Technical Report

UCAM-CL-TR-632
ISSN 1476-2986

Number 632

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Between shallow and deep: an experiment in automatic summarising

R.I. Tucker and K. Spärck Jones

April 2005

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/TechReports/

ISSN 1476-2986

Between shallow and deep: an experiment in automatic summarising

R.I. Tucker and K. Spärck Jones

Abstract

This paper describes an experiment in automatic summarising using a general-purpose strategy based on a compromise between shallow and deep processing. The method combines source text analysis into simple logical forms with the use of a semantic graph for representation and operations on the graph to identify summary content. The graph is based on predications extracted from the logical forms, and the summary operations apply three criteria, namely importance, representativeness, and cohesiveness, in choosing node sets to form the content representation for the summary. This is used in different ways for output summaries. The paper presents the motivation for the strategy, details of the CLASP system, and the results of initial testing and evaluation on news material.

Note 2005, on text of 2000

This paper reports a fairly early attempt to automate graph-based summarising. It was not wholly successful in practice, but this is attributable to detail rather than problems with the generic approach. The evaluation was also very limited, since at the time the work was done in the late 1990s, there was no serious evaluation data available. The paper is printed now since there has been growing interest in this generic approach to summarising.

1 Introduction

In general, approaches to automatic summarising have fallen into two classes: those based on text extraction, i.e. *shallow* approaches (e.g. Hovy and Lin's SUMMARIST and other examples in Mani and Maybury (1999), Goldstein et al. (1999)), and those based on domain frame instantiation, i.e. *deep* approaches (e.g. Cullingford (1982), Hahn (1990)). The study reported here explored an intermediate strategy designed to avoid the well-known problems of shallow methods without calling for the effort and resources that domain methods require. Specifically, it was intended, like the former, to be a general technique, not an application-dependent one. The essence of the strategy was therefore to go below the given input text to build a source content representation, using a general-purpose parser to identify predicate-argument structures; to form a predication network from these; to operate on the network to identify a substructure satisfying criteria for

summary content; and to produce text from the selected substructure as the output summary.

Previous research aimed at overcoming the defects of statistically-based sentence extraction has pursued other directions, for instance exploiting discourse structure, in a variety of forms, along with or instead of statistical data (Barzilay and Elhadad (1999), Lehmam (1999), Saggion and Lapalme (2000), Marcu (1999), Moens and Dumortier (2000)); working with subsentential fragments from key sentences (Boguraev and Kennedy (1999), Boguraev and Neff (2000), Oka and Ueda (2000)) or alternatively larger extracts (Strzalkowski et al. 1999); and applying discourse organisation and text generation techniques for a more coherent and readable output (McKeown, Robin and Kukich (1995), Maybury (1995)). Research aimed at escaping from the source text altogether, however, has required a much heavier-duty apparatus exploiting at least world knowledge (DeJong (1982), Hahn (1990), Hahn and Reimer (1999)), and possibly discourse structure specific to the text genre for a domain (Lehmam (1999), Moens and Dumortier (2000), Strzalkowski et al. (1999)). Practical considerations and opportunities, on the other hand, have prompted the use of output phrase lists and highlighting for summary 'keys' rather than full text summaries (Berger and Mittal (2000), Boguraev and Neff (2000), Witbrock and Mittal (1999)). There are problems with all of these: tweaking extracted source text, or motivating extraction by invoking discourse structure cues, has not so far delivered high-class output summaries other than in limited applications; domain frame approaches are costly; and phrase lists are adequate only for some tasks, like document retrieval where the full text is also to hand. The research on summarising described below, adopting an approach which has not been investigated elsewhere, was designed to avoid all these problems. It did not wholly succeed, but as the experiments were only initial ones and we believe the motivating arguments for our strategy are sound, our findings are a fair base for further research.

Section 2 outlines the context for our work in relation to the factors affecting summarising and a framework model for summarising systems. Section 3 gives an overview of the CLASP system, which is described in more detail in Sections 4, 5 and 6. Our tests and evaluation are reported in Section 7 and we assess our approach and findings in Section 8. The work as a whole is fully detailed in Tucker (1999).

2 Background

We have made use, in motivating our approach, of a *general model* of summarising systems, and we have responded in a particular way to the *context factors* that determine specific system design. The model and factors are introduced in Spärck Jones (1999), and their relation to CLASP and our tests is discussed in Tucker (1999), so what follows is only a reference summary of the main points involved,

2.1 General model

We assume a three-stage model of summarising with first, source text *analysis* to derive a source representation; second, *condensation* of the source representation to form a summary representation; and third, *synthesis* from the summary to produce a summary text. This very general model covers many specific variations, with the advantage that

the component processes can be considered independently: for example the source representation need not preempt the condensation step, as is the case with DeJong (1982), and may support different forms of condensation to meet different summarising requirements. Again, the source representation may be very close to, or far from, the input surface text. The synthesis step may be quite simple, or very complex. (Indeed, though we refer to input and output text, this need not imply running text in either case.)

It is, however, helpful to push the general characterisisation of summarising models further, as follows. Thus source representations may be shallow or deep: the text itself with associated sentence weights based on statistical criteria is a very shallow representation, an instantiated script typically a deep one. (This contrast is not necessarily correlated with whether the text processing to obtain the representation is shallow or deep.) The representation may be large- or small-grained in its basic units, for instance templates or words. Further, since summarising is explicitly concerned with the nature of the source text as a whole, it has to capture large scale discourse structure, but this may be linguistic, intentional, attentional or informational structure (Grosz and Sidner (1986), Spärck Jones (1993), Spärck Jones (1995)). Moreover, independent of the substantive structure, the representation itself may take different forms, for instance tree-like or graph-like.

The condensation step can be characterised, very broadly, as selective or generalising, for example according to whether it replicates source material at the same level of detail or abstracts from this. This is is clearly not an absolute distinction but is useful for design. Another broad cut on the summarising stage is whether it is responsive or prescriptive, that is whether what is regarded as important in the source is emergent or is subject to external judgement, as in much work on information extraction (Gaizauskas and Wilks 1998). This particular distinction is important because it is connected with with taking a global or a local approach to the source. The summary representation will reflect these generic choices for the transformation from the source one.

Finally, the output synthesis may be rigid or flexible, the former imposing a preestablished structure on the output, the latter allowing the summary content to organise the output. Of course here, as in the previous steps, systems may in fact take intermediate positions or use combinations. This would apply, for instance, with output having a fixed overall structure, say using standard headings, with a flexible treatment or output text under each.

2.2 Context factors

The factors affecting summarising, taking the initial account in Spärck Jones (1999) further, are listed, with brief examples, in Figure 1. These factors are complex and hard to define, but they have to be recognised for proper system design and evaluation. They fall into three classes: *input*, *purpose*, and *output* factors.

Thus in relation to input, in considering both appropriate methods of source analysis in their own right, e.g. dealing with different types of material, and their implication for later processing steps, it is necessary to take explicit account of form, subject type (as defined by readership) and also the subject matter of the input. Each of these is complex: form subsumes the structure, scale, medium, genre and style of the input; subject type and subject matter each have many possibilities with their own distinctive characteristics, for instance technical type for specialist readers, sporting or botanical subject matter, and

Input factors

Form		
	structure	e.g. with subheadings
	scale	e.g. large (e.g. book)
	medium	e.g. Japanese
	genre	e.g. narrative
	style	e.g. verbose
Subject type		e.g. technical
Subject matter		e.g. horseracing
Unit		e.g. multiple news stories

Purpose factors

Situation	e.g. research laboratory
Audience	e.g. sales managers
Use	e.g. preview
Type	e.g. evaluative
Coverage	e.g. focused

Output factors

Form	e.g. tabulated
Subject type	e.g. non-technical
Subject matter	e.g. science fiction

Figure 1: Summarising factors with examples

so forth. Summarising may also be over multiple sources, not just a single one.

More crucially, the central condensation step refers to the purpose for which summaries are intended, subsuming intended situation of use, audience, and use for the summary, and the summary type and coverage as these relate to the choice of summary content. Again these are complex factors: thus situation refers to environmental matters, which have implications for example for whether whether summaries are for long-term or short term use, to be written or spoken; audience refers to the users and their characteristics, e.g. professionals in a particular field; use to the functional purpose for which the summary is required, e.g. as an executive summary. Other factors characterising the condensation relation between source and summary include summary type, which covers at least such familiar distinctions as informative versus indicative or evaluative; and coverage, i.e. the extent to which a summary is required to be, for example, narrowly focused with respect to the source. As with input factors, while there are natural correlations between purpose factors, they are fundamentally independent, primarily because none can be characterised so tightly as to definitively constrain others. Moreover, while there may be a presumption that (subject to the nature of the input material) the intended use for summaries will guide choices for these 'relational' factors, there are very good heuristic reasons for checking them explicitly.

Finally, there are output factors bearing on the specific properties of the output summary text. These parallel the source ones, but of course need not replicate them, and they have a direct rather than indirect effect on the final summary. The output factors thus include form, for example producing output with headings, medium, style etc; subject type; and subject matter in the sense in which the subject matter of an evaluative summary is different from that of an informative summary. Input and purpose factors naturally guide output choices, but they do not unequivocally determine them, so these choices have to be specifically and explicitly made.

2.3 Design constraints on CLASP

Our work on CLASP was primarily motivated by the wish to explore certain technology possibilities for summarising. These implied that some factor alternatives were in practice, if not in principle excluded, for example producing evaluative summaries. However since this was laboratory work without a directly motivating external purpose requirement, we specified other factors as seemed reasonable and not too taxing for our initial system development and testing.

Thus we adopted the notion of *basic* summary, defined as assuming, or making, relatively undemanding settings of the various environment variables that the factors represent. This does not, however, imply that such basic summaries are appropriate for all summarising contexts.

As our input we used 'ordinary' running text, namely news material, implying non-specialist language in source texts designed for a wide readership and primarily devoted to conveying information on a wide range of topics. The source texts were modest in size, in English, varied in genre but typically straightforward in style. Under purpose we assumed that there were no external requirements referring to specific content categories or intended readers, or well-defined situations and uses for the output summaries. We took the summary readers to be of the same kind as the source text readers, and wanting

fairly brief summaries, for use in a variety of not previously specified situations, that could give their readers an idea of what the source texts are about. In a broad way we assumed a preview style of use. This characterisation implies an idea of *neutrality* and *reflectivity* in the summarisation which, though often believed to be central to summarising is not necessarily the case. This sort of summary is also general-purpose only in a weak sense. Such summaries could in principle be informative in type, but without heavy-duty interpretive apparatus the presumption was that we could only aim at *indicative* summaries. At the same time, since the summaries were to be reflective, the aim was reasonable coverage of all the important content of the source, not selection by *a priori* criteria defining what subject content is important. Finally, under output, the aim was running text for the summary, with style and subject matter the same as that of the source. However the limitations of the source interpretation apparatus led us to consider the modest 'phrase list' form as an alternative output, as this could be sufficient for indicative purposes.

3 CLASP in outline

3.1 Grounding

As noted, CLASP was intended to apply a robust processing strategy, able to withstand the limitations of source text interpretation that using a general-purpose purely linguistic analyser implies, essentially by relying on the redundancy in discourse that simple statistical methods also exploit. However the aim was also to construct an explicitly semantic source representation, as the proper basis for deriving a semantic representation for the summary. These aims precluded strategies explicitly relying on domain content, e.g. DeJong (1982), Hahn (1990), or application-specific genre, e.g. Lehmam (1999), Saggion and Lapalme (2000), Strzalkowski et al. (1999), and also those assuming general text interpretation capabilities far beyond the current state of the art, e.g. van Dijk and Kintsch (1983). They also precluded approaches requiring only simple linguistic processing which either do not construct a semantic representation, e.g. Goldstein et al. (1999), Teufel and Moens (1999), or depend on surface markers of underlying discourse structure that are not necessarily available, e.g. Marcu (1999), Marcu (2000), Moens and Dumortier (2000).

Thus CLASP combines substantive sentence processing with the extraction of a particular form of discourse structure, namely *attentional* structure. Large-scale discourse structure has different forms, intentional (cf. Grosz and Sidner (1986)), informational (Hahn (1990), DeJong (1982)), linguistic (Mann and Thompson (1988)), and attentional (Sidner (1983)), which may all be used as a means of identifying what is important in a source text (Spärck Jones (1993) Spärck Jones (1995)). However they are not equally accessible to relatively limited language processing. CLASP is based on the belief that attentional structure is so important to coherent discourse that it can be identified fairly easily and can be reliably used, even though it may not be fully captured, because it is redundant. Statistical approaches to summarising make the same assumption, very crudely, at the surface text level, though Barzilay and Elhadad (1999), Benbrahim and Ahmad (1994), and Boguraev and Kennedy (1999) make a more sophisticated use of surface lexical cohesion. CLASP seeks to capture attentional structure at a somewhat deeper level, grounded in conventional sentence analysis (though this is still fairly shallow com-

S-JAP1
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 2: Part of a source text, T-JAP

pared with the discourse structure embodied in full-blown script-based approaches like Cullingford (1982)). At the same time, the notion of attentional structure is less local, and also static rather than dynamic with respect to the source discourse as a whole, than Sidner's attentional structure (Sidner 1983). Summarising is then designed to pick out those elements in the source representation that had the most author attention in the source text.

3.2 Structure

The CLASP technology is therefore as follows.

The analyser uses the SRI Core Language Engine (Alshawi et al. 1992), a general-purpose, broad-coverage, robust and powerful processor to deliver a *quasi-logical form* representation for sentences, i.e. predicate-argument structures but ones without contextually-dependent references resolved. These sentence analyses are then processed to extract simple *predications*. The source representation for the input text as a whole is constructed as a *predication cohesion graph*, with individual predications linked through shared predicates or arguments, by applying specific identify and similarity criteria to determine acceptable links. The second condensation stage operates on the predication graph to derive a *weighted graph*, with links between predications that consolidate the information supplied by individual particular links, say between arguments or between predicates, in the initial graph. The weighting is a simple way of capturing 'co-attention' to predications. The weighted graph is then processed, using a greedy algorithm and suitable scoring functions, to identify subgraphs that satisfy three summarising criteria, namely *importance*, *representativeness*, and *cohesiveness*. The output synthesis stage produces text, or 'quasi-text', using the data supplied by the summary graph. In our experiments we found, as described later, that there were problems with generating a single proper text from the graph, so we either generated phrase lists, or used backlinks from the graph predication nodes to their source text origins to extract source sentences, hopefully for a well-motivated selection, though as is well known for such extractive methods, the overall summary may lack presentational coherence.

The next sections describe the three stages of processing in more detail, illustrated by a running example, and we then report on our performance evaluation. For a fuller account of the system processing, see Tucker (1999). The example uses one text, chiefly as the brief excerpt, T-JAP, shown in Figure 2: the full text has 19 long sentences. The excerpt was deliberately chosen to exhibit the different phases and aspects of processing in an economical way. Individual sentences are referred to as S-JAP1, etc.

4 Source text analysis

This has two major stages: (1) *sentence interpretation* and (2) *text representation*. Interpretation in turn covers first, sentence parsing and second, predication extraction for each sentence. Representation covers predication link typing and the formation of the predication cohesion graph for the source text as a whole.

The principles, coverage, and mechanisms used by the SRI Core Language Engine (CLE) have been fully described elsewhere (Alshawi et al. 1992). We used a fairly stable, though not necessarily the most recent, version, and are grateful to SRI Cambridge for making it available. The CLE has a wide coverage grammar and detailed core lexicon, which we backed up with the MRC Lexicon which proves basic dictionary data. The system applies a corpus-based preference mechanism to choose among competing analysis, and if unable to complete a sentence parse, returns a list of well-formed constituents. The parser applies a bottom-up, left-corner strategy using unification over feature values, with semantic interpretation compositionally related to syntactic analysis.

From the point of view of our summarising requirement for a more substantial semantic representation of the source text than the text itself provides, the CLE has the merit of delivering sentence representations that substantiality reduce the expressive variability of the text, though this is limited to syntactic variability and does not, for instance, merge lexical synonyms. The CLE's preference mechanisms and sortal hierarchy significantly reduce source ambiguity, specifically syntactic ambiguity, but sense disambiguation is very limited and without context reference there is no anaphor resolution. We consider CLE performance for the test data later.

4.1 Sentence interpretation

Sentence parsing

As mentioned, CLE sentence parsing delivers a quasi-logical form (QLF) of a generally familiar kind. In a predication of the form `[red_Coloured,I]`, `red` refers to the original text form and `Coloured` is the corresponding word sense label. For words not in the basic lexicon and taken from the MRC lexicon, which does not provide semantic information, the label indicates part of speech and source, as in `NounMRC`. QLFs in the CLE version we used included two types of referential expression, `terms` and `forms`, both requiring context for interpretation; the former is for quantified expressions, the latter for items like nominal compounds, possessives, etc. The QLF for S-JAP1 produced is shown in Figure 3.

Using a full-blown logically-motivated engine, particularly without proceeding to context resolution, may seem overkill. But the CLE had the advantage for our experiments of being a working engine that would deliver the type of primitive semantic unit we wanted, as well as a good deal of other potentially useful information. It is not clear that a more shallow analyser would have given us all we wanted, even if the CLE had the disadvantage of taking deconstruction of the source sentence to a depth that made it quite hard to connect things back together as required to build the predication graph.

```
[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 3: Quasi-logical form for sentence S-JAP1

Predication extraction

The second step of sentence processing traverses the QLF tree extracting simple predications. A simple predication has an atomic predicate and one or more arguments, each of which is either an atomic constant or a variable. Each variable has a semantic head. Thus for example, the simple predications derived from S-JAP1 include

```
in(A, B)
A: investment_NounMRC; B: Southeast Asia
```

and

```
economic_Financial(C)
C: integration_NounMRC
```

where the values given for `A,B,C` are semantic heads.

This processing is not completely straightforward, since it has to deal with incomplete and complex predications. Unresolved components like `forms` have to be given a default resolution, and arguments that are themselves predications unpacked. The unpacking process has to find variable or atomic arguments for predicates, and to replace variables so there is only one final variable per entity. At the same time, because we want simple predications that suppress variation which is confusing detail from a summarising point of view, like number and tense, some QLF information is ignored during extraction.

Extraction involves five sub-steps.

In the first, raw predications are extracted, either from existing predicate-argument lists or by applying rules to `form` substructures. These predications have atomic predicates but may have complex arguments. In the second sub-step, rules are applied to obtain appropriate atomic arguments for different complex argument types. The third sub-step suppresses the proliferation of discourse variables that the CLE initially generated because they might refer to different things. The fourth removes predications reflecting formal logical or syntactic rather than substantive content at the simple predication level, for instance ones with the predicate `and`. These cleaning up operations are followed by a process to identify the *semantic head* for each variable, which is done by finding the semantic head for each predication, and the *head predication* for each variable. These semantic heads are the key hooks when sentence predication sets are combined for the whole text graph. For instance the head predication for a common noun term is the predication corresponding to the head noun, and the semantic head for this predication is its predicate. (In this last analysis operation, some source information for future use in synthesis is also recorded: see below.)

The final output from sentence processing for S-JAP1 is illustrated Figure 4.

4.2 Text representation

The only information available to link sentence representations is that supplied by predicates and arguments that can be treated as 'really', or at least sufficiently, the same. Statistical summarising methods make strong assumptions about identity of lexical meaning. Halliday and Hasan's textual cohesion (Halliday and Hasan 1976) tacitly presupposes some understanding of a text. Without full text interpretation it is impossible to guarantee identity of meanings and referents. It is therefore necessary, while making the assumptions required to link two sentences, to be rather conservative. For our purposes we rely on the general assumption that, since discourse is topic oriented *some* continuity

Simple predications

```
  (S-JAP1)
 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)
```

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
```

Figure 4: Simple predications and semantic heads for sentence S-JAP1

of meaning and reference can be taken for granted, as the use of the same or closely related words indicates, even if we cannot be sure of every specific referent. While this could be problematic for informative summarising, we hoped it would be acceptable for indicative summarising.

Forming the predication graph to represent the entire source text is a single operation, but can be seen as involving two sub-processes, namely typing individual links, and then establishing all the links between the simple predications that are the nodes in the graph.

Link typing

CLASP uses three types of link between predications: *identity*, *similarity*, and *semantic stem links*. Each has several subtypes, as shown in Figure 5. Identity links hold between predications which have the same predicates or one or more of the same arguments. However, while these are the strongest cohesive links, they are too restrictive. In particular, while predicates are simple, we also allow linkage where arguments are not identical but are sufficiently similar. For this we use semantic heads. Thus we allow similarity links between predications whose arguments share semantic heads and between the head predications of such arguments. Finally, as a generalisation over fine-grained lexical variation we use the notion of *semantic stem* so, for instance, distinctions between common and proper nouns, or between word senses, are ignored. The semantic stem of the predicate `economic_Financial` is `economic`, of `United States of America` is `america`. The presumption is that this degree of generalisation is appropriate for discourse-wide indicative summarising. (Our actual link typing rules are somewhat more detailed than as given here: see Tucker (1999).)

Graph formation

With all the individual predication links established, the complete source graph can be formed. The graph obtained for T-JAP, is shown in Figure 6, simplified so that multiple links between nodes are shown only as one link, and the shaded areas represent maximally

Identity links:
argument link
predicate link

Similarity links:
similar argument link
similar head-predication link

Semantic stem links:
stem-similar argument link
stemmed predicate link
stem-similar head-predication link

Figure 5: Types of cohesive link

linked node sets. T-JAP was deliberately chosen to exhibit a highly connected text: in practice source graphs may have disconnected subgraphs and fewer links.

5 Condensation

Deriving the summary graph with the predication content for the summary from the source graph has two aspects: one is to characterise the graph in a way that 'signifies' for summarising, i.e. provides information about the graph that is suited to condensation requirements; and the other is using this information to derive the substructure for the summary graph, i.e. to select a a particular predication node set. The former, *graph characterisation*, applies summarising criteria to nodes and their relations; the latter, *node set selection*, is done by applying a greedy algorithm to the graph. There are many specific parameter settings in both cases, and our experiments have explored only some of the possibilities.

5.1 Graph characterisation

This has two component steps. The first, graph weighting, replaces the set of bottom-level individual links between a pair of predications by a single weighted link. In the second, node sets are scored with respect to three summarising requirements.

Link weighting

CLASP link weighting is based on the view that the various link types defined earlier have a different potential value, as indicators of subgraph properties, for summarising. Thus identity links are regarded as stronger than similarity ones, and these in turn as stronger than stem-similar links. The experiments described later assign weights of 1.0, 0.9 and 0.8 respectively to the three types of link, and where there are multiple links between predications, we tested both strongest individual weight and sum of all weights as the final link value.

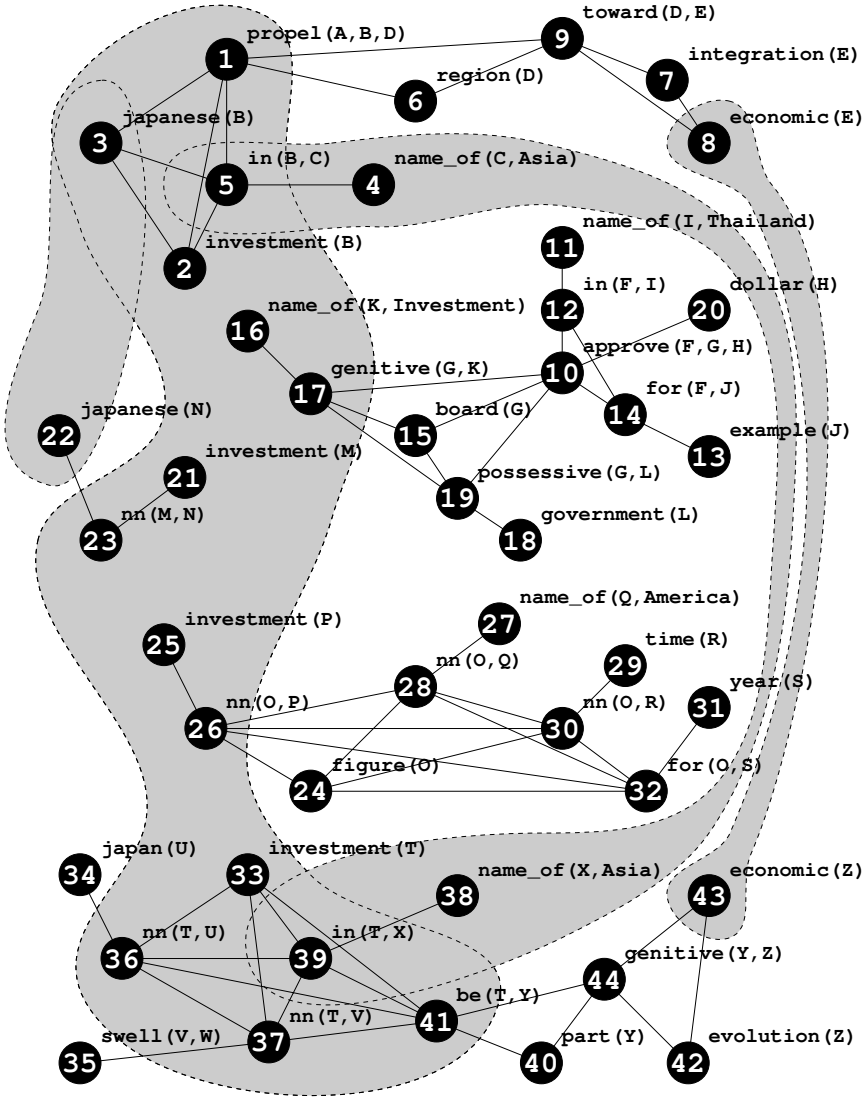

Figure 6: Predication cohesion graph for T-JAP

Node set scoring

This is the core of summarising. The material for a summary is chosen to reflect three criteria for a good summary, namely *importance*, *representativeness* and *cohesiveness*. Importance and representativeness are properties of the summary that refer back to the source and, specifically, are needed for reflective summaries. What is important in the summary should emerge from the source, and the summary should also seek to cover all the important content in the source. Cohesiveness, on the other hand, while it may be promoted by a coherent source, is designed to ensure a clear and readable summary as a discourse in its own right.

In establishing importance, we start by determining it for an individual node. This local importance can be naturally defined as a function of the number and weight of its links. Thus for a graph G and individual node g we define the sum of edge weights at g as

$$\sigma_1(g) \;=\; \Sigma_{h \epsilon G}\, w(h, g)$$

and, for subsets $H \subseteq G$,

$$\sigma_1(H) \;=\; \Sigma_{h \epsilon H}\, \sigma_1(h).$$

Then we define our first measure of importance as

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

This measure is closely related to those used by Skorokhod'ko (1972) and Benbrahim and Ahmad (1994), though in their approaches initial link weights are all 1, and Benbrahim and Ahmad's links are also directed (as they take account of source presentation structure).

We can extend these definitions to take account of the importance of nodes to which a given node is connected. Thus if we define

$$\sigma_2(g) \;=\; \Sigma_{h \epsilon G}\, \sigma_1(h)\, w(h, g)$$

we can then define $\sigma_2(H)$ by analogy with $\sigma_1(H)$ and $\mathbf{imp}_2(H)$ by analogy with $\mathbf{imp}_1(H)$. Indeed we can consider paths of any length n. But it is not necessary to choose just one to measure importance: we can combine them, weighting the combination in favour of shorter paths, by defining

$$\mathbf{imp}(H) \;=\; \Sigma_i\, a_i\, \mathbf{imp}_i(H)$$

where $a_1 \geq a_2 ...$ and $\Sigma_i\, a_i \;=\; 1$. Then finding the best measure is a matter of experiment, for a given type of data and purpose specification.

Representativeness is more complicated. Importance applies to individual nodes, and relative importance for two nodes is determined simply by comparing their scores. Representativeness, on the other hand, is designed to capture the extent to which a subset of nodes is connected with the graph for the text as a whole. We define the *n-neighbourhood* of a set of nodes H, $B^n(H)$, as the set of nodes reachable from H in n or fewer steps. Then, to calculate the representativeness of H, we look at the importance of the nodes connected to H, so we define

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

We can also, by analogy with the treatment of paths for importance, allow for neighbourhoods of different sizes in determining representativeness, so

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

and we then define

$$\mathbf{rep}(H) \;=\; \Sigma_i\, b_i\, \mathbf{rep}_i(H)$$

where $b_1 \geq b_2...$ and $\Sigma_i\, b_i \;=\; 1$.

Figure 7 illustrates the combination of $\mathbf{rep}_1$ with $\mathbf{imp} \;=\; 0.75\mathbf{imp}_1 + 0.25\mathbf{imp}_2$ where, comparing the two node sets ($\mathbf{c\,j\,d}$) and ($\mathbf{c\,g}$), the first set scores more than the second on importance, but the second scores more than the first on representativeness.

Both the importance and representativeness functions are monotonic. There therefore have to be independent grounds for limiting the summary graph size. However maximising the functions subject to a set size restriction helps to limit the summary graph to core material. We are not claiming that these functions are the only or best ones. But we believe that, subject to starting with a graph representation of source text content, that they are reasonable approaches to defining key summarising notions.

As noted, cohesiveness has a somewhat different motivation. It can nevertheless be defined in an analogous way. Thus we measure the extent to which each node g in a (sub)set H is connected to this set, and average over all the set members. Thus by analogy with σ_n we define

$$\gamma_1(g,\, H) \;=\; \Sigma_{h \epsilon H}\, w(h,\, g)$$

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

and

$$\gamma_n(H) \;=\; \Sigma_{h \epsilon H}\, \gamma_n(h, H) \,/\, |H|$$

and then

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

so

$$\mathbf{coh}(H) \;=\; \Sigma_i\, c_i\, \mathbf{coh}_i(H)$$

for $c_1 \geq c_2....$ A high cohesiveness score indicates a well-connected set of predications without superfluous detail. Overall, representativeness and cohesiveness are complementary, and as with representativeness and importance can be adjusted to reflect summarising requirements.

Thus the general formula used in CLASP to select the node set constituting the summary representation by combining information about importance, representativeness and cohesiveness is

$$\mathbf{score}(H) \;=\; A\; \mathbf{imp}(H) \;+\; B\; \mathbf{rep}(H) \;+\; C\; \mathbf{coh}(H)$$

where $A \;+\; B \;+\; C \;=\; 1$ and, just as a_i, b_i and c_i apply within **imp**, **rep** and **coh** respectively, A, B and C can be adjusted for a particular balance between **imp**, **rep** and **coh** in determining the set of nodes selected for the summary representation.

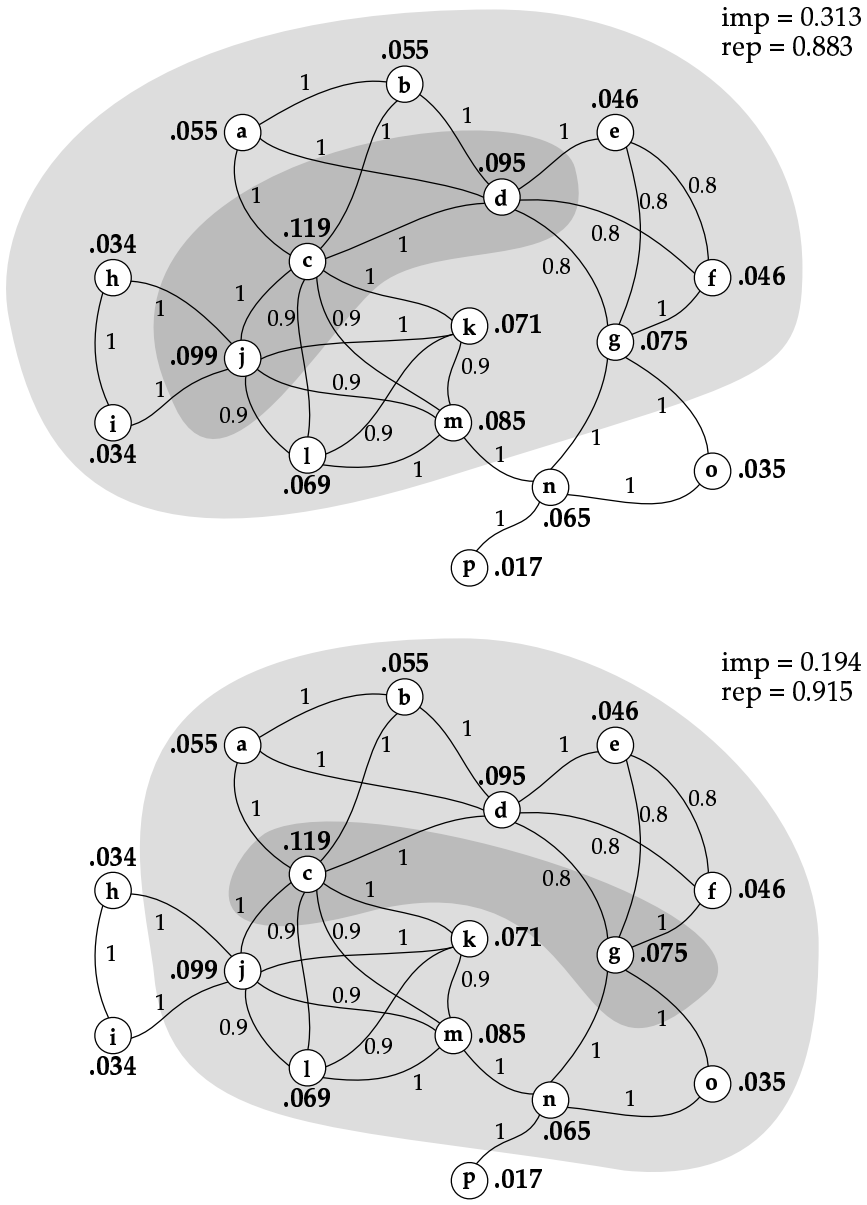

Figure 7: Importance and representativeness scoring: two subsets H, heavy shading; their neighbourhoods $B(H)$, light shading

5.2 Node set selection

This overall function score is applied using a *greedy algorithm* that adds nodes to the existing set, starting from the empty set and maximising the score at each step. An exhaustive search for the best set is clearly impossible. However since the algorithm has no natural stopping point it is constrained in two ways.

One depends on the form of the output summary. If the aim is to produce phrasal output, the algorithm can select any combination of predications satisfying its formal criteria. However if the intention is to produce output consisting of extracted sentences, a sentence has to be either selected or not, implying that the algorithm has to select all or reject all of the predications for a sentence. The other constraint, mentioned earlier, specifies the number of steps to be taken, i.e. for phrasal output the number of predications, or for sentence output the number of sentences. These can be seen as under user control, similar to degrees of compression as in Boguraev and Neff (2000). Both of these parameters lack elegance in detail, but have some rationale. Since the number of steps used is pertinent to later test results, we will refer to it here as *steps*.

6 Synthesis

As just mentioned, CLASP delivers output in two forms: as phrase lists and as extracted sentences.

6.1 Phrase generation

While it should in principle be possible to generate running text from a set of connected predications, we did not explore this. One reason is that it would have required the provision of an output planner to organise the material to be fed to the existing CLE generator. But the main reason was that it rapidly became apparent that the simple predications we were using were too primitive as well as, often, too fragmentary, to supply all the information needed for well-formed sentences. This was not just because the original QLFs were not fully interpreted, but also because the source and summary representations as wholes were not rich enough to support a 'creative' construction of new text with its own sensible provision of internal anaphors, for example. However as Boguraev and Kennedy (1999), Boguraev and Neff (2000), Goldstein et al. (1999), Oka and Ueda (2000), Witbrock and Mittal (1999), and also (Berger and Mittal 2000) suggest, there are task contexts where phrasal summaries can be perfectly adequate or even preferable for time-saving reasons.

The fact that the CLASP source representation could only be indicative of the source text content, rather than informative, would not of itself preclude well-formed output sentences. They could be of some such form: "The source says something about X; and about Y. And it says something about Z in relation to both X and Y." But while technically well-formed, a summary text of this kind would be tedious to read. Our production of output phrases was intended to be a more straightforward and economical, albeit less stylish, alternative. It should be emphasised, however, that by comparison with, e.g. Boguraev and Kennedy (1999), these phrases are generated, not simply extracted from the source.

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 8: Simple predications and summary phrases

Phrase generation has three aspects: choosing a surface form for individual predications, including providing determiners or quantifiers for nominal entities; combining multiple predications into a single phrase, and ordering the output phrase list.

The individual phrases are constructed quite easily. They are all output as simple noun phrases or, for verbal predications, in *-ing* form. The processing constructs a QLF for each predication and uses the source lexical information retained for its semantic heads to determine the appropriate form for each output word, as illustrated in Figure 8. CLASP cannot produce complex output forms, but this was not a practical problem for the test data. We considered different ways of providing appropriate determiners and quantifiers. Simply reproducing the original information in a *source* strategy can lead to dangling anaphors. A *safe* alternative replaces any definite determiner by an indefinite one, but this is not necessarily sensible, for instance in replacing "the time" by "a time". We experimented with a *mixed* strategy, which tried to distinguish exophoric from anaphoric references by looking for semantic head recurrence, but without notable success, and therefore simply applied the source method.

The selected simple predications have many overlaps, of predicates or arguments, so simply listing them makes heavy reading. CLASP therefore *clusters* predications, for integration in a single output phrase. The clustering was designed, however, to avoid promoting inappropriate causal inferences. It therefore starts with a single predication from the selected set and adds predications with argument links to this, i.e. predications which are derived from the same source sentence and so do not have different referents.

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.

Figure 9: Summary phrases for whole text including T-JAP

(The additional predications may not themselves have been selected ones.) In addition, to avoid unwieldy output phrases, clustering is subject to output syntactic constraints, so conjoined adjectival and adverbial phrases are allowed, but not conjoined verbal ones.

Finally, it is necessary to order the set of phrases for output, following a standard header of the form *This text says something about:*. As a preliminary, redundant phrases, i.e. ones with the same output realisation albeit distinct underlying predications, are removed. This is subject to constraints, for example not breaking up different clusters. We explored alternative ways of ordering the final phrase list for output: *selection* order, reflecting cluster formation; *source* order, reflecting the original text order; and *length* order, with the longest phrases first. There is no clearly preferable one, so for our tests we used selection order, which tends to order by cluster size and can be taken as somewhat correlated with relative content importance. Figure 9 illustrates the result for the whole of the text from which the excerpt T-JAP is taken.

6.2 Sentence extraction

As noted, CLASP was intended not to produce extractive summaries. However it could also be seen as providing the means of overcoming some major problems with sentence extraction on purely statistical grounds. Thus working with the source semantic representation rather than just the given text might leverage better sentence extraction. We therefore explored this form of output.

The condensation stage, when parametrised by the requirement for extracted output, selects all the predications for a sentence if it selects any. The corresponding sentences are then simply output, in source order. Ordering in importance order (as established from the graph) does not give more satisfactory output, and in general the source order is less potentially misleading about the content structure of the original. Unfortunately the extracted sentences retain unimportant detail, empty references, etc: we have not explored pruning, aggregation or other smoothing techniques (Jing (2000), McKeown et al. (1995), Mani, Gates and Bloedorn (1999)), or listing the sentences separately to minimise false inferences (Boguraev and Neff 2000).

7 Evaluation

7.1 Approaches to evaluation

It is evident from the earlier discussion of factors affecting summarising that the only proper form of summary evaluation is a task-based one, as illustrated by SUMMAC (Firmin and Chrzanowski (1999), Mani et al. (1998)), and also Brandow et al. (1995), Goldstein et al. (1999), Miike et al. (1994) Oka and Ueda (2000) and Strzalkowski et al. (1999) as well as, in a more rarified form through reading comprehension tests, by Morris, Kasper and Adams (1992). However because our experiments were not done within the *setup* (Spärck Jones and Galliers 1996) of a task context, so we could not attempt this preferable, extrinsic form of system evaluation, and also because we were interested in some more specific questions at the system level, we evaluated CLASP in a more limited way.

The main limited forms of summary evaluation make at most implicit reference to potential contexts of summary use. With *target-based* evaluation the presumption is that some independently-provided, usually human, summaries are appropriate for some task or range of tasks, so automatic summaries are compared with these. Quite apart from difficulties about precisely how such comparisons should be made, there are well-known problems with this approach as a generic one. Thus there is often only a presumption, without independent evidence, that the target summaries are suited to their notional task(s), even at the broad type level. More importantly, there is no reason to suppose that any particular target summary is the best one, in the putative task context, for its particular source document. There are further difficulties when, as with extractive summaries, the targets are not naturally available and have to be constructed specifically to support the evaluation, since this makes the connection with real summary-using tasks weaker. Target-based evaluation, though it has been widely used (cf. many examples in Mani and Maybury (1999)), thus makes many assumptions and needs to be used with caution. It is, however, helpful for initial system performance assessment, and we have used it for this purpose.

The alternative *direct* approach to evaluation, as illustrated by judging summaries for readability, makes its own strong assumptions about potential task utility, but it conveniently overcomes the practical difficulties of implementing target comparison, especially for non-extractive summarising technologies, and also recognises the dangers of using any specific summaries as targets. Direct evaluation has been used by others (e.g. Brandow et al. (1995), Okurowski et al. (2000), Strzalkowski et al. (1999)), and we also used it to evaluate CLASP output.

7.2 CLASP evaluation

Our evaluation was limited in scale for practical reasons (compared for example with Brandow et al. (1995) and DeJong (1982)), so our CLASP assessment was only an initial one. This limitation was partly because the CLASP system was slow, partly because of the need to obtain a range of human reference targets since none were already available.

The CLASP test data consisted of 20 source texts, between 300 and 1500 words long, representing a range of news material taken from the NIST Text REtrieval Conferences

12 news stories:
subject matter: politics, financial, transport, media
genre: mainly descriptive, some narrative
length: 300-900 words

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

2 review articles:
subject matter: novel, film
genre: descriptive, sometimes critical
length: 300-600 words

Figure 10: Wall Street Journal articles used as test data

(TREC) Wall Street Journal files, as shown in Figure 10. We thought it important to have several genres, not just news reports.

We evaluated CLASP phrase summaries by direct evaluation only: since CLASP does predication clustering, it would not have been practicable to attempt a comparison with humanly-selected target phrases. We evaluated CLASP sentence summaries by both target and direct methods. We also compared the two types of summary.

Phrase evaluation

The earlier description of CLASP indicated there are many alternative parameter settings, for example different link weighting and node set scoring schemes. These might in principle give quite different results, though in practice this was not always the case. Figure 11 shows the weighting options and importance scoring alternatives we explored (the scoring names simply reflect computation effort). For the phrase test we decided to compare two particular promising-looking combinations from those shown in the figure, *uniform-simplish* and *uniform-simplish-repstrong*, and to evaluate the output by direct judgements (by the first author) on whether the phrases made sense, and whether they were relevant to the source text. We defined *repstrong* as

$$\mathbf{repstrong}(H) = 0.75\mathbf{imp}(H) + 0.25\mathbf{rep}(H)$$

with $b_1 = 1$, other $b_i = 0$.

We therefore used each of the two strategies to select 20 simple predications from each test text and to produce up to 10 summary phrases from these. Each phrase was judged first as *clear*, i.e. as grammatical and with a clear meaning, or *unclear*. It was also independently judged as *relevant*, i.e. as mentioning salient topic(s) in the text not already covered by another phrase judged relevant, or *not relevant*. The results are shown in Figure 12. Clarity and relevance were strongly correlated. Most of the unclear phrases were attributable to failures in source text analysis rather than to later processing stages. Informal analysis shows that the *uniform-simplish* strategy delivers rather narrowly-focused summaries, the *uniform-simplish-repstrong* strategy rather broader ones. Unfortunately

GRAPH WEIGHTING

unweighted	every edge has weight 1
uniform	every link has weight 1
head	similar and stem-similar head links have weight 1, others weight 2
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 ignored

IMPORTANCE SCORING

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 11: Graph weighings and importance scores for tests

the overall quality of the the output, as illustrated earlier in Figure 9 cannot be described as high. The evaluation did not judge the quality of phrase summaries as wholes. We consider this later, and also the reasons for the poor performance.

Sentence evaluation

For the target-based evaluation of CLASP sentence summaries, we obtained our reference data by asking 5 readers to chose 3 sentences for each test text that contained information they felt should be included in a summary: this should be without regard for whether these sentences would fit together for a smooth overall summary, though in practice it appeared the selected sentences would combine reasonably. Since we wanted to allow for

	Strategy *uniform-simplish*:		*uniform-simplish-repstrong*:	
	165 summary phrases in total		183 phrases	
	relevant	*not relevant*	*relevant*	*not relevant*
clear	51 (31%)	40 (24%)	60 (33%)	32 (17%)
unclear	11 (7%)	63 (38%)	13 (7%)	78 (43%)

Figure 12: Summary phrase categorisation

	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 13: Comparison between CLASP and other sentence extraction methods

different system parameter settings, for summaries of different *steps* lengths, we used the *coverage* measure used by Miike et al. (1994), defined as the proportion of target sentences that are in the automatically selected set. However with 5 different target sets, we could use *average*, *pessimistic* or *optimistic* coverage for the selected set, though whichever we used had to be taken in the context of quite low intra-target set similarity (40% percent).

We produced a range of summaries of different lengths and with different parameter settings for condensation. Overall, there was little difference in the sentence selections per text, suggesting that the overall shape of the source predication graph is the dominant factor, though it appears, with respect to the scoring functions, that local graph structure, i.e. direct predication relations, is more useful than non-local.

As a simple comparison with another extractive method, we ran the BT NetSumm system (Preston and Williams 1994) on our test texts, and also made sentence selections by choosing 3 at *random* and by taking the 3 *initial* sentences per text. The comparison between all of these in Figure 13 shows CLASP, NetSumm and *initial* all better than *random*, but all of them more different from the targets than one target set from another. The data is too small for the differences between systems to be significant. To assess the effects of different summary lengths, particularly bearing in mind Brandow et al's (1995) findings for news material, we compared four choices of number of sentences to be selected for CLASP and *initial* against *random* selection. The performance difference between CLASP and *initial* decreases with the number of sentences, though this may be partly because the source texts are not really long. It should perhaps be noted that, as in the comparison with a simple statistically-based public-domain summariser that Saggion and Lapalme (2000) and others have made with the Microsoft summariser, own-system performance appears somewhat better.

Informal inspection of our summaries suggests that in good ones working with predications rather than surface text has meant that useful, but not lexically-repetitive, material has been included. Bad ones illustrate difficulties like that of dealing with source text with a topic that is indirectly rather than directly expressed, and has a large amount of 'list-like' content.

Our direct evaluation of CLASP sentence summaries was very limited, giving summaries a *relevance score* in the range 0-5. The first author read each source and noted key content, and then read and scored the alternative CLASP summaries for each text, taking them in random order. The results show that while CLASP does not in general do better than *initial*, it is more consistent.

Comparing phrase and sentence summaries

CLASP phrase summaries were intended to be more concise ('efficient') than extracted sentence ones. But their value clearly depended on how accurately the phrase selection captured important content. The relevance judgements for the 5-sentence summaries averaged 3.5 relevant sentences. A 5-phrase summary should contain more relevant phrases, but even 8/9-phrase summaries contained only 3 relevant phrases on average, though the proportion of relevant phrases among clear ones was higher.

The evaluations focused on the elements of the two types of summary and their quality as phrases or sentences, rather than on the summaries as wholes. This would have been hard to do given that the phrase summaries are only presented as lists, and the sentence ones are probably better treated that way. However the method of synthesising the phrase summaries, and particularly the phrase ordering, suggests that the phrases summaries could legitimately be viewed as wholes. This would have the advantage of providing a motivating context for small phrases, if only a loose one: there is less need to motivate whole sentences in this way. We should perhaps have done some simple direct evaluation of whole summaries for completeness. However since the CLASP summary elements in both cases were often unsatisfactory, it did not seem a sensible expenditure of effort.

8 Concluding assessment

The CLASP work was an experiment to explore a new base for summarising, by using a certain type of source representation. The motivation was to achieve a deeper analysis of the source text than simple extractive methods offer, but without relying on any domain or application-specific resources. It would have been interesting to compare the results with, on the one hand, output obtained by other graph-based methods like Skorochod'ko's (1972) or Benbrahim and Ahmad's (1994) and, on the other, with that obtained by extractive methods, especially for phrases, like Boguraev and Kennedy's (1999). But this was impracticable. It would also have been desirable to compare CLASP with other radically different methods, but this was equally impracticable. We can, however, consider similarities and differences between CLASP and other approaches to summarising that exploit salience, as captured by attentional *structure* not mere unit frequency, and that use graph-based source text representations. (This excludes discourse structure defined by rhetorical relations (cf. Marcu (1999), Marcu (2000), and Miike et al. (1994)), which captures salience only in part and indirectly, and is too different from the kinds of graph structure that CLASP builds.)

8.1 Comparable approaches

The lexical chains that Barzilay and Elhadad (1999) used represent a very basic form of attentional structure. They exploited WordNet to capture indirect lexical links, and scored chains over text segments, essentially for prominence and density; they then used selected chains to extract sentences. This is a strongly reflective approach to summarising. The attentional structure that chains represent is much simpler than CLASPs and wholly surface-oriented, where ours is designed to go behind the text surface. Boguraev and Kennedy (1999)'s approach to salience is similar but more sophisticated. They apply

surface parsing both to determine classes of equivalent phrases and to support anaphor resolution. They then generalise from the local salience that this level of analysis defines to discourse salience, to track concept importance through larger segments of the source text. The selected most salient concepts ('topic stamps') are presented with their immediately grounding context, producing an output similar to our phrasal summaries. As Boguraev and Kennedy note, the use of anaphor resolution helps to identify the 'real' discourse entities. However their structural model is still much simpler than the CLASP predication graph, and they use only a single criterion, salience, for summarising, not a combination of criteria like CLASP.

Hahn and Reimer (1999) adopt a much more sophisticated approach to salience, combining textual information supplied by parsing with structural information given by a domain terminological model. Their source text representation essentially instantiates (some of) this, with salience annotation. The salience information is the means of selecting concepts for the output summary. Since Hahn and Reimer's salience criteria cover various types of domain information, and allow for indirect contributions to salience by related concepts, their approach, as they note, covers some graph-like connectivity between concepts. But their approach requires an explicit domain model, parsing is domain oriented, and there is no independent text representation like CLASP's. Thus though they exploit a structured representation as CLASP does, and use salience weights, their form of representation is less flexible than CLASP's and they apply more prescriptive, because domain-guided, summarising criteria.

Skorokhod'ko (1972) was an early proposal for graph representations, in this case with sentences as nodes and links representing shared words. The graph structure would be directly representative of the surface attentional structure of the source text. Skorokhod'ko suggested that in principle summarising could respond to different types of structure as exhibited in the form of the graph. He also suggested using both importance and representativeness criteria to select summary nodes, by considering both the number of links to a node and the degree to which the node held a subgraph together (so removing it would make the graph fall apart). But he does not appear to have tested his ideas.

Benbrahim and Ahmad (1994) 's TELE-PATTAN builds a richer, though still surface, graph representation by using different types of lexical link between sentences (a thesaurus-based version of Hahn and Reimer). They categorise sentences by their cohesion *function*, which is determined not only by their links but also by their presentational 'status', e.g. as opening sentences, so Benbrahim and Ahmad's graph links are directed. This allows them to select sentences using both connectivity and status. It would be possible to make use of status in CLASP, but our approach places more emphasis on semantic structure. Taylor, Krulee and Henschen (1977), on the other hand, propose a deep graphical representation for source texts, using weighted case relationships. This is, like CLASP's, a semantic source representation, but a general one rather than Hahn and Reimer's domain-specific one. Their condensation process was designed to capture both importance and representativeness, and the algorithm they applied (signal flow analysis) allows for indirect as well as direct connections between nodes. It is difficult to judge how well it would work, particularly since the text analysis does not appear to have been automated.

8.2 CLASP reviewed

As these comparisons suggest, the two major fatures of CLASP are quite distinctive. The use of logical forms, though limited to what a general-purpose analyser can deliver, provides a more explicit sentence analysis than conventional shallow parsers and thus allows much stronger linking, through shared variables, than lexical linking alone. The predication graph, as a form of text reprentation, differs from the other types of graph structure mentioned. CLASP was also intended to make different summarising criteria, namely importance, representativeness and cohesiveness, explicit, and to allow them to be independently manipulated. While others have recognised that summarising has different aspects, CLASP addressed this key point in a systematic way, even if with only rather simple models in each case. The tests, especially for phrase summaries, showed clear tradeoffs between importance and representativeness. Cohesiveness did not play a significant role, perhaps primarily because the relatively short source documents and typically short summaries did not call for it. It could be much more critical with longer summaries of longer texts.

The main problems with CLASP as implemented were analysis failures (and speed in the CLE version used); it is also possible that a better general-purpose lexicon would have been useful, especially since some failures were because cohesive links were not identified (e.g. between "film" and "movie" or "japanese" and "Japan". It was also difficult to control summary length, since this was a byproduct of setting a predication selection parameter, or to respond to any graph features that might suggest selecting more or fewer predications. Finally, the predication clustering for phrases was rather crude.

Some obvious ways to try to improve CLASP thus follow. For analysis, one would be to speed up the CLE, by using a fast statistical parser as a preprocessor. (We consider a more radical alternative to the CLE below.) A better dictionary would help, and 'missing' cohesion links might be tackled by making use of a lexical classification like WordNet (Fellbaum 1999). For condensation, it is clearly desirable to explore the scoring functions further, and to introduce more flexibility in predication selection. For synthesis, some of the observed problems with unclear phrases would be removed by better initial source parsing. It could also be the case that, rather than generating from selected predications, better phrases could be obtained by recovering the corresponding source text fragments, to give output like Boguraev and Kennedy's (1999) but with presumed superior grounds for their selection.

The predication graph is the central idea in CLASP. The tests we have described did not show it to be a winner. But they did not show, either, that it is a loser, and it is intuitively very attractive. Thus since some of our results could be attributed to trying to make a parser designed for heavier-duty work, including the provision of sentence representations intended to support contextually-based ambiguity resolution, do a job for which it was not designed, it is natural to ask whether there is a better alternative mode of analysis. Thus a natural line would be to opt for a somewhat shallower analysis phase designed to deliver simple governor-dependent units, which would capture much of what the CLASP use of CLE QLFs was intended to capture, but with less hassle. A similar strategy has been used for document indexing, and fast parsers like the Collins parser (Collins 1999) deliver such outputs. The critical question is whether such an analysis output with, for instance, lexical normalisation and supplementation, could deliver a rich enough semantic graph representation for the source text, i.e. one with the predication

elements and linkage properties that using the CLE allowed. If this appeared to be the case it would, of course, also still be desirable to explore and develop the CLASP approach to condensation, i.e. the use of a predicate-argument graph for summarisation.

References

Alshawi, H. et al. *The Core Language Engine*, Cambridge, MA: MIT Press, 1992.

Aone, C. et al. 'A trainable summariser with knowledge acquired from robust NLP techniques', in *Advances in automatic text summarisation* (Ed. I. Mani and M. Maybury), Cambridge, MA: MIT Press, 1999, 71-80.

Barzilay, R. and Elhadad, M. 'Using lexical chains for text summarisation, in *Advances in automatic text summarisation* (Ed. I. Mani and M. Maybury), Cambridge, MA: MIT Press, 1999, 111-121.

Benbrahim, M. and Ahmad, K. 'Computer-aided lexical cohesion analysis and text abridgement', Report CS-94-11, Computer Sciences Department, University of Surrey, 1994.

Berger, A. and Mittal, V.O. 'OCELOT: a system for summarising web pages', *SIGIR-2000: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2000, 144-151.

Boguraev, B. and Kennedy, C. 'Salience-based content characterisation of text documents', in *Advances in automatic text summarisation* (Ed. I. Mani and M. Maybury), Cambridge, MA: MIT Press, 1999, 99-110.

Boguraev, B.K. and Neff, M.S. 'Lexical cohesion, discourse representation and document summarisation', *RIAO '2000, Content-based multimedia information access*, (Paris), 2000, 962-979.

Brandow, R., Mitze, K. and Rau, L.F. 'Automatic condensation of electronic publications by sentence selection', *Information Processing and Management*, 31 (5), 1995, 675-685.

Collins, M. *Head-driven statistical models for natural language parsing*, PhD Dissertation, University of Pennsylvania, 1999.

Cullingford, R.E. 'SAM', in *Inside Computer Understanding* (Ed. R.C. Schank and C.K. Riesbeck), Hillsdale, NJ: Lawrence Erlbaum, 1981, 75-119.

DeJong, G.F. *Skimming stories in real time: an experiment in integrated understanding*, PhD Thesis, Yale University, 1979; Technical Report 158, Department of Computer Science, Yale University, 1979.

DeJong, G.F. 'An overview of the FRUMP system', in *Strategies for natural language processing* (Ed. Lehnert and Ringle), Hillsdale, NJ: Lawrence Erlbaum, 1982, 149-176.

Donaway, R.L., Drummey, K.W. and Mather, L.A. 'A comparison of rankings produced by summarisation evaluation measures', *Automatic summarisation*, ANLP/NAACL 2000 Workshop, Somerset, NJ: Association for Computational Linguistics, 2000, 69-78.

Fellbaum, C. (Ed.) *WordNet: an electronic lexical database and some of its applications*, Cambridge, MA: MIT Press, 1999.

Firmin, T. and Chrzanowski, M.J. 'An evaluation of automatic text summarisation systems', in *Advances in automatic text summarisation* (Ed. I. Mani and M. Maybury), Cambridge, MA: MIT Press, 1999, 326-339.

Gaizauskas, R. and Wilks, Y. 'Information extraction: beyond document retrieval' *Journal of Documentation*, 54 (1), 1998, 70-105.

Goldstein et al. 'Summarising text documents: sentence selection and evaluation metrics', *SIGIR-99, Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, 121-128.

Grosz, B.J. and Sidner, C.L. 'Attentions, intentions and the structure of discourse', *Computational Linguistics*, 12 (3), 1986, 175-204.

Hahn, U. 'Topic parsing: accounting for text macro structures in full-text analysis', *Information Processing and Management*, 26 (1), 1990, 135-170.

Hahn, U. and Reimer, U. 'Knowledge-based text summarisation: salience and generalisation operators for knowledge base abstraction', in *Advances in automatic text summarisation* (Ed. I. Mani and M. Maybury), Cambridge, MA: MIT Press, 1999, 215-232.

Halliday, M.A.K. and Hasan, R. *Cohesion in English*, London: Longman, 1976.

Hovy, E. and Lin, C.Y. 'Automated text summarisation in SUMMARIST', in *Advances in automatic text summarisation* (Ed. I. Mani and M. Maybury), Cambridge, MA: MIT Press, 1999, 81-94.

Jing, H. 'Sentence reduction for automatic text summarisation', *Proceedings of the Sixth Applied Natural Language Processing Conference and the First Meeting of the North American Chapter of the Association for Computational Linguistics*, 2000, 310-315.

Lehmam, A. 'Text structuration leading to an automatic summarising system', *Information Processing and Management*, 35 (2), 1999, 181-191.

Mani, I. and Maybury, M. (Eds.) *Advances in automatic text summarisation*, Cambridge, MA: MIT Press, 1999.

Mani, I. et al. *TIPSTER text summarisation evaluation conference (SUMMAC). Final Report*, The MITRE Corporation, McClean, VA, 1998.
www.itl.nist.gov/div894/894.02/related_projects/tipster_summac/final_rpt.html

Mani, I., Gates, B. and Bloedorn, E. 'Improving summaries by revising them', *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 1999, 558-565.

Mani, I., Concepcion, K. and van Guilder, L. 'Using summarisation for automatic briefing generation', *Automatic summarisation*, ANLP/NAACL 2000 Workshop, Somerset, NJ: Association for Computational Linguistics, 2000, 89-98.

Mann, W.C. and Thompson, S.A. 'Rhetorical structure theory: towards a functional theory of text organisation', *Text*, 8 (3), 1988, 243-281.

Marcu, D. 'Discourse trees are good indicators of importance in text', in *Advances in automatic text summarisation* (Ed. I. Mani and M. Maybury), Cambridge, MA: MIT Press, 1999, 123-136.

Marcu, D. 'The rhetorical parsing of unrestricted texts: a surface-based approach', *Computational Linguistics*, 26 (3), 2000, 395-448.

Maybury, M.T. 'Generating summaries from event data', *Information Processing and Management*, 31 (5), 1995, 735-751.

Miike, S. et al. 'A full text retrieval system with a dynamic abstract generation function', *SIGIR-94, Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (London: Springer-Verlag), 1994, 152-161.

McKeown, K., Robin, J. and Kukich, K. 'Generating concise natural language summaries', *Information Processing and Management*, 31 (5), 1995, 703-733.

Moens, M.-F. and Dumortier, J. 'Use of a text grammar for generating highlight abstracts of magazine articles', *Journal of Documentation*, 56 (5), 2000, 520-239.

Morris, A.H., Kasper, G. and Adams, D. 'The effects and limitations of automated text condensing on reading comprehension performance', *Information Systems Research*, 3, 1992, 17-35.

Myaeng, S.H. and Jang, D. 'Development and evaluation of a statistically based document summarisation system', in *Advances in automatic text summarisation* (Ed. I. Mani and M. Maybury), Cambridge, MA: MIT Press, 1999, 61-70.

Oka, M. and Ueda, Y. 'Evaluation of phrase-representation summarisation based on information retrieval task', *Automatic summarisation*, ANLP/NAACL 2000 Workshop, Somerset, NJ: Association for Computational Linguistics, 2000, 59-68.

Okurowski, M.E. et al. 'Text summariser in use: lessons learned from real world deployment and evaluation', *Automatic summarisation*, ANLP/NAACL 2000 Workshop, Somerset, NJ: Association for Computational Linguistics, 2000, 49-58.

Piskorski, J. and Neumann, G. 'An intelligent text extractive and navigation system' *RIAO '2000, Content-based multimedia information access*, (Paris), 2000, 1015-1032.

Preston, K. and Williams, S. 'Managing the information overload', in *Physics in business*, London: Institute of Physics, 1994.

Saggion, H. 'Using linguistic knowledge in automatic abstracting' *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 1999, 596-598.

Saggion, H. and Lapalme, G. 'Concept identification and presentation in the context of technical text summarisation', *Automatic summarisation*, ANLP/NAACL 2000 Workshop, Somerset, NJ: Association for Computational Linguistics, 2000, 1-10.

Sidner, C.L. 'Focusing in the comprehension of definite anaphora', in *Computational models of discourse* (Ed. R.C. Berwick and M. Brady), Cambridge, MA: MIT Press, 1983, 267-330.

Skorokhod'ko, E.F. 'Adaptive method of automatic abstracting and indexing', *Information Processing 71* (IFIP Congress 71), Amsterdam: North-Holland, 1972.

Spärck Jones, K. 'What might be in a summary?', *Information Retrieval 93: Von der Modellierung zur Anwendung* (Ed. G. Knorz, J. Krause and C. Womser-Hacker), Konstanz: Universitatsverlag Konstanz), 1993, 9-26.
http://www.cl.cam.ac.uk/public/papers/ksj/ksj-whats-in-a-summary.ps.gz)

Spärck Jones, K. 'Discourse modelling for automatic summarising', Technical Report 290, Computer Laboratory, University of Cambridge, 1993; and in *Travaux du Cercle Linguistique de Prague* (Prague Linguistic Circle Papers), New Series, Volume 1, 1995, Amsterdam: John Benjamins, 201-227.

Spärck Jones, K. 'Automatic summarising: factors and directions', in *Advances in automatic text summarisation* (Ed. I. Mani and M. Maybury), Cambridge, MA: MIT Press, 1999, 1-12.

Spärck Jones, K. and Galliers, J.R. *Evaluating natural language processing systems*, Lecture Notes in Artificial Intelligence 1083, Berlin: Springer, 1996.

Strzalkowski, T. et al. 'A robust practical text summariser', in *Advances in automatic text summarisation* (Ed. I. Mani and M. Maybury), Cambridge, MA: MIT Press, 1999, 137-154.

Taylor, S.L., Krulee, G.K. and Henschen, L.T. 'Automatic abstracting of technical material', *IJCAI-77, Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, 1977, 117-118.

Teufel, S. and Moens, M. 'Argumentative classification of extracted sentences as a first step towards flexible abstracting', in *Advances in automatic text summarisation* (Ed. I. Mani and M. Maybury), Cambridge, MA: MIT Press, 1999, 155-171.

Tucker, R.I. *Automatic summarising and the CLASP system*, PhD Thesis, University of Cambridge, 1999; Technical Report 484, Computer Laboratory, University of Cambridge, 2000.

van Dijk, T.A. and Kintsch, W. *Strategies of discourse comprehension*, New York: Academic Press, 1983.

Witbrock, M.J. and Mittal, V.O. 'Ultra-summarisation: a statistical approach to generating highly condensed non-extractive summaries', *SIGIR-99, Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, 315-316.