A SPEECH TO SPEECH TRANSLATION SYSTEM BUILT FROM STANDARD COMPONENTS

Manny Rayner[1], *Hiyan Alshawi*[1], *Ivan Bretan*[3], *David Carter*[1],
Vassilios Digalakis[2], *Björn Gambäck*[3], *Jaan Kaja*[4], *Jussi Karlgren*[3],
Bertil Lyberg[4], *Steve Pulman*[1], *Patti Price*[2] *and Christer Samuelsson*[3]

(1) SRI International, Cambridge, UK (2) SRI International, Menlo Park, CA
(3) SICS, Stockholm, Sweden (4) Telia Research AB, Haninge, Sweden

ABSTRACT

This paper[1] describes a speech to speech translation system using standard components and a suite of generalizable customization techniques. The system currently translates air travel planning queries from English to Swedish. The modular architecture is designed to be easy to port to new domains and languages, and consists of a pipelined series of processing phases. The output of each phase consists of multiple hypotheses; statistical preference mechanisms, the data for which is derived from automatic processing of domain corpora, are used between each pair of phases to filter hypotheses. Linguistic knowledge is represented throughout the system in declarative form. We summarize the architectures of the component systems and the interfaces between them, and present initial performance results.

1. INTRODUCTION

From standard components and a suite of generalizable customization techniques, we have developed an English to Swedish speech translation system in the air travel planning (ATIS) domain. The modular architecture consists of a pipelined series of processing phases that each output multiple hypotheses filtered by statistical preference mechanisms.[2] The statistical information used in the system is derived from automatic processing of domain corpora. The architecture provides greater robustness than a 1-best approach, and yet is more computationally tractable and more portable to new languages and domains than a tight integration, because of the modularity of the components: speech recognition, source language processing, source to target language transfer, target language processing, and speech synthesis.

Some aspects of adaptation to the domain task were fairly simple: addition of new lexical entries was facilitated by existing tools, and grammar coverage required adding only a few very domain-specific phrase structure rules, as described in Section 3.1. Much of the effort in the project, however, has focussed on the development of well-specified methods for adapting and customizing other aspects of the existing modules, and on tools for guiding the process. In addition to the initial results (Section 5), the reported work makes several contributions to speech translation in particular and to language processing in general:

- A general method for training statistical preferences to filter multiple hypotheses, for use in ranking both analysis and translation hypotheses (Section 3.2);
- A method for rapid creation of a grammar for the target language by exploiting overlapping syntactic structures in the source and target languages (Section 3.3);
- An Explanation Based Learning (EBL) technique for automatically chunking the grammar into commonly occurring phrase-types, which has proven valuable in maximizing return on effort expended on coverage extension, and a set of procedures for automatic testing and reporting that helps to ensure smooth integration across aspects of the effort performed at the various sites involved (Section 4).

2. COMPONENTS AND INTERFACES

The speech translation process begins with SRI's DECIPHER(TM) system, based on hidden Markov modeling and a progressive search [12, 13]. It outputs to the source language processor a small lattice of word hypotheses generated using acoustic and language model scores. The language processor, for both English and Swedish, is the SRI Core Language Engine (CLE) [1], a unification-based, broad coverage natural language system for analysis and generation. Transfer occurs at the level of quasi logical form (QLF); transfer rules are defined in a simple declarative formalism [2]. Speech synthesis is performed by the Swedish Telecom PROPHON

[1] The research reported in this paper was sponsored by Swedish Telecom (Televerket Nät). Several people not listed as co-authors have also made contributions to the project: among these we would particularly like to mention Marie-Susanne Agnäs, George Chen, Dick Crouch, Barbro Ekholm, Arnold Smith, Tomas Svensson and Torbjörn Åhs.

[2] The preference mechanism between target language text output and speech synthesis has not yet been implemented.

system [8], based on stored polyphones. This section describes in more detail these components and their interfaces.

2.1. Speech Recognition

The first component is a fast version of SRI's DECIPHER(TM) speaker-independent continuous speech recognition system [12]. It uses context-dependent phonetic-based hidden Markov models with discrete observation distributions for 4 features: cepstrum, delta-cepstrum, energy and delta-energy. The models are gender-independent and the system is trained on 19,000 sentences and has a 1381-word vocabulary. The progressive recognition search [13] is a three-pass scheme that produces a word lattice and an N-best list for use by the language analysis component. Two recognition passes are used to create a word lattice. During the forward pass, the probabilities of all words that can end at each frame are recorded, and this information is used to prune the word lattice generated in the backward pass. The word lattice is then used as a grammar to constrain the search space of a third recognition pass, which produces an N-best list using an exact algorithm.

2.2. Language Analysis and Generation

Language analysis and generation are performed by the SRI Core Language Engine (CLE), a general natural-language processing system developed at SRI Cambridge [1]; two copies of the CLE are used, equipped with English and Swedish grammars respectively. The English grammar is a large, domain-independent unification-based phrase-structure grammar, augmented by a small number of domain-specific rules (Section 3.1). The Swedish grammar is a fairly direct adaptation of the English one (Section 3.3).

The system's linguistic information is in declarative form, compiled in different ways for the two tasks. In analysis mode, the grammar is compiled into tables that drive a left-corner parser; input is supplied in the form of a word hypothesis lattice, and output is a set of possible semantic analyses expressed in Quasi Logical Form (QLF). QLF includes predicate-argument structure and some surface features, but also allows a semantic analysis to be only partially specified [3].

The set of QLF analyses is then ranked in order of *a priori* plausibility using a set of heuristic preferences, which are partially trainable from example corpus data (Section 3.2). In generation mode, the linguistic information is compiled into another set of tables, which control a version of the Semantic Head-Driven Generation algorithm [16]. Here, the input is a QLF form, and the output is the set of possible surface strings which realize the form. Early forms of the analysis and generation algorithms used are described in [1].

2.3. Speech/Language Interface

The interface between speech recognition and source language analysis can be either a 1-best or an N-best interface. In 1-best mode, the recognizer simply passes the CLE a string representing the single best hypothesis. In N-best mode, the string is replaced by a list containing all hypotheses that are active at the end of the third recognition pass. Since the word lattice generated during the first two recognition passes significantly constrains the search space of the third pass, we can have a large number of hypotheses without a significant increase in computation.

As the CLE is capable of using lattice input directly [6], the N-best hypotheses are combined into a new lattice before being passed to linguistic processing; in cases where divergences occur near the end of the utterance, this yields a substantial speed improvement. The different analyses produced are scored using a weighted sum of the acoustic score received from DECIPHER and the linguistic preference score produced by the CLE. When at least one linguistically valid analysis exists, this implicitly results in a selection of one of the N-best hypotheses. Our experimental findings to date indicate that N=5 gives a good tradeoff between speed and accuracy, performance surprisingly being fairly insensitive to the setting of the relative weights given to acoustic and linguistic scoring information. Some performance results are presented in Section 5.

2.4. Transfer

Unification-based QLF transfer [2], compositionally translates a QLF of the source language to a QLF of the target language. QLF is the transfer level of choice in the system, since it is a contextually unresolved semantic representation reflecting both predicate-argument relations and linguistic features such as tense, aspect, and modality. The translation process uses declarative transfer rules containing cross-linguistic data, i.e., it specifies only the differences between the two languages. The monolingual knowledge of grammars, lexica, and preferences is used for ranking alternative target QLFs, filtering out ungrammatical QLFs, and finally generating the source language utterance.

A transfer rule specifies a pair of QLF patterns; the left hand side matches a fragment of the source language QLF and the right hand side the corresponding target QLF. Table 1 breaks down transfer rules by type. As can been seen, over 90% map atomic constants to atomic constants; of the remainder, about half relate to spe-

Table 1: Transfer rule statistics

Atom to atom	649	91%
Complex (lexical)	27	4%
Complex (non-lexical)	34	5%
Total	710	100%

cific lexical items, and half are general structural transfer rules. For example, the following rule expresses a mapping of English NPs postnominally modified by a progressive VP (*"Flights going to Boston"*) to Swedish NPs modified by a relative clause (*"Flygningar som går till Boston"*):

```
[and,tr(head),
 form(verb(tense=n,perf=P,prog=y),
      tr(mod))]
>=
[and,tr(head),
 [island,form(verb(tense=pres,perf=P,prog=n),
      tr(mod))]]
```

Transfer variables, of the form `tr(`*atom*`)`, show how subexpressions in the source QLF correspond to subexpressions in the target QLF. Note how the transition from a tenseless, progressive VP to a present tense, non-progressive VP can be specified directly through changing the values of the slots of the "verb" term. This fairly simple transfer rule formalism seems to allow most important restructuring phenomena (e.g., change of aspect, object raising, argument switching, and to some extent also head switching) to be specified succinctly. The degree of compositionality in the rule set currently employed is high; normally no special transfer rules are needed to specify combinations of complex transfer. In addition, the vast majority of the rules are reversible, providing for future Swedish to English translation.

2.5. Speech Synthesis

The Prophon speech synthesis system, developed at Swedish Telecom, is an interactive environment for developing applications and conducting research in multilingual text-to-speech conversion. The system includes a large lexicon, a speech synthesizer and rule modules for text formatting, syntactic analysis, phonetic transcription, parameter generation and prosody. Two synthesis strategies are included in the system, formant synthesis and polyphone synthesis, i.e., concatenation of speech units of arbitrary size. In the latter case, the synthesizer accesses the database of polyphone speech waveforms according to the allophonic specification derived from the lexicon and/or phonetic transcription rules. The polyphones are concatenated and the prosody of the utterance is imposed via the PSOLA (pitch synchronous overlap add) signal processing technique [11]. The Prophon system has access to information other than the text string, in particular the parse tree, which can be used to provide a better, more natural prosodic structure than normally is possible.

3. ADAPTATION

In this section, we describe the methods used for adapting the various processing components to the English-Swedish ATIS translation task. Section 3.1 describes the domain customization of the language component, and section 3.2 the semi-automatic method developed to customize the linguistic preference filter. Finally, section 3.3 summarizes the work carried out in adapting the English-language grammar and lexicon to Swedish.

3.1. CLE Domain Adaptation

We begin by describing the customizations performed to adapt the general CLE English grammar and lexicon to the ATIS domain. First, about 500 lexical entries needed to be added. Of these, about 450 were regular content words (*airfare*, *Boston*, *seven forty seven*, etc.), all of which were added by a graduate student[3] using the interactive VEX lexicon acquisition tool [7]. About 55 other entries, not of a regular form, were also added. Of these, 26 corresponded to the letters of the alphabet, which were treated as a new syntactic class, 15 or so were interjections (*Sure*, *OK*, etc.), and seven were entries for the days of the week, which turned out to have slightly different syntactic properties in American and British English. The only genuinely new entries were for *available*, *round trip*, *first class*, *nonstop* and *one way*, all of which failed to fit syntactic patterns previously implemented within the grammar, (e.g. *"Flights available from United"*, *"Flights to Boston first class"*).

Sixteen domain-specific phrase-structure rules were also added, most of them by the graduate student. Of these, six covered 'code' expressions (e.g. *"Q X"*), and eight covered 'double utterances' (e.g. *"Flights to Boston show me the fares"*). The remaining two rules covered ordinal expressions without determiners (*"Next flight to Boston"*), and PP expressions of the form '*Name* to *Name*' (e.g. *"Atlanta to Boston Friday"*). Finally, the preference metrics were augmented by a preference for attaching 'from-to' PP pairs to the same constituent, (this is a domain-independent heuristic, but is particularly important in the context of the ATIS task), and the semantic collocation preference metrics (Section 3.2)

[3] Marie-Susanne Agnäs, the graduate student in question, was a competent linguist but had no previous experience with the CLE or other large computational grammars.

were retrained with ATIS data. The grammar and lexicon customization effort has so far consumed about three person-months of specialist time, and about two and a half person-months of the graduate student. The current level of coverage is indicated in Section 5.

3.2. Training Preference Heuristics

Grammars with thorough coverage of a non-trivial sublanguage tend to yield large numbers of analyses for many sentences, and rules for accurately selecting the correct analysis are difficult if not impossible to state explicitly. We therefore use a set of about twenty preference metrics to rank QLFs in order of *a priori* plausibility. Some metrics count occurrences of phenomena such as adjuncts, ellipsis, particular attachment configurations, or balanced conjunctions. Others, which are trained automatically, reflect the strengths of semantic collocations between triples of logical constants occurring in relevant configurations in QLFs.

The overall plausibility score for a QLF under this scheme is a weighted (scaled) sum of the scores returned by the individual metrics. Initially, we chose scaling factors by hand, but this became an increasingly skilled and difficult task as more metrics were added, and it was clear that the choice would have to be repeated for other domains. The following semi-automatic optimization procedure [4] was therefore developed.

QLFs were derived for about 4600 context-independent and context-dependent ATIS sentences of 1 to 15 words. It is easy to derive from a QLF the set of segments of the input sentence which it analyses as being either predications or arguments. These segments, taken together, effectively define a tree of roughly the form used by the Treebank project [5]. A user presented with all strings derived ¿from any QLF for a sentence selected the correct tree (if present). A skilled judge was then able to assign trees to hundreds of sentences per hour.

The "goodness" of a QLF Q with respect to an approved tree T was defined as $I(Q,T) - 10 * A(Q,T)$, where $I(Q,T)$ is the number of string segments induced by Q and present in T, and $A(Q,T)$ is the number induced by Q but absent from T. This choice of goodness function was found, by trial and error, to lead to a good correlation with the metrics. Optimization then consisted of minimizing, with respect to scaling factors c_j for each preference metric m_j, the value of

$$\sum_i (g_i - \sum_j c_j s_{ij})^2$$

where g_i is the goodness of QLF i and s_{ij} is the score assigned to QLF i by metric f_j; to remove some "noise" from the data, all values were relativized by subtracting the (average of the) corresponding scores for the best-scoring QLF(s) for the sentence.

The kth simultaneous equation, derived by setting the derivative of the above expression with respect to c_k to zero for the minimum, is

$$\sum_i s_{ik}(g_i - \sum_j c_j s_{ij}) = 0$$

These equations can be solved by Gaussian elimination.

The optimized and hand-selected scaling factors each resulted in a correct QLF being selected for about 75% of the 157 sentences from an unseen test set that were within coverage, showing that automatic scaling can produce results as good as those derived by labour- and skill-intensive hand-tuning. The value of Kendall's ranking correlation coefficient between the relativized "goodness" values and the scaled sum (reflecting the degree of agreement between the orderings induced by the two criteria) was also almost identical for the two sets of factors. However, the optimized factors achieved much better correlation (0.80 versus 0.58) under the more usual product-moment definition of correlation, $\sigma_{xy}/\sigma_x\sigma_y$, which the least-squares optimization used here is defined to maximize. This suggests that optimization with respect to a (non-linear) criterion that reflects ranking rather than linear agreement could lead to a still better set of scaling factors that might outperform both the hand-selected and the least-squares-optimal ones. A hill-climbing algorithm to determine such factors is therefore being developed.

The training process allows optimization of scaling factors, and also provides data for several metrics assessing semantic collocations. In our case, we use semantic collocations extracted from QLF expressions in the form of $(H1, R, H2)$ triples where $H1$ and $H2$ are the head predicates of phrases in a sentence and R indicates the semantic relationship (e.g. a preposition or an argument position) between the two phrases in the proposed analysis. We have found that a simple metric, original to us, that scores triples according to the average treebank score of QLFs in which they occur, performs about as well as a chi-squared metric, and better than one based on mutual information (cf [9]).

3.3. CLE Language Adaptation

The Swedish-language customization of the CLE (S-CLE) has been developed at SICS from the English-language version by replacing English-specific modules with corresponding Swedish-language versions.[4] Swedish is a Germanic language, linguistically about as "far" from English as German is. Our experience sug-

[4] The S-CLE and the adaptation process is described in detail in [10].

gests that adapting the English system to close languages is fairly easy and straight-forward. The total effort spent on the Swedish adaptation was about 14 person-months (compared with about 20 person-years for the original CLE), resulting in coverage only slightly less than that of the English version.

The amount of work needed to adapt the various CLE modules to Swedish declined steadily as a function of their "distance" from surface structure. Thus the morphology rules had to be nearly completely rewritten; Swedish morphology is considerably more complex than English. In contrast, only 33 of the 401 Swedish function word entries were not derived from English counterparts, the differences being confined to variance in surface form and regular changes to the values of a small number of features. At the level of syntax, 97 (81%) of a set of 120 Swedish syntax rules were derived from exact or very similar English rules. The most common difference is some small change in the features; for example, Swedish marks for definiteness, which means that this feature often needs to be added. 11 rules (9%) originated in English rules, but had undergone major changes, e.g., some permutation or deletion of the daughters; thus Swedish time rules demand a word-order which in English would be "o'clock five", and there is a rule that makes an NP out of a bare definite NBAR. This last rule corresponds to the English NP → DET NBAR rule, with the DET deleted but the other features instantiated as if it were present. Only 12 (10%) Swedish syntax rules were completely new. The percentage of changed semantic rules was even smaller.

The most immediately apparent surface divergences between Swedish and English word-order stem from the strongly verb-second nature of Swedish. Formation of both YN- and WH-questions is by simple inversion of the subject and verb without the introduction of an auxiliary, thus for example *"Did he fly with Delta?"* is *"Flög han med Delta?"*, lit. *"Flew he with Delta?"*. It is worth noting that these changes can all be captured by doing no more than adjusting features. The main rules that had to be written "from scratch" are those that cover adverbials, negation, conditionals, and the common *vad ...för* construction, e.g., *"Vad finns det för flygningar till Atlanta"* (lit. *"What are there for flights to Atlanta"*, i.e., *"What flights are there to Atlanta?"*).

4. RATIONAL DEVELOPMENT METHODOLOGY

In a project like this one, where software development is taking place simultaneously at several sites, regular testing is important to ensure that changes retain inter-component compatibility. Our approach is to maintain a set of test corpora to be run through the system (from text analysis to text generation) whenever a significant change is made to the code or data. Changes in the status of a sentence – the translation it receives, or the stage at which it fails if it receives no translation – are notified to developers, which facilitates bug detection and documentation of progress.

The most difficult part of the exercise is the construction of the test corpora. The original training/development corpus is a 4600-sentence subset of the ATIS corpus consisting of sentences of length not more than 15 words. For routine system testing, this corpus is too large to be convenient; if a randomly chosen subset is used instead, it is often difficult to tell whether processing failures are important or not, in the sense of representing problems that occur in a large number of corpus sentences. What is needed is a sub-corpus that contains all the commonly occurring types of construction, together with an indication of how many sentences each example in the sub-corpus represents.

We have developed a systematic method for constructing representative sub-corpora, using "Explanation Based Learning" (EBL) [15]. The original corpus is parsed, and the resulting analysis trees are grouped into equivalence classes; then one member is chosen from each class, and stored with the number of examples it represents. In the simplest version, trees are equivalent if their leaves are of the same lexical types. The criterion for equivalence can be varied easily: we have experimented with schemes where all sub-trees representing NPs are deemed to be equivalent. When generalization is performed over non-lexical classes like NPs and PPs, the method is used recursively to extract representative examples of each generalized class.

At present, three main EBL-derived sub-corpora are used for system testing. Corpus 1, used most frequently, was constructed by generalizing at the level of lexical items, and contains one sentence for each class with at least three members. This yields a corpus of 281 sentences, which together represent 1743 sentences from the original corpus. Corpus 2, the "lexical" test corpus, is a set with one analyzable phrase for each lexical item occuring at least four times in the original corpus, comprising a total of 460 phrases. Corpus 3 generalizes over NPs and PPs, and analyzes NPs by generalizing over non-recursive NP and PP constituents; one to five examples are included for each class that occurs ten or more times (depending on the size of the class), giving 244 examples. This corpus is useful for finding problems linked with constructions specific to either the NP or the sentence level, but not to a combination. The time needed to process each corpus through the system is on the order of an hour.

5. RESULTS OF SYSTEM EVALUATION

In this final section we present evaluation results for the current version of the system running on data previously unseen by the developers. There is so far little consensus on how to evaluate spoken language translation systems; for instance, no evaluation figures on unseen material are cited for the systems described in [17] and [14]. We present the results below partly in an attempt to stimulate discussion on this topic.

The sentences of lengths 1 to 12 words from the Fall 1992 test set (633 sentences from 1000) were processed through the system from speech signal to target language text output, and the translations produced were evaluated by a panel fluent in both languages. Points were awarded for meaning preservation, grammaticality of the output, naturalness of the output, and preservation of the style of the original, and a translation had to be classified as acceptable on all four counts to be regarded as acceptable in general. Judgements were also elicited for intermediate results, in particular whether a speech hypothesis could be judged as a valid variant of the reference sentence in the context of the translation task, and whether the semantic analysis sent to the transfer stage was correct. The criteria used to determine whether a speech hypothesis was a valid variant of the reference were strict, typical differences being substitution of *all the* for plural *the*, *what's* for *what is*, or *I want* for *I'd like*.

The results were as follows. For 1-best recognition, 62.4% of the hypotheses were equal to or valid variants of the reference, and 55.3% were valid and also within grammatical coverage. For 5-best recognition, the corresponding figures were 78.2% and 69.0%. Selecting the acoustically highest-ranked hypothesis that was inside grammatical coverage yielded an acceptable choice in 61.1% of the examples; a scoring scheme that chose the best hypothesis using a weighted combination of the acoustic and linguistic scores did slightly better, increasing the proportion to 63.0%. 54% of the examples received a most preferred semantic analysis that was judged correct, 45.3% received a translation, and 41.8% received an acceptable translation. The corresponding error rates for each component are shown in table 2.

Table 2: Component error rates	
(1-best recognition)	(37.4%)
5-best recognition	21.8%
Speech/language interface	8.7%
Source linguistic analysis	11.8%
Source analysis preferences	13.4%
Transfer and generation	22.7%

References

1. Alshawi, H. (ed.), *The Core Language Engine*, MIT Press, 1992.
2. Alshawi, H., Carter, D., Rayner, M. and Gambäck, B., "Transfer through Quasi Logical Form", *Proc. 29th ACL*, Berkeley, 1991.
3. Alshawi, H. and Crouch, R., "Monotonic Semantic Interpretation", *Proc. 30th ACL*, Newark, 1992.
4. Alshawi, H., and Carter, D., "Optimal Scaling of Preference Metrics", SRI Cambridge Research Report, 1992.
5. Black, E., *et al.*, "A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars," *Proc. Third DARPA Speech and Language Workshop*, P. Price (ed.), Morgan Kaufmann, June 1991.
6. Carter, D.M., "Lattice-based Word Identification in CLARE", *Proc 30th ACL*, Newark, 1992.
7. Carter, D.M., "Lexical Acquisition in the Core Language Engine", *Proc. 4th European ACL*, Manchester, 1989.
8. Ceder, K. and Lyberg, B., "Yet Another Rule Compiler for Text-to-Speech Conversion?", *Proc. ICSLP*, Banff, 1993.
9. Church, K.W. and Hanks, P., "Word Association Norms, Mutual Information, and Lexicography", *Computational Linguistics* 16:22-30, 1990.
10. Gambäck, B. and Rayner, M., "The Swedish Core Language Engine", *Proc. 3rd NOTEX*, Linköping, 1992.
11. Moulines, E. and Charpentier, F., "Pitch-Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis Using Diphones", *Speech Communication* Vol. 9, 1990.
12. Murveit, H., Butzberger, J. and Weintraub, M., "Speech Recognition in SRI's Resource Management and ATIS Systems", *Proc. DARPA Workshop on Speech and Natural Language*, 1991.
13. Murveit, H., et al., "Large Vocabulary Dictation using SRI's DECIPHER(TM) Speech Recognition System: Progressive Search Techniques", *Proc. ICASSP*, 1993.
14. Roe, D.B., Pereira, F.C.N., Sproat, R.W., Riley, M.D. and Moreno, P.J., "Towards a Spoken-Language Translator for Restricted-Domain Context-Free Languages", *Proc. Eurospeech*, 1991.
15. Samuelsson, C. and Rayner, M., "Quantitative Evaluation of Explanation-Based Learning as an Optimization Tool for a Large-Scale Natural Language System", *Proc. 12th IJCAI*, Sydney, 1991.
16. Shieber, S. M., van Noord, G., Pereira, F.C.N and Moore, R.C., "Semantic-Head-Driven Generation", *Computational Linguistics*, 16:30-43, 1990.
17. Woszczyna, M. et al., "Recent advances in JANUS: A Speech Translation System", ARPA Workshop on Human Language Technology, Plainsboro, NJ, 1993.