estimation the best wine Stag's Leap has produced, and with fewer than 700 cases available, it is sure to sell quickly. The price is a new high for California Cabernet Sauvignon, but it is not the highest. Diamond Creek 1985 Lake Vineyard Cabernet weighed in this fall with a sticker price of $100 a bottle.

One of the fastest growing segments of the wine market is the category of superpremiums – wines limited in production, of exceptional quality (or so perceived, at any rate), and with exceedingly high prices. For years, this group included a stable of classics – Bordeaux first growths (Lafite-Rothschild, Latour, Haut-Brion, Petrus), Grand Cru Burgundies (Romanee-Conti and La Tache) deluxe Champagnes (Dom Perignon or Roederer Cristal), rarefied sweet wines (Chateau Yquem or Trockenbeerenauslesen Rieslings from Germany, and Biondi-Santi Brunello Riserva from Tuscany). These first magnitude wines ranged in price from $40 to $125 a bottle.

In the last year or so, however, this exclusive club has taken in a host of flashy new members. The classics have zoomed in price to meet the competition, and it almost seems that there's a race on to come up with the priciest single bottle, among current releases from every major wine region on the globe.

France can boast the lion's share of high-priced bottles. Bordeaux's first growths from 1985 and 1986 are $60 to $80 each (except for the smallest in terms of production, Chateau Petrus, which costs around $250]). These prices seem rather modest, however, in light of other French wines from current vintages. Chateau Yquem, the leading Sauternes, now goes for well over $100 a bottle for a lighter vintage like 1984; the spectacularly rich 1983 runs $179.

In Champagne, some of the prestige cuvees are inching toward $100 a bottle. The first Champagne to crack that price barrier was the 1979 Salon de Mesnil Blanc de Blancs. The '82 Salon is $115. Roederer Cristal at $90 a bottle sells out around the country and Taittinger's Comtes de Champagne Blanc de Blancs is encroaching upon that level. The great reds of the Rhone Valley have soared in price as well. E. Guigal's 1982 Cote Rotie La Landonne, for example, is $120.

None of France's wine regions can steal a march on Burgundy, however. The six wines of the Domaine de la Romanee-Conti, 72 of the most precious acres of vineyard anywhere in the world, have commanded three-digit price tags for several years now. With the 1985 vintage, they soared higher: La Tache, $195; Richebourg, $180; Romanee-Conti, $225. Another small Burgundy estate, Coche-Dury, has just offered its 1987 Corton-Charlemagne for $155.

From Italy there is Angelo Gaja Barbaresco at $125 a bottle, Piero Antinori's La Solaia, a $90 Cabernet from Tuscany, and Biondi-Santi Brunello at $98. Spain's Vega Secilia Unico 1979 (released only in its 10th year) is $70, as is Australia's Grange Hermitage 1982.

"There are certain cult wines that can command these higher prices," says Larry Shapiro of Marty's, one of the largest wine shops in Dallas. "What's different is that it is happening with young wines just coming out. We're seeing it partly because older vintages are growing more scarce."

Wine auctions have almost exhausted the limited supply of those wines, Mr Shapiro continued: "We've seen a dramatic decrease in demand for wines from the '40s and '50s, which go for $300 to $400 a bottle. Some of the newer wines, even at $90 to $100 a bottle or so, almost offer a bargain."

Take Lake Vineyard Cabernet from Diamond Creek. It's made only in years when the grapes ripen perfectly (the last was 1979) and comes from a single acre of grapes that yielded a mere 75 cases in 1987. Owner Al Brownstein originally planned to sell it for $60 a bottle, but when a retailer in Southern California asked, "Is that wholesale or retail?" he re-thought the matter. Offering the wine at roughly $65 a bottle wholesale ($100 retail), he sent merchants around the country a form asking them to check one of three answers: 1) no, the wine is too high (2 responses); 2) yes, it's high but I'll take it (2 responses); 3) I'll take all I can get (58 responses). The wine was shipped in six-bottle cases instead of the usual 12, but even at that it was spread thin, going to 62 retailers in 28 states.

"We thought it was awfully expensive," said Sterling Pratt, wine director at Schaefer's in Skokie, Ill., one of the top stores in suburban Chicago, "but there are people out there with very different opinions of value. We got our two six-packs – and they're gone."

Mr Pratt remarked that he thinks steeper prices have come about because producers don't like to see a hit wine dramatically increase in price later on. Even if there is consumer resistance at first, a wine that wins high ratings from the critics will eventually move. "There may be sticker-shock reaction initially," said Mr Pratt, "but as the wine is talked about and starts to sell, they eventually get excited and decide it's worth the astronomical price to add it to their collection."

"It's just sort of a one-upsmanship thing with some people," added Larry Shapiro. "They like to talk about having the new Red Rock Terrace {of Diamond Creek's Cabernets} or the Dunn 1985 Cabernet, or the Petrus. Producers have seen this market opening up and they're now creating wines that appeal to these people."

That explains why the number of these wines is expanding so rapidly. But consumers who buy at this level are also more knowledgeable than they were a few years ago. "They won't buy if the quality is not there," said Cedric Martin of Martin Wine Cellar in New Orleans. "Or if they feel the wine is overpriced and they can get something equally good for less." Mr Martin has increased prices on some wines (like Grgich Hills Chardonnay, now $32) just to slow down movement, but he is beginning to see some resistance to high-priced red Burgundies and Cabernets and Chardonnays in the $30 to $40 range.

Image has, of course, a great deal to do with what sells and what doesn't, and it can't be forced. Wine merchants can't keep Roederer Cristal in stock, but they have to push Salon le Mesnil, even lowering the price from $115 to $90. It's hardly a question of quality – the 1982 Salon is a beautiful wine, but, as Mr. Pratt noted, people have their own ideas about value.

It's interesting to find that a lot of the expensive wines aren't always walking out the door. In every major market in the U.S., for instance, you can buy '86 La Tache or Richebourg, virtually all of the first growth Bordeaux (except Petrus), as well as Opus One and Dominus from California and, at the moment, the Stag's Leap 1985 Cask 23.

With the biggest wine-buying period of the year looming as the holidays approach, it will be interesting to see how the superpremiums fare. By January it should be fairly clear what's hot – and what's not.

Ms Ensrud is a free-lance wine writer in New York.

APPENDIX B — EXAMPLE SUMMARIES

Here are sentence extraction summaries and summary phrases produced by CLASP for each of the stories in appendix A. The sentence extracts were obtained by selecting five sentences with the scoring function *uniform-simplish.* The summary phrases were produced with *uniform-simplish-repstrong*, and are presented in order of selection. (See figure 8.3 for definitions of these scoring functions.)

B.1 JAPINV

Sentence extraction:

In Thailand, for example, the government's Board of Investment approved $705 million of Japanese investment in 1988, 10 times the US investment figure for the year. Asia's other cash-rich countries are following Japan's lead and pumping capital into the region. Japan's swelling investment in Southeast Asia is part of its economic evolution. Rather, Japanese investment will spur integration of certain sectors, says Kent Calder, a specialist in East Asian economies at the Woodrow Wilson School for Public and Internatonal Affairs at Princeton University. Japan not only outstrips the US in investment flows but also outranks it in trade with most Southeast Asian countries (although the US remains the leading trade partner for all of Asia).

Summary phrases:

Japanese investment in Southeast Asia propelling the region toward economic integration,
America encouraging Japan,
Asia's cash-rich countries,
Asian nations,
Japan's commitment in Southeast Asia including steep increases in foreign trade, in foreign assistance,
The University of Southern California's Graduate School,
Starts for decades,
Japanese capital producing the economic cooperation,
Washington's demanding that Tokyo step,
Asian cooperation parallelling the European Common Market likely,
Thailand Indonesia,
Trade with most Southeast Asian countries,
Countries in the region,
Economic burdens in the region.

B.2 LANEFILM

Sentence extraction:

In 1976, as a film student at the Purchase campus of the State University of New York, Mr Lane shot A Place in Time, a 36 minute black-and-white film about a sketch artist, a man of the streets. Now, 13 years later, Mr Lane has revived his Artist in a full-length movie called Sidewalk Stories, a poignant piece of work about a modern-day tramp. Composer Marc Marder, a college friend of Mr Lane's who earns his living playing the double bass in classical music ensembles, has prepared an exciting, eclectic score that tells you what the characters are thinking and feeling far more precisely than intertitles, or even words, would. To further load the stakes, Mr Lane dreamed up a highly improbable romance for the Artist, with a young woman who owns her own children's shop and who lives in an expensive high-rise apartment building. Mr Lane's final purpose isn't to glamorize the Artist's vagabond existence.

Summary phrases:

Mr Lane reviving an Artist in a full-length,
A man of the streets,
Sidewalk Stories about a modern-day tramp,
The film containing dialogue,
Movie director,
The double bass in classical music preparing an eclectic score,
Woman walk,
Charlie Chaplin's spirit,
Cute child turning out This,
Story line resonating This.

B.3 PACKAGES

Sentence extraction:

You're dead in the water if you aren't segmenting the market, says Anne Moore, president of Synergistics Research Corp, a bank consulting firm in Atlanta. In September, Union Planters Corp of Memphis, Tenn, launched The Edge account, a package designed for the thirtysomething crowd with services that include a credit card and line of credit with no annual fees, and a full percentage point off on installment loans. Today, a banker is worrying about local, regional and money-center banks, as well as thrifts and credit unions, says Ms Moore at Synergistics Research. Packages encourage loyalty by rewarding customers for doing the bulk of their banking in one place. Chemical Bank spent more than $50 million to introduce its ChemPlus line, several packages aimed at different segments, in 1986, according to Thomas Jacob, senior vice president of marketing.

Summary phrases:

Customers banking,
A package designing Union,
Hefty fees on services,
President of Synergistics Research Corp consulting firm in Atlanta,
A popular Partner Senior Program,
A good deal on loans,
Well as say Ms Moore, as credit union,
Personal line of credit,
Strict price competition.

B.4 SHEEPCHASE

Sentence extraction:

Judging from the Americana in Haruki Murakami's A Wild Sheep Chase (Kodansha, 320 pages, $19), baby boomers on both sides of the Pacific have a lot in common. For an American reader, part of the charm of this engaging novel should come in recognizing that Japan isn't the buttoned-down society of contemporary American lore. A disaffected, hard-drinking, nearly-30 hero sets off for snow country in search of an elusive sheep with a star on its back at the behest of a sinister, erudite mobster with a Stanford degree. A more recent novel, Norwegian Wood (every Japanese under 40 seems to be fluent in Beatles lyrics), has sold more than four million copies since Kodansha published it in 1987. But he is just one of several youthful writers – Tokyo's brat pack – who are dominating the best-seller charts in Japan.

Summary phrases:

Set in Japan, in the novel,
A solicitous Christian chauffeur offering the hero God's telephone number,
A book carrying hefty dashes usually,
Society of American lore,
The behest of an erudite mobster with a Stanford degree,
A Wild Sheep Chase carrying an implicit message for relations,
Search of an elusive sheep with a star,
Baby boomers on both sides of the Pacific having a lot in common,
The roughhewn Sheep Man figure,
A prescient girlfriend whose sassy retorting,
A sensation in Japan publishing,
Talk about Harpo, about Groucho,
Idiomatic language,
Part of the charm.

B.5 WINEMARKET

Sentence extraction:

These first magnitude wines ranged in price from $40 to $125 a bottle. There are certain cult wines that can command these higher prices, says Larry Shapiro of Marty's, one of the largest wine shops in Dallas. We've seen a dramatic decrease in demand for wines from the 40 and 50, which go for $300 to $400 a bottle. Mr Martin has increased prices on some wines (like Grgich Hills Chardonnay, now $32) just to slow down movement, but he is beginning to see some resistance to high-priced red Burgundies and Cabernets and Chardonnays in the $30 to $40 range. Wine merchants can't keep Roederer Cristal in stock, but they have to push Salon le Mesnil, even lowering the price from $115 to $90.

Summary phrases:

Certain cult wines commanding these higher prices,
Cabernets in the 30 dollars to 40 dollars ranging high-priced,
Share of high-priced bottling the lion's,
People having a Pratt.

BIBLIOGRAPHY

ARPA (Advanced Research Projects Agency). *Fifth Message Understanding Conference* (MUC-5), Baltimore, Maryland. Software and Intelligent Systems Technology Office, 1993.

D Allport. The TICC: parsing interesting text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 211–218. ACL, 1988.

H Alshawi. *The Core Language Engine*. MIT press, Cambridge, MA, 1992.

H Alshawi, D Carter, R Crouch, S Pulman, M Rayner, A Smith. CLARE-3 *software manual*. SRI International, Cambridge, 1992.

C Aone, M E Okurowski, J Gorlinsky, B Larsen. A scalable summarization system using robust NLP. In Mani & Maybury (editors), *Proceedings of the* ACL/EACL'97 *workshop on Intelligent Scalable Text Summarization*, pages 66–73, Madrid, 1997.

R Barzilay, M Elhadad. Using lexical chains for text summarisation. In Mani & Maybury (editors), *Proceedings of the* ACL/EACL'97 *workshop on Intelligent Scalable Text Summarization*, pages 10–17, Madrid, 1997.

P B Baxendale. Man-made index for technical literature – an experiment. *IBM Journal of Research and Development*, 2(4):354–361, 1958.

M Benbrahim, K Ahmad. Computer-aided lexical cohesion analysis and text abridgement. Computing Sciences Report CS-94-11, University of Surrey, 1994.

B Boguraev, C Kennedy. Salience-based content characterisation of test documents. In Mani & Maybury (editors), *Proceedings of the* ACL/EACL'97 *workshop on Intelligent Scalable Text Summarization*, pages 2–9, Madrid, 1997.

R Brandow, K Mitze, L F Rau. Automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 31(5):675–685, 1995.

T Briscoe, J Carroll. Generalised LR parsing of natural language with unification-based grammars. *Computational Linguistics* 19(1):25–60, 1993.

N H M Caldwell. An investigation into shallow processing for summarisation. Computer science tripos part II project, University of Cambridge Computer Laboratory, 1994.

M Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 184–191, Santa Cruz, CA, 1996.

R E Cullingford. SAM. In Schank & Riesbeck (editors), *Inside Computer Understanding*, Lawrence Erlbaum Associates, Hillsdale NJ, 1981.

G F DeJong. An overview of the FRUMP system. In Lehnert & Ringle (editors), *Strategies for natural language processing*. Lawrence Erlbaum Associates, Hillsdale HJ, 1982.

G F DeJong. Skimming stories in real time: an experiment in integrated understanding. Research report 158, Yale University Department of Computer Science, 1979.

J Dersy. Producing summary content indicators for retrieved texts. MPhil dissertation, University of Cambridge Department of Engineering, 1996.

L L Earl. Experiments in automatic extracting and indexing. *Information Storage and Retrieval* 6:313–334, 1970.

H P Edmundson. New methods in automatic abstracting. *Journal of the Association for Computing Machinery* 16(2):264–285, 1969.

B Endres-Niggemeyer, J Hobbs, K Sparck Jones. *Summarizing text for intelligent communication*, Dagstuhl seminar report 79, 1993.

G W Furnas, T K Landauer, L M Gomez, S T Dumais. The vocabulary problem in human-system communication. *Communications of the ACM* 30(11):964–971, 1987.

P Gladwin, S Pulman, K Sparck Jones. Shallow processing and automatic summarising: a first study. Technical report 223, University of Cambridge Computer Laboratory, 1991.

B J Grosz, A K Joshi, S Weinstein. Centering: a framework for modeling the local coherence of discourse. *Computational Linguistics* 21(2):203–225, 1995.

B J Grosz, C L Sidner. Attentions, intentions and the structure of discourse. *Computational Linguistics* 12:175–204, 1986

U Hahn. Topic parsing: accounting for text macro structures in full-text analysis. *Information processing and management* 26(1):135–170, 1990.

M A K Halliday, R Hasan. *Cohesion in English*. Longman, London, 1976.

T F Hand. A proposal for task-based evaluation of text summarisation systems. In Mani & Maybury (editors), *Proceedings of the* ACL/EACL'97 *workshop on Intelligent Scalable Text Summarization*, pages 31–38, Madrid, 1997.

D K Harman. The first text retrieval conference (TREC-1). NIST, US Department of Commerce, 1993.

J R Hobbs. *Literature and cognition*. Center for the study of language and information, Stanford CA, 1990.

E Hovy, C Y Lin. Automated text summarisation in SUMMARIST. In Mani & Maybury (editors), *Proceedings of the* ACL/EACL'97 *workshop on Intelligent Scalable Text Summarization*, pages 18–24, Madrid, 1997.

J Janos. Theory of functional sentence perspective and its application for the purposes of automatic extracting. *Information Processing and Management* 15:19–25, 1979.

F C Johnson, C D Paice, W J Black, A P Neal. The application of linguistic processing to automatic abstract generation. *Journal of document and text management* 1(3):215–241, 1993.

W Kintsch, T A van Dijk. Toward a model of text comprehension and production. *Psychologial Review* 85(5):363–394, 1978.

J Kupiec, J Pedersen, F Chen. A trainable document summariser. In *Proceedings of the 18th ACM SIGIR conference,* pages 68–73, Springer, 1995.

W G Lehnert. Plot units: a narrative summarization strategy. In Lehnert & Ringle (editors), *Strategies for natural language processing*. Lawrence Erlbaum Associates, Hillsdale HJ, 1982.

E D Liddy. The discourse-level structure of empirical abstracts: an exploratory study. *Information processing and management* 27(1):55–81, 1991.

H P Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

D Magerman. Statistical decision-tree models for parsing. In *Proceedings of the 33rd Meeting of the Association for Computational Linguistics*, pages 276–283. Boston, MA, 1995.

W C Mann, S Thompson. Rhetorical structure theory: description and constructions of text structures. In Kempen (editor), *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics*, pages 85–96. Martinus Ninjhoff Publishers, 1987.

D Marcu. From discourse structures to text summaries. In Mani & Maybury (editors), *Proceedings of the* ACL/EACL'97 *workshop on Intelligent Scalable Text Summarization*, pages 82–88, Madrid, 1997.

S J Mason, H J Zimmerman. *Electronic circuits, signals and systems*. Wiley, New York, 1960.

P H Matthews. *Syntax*. Cambridge University Press, 1981.

M Maybury. Generating summaries from event data. *Information processing and management* 31(5):735–751, 1995.

R Merchant. Tipster Program Overview. In *Tipster Text Program*, pages 1–2, Fredericksburg, VA, 1993.

S Miike, E Itoh, K Ono, K Sumita. A full-text retrieval system with a dynamic abstract generation function. In *Proceedings of the 17th ACM SIGIR conference,* pages 152–161, Springer, 1994.

G A Miller, G R Beckwith, C Fellbaum, D Gross, K J Miller. Five papers on WordNet. CSL report 43, Cognitive Science Laboratory, Princeton University, Princeton NJ, 1990.

J Minel, S Nugier, G Piat. How to appreciate the quality of automatic text summarisation? In Mani & Maybury (editors), *Proceedings of the* ACL/EACL'97 *workshop on Intelligent Scalable Text Summarization*, pages 25–30, Madrid, 1997.

M Mitra, A Singhal, C Buckley. Automatic text summarization by paragraph extraction. In Mani & Maybury (editors), *Proceedings of the* ACL/EACL'97 *workshop on Intelligent Scalable Text Summarization*, pages 39–46, Madrid, 1997.

A H Morris, G M Kasper, D A Adams. The effects and limitations of automated text condensing on reading comprehension performance. *Information Systems Research* 3(1):17–35.

C D Paice. Constructing literature abstracts by computer: techniques and prospects. *Information processing and management* 26(1):176–186, 1990.

C D Paice. The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases. In Oddy, Robertson, van Rijsbergen & P W Williams (editors), *Information Retrieval Research*, pages 172–191. Butterworths, 1981.

C D Paice, P A Jones. The identification of important concepts in highly structured technical papers. In *Proceedings of the 16th Annual International ACM SIGIR Conference of Research and Development in Information Retrieval,* pages 69–78, 1993.

J Pollock, A Zamora. Automatic abstracting research at chemical abstracts service. *Journal of Chemical Information and Computer Sciences* 15(4), 1975.

M F Porter. An algorithm for suffix-stripping. *Program* 14(3):130–137, 1980.

K Preston, S Williams. Managing the information overload. *Physics in Business*. Institute of Physics, June 1994.

C V Ramamoorthy. Analysis of graphs by connectivity considerations. *Journal of the Association for Computing Machinery* 13(2):211–222, 1966.

G J Rath, A Resnick, R Savage. The formation of abstracts by the selection of sentences: Part I: sentence selection by man and machines. *American Documentation* 12(2):139–141, 1961.

L F Rau, P S Jacobs, U Zernik. Information extraction and text summarization using linguistic knowledge acquisition. *Information Processing and Management* 25(4):419–428, 1989.

D E Rumelhart. Understanding and summarising brief stories. In Laberge & Samuels, *Basic processes in reading: perception and comprehension*, pages 265–303. Lawrence Erlbaum Associates, 1977.

J E Rush, R Salvador, A Zamora. Automatic abstracting and indexing. II. Production of indicative abstracts by application of contextual inference and syntactic coherence criteria. *Journal of the American Society for Information Science*, 22(4):260–274, 1971.

G Salton, J Allan, C Buckley, A Singhal. Automatic analysis, theme generation and summarization of machine-readable texts. *Science* 264:1421–1426, 1994.

G Salton, J Allan. Selective text utilization and text traversal. Technical report 93–1366, Department of Computer Science, Cornell University, Ithaca, NY, 1993.

G Salton, A Singhal, M Mitra, C Buckley. Automatic text structuring and summarization. *Information Processing and Management* 33(2):193–207, 1997.

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

C L Sidner. Focusing in the comprehension of definite anaphora. In Brady & Berwick (editors), *Computational Models of Discourse*, pages 267–330, MIT Press, Cambridge, MA, 1983.

E F Skorochod'ko. Adaptive method of automatic abstracting and indexing. *Information Processing* 71, North–Holland, 1971.

K Sparck Jones. Discourse modelling: where are we now and where should we be going? *Working Notes. AAAI Fall Symposium on Discourse Structure in Natural Language Understanding and Generation*, pages 142–145. Menlo Park, CA, 1991.

K Sparck Jones. 'Notes on summarising, 1989–92'. (Unpublished working notes), 1989–1992.

K Sparck Jones. What might be in a summary? In Knorz, Krause & Womser-Hacker (editors), *Information Retrieval 93: Von der Modellierung zur Anwendung*, pages 9–26. Universitatsverlag, Konstanz, 1993.

K Sparck Jones. Discourse modelling for automatic summaries. In Hajicova, Cervenka, Leska and Sgall (editors), *Prague Linguistic Circle Papers*, volume 1, pages 201–227, 1995.

K Sparck Jones. Automatic summarising: factors and directions. In Mani & Maybury (editors), *Advances in Automatic Text Summarization*, MIT Press, 1999.

K Sparck Jones, J Galliers. *Evaluating natural language processing systems*. Lecture Notes in Artificial Intellingence 1083, Springer, 1996.

K Sumita, L Ono, T Chino, T Ukita, S Amano. A discourse structure analyser for Japanese text. In *Proceedings of the International Conference on Fifth Generation Computer Systems* 1992, pages 1133–1140, 1992.

J I Tait. Automatic summarizing of English texts. Technical report 47, University of Cambridge Computer Laboratory, 1983.

S L Taylor, G K Krulee. Experiments with an automatic abstracting system. In *Proceedings of the ASIS annual meeting,* vol 14, Chicago, 1977.

S L Taylor. Automatic abstracting by applying graphical techniques to semantic networks. Doctoral dissertation, Northwestern University, 1975.

S R Young, P J Hayes. Automatic classification and summarisation of banking telexes. *Proceedings: Second Conference on Artificial Intelligence Applications*, pages 402–408, 1985.

T A van Dijk, W Kintsch. *Strategies of Discourse Comprehension.* Academic Press, New York, 1983.