Machine Translation **15:** 149–185, 2000.

Nine Issues in Speech Translation

MARK SELIGMAN
GETA, Université Joseph Fourier, 385 rue de la Bibliothèque, 38041 Grenoble Cedex 9, France
(E-mail: markseligman@earthlink.net)

Abstract. This paper sketches research in nine areas related to spoken language translation: interactive disambiguation (two demonstrations of highly interactive, broad-coverage speech translation are reported); system architecture; data structures; the interface between speech recognition and analysis; the use of natural pauses for segmenting utterances; example-based machine translation; dialogue acts; the tracking of lexical co-occurrences; and the resolution of translation mismatches.

Key words: dialogue acts, example-based machine translation, interactive disambiguation, pauses, speech recognition, spoken language translation

1. Introduction

This paper reviews some aspects of the author's research in spoken language translation (SLT) since 1992. Since the purpose is to prompt discussion, the treatment is informal, programmatic, and speculative. There is frequent reference to work in progress – in other words, work for which evaluation is incomplete.

The paper sketches work in nine areas: interactive disambiguation; system architecture; data structures; the interface between speech recognition (SR) and analysis; the use of natural pauses for segmenting utterances; example-based machine translation; dialogue acts; the tracking of lexical co-occurrences; and the resolution of translation mismatches. There is no attempt to provide a balanced survey of the SLT scene. Instead, the hope is to provide a provocative and somewhat personal look at the field by spotlighting it from nine directions – in some respects, to offer an editorial rather than purely a report.

One of the most significant and difficult aspects of the SLT problem is the need to integrate effectively many different sorts of knowledge: phonological, prosodic, morphological, syntactic, semantic, discourse, and domain knowledge should ideally work together to produce the most accurate and helpful translation. Thus a trend toward greater integration of knowledge sources is visible in current SLT research (e.g., in the Verbmobil project, Wahlster, 1993), and most of the work described below is in this integrative direction. Many of the issues to be discussed here could in fact be addressed by dedicated pieces of software playing parts in an integrated SLT system. The paper's conclusion will review the issues by sketching an idealized system of this sort – a kind of personal dream team in which the components are team members.

However, the first topic to be discussed is a renegade, headed in exactly the opposite direction. This is because, while continuing my concern with integration of SLT system components, I have become interested in an alternative system design which, in sharp contrast, stresses a clean separation between SR and translation. The thrust of this alternative "low road" or "quick and dirty" approach is to substitute temporarily intensive user interaction for system integration, thereby attempting a radical design simplification in hopes of fielding practical, broad-coverage systems as soon as possible.

To accommodate this renegade on the one hand and the team players on the other, the paper will be not only somewhat personal, but also two-faced. I will begin by advocating a "low road", non-integrated approach for the near term throughout Section 2. Two demonstrations of highly interactive, broad-coverage SLT will be reported and discussed. Then, performing an about-face, I will go on in the remaining sections to consider elements of a more satisfying integrated approach for the longer term.

2. Interactive Disambiguation

In the present state of the art, several stages of SLT leave ambiguities which current techniques cannot yet resolve correctly and automatically. Such residual ambiguity plagues SR, analysis, transfer, and generation alike.

Since users can generally resolve these ambiguities, it seems reasonable to incorporate facilities for interactive disambiguation into SLT systems, especially those aiming for broad coverage. A good idea of the range of work in this area can be gained from Boitet (1996a).

In fact, Seligman (1997) suggests that, by stressing such interactive disambiguation – for instance, by using highly interactive commercial dictation systems for input, and by adapting existing techniques for interactive disambiguation of text translation (Boitet, 1996b; Blanchon, 1996) – practically usable SLT systems may be constructable in the near term. In such "quick and dirty" or "low road" SLT systems, user interaction is substituted for system integration. For example, the interface between SR and analysis can be supplied entirely by the user, who can correct SR results before passing them to translation components, thus bypassing any attempt at effective communication or feedback between SR and MT.

The argument, however, is not that the "high road" toward integrated and maximally automatic systems should be abandoned. Rather, it is that the low road of forgoing integration and embracing interaction may offer the quickest route to widespread usability, and that experience with real use is vital for progress. Clearly, the high road is the most desirable for the longer term: integration of knowledge sources is a fundamental issue for both cognitive and computer science, and maximally automatic use is intrinsically desirable. The suggestion, then, is that the low and high roads be traveled in tandem; and that even systems aiming for full automaticity recognize the need for interactive resolution when automatic

resolution is insufficient. As progress is made along the high road and increasing knowledge can be applied to automatic ambiguity resolution, interactive resolution should be necessary less often. When it is necessary, its quality should be improved: Questions put to the user should become more sensible and more tightly focused.

2.1. TWO INTERACTIVE DEMOS

These design concepts have been informally and partly tested in two demos, first at the *Machine Translation Summit* in San Diego in October, 1997, and a second time at the meeting of C-STAR II (Consortium for Speech Translation Advanced Research) in Grenoble, France, in January, 1998. Both demos were organized and conducted under the supervision of Mary Flanagan, and both demo systems were based upon a text-based chat translation system previously built by Flanagan's team at CompuServe, Inc. The company's proprietary on-line chat technology was used, as distinct from Internet Relay Chat, or IRC (Pyra, 1995).[1]

In an on-line chat session, users most often converse as a group, though one-on-one conversations are also easy to arrange. Each conversant has a small window used for typing input. Once the input text is finished, the user sends it to the chat server by pressing Return. The text comes back to the sender after an imperceptible interval, and appears in a larger window, prefaced by a header indicating the author. Since this larger window receives input from all parties to the chat conversation, it soon comes to resemble the transcript of a cocktail party, often with several conversations interleaved.

Each party normally sees the "same" transcript window. However, prior to the SLT demos, CompuServe had arranged to place at the chat server a commercial translation system of the direct variety, enabling several translation directions. Once the user of this experimental chat system had selected a direction (say English–French), all lines in the transcript window would appear in the source language (in this case, English), even if some of the contributions originated in the target language (here, French). Bilingual text conversations were thus enabled between English typists and writers of French, German, Spanish, or Italian.

At the time of the demos, total delay from the pressing of Return until the arrival of translated text in the interlocutor's transcript window averaged about six seconds, well within tolerable limits for conversation.[2]

At the author's suggestion and with his consultation, highly interactive SLT demos were created by adding SR front ends and speech-synthesis back ends to CompuServe's text-based chat-translation system. Two laptops were used, one running English input and output software (in addition to the CompuServe client, modified as explained below), and one running the comparable French programs.

Commercial dictation software was employed for SR. For the first demo, both sides used discrete dictation, in which short pauses are required between words; for the second demo, English was dictated continuously – that is, without required pauses – while French continued to be dictated discreetly.[3]

At the time of the demos, the discrete products allowed dictation directly into the chat input buffer, but the continuous products required dictation into their own dedicated window. Thus for continuous English input it became necessary to employ third-party software[4] to create a macro which (a) transferred dictated text to the chat input buffer and (b) inserted a Return as a signal to send the chat.[5]

Commercial speech synthesis programs packaged with the discrete dictation products were used for voice synthesis. Using development software sold separately by the dictation vendor, CompuServe's chat client software was customized so that, as each text string returning from the chat server was written to the transcript window, it was simultaneously sent to the speech synthesis engine to be pronounced in the appropriate language. The text read aloud in this way was either the user's own, transmitted without changes, or the translation of an interlocutor's input.

The first demo took place in an auditorium before a quiet audience of perhaps 100, while the second was presented to numerous small groups in a booth in a noisy room of medium size. Each demo began with ten scripted and pre-tested utterances, and then continued with improvised utterances, sometimes solicited from the audience – perhaps six in the first demo, and 50 or more in the second. Some examples of improvised sentences are given in (1)–(2).

(1) **French:** *Qu'est-ce que vous étudiez?* (What do you study?)
English: Computer science. (*L'informatique.*)

(2) **French:** *Qu'est-ce que vous faites plus tard?* (What are you doing later?)
English: I'm going skiing. (*Je vais faire du ski.*)
French: *Vous n'avez pas besoin de travailler?* (You don't need to work?)
English: I'll take my computer with me. (*Je prendrai mon ordinateur avec moi.*)
French: *Où est-ce que vous mettrez l'ordinateur pendant que vous skiez?* (Where will you put the computer while you ski?)
English: In my pocket. (*Dans ma poche.*)

As these examples suggest, the level of language remained basic, and sentences were purposely kept short, with standard grammar and punctuation.

2.2. DISCUSSION OF DEMOS

A primary purpose of the chat SLT demos was to show that SLT is both feasible and suitable for on-line chat users, at least at the proof-of-concept level.

In my own view, the demos were successful in this respect. The basic feasibility of the approach appears in the fact that most demo utterances were translated

comprehensibly and within tolerable time limits. It is true that the language, while mostly spontaneous, was consciously kept quite basic and standard. It is also true that there were occasional translation errors (discussed below). Nevertheless, the demos can plausibly be claimed to show that chatters making a reasonable effort could successfully socialize in this way. As preliminary evidence that many users could adjust to the system's limitations, we can remark that the dozen or so utterances suggested by the audience, once repeated verbatim by the demonstrators, were successfully recognized, translated, and pronounced in every case.

In addition to the general demo goals just mentioned, the author also had his own, more specific axes to grind from the viewpoint of SLT research. I hoped the demos would be the first to show broad-coverage SLT of usable quality; and I hoped they would highlight the potential usefulness of interactive disambiguation in moving toward practical broad-coverage systems.[6]

I believe that these goals, too, were reached. Coverage was indeed broad by contemporary standards. There was no restriction on conversational topic – no need, for instance, to remain within the area of airline reservations, appointment scheduling, or street directions. As long as the speakers stayed within the dictation and translation lexica (each in the tens of thousands of words), they were free to chat and banter as they liked.

The usefulness of interaction in achieving this breadth was also clear: verbal corrections of dictation results were indeed necessary for perhaps 5% to 10% of the input words. To give only the most annoying example, *Hello* was once initially transcribed as *Hollow*. Here we see with painful clarity the limitations of an approach which substitutes interactive disambiguation for automatic knowledge-based disambiguation: even the most rudimentary discourse knowledge should have allowed the program to judge which word was more likely as a dialogue opener. On the other hand, the approach's capacity to compensate for lack of such knowledge was also clear: a verbal correction was quickly made, using facilities supplied by the dictation vendor.

It should be stressed that the SLT system of the CompuServe demos was by no means the first or only system to permit interactive monitoring of SR output before translation. As far back as the C-STAR I international SLT demonstrations of 1993,[7] selection among SR candidates was essential for most participating systems. Similarly, selection among, or typed correction of, candidates is possible in most of the systems shown in the recent C-STAR II demos of July 22, 1999.[8]

The CompuServe experiments were, however, the first to demonstrate that a broad-coverage SLT system of usable quality – a system capable of extending coverage beyond specialized domains toward unrestricted discourse – could be constructed by enabling users to correct ergonomically the output of a broad-coverage SR component before passing the results to a broad-coverage MT component.

Ergonomic operation was an important element in the system's success. The SR correction facilities used in the experiments – the set of verbal revision commands

supplied by the dictation product, including "scratch that", "correct ⟨word⟩", etc. – were designed for general use in a competitive market, and thus of necessity show considerable attention to ergonomic issues. (By contrast, the SR components of other research systems continue to rely on typed correction or menu selection.) Of course, a smooth human interface between SR and MT cannot by itself yield broad coverage; what it can do is to permit the unexpected combination of SR and MT components developed separately, with broad coverage rather than SLT in mind.

This reliance on interactive correction raises obvious questions: Is the current amount and type of dictation correction tolerable for practical use? Would additional interaction for guiding or correcting translation be useful? Even if potentially useful, would it be tolerated, or would it break the camel's back?

2.2.1. *Correction of Dictation*

The interaction required in the current demos for correcting dictation is just that currently required for correcting text dictation in general. All current dictation products require interactive correction. The question is, do the advantages of dictation over typing nevertheless justify the cost of these products, plus the trouble of acquiring them, training them, and learning to use them? Their steadily increasing user base indicates that many users think so. (For the record, portions of this paper were produced using continuous dictation software.) My own impression is that, during the demos, continuous dictation with spoken corrections supplied correct text at least twice as fast as my own reasonably skilled typing would have done.

For readers who have never tried dictating, a description of the dictation correction process available in Seligman et al. (1998b) may help to realistically estimate the correction burden.

While a strict hands-off policy was adopted for the demos, it is worth noting that typed text and commands can be freely interspersed with spoken text and commands. It is sometimes handy, for instance, to select an error using the mouse, and then verbally apply any of the above-mentioned correction commands. Similarly, when spelling becomes necessary, typing often turns out to be faster than spoken spelling. Thus verbal input becomes one option among several, to be chosen when – as often happens – it offers the easiest or fastest path to the desired text. The question, then, is no longer whether to type or dictate the discourse as a whole, but which mode is most convenient for the input task immediately at hand. As broad-coverage SLT systems in the near term are likely to remain multi-modal rather than exclusively telephonic, they can and should take advantage of this flexibility.

2.2.2. *Correction of Translation*

The current demos were not intended to demonstrate the full range of interactive possibilities. In particular, while *dictation* results were corrected on-line as just discussed, there was no comparable attempt at interactive disambiguation of *trans-*

lation. Thus, when ambiguities occurred, the speaker had no way to control or check the translation results.

For example, when the English partner concluded one dialog by saying (3a), the French partner saw and heard (3b), which might be literally rendered as (3c).

(3)a. It was a pleasure working with you.

b. *C'était un plaisir fonctionner avec vous.*

c. It was a pleasure functioning with you.

The word *work*, in other words, had been translated as *fonctionner*, as would be appropriate for an input like (4).

(4) This program is not working.

Such translation errors were not disruptive during the demos: they were infrequent, and many of the errors which did appear might be more amusing than bothersome in the sort of informal socializing seen in most on-line chat today.

However, errors arising from lexical and structural ambiguities might well be more numerous and more disruptive in future, more sensitive chat-translation applications. Further, it seems doubtful that they can be eliminated in near-term systems aiming for both broad coverage and high quality, even assuming effective use of multiple knowledge sources like those described below. Thus my own guess is that interactive resolution of ambiguities during chat translation would in fact prove valuable. Feedback concerning the translation, via some form of back-translation, would probably prove useful as well. Again, for discussion of possible techniques, see Boitet (1996a) and Blanchon (1996).

But even granting that interactive correction could raise the quality of SLT, would users be willing to supply it? There is some indication that the degree of interaction now required in the demos to correct dictation may already be near the tolerable limit for chat as it is presently used (Flanagan, 1997). A healthy skepticism concerning the practicality of real-time translation correction is thus warranted. I suspect, however, that users' tolerance for interactive correction will turn out to depend on the application and the value of correct translation: to the extent that real-time MT can move beyond socializing into business, emergency, military, or other relatively crucial and sensitive applications, user tolerance for interaction can be expected to increase.

Ultimately, though, questions about the trade-off between the burden of interaction and its worth should be treated as topics for research: using a specified system, what level of quality is required for given applications (specified in terms of tasks to be accomplished within specified time limits), and what types and amounts of interaction are required on average to achieve that quality level? Clearly, until SLT systems with translation correction capabilities are built, no such experiments will be possible.

Having discussed the role of interactive disambiguation in SLT, and having described two experiments with highly interactive SLT, we now turn on our heels as forecast toward more integration-oriented studies. We begin with considerations of SLT system architecture.

3. System Architecture

An ideal architecture for "high road", or highly integrated, SLT systems would allow global coordination of, cooperation between, and feedback among, components (SR, analysis, transfer, etc.), thus moving away from linear or pipeline arrangements. For instance, SR, as it moves through an utterance, should be able to benefit from preliminary analysis results for segments earlier in the utterance. The architecture should also be modular, so that a variety of configurations can be tried: it should be possible, for instance, to exchange competing SR components; and it should be possible to combine components not explicitly intended for work together, even if these are written in different languages or running on different machines.

Blackboard architectures have been proposed (Erman & Lesser, 1990) to permit cooperation among components. In such systems, all participating components read from and write to a central set of data structures, the blackboard. To share this common area, however, the components must all "speak a common (software) language". Modularity thus suffers, since it is difficult to assemble a system from components developed separately. Further, blackboard systems are widely seen as difficult to debug, since control is typically distributed, with each component determining independently when to act and what actions to take.

In order to maintain the cooperative benefits of a blackboard system while enhancing modularity and facilitating central coordination or control of components, Seligman and Boitet (1993) and Boitet and Seligman (1994) proposed and demonstrated a "whiteboard" architecture for SLT. As in the blackboard architecture, a central data structure is maintained which contains selected results of all components. However, the components do not access this whiteboard directly. Instead, only a privileged program called the Coordinator can read from it and write to it. Each component communicates with the Coordinator and the whiteboard via a go-between program called a Manager, which handles messages to and from the Coordinator in a set of mailbox files. Because files are used as data-holding areas in this way, components (and their managers) can be freely distributed across many machines.[9]

Managers are not only mailmen, but interpreters: they translate between the reserved language of the whiteboard and the native languages of the components, which are thus free to differ. In our demo, the whiteboard was maintained in a commercial Lisp-based object-oriented language, while components included independently-developed SR, analysis, and word-lookup components written in C. Overall, the whiteboard architecture can be seen as an adaptation of blackboard

architectures for client-server operations: the Coordinator becomes the main client for several components behaving as servers.

Since the Coordinator surveys the whiteboard, in which are assembled the selected results of all components, all represented in a single software interlingua, it is indeed well situated to provide central or global coordination. However, any degree of distributed control can also be achieved by providing appropriate programs alongside the Coordinator which represent the components from the whiteboard side. That is, to dilute the Coordinator's omnipotence, a number of demi-gods can be created. In one possible partly distributed control structure, the Coordinator would oversee a set of agendas, one or more for each component.

A closely related effort to create a modular "agent-based" (client-server-style) architecture with a central data structure, usable for many sorts of systems including SLT, is described by Julia et al. (1997). Lacking a central board but still aiming in a similar spirit for modularity in various sorts of translation applications is the project described by Zajac and Casper (1997). Further discussion of SLT architecture from the alternative viewpoint of the Verbmobil system appears in Görz et al. (1996). For discussion of a recent DARPA initiative stressing modular switching of components for experimentation, see Aberdeen et al. (1999).

4. Data Structures

We have argued the desirability for system coordination of a central data structure where selected results of various components are assembled. The question remains how that data structure should be arranged. The ideal structure should clarify all of the relevant relationships, in particular clearing up the matter of representational "levels" – a confusing term with several competing interpretations.

Boitet and Seligman (1994) presented several arguments for the use of interrelated lattices for maintaining components' results. Here I present one possible elaboration, suggesting a multi-dimensional set of structures in which three meanings of "level" are kept distinct (Figure 1).

We first distinguish an arbitrary number of "Stages of Translation", with each stage viewable as a long scroll of paper extending across our view from left to right. Left–right is the time dimension, with earlier elements on the left. The Stage 0 scroll represents the raw input to the SLT system, including for example the unprocessed speech input from both speakers and the record of one speaker's mouse clicks on an on-screen map, such as might be used for a direction-finding task. In its full extent from left to right, Stage 0 would thus include the raw input for a translation session once complete, for example, for a dialogue to be translated.

Stage 1 contains the results of the first stage of processing, whatever processes might be involved. This scroll, viewed as unrolling behind Stage 0, might for instance include twin sets of lattices representing the results of phoneme spotting within both speakers' raw input. Stages 2, 3, ..., n unroll in turn behind Stage 1, receding in depth. Stage 2 might include source-language syntactic trees; Stage 3

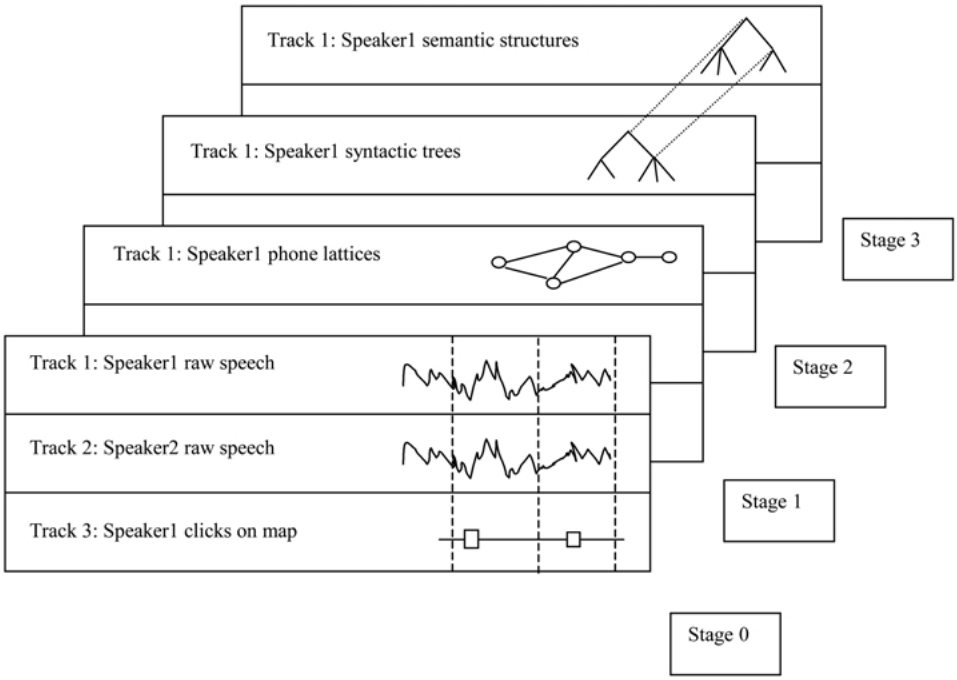

Figure 1. Multi-dimensional data structures for speech translation.

might include semantic structures derived from these trees; and so on, through MT transfer and generation to the final stage, a scroll behind all other scrolls, which might contain translated text annotated for speech synthesis. Pointers (diagonal light lines) would indicate relationships between elements in subsequent stages.

Each stage can be subdivided both vertically and horizontally. Vertical boundaries (vertical dashed lines) represent appropriate time or segment divisions, probably including utterances. Horizontal divisions represent "Tracks", since at each stage several separable signal sources may be under consideration. As already pointed out, Stage 1 in the figure includes two raw speech tracks and a track indicating mouse clicks on a map. Stage 2 might contain, in addition to tracks for the phoneme lattices already mentioned, other tracks (hidden from view) containing F0 curves extracted from the respective speech signals. Different stages may have different numbers of tracks, depending on the processes which define them.

Finally, within each track at a given stage, we can distinguish varying levels of "Height" on the page – that is, various values on the y-axis corresponding to given time values along the x-axis. These can be given various interpretations as appropriate for the type of track in question. When the track contains syntactic trees, height corresponds to syntactic rank, i.e., dominance, with dominant nodes usually covering longer time spans than dominated ones.

Confusion regarding the meaning of "level" bedevils many discussions of MT: it sometimes means a stage of processing, sometimes a mode or type of information,

and sometimes a gradation of dominance or span. The hope is that, by clearly distinguishing these meanings as stages, tracks, or height within tracks, we can help both programmers and programs keep their bearings amid a welter of information.

The multi-dimensional structures just described bear some resemblance to the three-dimensional charts of Barnett et al. (1990), used to track relationships between syntactic and semantic structures during analysis of queries to CYC knowledge bases (Lenat & Guha, 1990). They were developed independently, however. Three-dimensional charts were restricted to two depths or stages (syntactic and semantic), lacked tracks, and made no explicit reference to height or rank.

The whiteboard demo reported in Seligman and Boitet (1993) likewise made only partial use of the multi-dimensional structure: stages and height were explicitly represented and shown in the graphical user interface, with explicit representation of relations between structures in subsequent stages; but tracks were not yet included.

5. Interface between SR and MT Analysis

In a certain sense, SR and analysis for MT are comparable problems. Both require the recognition of the most probable sequences of elements. In SR, sequences of short speech segments must be recognized as phones, and sequences of phones must be recognized as words. In analysis, sequences of words must be recognized as phrases, sentences, and utterances.

Despite this similarity, current SLT systems use quite different techniques for phone, word, and syntactic recognition. Phone recognition is generally handled using Hidden Markov Models (HMMs); word recognition is often handled using Viterbi-style search for the best paths in phone lattices; and sentence recognition is handled through a variety of parsing techniques.

It can be argued that these differences are justified by differences of scale, perplexity, and meaningfulness. On the other hand, they introduce the need for interfaces between processing levels. The processors may thus become black boxes to each other, when seamless connection and easy communication might well be preferable. In particular, word recognition and syntactic analysis (of phrases, sentences, and utterances) should have a lot to say to each other: the probability of a word should depend on its place in the top-down context of surrounding words, just as the probability of a phrase or larger syntactic unit should depend on the bottom-up information of the words which it contains.

To integrate SR and analysis more tightly, it is possible to employ a single grammar for both processes, one whose terminals are phones and whose non-terminals are words, phrases, sentences, etc.[10] This phone-grounded strategy was used to good effect for example in the HMM-LR SR component of the ASURA SLT system (Morimoto et al., 1993), in which an LR parser extended a parse phone by phone and left to right while building a full syntactic tree.[11] The technique worked well for scripted examples. For spontaneous examples, however, performance was un-

satisfactory, because of the gaps, repairs, and other noise common in spontaneous speech. To deal with such structural problems, an island-driven parsing style might well be preferable. An island-based chart parser, like that of Stock et al. (1989), would be a good candidate.

However, chart initialization presents some technical problems. There is no difficulty in computing a lattice from spotted phones, given information regarding the maximum gap and overlap of phones. But it is not trivial to convert that lattice into a "chart" (i.e., a multi-path finite-state automaton) without introducing spurious extra paths. The author has implemented a Common Lisp program which does so correctly, based on an algorithm by C. Boitet (Seligman et al., 1998a). The algorithm tracks, for each node of an automaton under construction, the lattice arcs which it reflects and the lattice nodes at their origins and extremities. An extension of the procedure permits the inclusion of null, or epsilon, arcs in the output automaton. The method has been successfully applied to lattices derived from dictionaries, i.e., very large corpora of strings. (Full source code and pseudocode are available from the author.) Experiments with bottom-up island-driven chart parsing from charts initialized with phones are anticipated.

6. Use of Pauses for Segmentation

It is widely believed that prosody can prove crucial for SR and analysis of spontaneous speech.[12] Several aspects of prosody might be exploited: pitch contours, rhythm, volume modulation, etc. However, Seligman et al. (1997) proposed focusing on natural pauses as an aspect of prosody which is both important and relatively easy to detect automatically.[13]

Given the frequency of utterances in spontaneous speech which are not fully well-formed – which contain repairs, hesitations, and fragments – strategies for dividing and conquering utterances would be quite useful. The suggestion is that natural pauses can play a part in such a strategy: that "pause units", or segments within utterances bounded by natural pauses, can provide chunks which (a) are reliably shorter and less variable in length than entire utterances and (b) are relatively well-behaved internally from the syntactic viewpoint, though analysis of the relationships among them appears more problematic.

Our investigation began with transcriptions of four spontaneous Japanese dialogues concerning a simulated direction-finding task. The dialogues were carried out in the EMMI-ATR Environment for Multi-modal Interaction (Loken-Kim et al., 1993; Furukawa et al., 1993), two using telephone connections only, and two employing on-screen graphics and video as well. In each 3- to 7-minute dialogue, a caller pretending to be at Kyoto station received from a pre-trained "agent" directions to a conference center and/or hotel. In the multimedia setup, both the caller and agent could draw on on-screen maps and exchange typed information.

Morphologically tagged transcripts of the conversations were divided into turns by the transcriber, and included hesitation expressions and other natural speech

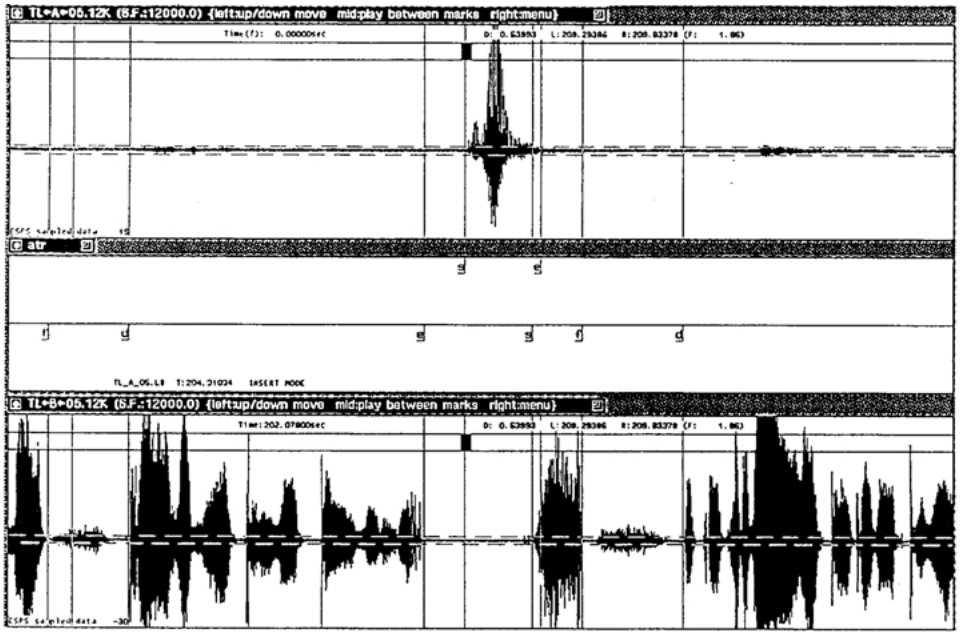

Figure 2. Interface used by the pause tagger.

features. We then added to the transcripts information concerning the placement and length of significant pauses. For our purposes, a significant pause was either a juncture of any length where breathing was clearly indicated (sometimes a bit less than 300 milliseconds) or a silence lasting approximately 400 milliseconds or more.

To facilitate pause tagging, we prepared a customized configuration of the X waves speech display program[14] so that it showed synchronized but separate speech tracks of both parties on screen (Figure 2). The pause tagger, referring to the transcript, could use the mouse to draw labeled lines through the tracks indicating the starts and ends of turns; the starts and ends of segments within turns; and the starts and ends of response syllables which occur during the other speaker's turn. Visual placement of labels was quite clear in most cases. As a secondary job, the tagger inserted a special character into a copy of the transcript text wherever pauses occurred within turns.

After tagging, labels bearing exact timing information were downloaded to separate files. Because there should be a one-to-one mapping between labeled pauses within turns and marked pause locations in the transcript, it was then possible to create augmented transcripts by substituting accurate pause-length information into the transcripts at marked pause points.

In studying the augmented transcripts, four specific questions were addressed:

1. Are pause units reliably shorter than whole utterances? If they were not, they could hardly be useful in simplifying analysis. It was found however, that,

in the corpus investigated, pause units are in fact about 60% the length of entire utterances, on the average, when measured in Japanese morphemes. The average length of pause units was 5.89 morphemes, as compared with 9.39 for whole utterances. Further, pause units are less variable in length than entire utterances: the standard deviation is 5.79 as compared with 12.97.

2. Would hesitations give even shorter, and thus perhaps even more manageable, segments if used as alternate or additional boundaries? The answer seems to be that because hesitations so often coincide with pause boundaries, the segments they mark out are nearly the same as the segments marked by pauses alone. No combination of expressions was found which gave segments as much as one morpheme shorter than pause units on average.
3. Is the syntax within pause units relatively manageable? A manual survey showed that, once hesitation expressions are filtered from them, some 90% of the pause units studied can be parsed using standard Japanese grammars; a variety of special problems appear in the remaining 10%.
4. Is translation of isolated pause units a possibility? We found that a majority of the pause units in four dialogues gave understandable translations into English when translated by hand.

The study provided encouragement for a "divide and conquer" analysis strategy, in which parsing and perhaps translation of pause units is carried out before, or even without, attempts to create coherent analyses of entire utterances.

As mentioned, parsability of spontaneous utterances might be enhanced by filtering hesitation expressions from them in preprocessing. Research on spotting techniques for such expressions would thus seem to be worthwhile. Researchers can exploit a speaker's tendency to lengthen hesitations, and to use them just before or after natural pauses.

Use of pause information for "dividing utterances into meaningful chunks" during SLT of Japanese is described by Takezawa et al. (1999). Pauses are used as segment boundaries in several commercial dictation products, but no descriptions are available.

7. Example-Based SLT

Example-based MT (EBMT) (Nagao, 1984; Sato, 1991) is translation by analogy. An EBMT system translates source-language sentences by reference to an "example base", or set of source-language utterances paired with their target-language equivalents. In developing such a system, the hope is to improve translation quality by reusing correct and idiomatic translations; to partly automate grammar development; and to gain insight into language learning.

Two EBMT systems are now being applied to SLT: the TDMT (Transfer-driven MT) system developed at ATR (Furuse & Iida, 1996; Iida et al., 1996; Sumita & Iida, 1992), used in the ATR-Matrix SLT system (Takezawa et al., 1999); and the PanEBMT system (Brown, 1996) of CMU, used along with transfer-based MT

within the multi-engine MT architecture in the Diplomat SLT system (Frederking et al., 1997).

Despite their common aims, the two systems differ substantially. The ATR system aims to supply a complete translation single-handed, and accordingly includes a full parser for utterances and a hand-built grammar (set of language patterns) to go with it. The CMU system, by contrast, operates as a component of a larger system: in general, its aim is to supply possible partial translations, or translation chunks, to be placed on a chart along with chunks supplied by other translation engines.[15] For this mission, the system requires neither parser nor grammar, relying instead on heuristics to align sub-elements of sentences in the example base at training time. Once it has put its chunks in place during translation, a separate process, belonging to the Multi-Engine MT architecture, will employ a statistical language model to select the best path through the pre-stocked chart in order to assemble the final output.

As the suggestions below relate to a tree-oriented and end-to-end view of example-based processing, the primary concern will be with systems of the ATR type. We begin with a sketch of this methodology.

Consider the Japanese noun phrase (5a). Its literal translation might be (5b), but a more graceful translation would be (5c) or (5d).

(5)a. *kyōto no kaigi*

b. conference of Kyoto

c. conference in Kyoto

d. Kyoto conference

We could hope to provide such improved translations if we had an example base showing for instance that (6a) had been translated as (6b) or (6c), and that (7a) had been rendered as (7b) or (7c).

(6)a. *tōkyō no kaigi*

b. conference in Tokyo

c. Tokyo conference

(7)a. *nyū yōku no kaigi*

c. conference in New York

d. New York conference.

The strategy would be to recognize a close similarity between the new input (5a) and these previously translated noun phrases, based on the semantic similarity

between *kyōto* on one hand and *tōkyō* and *nyū yōku* on the other. The same sort of pattern matching could be performed against a noun phrase in the example base differing from the input at more than one point, for example (8), where *miitingu* ('meeting') is semantically similar to *kaigi* ('conference').

(8). *tōkyō no miitingu*
'meeting in Tokyo'

At any number of such comparison points, semantic similarity of the relevant expressions can be assessed by reference to a semantic hierarchy – for example, a type hierarchy of semantic tags supplied by a thesaurus. A thesaurus associates a lexical item like *kaigi* with one or more semantic tags (e.g., CITY, SOCIAL-EVENT); and the similarity of two semantic tags can be defined as the distance one must rise in the relevant semantic hierarchy to reach a node which dominates both tags: the further, the more semantically distant. The four-level hierarchy of the *Kadokawa Ruigo Shin-jiten* (Ohno & Hamanishi, 1981) has been used in this way in several studies.

Different translations of the Japanese genitive *no* construction, for example as the English possessive (*tanaka-san no kuruma*, 'Tanaka's car') would be distinguished by the distinct semantic types of their respective comparison points – in this case, for example, PERSON and VEHICLE.

By replacing each comparison point in an expression like (5a) with a variable, we can obtain a pattern like (9a). Such patterns can be embedded, giving (9b) or (9c).

(9)a. [?X *no* ?Y]

b. [[?X *no* ?Y] *no* ?Z]

c. [?X *no* [?Y *no* ?Z]]

If we then receive an input like (10) we can determine which bracketing is most sensible – that is, we can parse the input – by extending the techniques already discussed for gauging semantic similarity.

(10) *kyōto no kaigi no ronbun*
KYOTO- gen CONFERENCE- gen PAPER
paper at the conference in Kyoto

One possibility is to designate a "head" for each pattern, and to posit that a pattern's overall semantic type is the type of its head. Then semantic similarity scores can be calculated between an input like (10) and an entire set of embedded patterns – that is, an entire pattern tree – by propagating similarity scores outward (upward). One

can calculate similarity scores for several possible bracketings (trees), and choose the bracketing most semantically similar to the input. In this way, the calculation of semantic similarity guides structural disambiguation during analysis.

Having outlined the essentials of example-based processing in the tree-oriented style, we are now ready to discuss possible elaborations. The first involves the degree of separation between stages of EBMT.

7.1. SEPARATION OF EXAMPLE-BASED ANALYSIS, TRANSFER, AND GENERATION

Recall that semantic similarity calculation can be used to select an embedded set of patterns (a parse tree) from among several competitors. If each source-language pattern (i.e., subtree) is associated with a unique target-language pattern which provides its translation, then the selection of a complete source-language tree will simultaneously and automatically provide a corresponding target-language tree. In this way, an example-based analysis process can be made to provide automatically a *transfer* process as well – that is, a mapping of source-language structures into target-language structures. TDMT intentionally combines analysis and transfer in this way. The combination is seen as an advantage: the same mechanism which handles structural disambiguation simultaneously selects the right translation from among several candidates. However, the combination of phases does raise issues concerning the role of transfer in handling translation ambiguity and structural mismatches.

First, some translation applications may require an explicit account of translation ambiguity – that is, of the possibility of translating a given subtree or node in more than one way. For such applications, transfer might be treated as a separate phase of translation from source-language parsing. That is, since considerations of semantic similarity can guide the selection of target structure – just as they can guide the choice of analysis tree – we can recognize the possibility of example-based *transfer* as separate from example-based *analysis*. Furthermore, depending on the depth of analysis, even once a target-language tree has been selected, ambiguity may arise in selecting target-language surface forms to express it. Thus a separate example-based *generation* phase also becomes a possibility.

A second issue relates to structural mismatches between source and target. Should they be handled in the transfer phase of translation? Consider the translations in (11)–(13), for example.

(11) *Zō wa, hana ga nagai.*
ELEPHANT topic NOSE subj BE-LONG
Elephants have long noses.

(12) *Taeko wa, kaminoke wa nagai*
(NAME) topic HAIR subj BE-LONG
Taeko's hair is long.
Taeko has long hair.

(13) *Watashi wa, taeko ga sukidesu*
ME topic, (NAME) subj IS-LOVED
I like/love Taeko.

In these cases, language-internal considerations dictate non-flat analyses on both the source and target sides. However, in each case, the source tree is differently configured from the target tree. Thus, to represent the correspondences completely, it is insufficient simply to map one source node (one source pattern) onto one target node (one target pattern); rather, we need to intermap arbitrary subtree configurations (embedded pattern sets). In current implementations of tree-oriented EBMT, such general mappings between subtrees are not supported during transfer; rather, they are handled by special-purpose post-processing routines. It might prove easier to arrange a more general treatment for such intermappings if transfer were treated as a separate translation phase.

An experiment reported by Sobashima and Iida (1995) and Sobashima and Seligman (1994) takes a first step toward clear separation of EBMT phases: it presents an example-based treatment of analysis only. (Further information is given below.) A distinct example-based transfer phase including facilities for intermapping embedded patterns was envisaged, but has not yet been implemented.

7.2. MULTIPLE DIMENSIONS OF SIMILARITY

So far we have discussed the measurement of similarity along the semantic scale only. But utterances and structures can be compared along other dimensions as well. Thus for example, when assessing the similarity between a given pattern and the input pattern to be translated, we could ask not only how *semantically* similar its elements are to those of the input pattern, but how *syntactically* similar as well, or how *graphologically* or *phonologically similar*.

Sobashima and Iida (1995) and Sobashima and Seligman (1994) describe facilities for measuring and combining several sorts of similarity. Syntactic similarity, for instance, is measured with reference to a syntactic ontology, comparable to the thesaurus-based semantic hierarchy discussed above; and a score indicating overall similarity of respective variable elements in two patterns is calculated by combining syntactic and semantic similarity scores. The reported implementation also considered, as a factor in overall similarity, a score indicating graphological similarity: 1 for a complete match, and 0 in other cases. Future versions, however, might instead measure phonological similarity – for instance, by means of a phonetype ontology indicating, for example, that [ʃ] and [t ʃ] are similar sounds, while

[ʃ] and [k] are more different. Below, we briefly indicate how multiple similarity dimensions entered into the calculation of overall similarity.

Once we recognize the possibility of considering phonological similarity as a factor in overall similarity between patterns, we move EBMT beyond text translation into the area of SLT. We could, for instance, attempt to disambiguate the speech act of an utterance by comparing the prosodic contours of its elements with the contours of elements of labeled utterances in a database. Such prosodic comparisons might help, for example, to distinguish politely hesitant statements and yes–no questions in Japanese. These utterance types are syntactically marked by final particles *ga* and *ka*, which are phonologically quite difficult to distinguish; their prosodies, however, tend to be quite distinct.

In any case, use of similarity measurements along multiple dimensions as an aid to disambiguation would be very much in the spirit of the "high road", or integrative, approach to SLT discussed throughout.[16]

7.3. BOTH TOP-DOWN AND BOTTOM-UP

In most current example-based systems, the applicability of a pattern is judged by the semantic match of its sub-elements against those of the input. These are *bottom-up* similarity judgments: the sub-elements provide evidence for the presence of the pattern as a whole. Usually absent, however, are corresponding *top-down* similarity judgments whereby the patterns give evidence for the sub-elements. Sobashima and Iida (1995) and Sobashima and Seligman (1994), however, do demonstrate application of both bottom-up and top-down similarity constraints. Further, similarity is measured in both directions along several dimensions (syntactic, semantic, and others), as suggested above. We now briefly describe the method.

First, some necessary background. Consider a "linguistic expression", which may be either atomic or complex. Complex expressions are composed of variables and/or fixed lexical elements, as in (9a) above. We calculate the "elemental similarity", or **E-Sim**, of two expressions as a combined function of their syntactic, semantic, and phonological or graphological similarites.[17]

Now we are ready to consider top-down vs. bottom-up similarity measurement. We calculate the "structural similarity", or "bottom-up similarity", of two complex expressions by combining the elemental similarities of their respective elements. By contrast, the top-down factor in the similarity of two expressions A and B is a measure of the similarity of their respective contexts. We call this factor the "contextual similarity" **C-Sim** of expressions A and B, and calculate it as the sum of the elemental similarities of their respective left and right neighbor expressions L and R (14).

(14) $$\textbf{C-Sim}(A, B) = \textbf{E-Sim}(L_A, L_B) + \textbf{E-Sim}(R_A, R_B)$$

The final, or integrated, similarity score **Sim** for expressions S_1 and S_2, then, is the combination of their structural (bottom-up) similarity and their contextual (top-down) similarity (15).

(15) $\mathbf{Sim}(S_1, S_2) = \mathbf{S\text{-}Sim}(S_1, S_2) * \mathbf{C\text{-}Sim}(S_1, S_2)$

We have seen that **Sim** incorporates multi-dimensional similarity measurements applied both top-down and bottom-up. The next question is how to apply this score for example-based analysis. We now outline the method proposed in the cited papers.

7.4. ANALYSIS WITH MULTI-DIMENSIONAL, TOP-DOWN AND BOTTOM-UP SIMILARITY MEASUREMENTS

We can consider the training stage first. In this stage, an example base is prepared by bracketing and labeling the training corpus by hand. The labeling entry for a complex expression includes the number of elements it contains; the set of syntactic, semantic, and other classifying features of the complex structure as a whole; the classifying features of each sub-element; and the classifying features of the left and right contexts.

Now on to the analysis itself. After morphological processing, with access to a lexicon giving classifying feature sets (perhaps multiple sets) for each terminal, the main routine proceeds as follows:

1. Search the example base for the expression most similar to any contiguous subsequence in the input: find the longest similar matches from position 1 in the input, then from position 2, and so on, terminating if a perfect match is found.
2. Reduce, or rewrite, the covered subsequence, passing its similarity features to the rewritten structure.
3. Go to 1. Continue the cycle until no further reduction is possible.

A preliminary experiment was conducted on 132 English and 129 Japanese sentences. This corpus was too small to permit meaningful statistical evaluation, but we can say that numerous sentences were successfully analyzed which might have yielded massive structural ambiguity, for example (16).

(16) However, we do have single rooms with a shower for eighty dollars and night and twin rooms with a bath for a hundred and forty dollars a night.

Here, many spurious combinations, e.g., *shower for eighty dollars a night and twin rooms*, were ignored in favor of the proper interpretations. Successful analysis of various uses of the article *a* was particularly notable. A full trace appears in the cited papers.

7.5. SIMILARITY VS. FREQUENCY

We have been discussing the uses of similarity calculations for the resolution of various sorts of ambiguity. We conclude this section by contrasting similarity-based disambiguation and *probability*-based disambiguation, an approach which is more widely studied at present. Several current parsers (e.g., Black et al., 1993) are trained to resolve conflicts among competing analyses by using information about the relative frequencies, and thus probabilities, of the combinations of elements in question. At short range, *n*-gram statistics are used; at longer ranges, stochastic rules.

Several of the considerations raised above with respect to similarity-based disambiguation apply equally to probability-based disambiguation. For example, Jurafsky (1993) stresses the need for multidimensional processing: in his parser – based upon the theory of grammatical constructions (Fillmore et al., 1988; Kay, 1990) and claimed to model several features of human parsing as observed in psycholinguistic experiments – semantic as well as syntactic frequencies and probabilities are brought to bear in selecting the proper parse. Also stressed is the need for both top-down and bottom-up statistics in evaluating parse trees as a whole.

Ideally, disambiguation approaches based upon similarity and approaches based upon occurrence probability should complement each other. However, I am aware of no attempts to combine the two.

8. Cue-based Speech Acts

Speech-act analysis (Searle, 1969) – analysis in terms of illocutionary acts like INFORM, WH-QUESTION, REQUEST, and so on – can be useful for SLT in numerous ways. Six uses, three related to translation and three to speech processing, will be mentioned here. Concerning translation, the following tasks must be performed:

1. *Identify the speech acts of the current utterance*. Speech-act analysis of the current utterance is necessary for translation. For instance, the English pattern *can you* (VP, bare infinitive)? may express either an ACTION-REQUEST or a YN-QUESTION (yes-no question). Resolution of this ambiguity will be crucial for translation.
2. *Identify related utterances*. Utterances in dialogues are often closely related: for instance, one utterance may be a prompt and another utterance may be its response; and the proper translation of a response often depends on identification and analysis of its prompt. For example, Japanese *hai* can be translated as *yes* if it is the response to a YN-QUESTION, but as *all right* if it is the response to an ACTION-REQUEST. Further, the syntax of a prompt may become a factor in the final translation. Thus, in a responding utterance *hai, sō desu* (lit. 'yes, that's right'), the segment *sō desu* may be most naturally translated as *he can, you will, she does*, etc., depending on the structure and content of the prompting question. The recognition of such prompt–response relationships will require analysis of typical speech-act sequences.

3. *Analyze relationships among segments and fragments*. Early processing of utterances may yield fragments which must later be assembled to form the global interpretation for an utterance. Speech-act sequence analysis should help fit fragments together, since we hope to learn about typical act groupings.

Concerning speech processing, it is necessary to do the following:

4. *Predict speech acts to aid SR*. If we can predict the coming speech acts, we can partly predict their surface patterns. This prediction can be used to constrain SR. As already mentioned, for instance, Japanese utterances ending in *ka* and *ga* – respectively, YN-QUESTIONs and INFORMs – are difficult to distinguish phonologically. We earlier considered the use of prosodic information in resolving this uncertainty. Predictions as to the relative likelihood of these speech acts in a given context should further aid recognition.
5. *Provide conventions for prosody recognition*. Once spontaneous data is labeled, SR researchers can try to recognize prosodic cues to aid in speech-act recognition and disambiguation. For instance, they can try to distinguish segments expressing INFORMs and YN-QUESTIONs according to the F0 curves associated with them – a distinction which would be especially useful for recognizing YN-QUESTIONs with no morphological or syntactic markings.
6. *Provide conventions for speech synthesis*. Similarly, speech synthesis researchers can try to provide more natural prosody by exploiting speech-act information. Once relations between prosody and speech acts have been extracted from corpora labeled with speech-act information, researchers can attempt to supply natural prosody for synthesized utterances according to the specified speech acts. For instance, more natural pronunciations can be attempted for YN-QUESTIONs, or for CONFIRMATION-QUESTIONs (including tag questions in English, as in

(17) The train goes east, doesn't it?

While a well-founded set of speech-act labels would be useful, it has not been clear what the theoretical foundation should be. As a result, no speech-act set has yet become standard, despite considerable recent effort.[18] Labels are still proposed intuitively, or by trial and error.

Speakers' goals can certainly be analyzed in many ways. However, Seligman et al. (1995) hypothesize that only a limited set of goals is conventionally expressed in a given language. For just these goals, relatively fixed expressive patterns are learned by speakers when they learn the language. In English, for instance, it is conventional to express certain suggestions or invitations using the patterns *Let's* V or *Shall we* V*?* In Japanese, one conventionally expresses similar goals via the patterns V[combining stem]*mashō* or V[combining stem]*masen ka?*

The proposal is to focus on discovery and exploitation of these conventionally expressible speech acts, or "Cue-based Speech Acts" (CBSAs).[19] The relevant expressive patterns and the contexts within which they are found have the great virtue of being objectively observable; and assuming the use of these patterns is

common to all native speakers, it should be possible to reach a consensus classification of the patterns according to their contextualized meaning and use. This functional classification should yield a set of language-specific speech-act labels which can help to put speech-act analysis for SLT on a firmer foundation.

The first reason to analyze speech acts in terms of observable linguistic patterns, then, is the measure of objectivity thus gained: the discovery process is to some degree empirical, data-driven, or corpus-based. A second reason is that automated cue-based analysis, being shallow or surface-bound, should be relatively quick as opposed to plan-based analysis. Plan-based analysis may well prove necessary for certain purposes, but it is quite expensive. For applications like SLT which must be carried out in nearly real time, it seems wise to exploit shallow analysis as far as possible.

With these advantages of cue-based processing – empirical grounding and speed – come certain limitations. When analyzing in terms of CBSAs, we cannot expect to recognize all communicative goals. Instead, we restrict our attention to communicative goals which can be expressed using conventional linguistic cue patterns. Communicative goals which cannot be described as CBSAs include utterance goals which are expressed non-conventionally (compare the non-conventional warning (17a) to the conventional WARNING (17b); or goals which are expressed only implicitly ((18) as an implicit request to shut the window); or goals which can only be defined in terms of relations between utterances.[20]

(17)a. May I call your attention to a potentially dangerous dog?

b. Look out for the dog!

(18) It's cold outside.

Given that the aim is to classify expressive patterns according to their meaning and function, how should this be done? Seligman (1991) and Seligman et al. (1995) describe a paraphrase-based approach: native speakers are polled as to the essential equivalence of expressive patterns in specified discourse contexts. If by consensus several patterns can yield paraphrases which are judged equivalent in context, and if the resulting pattern set is not identical to any competing pattern set, then it can be considered to define a CBSA. (Knott and Dale (1995) and Knott (1996) describe a similar *substitution*-based approach to the discovery of discourse relations, as opposed to speech acts.)

CBSAs are defined in terms of monolingual conventions for expressing certain communicative goals using certain cue patterns. For translation purposes, however, it will be necessary to compare the conventions in one language with those in the other. With this goal in mind, the discovery procedure was applied to twin corpora of Japanese–Japanese and English–English spontaneous dialogues concerning transportation directions and hotel accommodation (Loken-Kim et al., 1993). CBSAs were first identified according to monolingual criteria. Then, by observing

translation relations among the English and Japanese cue patterns, the resulting English and Japanese CBSAs were compared. Interestingly, it was found that most of the proposed CBSAs seem valid for both English and Japanese: only two out of 27 seem to be monolingual for the corpus in question.

We have been outlining a cue-based approach to recognition of speech or discourse acts, with the assumption that some sort of parsing would be employed to recognize cue patterns. This methodology can be compared with statistical recognition approaches: speech- or discourse-act labels are posited in advance, and statistical models are subsequently built which attempt to identify the acts according to their sequence (Reithinger, 1995; Nagata & Morimoto, 1993) or according to the words they contain (Alexandersson et al., 1997; Reithinger & Klesen, 1997).

Certain speech-act sequences may indeed turn out to be typical; and certain words may indeed prove to be unusually common in, and thus symptomatic of, arbitrarily defined speech acts. Thus statistical techniques are indeed likely to be helpful for recognition of conventional speech acts when they are implied or expressed non-conventionally, or for recognition of speech acts which are not conventional but nevertheless appear useful for some applications. Further, even for conventional speech acts which are conventionally expressed, efficiency considerations may sometimes favor statistical recognition techniques over pattern recognition: once CBSAs were identified using our methods and a sufficiently large training corpus had been hand-labeled, statistical models might certainly be built to permit efficient identification in context. However, statistical recognition approaches alone cannot provide a principled way to *discover* (that is, posit or hypothesize) the labels in the first place, and this is what we seek. The current CBSA set has been applied in three studies: Black (1997) attempted to associate speech acts, including CBSAs, with intonation contours in hopes of improving speech synthesis; Iwadera et al. (1995) employed the CBSA set in attempts to parse discourse structure; and Jokinen et al. (1998) used CBSAs in topic-tracking experiments.

9. Tracking Lexical Co-occurrences

In the processing of spontaneous language, the need for predictions at the morphological or lexical level is clear. For bottom-up parsing based on phones or syllables, the number of lexical candidates is explosive. It is crucial to predict which morphological or lexical items are likely so that candidates can be weighted appropriately. (Compare such lexical prediction with the predictions from CBSAs discussed above. In general, it is hoped that by predicting CBSAs we can in turn predict the structural elements of their cue patterns. We are now shifting the discussion to the prediction of open-class elements instead. The hope is that the two sorts of prediction will prove complementary.)

N-grams provide such predictions only at very short ranges. To support bottom-up parsing of noisy material containing gaps and fragments, longer-range

predictions are needed as well. Some researchers have proposed investigation of associations beyond the n-gram range, but the proposed associations remain relatively short-range (about five words). While stochastic grammars can provide somewhat longer-range predictions than n-grams, they predict only within utterances. Our interest, however, extends to predictions on the scale of several utterances.

Thus Seligman (1994a) and Seligman et al. (1999) propose to permit the definition of windows in a transcribed corpus within which co-occurrences of morphological or lexical elements can be examined. A flexible set of facilities (CO-OC) has been implemented in Common Lisp to aid collection of such discourse-range co-occurrence information and to provide quick access to the statistics for on-line use.

A window is defined as a sequence of minimal segments, where a segment is typically a turn, but can also be a block delimited by suitable markers in the transcript.

Sparse data is somewhat less problematic for long-range than for short-range predictions, since it is in general easier to predict what is coming "soon" than what is coming next. Even so, there is never quite enough data; so smoothing will remain important. CO-OC can support various statistical smoothing measures. However, since these techniques are likely to remain insufficient, a new technique for semantic smoothing is proposed and supported: researchers can track co-occurrences of semantic tokens associated with words or morphs in addition to co-occurrences of the words or morphs themselves. The semantic tokens are obtained from standard on-line thesauri. The benefits of such semantic smoothing appear especially in the possibility of retrieving reasonable semantically-mediated associations for morphs which are rare or absent in a training corpus.

Sections 9.1–9.3 describe CO-OC's operations in somewhat greater detail. Section 9.4 sketches possible applications for the co-occurrence information harvested by the program.

9.1. WINDOWS AND CONDITIONAL PROBABILITIES

As mentioned, we first permit the investigator to define minimal segments within the corpus: these may be utterances, sections bounded by pauses or significant morphemes such as conjunctions, hesitations, postpositions, and so on. Windows composed of several successive minimal segments can then be recognized: Let S_i be the current segment and N be the number of additional segments in the window as it extends to the right. N=2 would, for instance, give a window three segments long with S_i as its first segment. Then if a given word or morpheme M_1 occurs (at least once) in the initial segment, S_i, we attempt to predict the other words or morphemes which will co-occur (at least once) anywhere in the window. Specifically, a conditional probability Q can be defined as in (20),

$$(20) \qquad Q(M_1, M_2) = P(M_2 \in S_i \cup S_{i+1} \cup S_{i+2} \cup \cdots \cup S_{i+N} | M_1 \in S_i)$$

where M_j are morphemes, S_j are minimal segments, and N is the width of window in segments. Q is thus the conditional probability that M_2 is an element of the union of segments S_i, S_{i+1}, S_{i+2}, and so on up to S_{i+N}, given that M_1 is an element of S_i. Both the segment definition and the number of segments in a window can be adjusted to vary the range over which co-occurrence predictions are attempted.

For initial experiments, we used a morphologically tagged corpus of 16 spontaneous Japanese dialogues concerning direction-finding and hotel arrangements (Loken-Kim et al., 1993). We collected common-noun/common-noun, common-noun/verb, verb/common-noun, and verb/verb conditional probabilities in a three-segment window (N=2). Conditional probability Q was computed among all morph pairs for these classes and stored in a database; pairs scoring below a threshold (0.1 for the initial experiments) were discarded. We also computed and stored the mutual information for each morph pair, using the standard definition as in Fano (1961).

Fast queries of the database are then enabled. A central function is GET-MORPH-WINDOW-MATES, which provides all the window mates for a specified morph which belong to a specified class and have scores above a specified threshold for the specified co-occurrence measure (conditional probability or mutual information).

The intent is to use such queries in real time to support bottom-up, island-driven SR and analysis. To support the establishment of island centers for such parsing, we also collect information on each corpus morph in isolation: its hit count and the segments it appears in, its unigram probability and probability of appearance in a given segment, etc. Once island hypotheses have been established based on this foundation, co-occurrence predictions will come into play for island extension.

9.2. SEMANTIC SMOOTHING

As mentioned, CO-OC supports the use of standard statistical techniques (Nadas, 1985) for smoothing both conditional probability and mutual information. In addition, however, we enable semantic smoothing in an innovative way. Thesaurus categories – "cats" for short – are sought for each corpus morph (and stored in a corpus-specific customized thesaurus for fast access). The common noun *eki* ('station'), for instance, has among others the cat label "725a" (representing a semantic class of POSTS-OR-STATIONS in the standard *Kadokawa* Japanese thesaurus (Ohno & Hamanishi, 1981).

Equipped with such information, we can study the co-occurrence within windows of cats as well as morphs. For example, using N=2, GET-CAT-WINDOW-MATES finds 36 cats co-occurring with "725a", one of the cats associated with *eki*, with a conditional probability $Q > 0.10$, including "459a" (*sewa* 'taking care of' or 'looking after'), "216a" (*henkō* 'transfer'), and "315b" (*ori* 'getting off'). Since we have prepared an indexed reverse thesaurus for our corpus, we can quickly find the corpus morphs which have these cat labels, respectively *miru* 'look', *mieru* 'can see/be visible', *magaru* 'turn', and *oriru* 'get off'. The resulting morphs are related

to the input morph *eki* via semantic rather than morph-specific co-occurrence. They thus form a broader, smoothed group.

This semantic-smoothing procedure – morph to related cats, cats to co-occurring category window-mates, cats to related morphs – has been encapsulated in the function GET-MORPH-WINDOW-MATES-VIA-CATS. It permits filtering, so that morphs are output only if they belong to a desired morphological class and are mediated by cats whose co-occurrence likelihood is above a specified threshold.

Thesaurus categories are often arranged in a type hierarchy. In the *Kadokawa* thesaurus, there are four levels of specificity: "725a" (POSTS-OR-STATIONS), mentioned above, belongs to a more general category "725" (STATIONS-AND-HARBORS), which in turn belongs to "72" (INSTITUTIONS), which belongs to "7" (SOCIETY). Accordingly, we need not restrict co-occurrence investigation to cats at the level given by the thesaurus. Instead, knowing that "725a" occurred in a segment S_i, we can infer that all of its ancestor cats occurred there as well; and we can seek and record semantic co-occurrences at every level of specificity. This has been done; and GET-MORPH-WINDOW-MATES-VIA-CATS has a parameter permitting specification of the desired level of semantic smoothing. The more abstract the level of smoothing, the broader the resulting group of semantically-mediated morpheme co-occurrences. The most desirable level for semantic smoothing is a matter for future experimentation.

9.3. EVALUATION

We are presently reporting the implementation of facilities intended to enable many experiments concerning morphological and morpho-semantic co-occurrence; the experiments themselves remain for the future. Clearly, further testing is necessary to demonstrate the reliability and usefulness of the approach. A principle aim would be to determine how large the corpus must be before consistent co-occurrence predictions are obtained. Nevertheless, some indication of the basic usability of the data is in order.

Tools have been provided for comparing two corpora with respect to any of the fields in the records relating to morphs, morph co-occurrences, cats, or cat co-occurrences. Using these, we treated 15 of our dialogues as a training corpus, and the one remaining dialogue as a test corpus. We compared the two corpora in terms of conditional probabilities for morph co-occurrences. In both cases, statistically unsmoothed scores were used for simplicity of interpretation.

We found 5,162 co-occurrence pairs above a conditional probability threshold of 0.10 in the training corpus and 1,552 in the test. Since 509 pairs occurred in both corpora, the training corpus covered 509 out of 1,552, or 33%, of the test corpus. That is, one third of the morph co-occurrences with conditional probabilities above 0.10 in the test corpus were anticipated by the training corpus.

This coverage seems respectable, considering that the training corpus was small and that neither statistical nor semantic smoothing was used. More important

than coverage, however, is the presence of numerous pairs for which good co-occurrence predictions were obtained. Such predictions differ from those made using *n*-grams in that they need not be chained, and thus need not cover the input to be useful: if consistently good co-occurrence predictions can be recognized, they can be exploited selectively.

The figures obtained for cats and cat co-occurrences are comparable.

9.4. POSSIBLE APPLICATIONS

A weighted co-occurrence between morphemes or lexemes can be viewed as an association between these items; so the set of co-occurrences which CO-OC discovers can be viewed as an associative or semantic network. Spreading activation within such networks is often proposed as a method of lexical disambiguation.[21] Thus disambiguation becomes a second possible application of CO-OC's results, beyond the abovementioned primary use for constraining SR.[22]

A third possible use is in the discovery of topic transitions: we can hypothesize that a span within a dialogue where few co-occurrence predictions are fulfilled is a topic boundary.[23] Once the new topic is determined, appropriate constraints can be exploited, for example by selecting a relevant subgrammar.

10. Translation Mismatches

During translation, when the source and target expressions contain differing amounts of information, a "translation mismatch" is said to occur. For example, the English sentence (21a) may be translated by Japanese (21b). In this case, because the explicit pronoun is suppressed, information concerning person and number is lost. Similarly, (22a) may be translated as (22b). Here, the pronoun is once again suppressed, and information about the object of the verb is lost as well: Japanese does not express either its number or its definiteness.

(21)a. He ate.

b. *Tabemashita.*

EAT-past

(22)a. He bought the books.

b. *Hon o kaimashita.*
BOOK obj BUY-past

Suppressing such information during translation is less difficult than arranging for its addition when translating in the opposite direction. When translating from Japanese to English, for instance, how is a program to determine whether an entity

is definite, or plural, or third-person? Of course, such problems are not unique to SLT – they are equally present in text translation. In spoken translation, though, there is the added difficulty of resolving them in real time.

The first observation we can make about mismatch resolution is that it is in some respects akin to ambiguity resolution. In both cases, information is missing which must be supplied somehow: in translation mismatches, missing information must be filled in; in ambiguity resolution, missing information must guide a choice. In light of this similarity, the interactive resolution techniques suggested above for ambiguity resolution can be suggested for mismatches as well. For example, it would be relatively straightforward to put up a menu offering a choice between singular and plural – or "one vs. many", etc. Granted, other sorts of information, for example concerning definiteness, would be trickier to elicit in non-technical terms.[24] Of course, handling many such requests would be tedious, so interface design would be crucial. And again, the hope is that the need for interaction will shrink as knowledge-source integration advances.

A second observation about mismatch resolution is that, when missing information cannot be accurately computed and is excessively burdensome for users to supply, it can simply be left missing. For example, for translating (21b) when the correct English would be (21a), the incomplete translation (23) could be produced. This broken English would at least allow the hearer to infer the correct meaning from context, more or less as a hearer of the original Japanese (or a hearer of "real" broken English) would have to do. Further, supplying insufficient information is usually better than supplying incorrect information: in the same situation, (23) would be far less confusing than, (24), say, which is wrong on several counts. Thus far, however, I am aware of no SLT programs which deliberately abstain when in doubt.

(23) * bought book

(24) I bought a book.

Ideally, however, translation software will do its best to resolve mismatches before requesting help from the user or throwing in the towel. Researchers in this area have tended to create programs focusing on a specific sort of mismatch. For example, Murata and Nagao (1993) proposed an expert system for supplying number and definiteness information, and thus articles, during Japanese–English translation.

In a similar spirit, Seligman (1994b) describes a program for resolving the references of zero pronouns in the Asura SLT system (Morimoto et al., 1993), thus supplying the missing pronouns for translation. The program, based upon the theory of centering (Sidner, 1979; Grosz et al., 1983; Joshi & Weinstein, 1981; Takeda & Doi, 1994), follows unpublished work by Masaaki Nagata. It is invoked from within specially modified transfer rules for verbs, and can work alongside

other pronoun-resolution techniques, for example those making use of Japanese honorific information (Dohsaka, 1990). No evaluations have yet been made.

Other mismatch problems to be addressed are surveyed in Seligman et al. (1993) in the context of Japanese–English or Japanese–German transfer. These include the determination of tense (Japanese, for example, does not have an explicit future tense); aspect (Japanese lacks explicit cues which would license a choice between (25a, b)); intimacy (as required for a choice between German *du* and *Sie* – Japanese does supply a great deal of information concerning politeness, formality, relative status, etc., but none of these map cleanly into the German distinction); choice of possessive determiners (Japanese often uses only *namae*, 'name', where English would have *your name*); and several other sorts of mismatch.

(25)a. He has been studying.

b. He is studying.

11. Conclusions

The first section of the paper described a "low road" or "quick and dirty" approach to SLT, in which interactive disambiguation of SR and translation is temporarily substituted for system integration. This approach, I believe, is likely to yield broad-coverage systems with usable quality sooner than approaches which aim for maximally automatic operation based upon tight integration of knowledge sources and components.

Two demonstrations of "quick and dirty" SLT over the Internet were reported. For the demos, an experimental chat translation system created by CompuServe, Inc. was provided with front and back ends, using commercial dictation products for speech input and commercial speech-synthesis engines for speech output. The dictation products' standard interfaces were used to debug dictation results interactively. While evaluation of these experiments remained informal, coverage was much broader than in most SLT experiments to date – in the tens of thousands of words. While interactive control of translation was lacking, output quality was probably sufficient for many social exchanges.

But while the "low road" may offer the fastest route to usable broad-coverage SLT systems, automatic operation based upon knowledge-source integration is certain to remain desirable in the longer run. Hence the balance of the paper has concentrated on aspects of integrated systems.

Taken together, the nine areas of research examined in the paper suggest a nine-item wish list for an experimental SLT system.

1. The system would include facilities for interactive disambiguation of both speech and translation candidates.
2. Its architecture would allow modular reconfiguration and global coordination of components.

3. It would employ a perspicuous set of data structures for tracking information from multiple processes: stages of translation, multiple tracks, and height, span, or dominance of nodes would be clearly distinguished.
4. The system would employ a grammar whose terminals were phones, recognizing both words and syntactic structures in a uniform and integrated manner, e.g., via island-driven chart parsing.
5. Natural pauses and other aspects of prosody would be used to segment utterances and otherwise aid analysis.
6. Similarity-based techniques for resolving ambiguities, comparable to those of EBMT, would be effectively used. Stages of translation yielding potential ambiguities would be kept distinct; similarity would be measured along several dimensions (e.g., syntactic and phonological in addition to semantic); top-down as well as bottom-up constraints would be exercised; and disambiguation using both probability-based and similarity-based techniques would be used in complementary fashion.
7. Speech or dialogue acts would be defined in terms of their cue patterns, and analyses based upon them would be exploited for SR and analysis.
8. Semantically smoothed tracking of lexical co-occurrences would provide a network of associations useful for SR, lexical disambiguation, and topic-boundary recognition.
9. A suite of specialized programs would help to resolve translation mismatches, for instance to supply referents for zero pronouns.

Acknowledgements

Warmest appreciation to CompuServe, Inc. for making the chat-based SLT demonstrations possible. In particular, thanks are due to Mary Flanagan, then Manager, Advanced Technologies, and to Sophie Toole, then Supervisor, Language Support. Ms. Flanagan authorized and oversaw both demos. Ms. Toole organized and conducted the Grenoble demo and played an active role in making the SR and speech-synthesis software operational. Thanks also to Phil Jensen and Doug Chinnock, translation system engineers. The demos made use of pre-existing proprietary software.

Work on all nine of the issues discussed here began at ATR Interpreting Telecommunications Laboratories in Kyoto, Japan. I am very grateful for the support and stimulation I received there.

Thanks also to numerous colleagues at GETA (Groupe d'Études pour la Traduction Automatique) at the Université Joseph Fourier in Grenoble, France; and at DFKI (Deutsches Forschungszentrum für Künstliche Intelligenz) in Saarbrücken, Germany.

The opinions expressed throughout are mine alone.

Notes

[1] CompuServe's chat translation project was discontinued in early 1998. All trademarks are hereby acknowledged.

[2] A later commercial chat translation service, that of Uni-verse, Inc. (now discontinued), gave a comparable throughput in 2–3 seconds.

[3] Continuous French was released just before the second demo, but because little testing time was available, a decision was made to forego its use.

[4] SpeechLinks software from SpeechOne, Inc.

[5] By March 1998, upgrades of the continuous software had already made this macro less necessary. Direct dictation to the chat window would then have been possible without it, with some sacrifice of advanced features for voice-driven interactive correction of errors.

[6] Kowalski et al. (1995) arranged the only previous demonstration known to the author of SLT using commercial dictation software for input (though at least one group (Miike et al., 1988) had previously *simulated* SLT after a fashion by automatically translating typed conversations). Since Kowalski's users (spectators at twin exposition displays in Boston, Massachusetts and Lyons, France) were untrained, little interactive correction of dictation was possible. For this and other reasons, translation quality was generally low (Burton Rosenberg, personal communication); but as the main purpose of the demo was to make an artistic and social statement concerning future hi-tech possibilities for cross-cultural communication, this was no great cause for concern. Text was transmitted via FTP, rather than via chat as in the experiments reported here. See Seligman (1997) for a fuller account.

[7] `www.itl.atr.co.jp/matrix/c-star/matrix.en.html`

[8] `www.c-star.org`

[9] Mailbox files were extensively and successfully used in the French entry in the C-STAR II SLT demo of July 22, 1999 (`www.c-star.org`).

[10] Inclusion of other levels is also possible. At the lower limit, assuming the grammar were stochastic, one could even use sub-phone speech segments as grammar terminals, thus subsuming even HMM-based phone recognition in the parsing regime. At an intermediate level between phones and words, syllables could be used.

[11] The parse tree was not used for analysis, however. Instead, it was discarded, and a unification-based parser began a new parse for MT purposes on a text string passed from speech recognition.

[12] For one example of extensive related work in the framework of the Verbmobil system, see Kompe et al. (1997).

[13] A related but distinct proposal appears in Hosaka et al. (1994).

[14] Entropic Research Laboratory, Washington, DC, 1993.

[15] PanEBMT operates solo only when the entire source expression can be rendered with a single memorized target expression.

[16] The sort of generalization suggested here – from graded semantic similarity measurements to graded measurements of similarity along multiple dimensions – should not be confused with that of Generalized EBMT, the example-based technique proposed for CMU's PanEBMT engine. That engine utilizes no graded similarity measurements along any scale. Its generalization instead involves substitution of semantic tags for lexical items in examples and in input, so that for example *John Hancock was in Washington* becomes ⟨PERSON⟩ *was in* ⟨CITY⟩.

[17] In this calculation, fixed elements are treated differently from variable elements, and variable elements can be weighted to varying degrees: the heads of complex structures are differently weighted than non-heads.

[18] See for example the website of the Discourse Resource Initiative, `www.georgetown.edu/luper-foy/Discourse-Treebank/dri-home.html`, with links to recent workshops, or browse Walker (1999), especially regarding attempted standardization of Japanese discourse labeling (Ichikawa et al., 1999).

[19] Called "Communicative Acts" in Seligman et al. (1995) and "Situational Formulas" in Seligman (1991).
[20] While speakers often repeat an interlocutor's utterance to confirm it, we do not use a REPEAT-TO-CONFIRM CBSA, since it is apparently signaled by no cue patterns, and thus could only be recognized by noting inter-utterance repetition.
[21] For example, if the concept MONEY has been observed, then the lexical item *bank* has the meaning closest to MONEY in the network: 'savings institution' rather than 'edge of river', etc.
[22] See Schütze (1998) or Veling and van der Werd (1999) concerning the use of co-occurrence networks for disambiguation, though without comparable segmentation or semantic smoothing.
[23] Compare, for example, Morris and Hirst (1991), Hearst (1994), Nomoto and Nitta (1994), or Kozima and Furugori (1994).
[24] One possible formulation: "Can the audience easily identify which one is meant?" See Boitet (1996a) for discussion.

References

Aberdeen, John, Sam Bayer, Sasha Caskey, Laurie Damianos, Alan Goldschen, Lynette Hirschman, Dan Loehr and Hugo Trappe: 1999, 'Implementing Practical Dialogue Systems with the DARPA Communicator Architecture', *IJCAI-99 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Stockholm, Sweden, pp. 81–86.

Alexandersson, Jan, Norbert Reithinger and Elisabeth Maier: 1997, 'Insights into the Dialogue Processing of VERBMOBIL', *Fifth Conference on Applied Natural Language Processing*, Washington, DC, pp. 33–40.

Barnett, Jim, Kevin Knight, Inderjeet Mani and Elaine Rich: 1990, 'Knowledge and Natural Language Processing', *Communications of the ACM* **33**(8), 50–71.

Black, Alan: 1997, 'Predicting the Intonation of Discourse Segments from Examples in Dialogue Speech', in Y. Sagisaka, N. Campbell and N. Higuchi (eds), *Computing Prosody*, Springer Verlag, Berlin, pp. 117–128.

Black, Ezra, Roger Garside and Geoffrey Leech: 1993, *Statistically-driven Computer Grammars of English: The IBM/Lancaster Approach*, Rodopi, Amsterdam.

Blanchon, Hervé: 1996, 'A Customizable Interactive Disambiguation Methodology and Two Implementations to Disambiguate French and English Input', in C. Boitet (1996a), pp. 190–200.

Boitet, Christian (ed.): 1996a, *Proceedings of MIDDIM-96 Post-COLING Seminar on Interactive Disambiguation*, Le Col de Porte, France.

Boitet, Christian: 1996b, 'Dialogue-based Machine Translation for Monolinguals and Future Self-explaining Documents', in C. Boitet (1996a), pp. 75–85.

Boitet, Christian and Mark Seligman: 1994, 'The "Whiteboard" Architecture: A Way to Integrate Heterogeneous Components of NLP Systems', *COLING 94, The 15th International Conference on Computational Linguistics*, Kyoto, Japan, pp. 426–430

Brown, Ralph D.: 1996, 'Example-based Machine Translation in the Pangloss System', *COLING-96, The 16th International Conference on Computational Linguistics*, Copenhagen, Denmark, pp. 169–174.

Dohsaka, K.: 1990, 'Identifying the Referents of Zero-pronouns in Japanese Based on Pragmatic Constraint Interpretation', *9th European Conference on Artificial Intelligence, ECAI '90*, Stockholm, Sweden, pp. 240–245.

Erman, Lee D. and Victor R. Lesser: 1990, 'The Hearsay-II Speech Understanding System: A Tutorial', in A. Waibel and K.-F. Lee (eds), *Readings in Speech Recognition*, Morgan Kaufmann, San Mateo, CA, pp. 235–245.

Fano, Robert M.: 1961, *Transmission of Information: A Statistical Theory of Communications*, MIT Press, Cambridge, MA.

Fillmore, Charles J., Paul Kay and Catherine O'Connor: 1988, 'Regularity and Idiomaticity in Grammatical Constructions: The Case of Let Alone', *Language* **64**, 501–538.

Flanagan, Mary: 1997, 'Machine Translation of Interactive Texts', In *MT Summit VI, Machine Translation: Past Present Future*, San Diego, CA, p. 50.

Frederking, Robert, Alexander Rudnicky, and Christopher Hogan: 1997, 'Interactive Speech Translation in the DIPLOMAT Project', *Spoken Language Translation: Proceedings of a Workshop Sponsored by the Association for Computational Linguistics and by the European Network in Language and Speech (ELSNET)*, Madrid, Spain, pp. 61–66.

Furukawa Ryo, Yato Fumihiro and Loken-Kim Kyung-ho: 1993, *Denwakaiwa o Maruchimedia Kaiwa no Tokuchōbunseki* [Multimedia Dialogue Feature Analysis of Telephone Conversations]. Technical Report TR-IT-0020, ATR Interpreting Telecommunications Laboratories, Kyoto, Japan.

Furuse, Osamu and Hitoshi Iida: 1996, 'Incremental Translation Using Constituent Boundary Patterns', *COLING-96, The 16th International Conference on Computational Linguistics*, Copenhagen, Denmark, pp. 412–417.

Görz, Günther, Marcus Kesseler, Jörg Spilker and Hans Weber: 1996, 'Research on Architectures for Integrated Speech/Language Systems in Verbmobil', *COLING-96, The 16th International Conference on Computational Linguistics*, Copenhagen, Denmark, pp. 484–489.

Grosz, Barbara J., Aravind K. Joshi and Scott Weinstein: 1983, 'Providing a Unified Account of Definite Noun Phrases in Discourse', *21st Annual Meeting of the Association for Computational Linguistics*, Cambridge, MA, pp. 44–50.

Hearst, Marti A.: 1994, 'Multi-paragraph Segmentation of Expository Text', *32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, NM, pp. 9–16.

Hosaka, Junko, Mark Seligman and Harald Singer: 1994, 'Pause as a Phrase Demarcator for Speech and Language Processing', *COLING 94, The 15th International Conference on Computational Linguistics*, Kyoto, Japan, pp. 987–991.

Ichikawa, A., M. Araki, Y. Horiuchi, M. Ishizaki, S. Itabashi, T. Itoh, H. Kashioka, K. Kato, H. Kikuchi, H. Koiso, T. Kumagai, A. Kurematsu, K. Maekawa, S. Nakazato, M. Tamoto, S. Tutiya, Y. Yamashita and T. Yoshimura: 1999, 'Evaluation of Annotation Schemes for Japanese Discourse', in M. Walker (1999), pp. 26–34.

Iida, Hiroshi, Eichiro Sumita and Osamu Furuse: 1996, 'Spoken Language Translation Method Using Examples', *COLING-96, The 16th International Conference on Computational Linguistics*, Copenhagen, Denmark, pp. 1074–1077.

Iwadera, T., M. Ishizaki and T. Morimoto: 1995, 'Recognizing an Interactional Structure and Topics of Task-oriented Dialogues', *Proceedings of the European Workshop on Spoken Dialogue Systems*, Vigsø, Denmark, pp. 41–44.

Jokinen, Kristiina, Hideki Tanaka and Akio Yokoo: 1998, 'Context Management with Topics for Spoken Dialogue Systems', *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Canada, pp. 631–637.

Joshi, Aravind K. and Scott Weinstein: 1981, 'Control of Inference: Role of some Aspects of Discourse Structure – Centering', *Seventh International Joint Conference on Artificial Intelligence (IJCAI-81)*, Vancouver, BC, pp. 385–387.

Julia, L., L. Neumeyer, M. Charafeddine, A. Cheyer, and J. Dowding: 1997, 'HTTP://WWW.SPEECH.SRI.COM/DEMOS/ATIS.HTML', *Working notes of the AAAI'97 Spring Symposium Workshop on Natural Language Processing for the Web*, Stanford, CA, pp. 72–76.

Jurafsky, Daniel: 1993, *A Cognitive Model of Sentence Interpretation: The Construction Grammar Approach*. Technical Report TR-93-077. International Computer Science Institute and Department of Linguistics, University of California, Berkeley.

Kay, Paul: 1990, 'Even', *Linguistics and Philosophy* **13**, 59–216.

Knott, Alistair: 1996, *A Data-driven Methodology for Motivating a Set of Coherence Relations*, Ph.D. thesis, Department of Artificial Intelligence, University of Edinburgh.

Knott, Alistair and Robert Dale: 1995, 'Using Linguistic Phenomena to Motivate a Set of Rhetorical Relations', *Discourse Processes* **18**, 35–62.

Kompe, R., A. Kiessling, H. Niemann, E. Noeth, A. Batliner, S. Schachtl, R. Ruland and H. U. Block: 1997, 'Improving Parsing of Spontaneous Speech with the Help of Prosodic Boundaries', *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)*, Munich, Germany, pp. 811–814.

Kowalski, Piotr, Burton Rosenberg and Jeffrey Krause: 1995, 'Information Transcript', *Biennale de Lyon d'Art Contemporain*, Lyon, France.

Kozima, Hideki and Teiji Furugori: 1994, 'Segmenting Narrative Text into Coherent Scenes', *Literary and Linguistic Computing* **9**, 13–19.

Lenat, Douglas B. and R. V. Guha: 1990, *Building Large Knowledge-based Systems*, Addison-Wesley, Reading, MA.

Loken-Kim, Kyung-ho, Fumihiro Yato, Kazuhiko Kurihara, Laurel Fais and Ryo Furukawa: 1993, *AMUSE – ATR Multi-media Simulation Environment*. Technical Report TR-IT-0018, ATR Interpreting Telecommunications Laboratories, Kyoto, Japan.

Mahesh, Kavi (ed.): 1997, *Natural Language Processing for the World Wide Web: Papers from the 1997 AAAI Spring Symposium*, The AAAI Press, Cambrdige, MA.

Miike, Seiji, Koichi Hasebe, Harold Somers and Shin-ya Amano: 1988, 'Experiences with an On-line Translating Dialogue System', *The 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY, pp. 155–162.

Morimoto, T., T. Takezawa, F. Yato, S. Sagayama, T. Tashiro, M. Nagata and A. Kurematsu: 1993, 'ATR's Speech Translation System: ASURA', *European Conference on Speech Communication and Technology*, Berlin, Germany, pp. 1295–1298.

Morris, Jane and Graeme Hirst: 1991, 'Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text', *Computational Linguistics* **17**, 21–48.

Murata, Masaki and Makoto Nagao: 1993, 'Determination of Referential Property and Number of Nouns in Japanese Sentences for Machine Translation into English', *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation TMI '93 – MT in the Next Generation*, Kyoto, Japan, pp. 218–225.

Nadas, Arthur: 1985, 'On Turing's Formula for Word Probabilities', *IEEE Transactions on Acoustics, Speech and Signal Processing* **33**, 1414–1416.

Nagao, Makoto: 1984, 'A Framework of a Mechanical Translation between Japanese and English by Analogy Principle', in A. Elithorn and R. Banerji (eds), *Artificial and Human Intelligence*, North-Holland, Amsterdam, pp. 173–180.

Nagata, Masaaki and Tsuyoshi Morimoto: 1993, 'An Experimental Statistical Dialogue Model to Predict the Speech Act Type of the Next Utterance', *Proceedings of ISSD-93, International Symposium on Spoken Dialogue – New Directions in Human and Man-machine Communication*, Tokyo, Japan, pp. 83–86.

Nomoto, Tadashi and Yoshihiko Nitta: 1994, 'A Grammatico-statistical Approach to Discourse Partitioning', *COLING 94, The 15th International Conference on Computational Linguistics*, Kyoto, Japan, pp. 1145–1150.

Ohno Susumu and Hamanishi Masando: 1981, *Kadokawa Ruigo Shin-jiten* [Kadokawa New Word Category Dictionary], Kadokawa Shoten, Tōkyō.

Pyra, Marianne: 1995, *Using Internet Relay Chat*, Que Corporation, Indianapolis, IN.

Reithinger, Norbert: 1995, 'Some Experiments in Speech Act Prediction', in Johanna Moore and Marilyn Walker (eds), *Empirical Methods in Discourse: Interpretation & Generation: Papers from the 1995 AAAI Symposium*, AAAI Press, Cambridge, MA, pp. 126–131.

Reithinger, Norbert and Martin Klesen: 1997, 'Dialogue Act Classification Using Language Models', *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech)*, Rhodes, Greece, pp. 2235–2238.

Sato, Satoshi: 1991, *Example-based Machine Translation*, Doctoral thesis (in Japanese), Kyoto University, Japan.

Schütze, Hinrich: 1998, 'Automatic Word Sense Discrimination', *Computational Linguistics* **24**, 97–124.

Searle, J.: 1969, *Speech Acts*, Cambridge University Press, Cambridge, England.

Seligman, Mark: 1991, *Generating Discourses from Networks Using an Inheritance-based Grammar*, Dissertation, Department of Linguistics, University of California, Berkeley.

Seligman, Mark: 1994a, *CO-OC: Semi-automatic Production of Resources for Tracking Morphological and Semantic Co-occurrences in Spontaneous Dialogues*. Technical Report TR-IT-0084, ATR Interpreting Telecommunications Laboratories, Kyoto, Japan.

Seligman, Mark: 1994b, *CNTR: Basic Functions for Centering Experiments with ASURA*. Technical Report TR-IT-0085, ATR Interpreting Telecommunications Laboratories, Kyoto, Japan.

Seligman, Mark: 1997, 'Interactive Real-time Translation via the Internet', in K. Mahesh (1997), pp. 142–148.

Seligman, Mark, Jan Alexandersson and Kristiina Jokinen: 1999, 'Tracking Morphological and Semantic Co-occurrences in Spontaneous Dialogues', *IJCAI-99 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Stockholm, Sweden, pp. 105–111.

Seligman, Mark and Christian Boitet: 1993, 'A "Whiteboard" Architecture for Automatic Speech Translation', *Proceedings of ISSD-93, International Symposium on Spoken Dialogue – New Directions in Human and Man-machine Communication*, Tokyo, Japan, pp. 243–246.

Seligman, Mark, Christian Boitet and Boubaker Meddeb-Hamrouni: 1998a, 'Transforming Lattices into Non-deterministic Automata with Optional Null Arcs', *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Canada, pp. 1205–1211.

Seligman, Mark, Laurel Fais and Mutsuko Tomokiyo: 1995, *A Bilingual Set of Communicative Act Labels for Spontaneous Dialogues*. Technical Report TR-IT-0081, ATR Interpreting Telecommunications Laboratories, Kyoto, Japan.

Seligman, Mark, Mary Flanagan and Sophie Toole: 1998b, 'Dictated Input for Broad-coverage Speech Translation', *Association for Machine Translation in the Americas (AMTA-98), Workshop on Embedded MT Systems: Design, Construction, and Evaluation of Systems with an MT Component*, Langhorne, PA.

Seligman, Mark, Junko Hosaka and Harald Singer: 1997, ' "Pause Units" and Analysis of Spontaneous Japanese Dialogues: Preliminary Studies', in E. Meier, M. Mast and S. Luperfoy (eds), *Dialogue Processing in Spoken Language Systems*, Springer, Berlin, pp. 110–112.

Seligman, Mark, Masami Suzuki and Tsuyoshi Morimoto: 1993. *Semantic-level Transfer in Japanese-German Speech Translation: Some Experiences*. Technical Report NLC93-13, Institute of Electronics, Information, and Communication Engineers (IEICE), Japan.

Sidner, Candace: 1979, *Toward a Computational Theory of Definite Anaphora Comprehension in English*. Technical Report AI-TR-537, MIT, Cambridge, MA.

Sobashima, Yauhiro and Hitoshi Iida: 1995, 'A Multi-dimensional Analogy-based, Context-dependent, Bottom-up Parsing Method for Spoken Dialogues', *Third Natural Language Processing Pacific Rim Symposium NLPRS'95*, Seoul, Korea, pp. 586–591.

Sobashima Yasuhiro and Mark Seligman: 1994, 'Yōrei to no tagenteki ruijido keisan ni motodzuku bunmyaku izon no kobun kaiseki hō', [Parsing Method for Example-based Analysis Integrating Multiple Knowledge Sources], *Shadan hōjin jōhō shori gakkai dai49 kai zenkoku taikai kōen ronbun shū*, Vol. 3, Sapporo, Japan, pp. 103–104.

Stock, Oliviero, Rino Falcone and Patrizia Insinnamo: 1989, 'Bi-directional Charts: A Potential Technique for Parsing Spoken Natural Language Sentences', *Computer Science and Language* **3**, 219–237.

Sumita, Eichiro and Hitoshi Iida: 1992, 'Example-based Transfer of Adnominal Particles into English', *IEICE Transactions on Information Systems* **E75-D**(4), 585–594.

Takeda, Shingo and Norihisa Doi: 1994, 'Centering in Japanese: A Step Towards Better Interpretation of Pronouns and Zero-pronouns', *COLING 94, The 15th International Conference on Computational Linguistics*, Kyoto, Japan, pp. 1151–1156.

Takezawa, Toshiyuki, Fumiaki Sugaya and Akio Yokoo: 1999, 'ATR-MATRIX: A Spontaneous Speech Translation System between English and Japanese', *ATR Journal* **2**, 29–33.

Veling, Anne and Peter van der Weerd: 1999, 'Conceptual Grouping in Word Co-occurrence Networks', *IJCAI-99 Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, pp. 694–699.

Wahlster, W.: 1993, *VERBMOBIL: Translation of Face-to-Face Dialogs*. Research Report RR-93-34, German Research Center for Artificial Intelligence (DFKI GmbH), Saarbrücken, Germany.

Walker, Marilyn (ed.): 1999, *Proceedings of the ACL '99 Workshop: Towards Standards and Tools for Discourse Tagging*, College Park, MD.

Zajac, Remi and Mark Casper: 1997, 'The Temple Web Translator', in K. Mahesh (1997), pp. 149–154.