Machine Translation **17:** 185–212, 2002.

MARS: A Statistical Semantic Parsing and Generation-Based Multilingual Automatic tRanslation System

YUQING GAO, BOWEN ZHOU, ZIJIAN DIAO, JEFFREY SORENSEN and MICHAEL PICHENY
IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

Abstract. We present MARS (Multilingual Automatic tRanslation System), a research prototype speech-to-speech translation system. MARS is aimed at two-way conversational spoken language translation between English and Mandarin Chinese for limited domains, such as air travel reservations. In MARS, machine translation is embedded within a complex speech processing task, and the translation performance is highly effected by the performance of other components, such as the recognizer and semantic parser, etc. All components in the proposed system are statistically trained using an appropriate training corpus. The speech signal is first recognized by an automatic speech recognizer (ASR). Next, the ASR-transcribed text is analyzed by a semantic parser, which uses a statistical decision-tree model that does not require hand-crafted grammars or rules. Furthermore, the parser provides semantic information that helps further re-scoring of the speech recognition hypotheses. The semantic content extracted by the parser is formatted into a language-independent tree structure, which is used for an interlingua based translation. A Maximum Entropy based sentence-level natural language generation (NLG) approach is used to generate sentences in the target language from the semantic tree representations. Finally, the generated target sentence is synthesized into speech by a speech synthesizer.

Many new features and innovations have been incorporated into MARS: the translation is based on understanding the meaning of the sentence; the semantic parser uses a statistical model and is trained from a semantically annotated corpus; the output of the semantic parser is used to select a more specific language model to refine the speech recognition performance; the NLG component uses a statistical model and is also trained from the same annotated corpus. These features give MARS the advantages of robustness to speech disfluencies and recognition errors, tighter integration of semantic information into speech recognition, and portability to new languages and domains. These advantages are verified by our experimental results.

1. Introduction

Robust systems for speech-to-speech translation (SST) have become more and more important for human communication between speakers who do not share a common language. However, construction of such systems is extremely complex and is considered a “grand challenge” of speech and natural language processing. SST involves research in Automatic Speech Recognition (ASR), Text-to-Speech

(TTS), Machine Translation (MT), Natural Language Understanding (NLU), and Generation (NLG). Although substantial progress has been made in each of these components individually over the last two decades, blindly integrating ASR, MT, and TTS components does not yield SST systems with acceptable results. Typical MT technologies and systems, designed for text translation, have not been designed to process text with imperfect syntax, disfluencies, and speech recognition errors that often characterize ASR-transcribed text from conversational speech. Accurate recognition of conversational spontaneous speech is still a major challenge. Spontaneous speech is inherently casual, often not very coherent, contains embedded disfluencies, and can be corrupted by channel or background noise in typical user scenarios. Finally, typical speech recognizers ignore the semantics of the underlying utterance; incorporating such information has the potential to improve performance.

The MARS (Multilingual Automatic tRanslation System) project tries to harness our separately developed technologies for speech recognition and synthesis, NLU and NLG to build a research prototype system that facilitates our exploration of new solutions to some of the unique SST problems mentioned above. Considering the increasing need for communication between Mandarin Chinese and English speaking populations, and the substantial technical challenges resulting from the significant dissimilarity between these two languages, our present focus is on two-way conversational spoken language translation between English and Mandarin Chinese for limited domains, such as air travel reservations.

The MARS system has the following features:

- The translation algorithm uses machine learning to understand and preserve the meaning of the sentence.
- The semantic parser uses a statistical model (in particular, a decision-tree model) and is trained from a semantically annotated corpus for the source language. No hand-crafted grammars or rules are used.
- The output of the semantic parser is used to determine the conversational dialog state; after that, a dialog-state-based language model and a turn-based language model are invoked to refine the speech recognition performance through a rescoring process.
- The NLG component uses a statistical model (in particular, a maximum entropy model), and is also trained from the same type of annotated corpus for the target language.

This paper is organized as follows. In Section 2, we briefly review the background of SST, by reviewing similar projects and approaches that have been used by others, as well as the history of the MARS project. In Section 3, an overview of our system structure is presented, and the advantages of our approach are described. In Section 4, we describe our system components, such as the basic speech recognizer and hypothesis rescoring processor, the NLU analyzer, including the class tagger (classer) and the semantic parser, the innovative statistical NLG approach,

and the TTS component. Section 5 describes the static knowledge sources and training data used in the system. In Section 6, we evaluate our system performance. Finally, Section 7 discusses current and future work.

2. Background

2.1. PRIOR WORK IN SST

There has been a significant amount of research effort in SST in the past. Many different translation approaches have been applied to or developed specifically for the SST problem, as opposed to the written-text translation problem. Many interesting systems have been built to demonstrate the feasibility of the concept. All these projects have constrained task domains.

Among the earliest efforts, C-STAR (http://www.c-star.org) is a research consortium among multi-national research groups pioneered by CMU and ATR that has attracted many groups to it in recent years. One of the translation approaches explored by members of C-STAR is interlingua-based. There, a knowledge representation system, IF, that is independent of any specific language, is used for translation. The interlinguas are tools for representing semantic meaning. With the introduction of an interlingua, the development complexity grows only linearly with the number of languages involved in the translation system. Nevertheless, there are still significant practical issues with the interlingua approach due to the difficulty of designing an efficient and comprehensive semantic knowledge representation formalism (Levin and Nirenburg, 1994). Prototype systems built by partner members of C-STAR, such as CMU (Lavie et al., 1997), ATR (Yamamoto, 2000), ITC-IRST (Lazarri, 2000), CLIPS (Blanchon and Boitet, 2000), were tested in the C-STAR '99 International Experiment.

VerbMobil (Wahlster, 2000) was a high-profile, eight-year effort for speech translation, supported by the German government during the 1990s. Since it involved over 30 research groups in Germany, multiple translation approaches, such as statistical MT, deep linguistic analysis using unification grammar, dialog-act based translation and others were widely investigated for the purpose of speech translation. Among those approaches, a statistical MT system (Brown et al., 1993) which was developed for written text translation, originally by an IBM research group, was for the first time applied to spoken language translation by the group in RWTH Aachen (Ney et al., 2000). This approach has a complex form that uses a channel decoding model that includes a seemingly naive method of word-to-word translation. This method provides a method for MT between arbitrary pairs of languages that is independent of individual language properties; however it requires a large amount of training data in the form of a parallel bilingual corpus and generally ignores any available linguistic knowledge.

Spoken Language Translator (SLT) is another early project in the area of speech translation (Rayner et al., 2000) developed mainly by researchers at SRI International. The main MT engine is a complex unification grammar-based system

intended to perform deep analysis and to produce high quality output. The hand-coded grammars used in SLT are supposedly linguistically motivated and generic in nature. Therefore, the grammar coverage is broad, and it is feasible to use these methods in serious applications, although the method was only evaluated in the domain of air travel reservations.

Finite-state transducers are another statistical and data-driven approach that has also been applied to speech translation recently (Bangalore and Riccardi, 2000; García-Varea et al., 2000; Alshawi et al., 2000). This approach has the potential of integrating speech recognition and MT into one search process by integrating the language model for source-language recognition and the translation model into a single finite-state network. This is in contrast to most other approaches, where speech is first recognized as a sequence of source-language words, and is then translated by a translation model (Casacuberta, 2002). However the size of the integrated finite-state network can grow very fast as the vocabulary size or domain coverage grows.

2.2. OTHER APPROACHES TO NLG

There are many approaches for surface NLG. Templates are the easiest way to generate sentences, but may not scale easily to complex domains in which hundreds or thousands of templates would be necessary. Templates may also have shortcomings as regards maintainability and text quality (Axelrod, 2000). Generation packages, such as FUF (Functional Unification Framework) (Elhadad and Robin, 1992) can be used for more sophisticated speech generation. These packages require sophisticated linguistic input in order to take advantage of the generative power. There are also corpus-based generation systems, for example described in Langkilde and Knight (1998). Purely statistical MT approaches (Berger et al., 1996) can be viewed as a way of generating the target-language sentence directly from the source language without the aid of an explicit semantic representation. The statistical generation approach we used in MARS is based on the approach introduced by Ratnaparkhi (2000) for noun phrase generation from a simple semantic representation: attribute–value pairs.

2.3. HISTORY OF MARS

The IBM SST project was one of the “Adventurous Research” projects in IBM Research, started in 2000 as part of IBM’s long-term commitment to speech and language technologies. The main purpose of this project is to explore new technologies and to attack the challenges present in SST. The MARS system is a prototype system of the project used as a test-bed to show research progress. One of reasons the air travel domain was chosen for MARS is that we have had experience with this task and have many existing components available, allowing us to focus on algorithmic work rather than on domain specific issues.

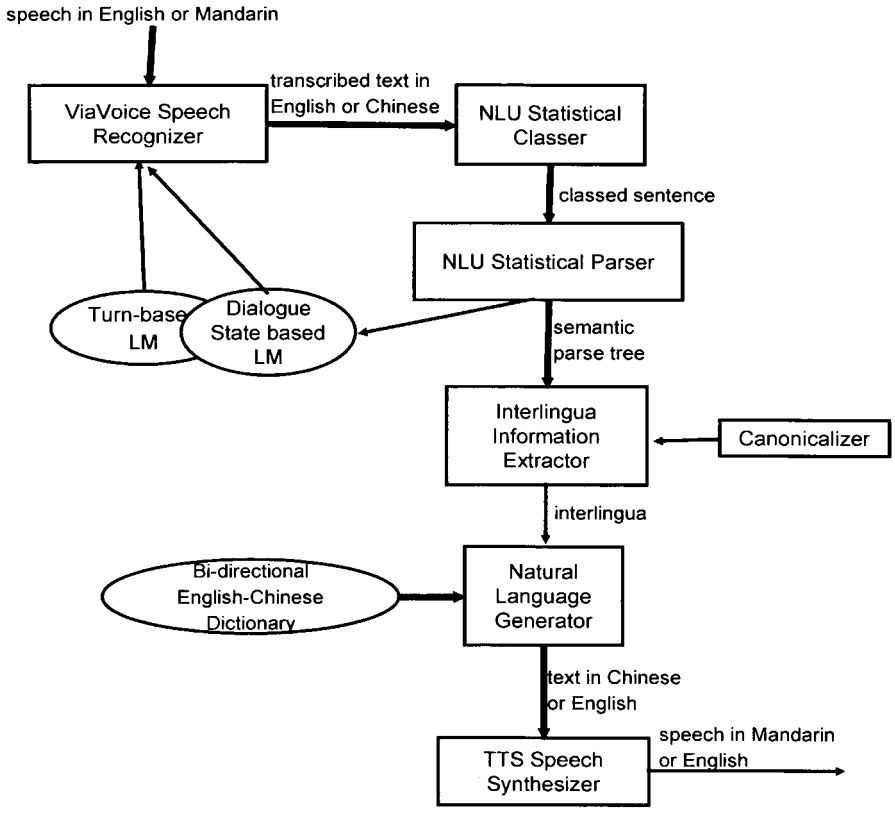


Figure 1. MARS system diagram.

While our new algorithms are formulated particularly for MARS, many of the components and techniques have been developed throughout the years for different projects. The core speech recognition model and engine for both English and Mandarin are inherited from the IBM large-vocabulary dictation system, ViaVoice (Chen et al., 1997; Gao et al., 2000). The NLU engine benefitted from work on the IBM Conversational Telephony system (Davies, 1999), while the NLU component that includes the statistical classer (or class tagger) and semantic parser models for the air travel task were developed for the IBM DARPA Communicator project (Luo et al., 2000; Visweswariah and Printz, 2001), a monolingual conversational dialog system; however, significant modifications were made allowing for the specific needs of the translation task. The TTS component is adopted from the IBM Phrase Splicing and Variable Substitution TTS system (Donovan et al., 1999).

3. System Description

Let us trace through the process steps in the system as indicated in Figure 1, and describe some of the important features of the system.

First, the speech signal is processed and recognized by a speech recognizer. The basic speech recognizer is adopted from the IBM ViaVoice Dictation product for both English and Mandarin Chinese. A language model trained using an air travel reservation domain corpus for English is used instead of the general English dictation language model found in the product.

Next, the ASR-transcribed text of the speech is analyzed by a statistical class tagger (also referred to in this paper as the classer) and a semantic parser. Both the classer and parser utilize statistical decision-tree models originally developed for NLU applications; they use no hand-crafted grammars or rules. Our approach does not include a syntactic component. We bypass an explicit syntactic analysis by using a component that takes the class-tagged sentence and builds a semantic parse tree. The approach emphasizes the preservation and transfer of meaning rather than literal translation. An example of the meaning translation is shown in Section 4.4 and Figures 4 and 5. When the text transcription is parsed, the parser focuses on meaningful phrases and automatically learns to ignore irrelevant portions or out-of-domain concepts. Because our system is trained on a corpus which includes text transcriptions (i.e., by humans) of recorded speech utterances from the task domain (See Section 5.1.2), the semantic parser has some capability to handle disfluencies in conversational speech.

Thirdly, the parser output is used to determine the dialog state, which is then used by a dialog-state-based language model (Visweswariah and Printz, 2001; Gao et al., 2001), and a turn-based language model (Sarikaya, 2002). These models are used to rescore the N-best hypotheses generated by the recognizer, thereby improving the recognition accuracy. In this way the recognizer and the parser are coupled, and the semantic analysis result helps to improve the recognition performance. If the best-path transcription picked by the rescoring process is different from the one generated in the first-pass recognition, the text transcription will again be class-tagged and parsed.

With the help of a canonicalizer and an information extractor, all the expressions in the parse tree are formalized, and the semantic content of the parser output is stored as a tree-structured interlingua-like semantic representation that is independent of the way any particular language expresses meaning, at least for concepts embodied in the limited domain of the application. The information extractor and the canonicalizer use rules to organize and format the data, but these components are the only components that require hand-crafted design. The design of these components is quite simple, relative to the complexity normally associated with rule-based translation systems.

A bi-directional English–Chinese dictionary has been designed to incorporate semantic information and uses the semantic parse context to provide word-sense information for words that have ambiguous meanings in either language. With this dictionary, the canonicalized values associated with the attribute–value pairs in the tree-structured representation can be translated into the target language. However, the attributes themselves may need to be reordered or otherwise changed

to accomplish fluent translation. For this we use a novel maximum entropy (ME) based generation model. The ME-based sentence-level NLG model receives the tree-structured semantic representation and the translated attribute–value pairs, and orders the concepts in each layer of the tree structure. Finally, the generated target sentence is synthesized into the target-language speech by a speech synthesizer.

Although at this stage our prototype system is a two-way English–Chinese translation system, its construction is intended to accommodate multiple languages. Hence we have emphasized the robustness of the overall system architecture, the universality of the methods used in each component, and the complexity of the development effort in terms of the languages involved. Building an equivalent system for another language pair would not require changing the architecture of our system. Furthermore, the same methods we adopt now can be used for any other combination of languages, just as we need only to annotate the domain-dependent text corpus for the new language and supply a minimum amount of language-specific knowledge such as a lexicon. All the components in the NLU analyzer are designed to be language independent.

4. System Components

In this section we describe each of the system components. The training and test data will be described in Sections 5.1 and 5.2.

4.1. SPEECH RECOGNIZER

The original baseline recognizer uses the English and Mandarin speech recognition systems developed for large vocabulary (approximately 64,000 words) continuous speech. The training data for the acoustic model and the language model will be described in Section 5.1. The English speech recognition system uses an alphabet of 52 phones, while the Mandarin system uses 162 phones, including some phones that are tone-dependent (Chen et al., 1997). Each phone is modeled with a 3-state left-to-right Hidden Markov Model. Both systems have approximately 3,000 context-dependent states modeled using 40,000 Gaussian distributions. The context-dependent states are generated using a decision-tree classifier, with a different splitting threshold used for English and Mandarin Chinese to keep the number of context-dependent states in both systems around the same range. The acoustic front-end uses a 24-dimensional cepstral feature vector extracted every 10 ms, which is transformed using linear discriminant analysis. The transformed feature vector is 40-dimensional. For Mandarin Chinese, one element of the 24-dimensional vector is the pitch contour. The out-of-the-box speaker-independent speech recognition accuracy for English and Mandarin is around 82% for a large variety of speakers, including young teenagers and non-native English speakers, and Mandarin speakers with accent.

To improve the recognition accuracy, a language model trained using an air travel domain corpus, EngText1, was substituted for the default general English language model. This improves the baseline recognition performance by a relative 20% (Gao et al., 2001). For the test set, EngSpTest (see Section 5.2), the word error rate is reduced from 17.0% to 13.6%. Unfortunately, for Mandarin Chinese, we do not currently have a similar domain-specific corpus; therefore the general dictation Chinese language model is used. In both cases, our recognizer has the ability to produce N-best hypothesis lists that can be post-processed by the statistical parser.

Another improvement for MARS is that we have also built a dialog-state-based language model (Gao et al., 2001) and a turn-based language model (Sarikaya, 2002) for English. These two models are used in a rescoring process after the text transcription has been parsed. We describe this rescoring process in Section 4.5.

4.2. CHINESE SEGMENTER

There is another issue for Chinese speech recognition: word segmentation. Since there are no word boundaries in written Chinese, a piece of Chinese text usually needs to be segmented into words prior to being processed by a class tagger or a parser. However, in our application, this issue is not as critical as in typical text-based Chinese processing. The text transcription obtained from the Mandarin Chinese recognizer is already segmented into "words" (Chen et al., 1997). The only thing requiring attention is that the segmentation of the text transcription generated by the recognizer may not be consistent with the segmentation of the corpus used for semantic processing. For example, a big recognition unit is often used to define a common expression in Mandarin ViaVoice. For example, 这 'this' 是 'is' is defined as two words in the classer and parser, but ViaVoice outputs them as one unit. When such a unit appears in the text transcription, the Chinese Segmenter needs to post-process the transcription and make sure the words appearing in the transcription are consistent with the words used in the classer and parser. After such post-processing, the consistency of the segmentation is found to be as high as 96%.The post-processing word segmentation is based on a modified version of the automatic segmentation program described in Chen et al. (1997).

4.3. STATISTICAL CLASS TAGGER (CLASSER)

The NLU unit is one of the most important components in our system. The NLU component includes a statistical class tagger (classer) as well as a semantic parser (Magerman, 1994; Davies et al., 1999; Luo and Franz, 2000); both are decision-tree based. The statistical classer examines the text transcription from the speech recognizer and identifies phrases that occur frequently and with high word variability and tokenizes them by replacing them with a class tag. For example, in the air travel domain, frequently occurring phrases such as cities, states, airports, dates, and times are good candidates to be identified and tagged with the class-specific

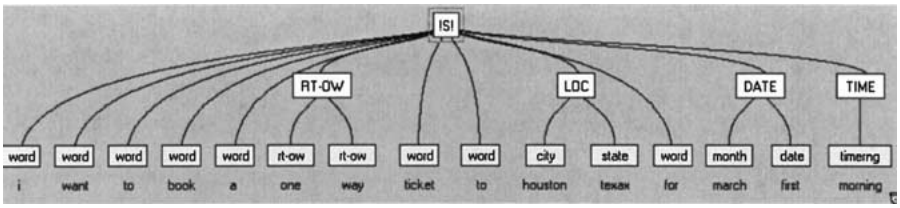


Figure 2. Example of classer annotation.

tokens. For example, the classer will process the following text transcription (1) as in Figure 2.

(1) I want to book a one way ticket to Houston Texas for March first morning

The classer would tokenize *one way* as **RT-OW**, *Houston Texas* as **LOC**, *tomorrow* as **DATE**, and *morning* as **TIME**. Thus, the output of the class tagger will be (2).

(2) I want to book a **RT-OW** ticket to **LOC** for **DATE TIME**

The classer uses 27 distinct tags for the air travel reservation task. Table I shows some examples. The number of distinct tags depends on the complexity of the task domain. For a financial transaction task, a similar number of class tags are used; however, other tasks have required more tags, depending upon the application complexity.

The English classer is trained from a corpus (EngText3) of 37,000 annotated sentences. This allows our system to achieve a performance rate of 96% recall and 95% precision for clean text. With text transcription of speech, the recall and precision rates become 91% and 87%, respectively.

The size of the air travel task corpus for Chinese is much smaller, containing only 2,000 sentences. The Chinese classer is trained only from this small corpus,

Table I. Examples of class tags used in air travel reservation domain

Class Fags	Description
AIR	Airline company name or code
CLASS	Air ticket class
DATE	A traveling date
LOC	A travel location
MEAL	A meal on flight (breakfast, lunch, dinner, snack, special diet meal, etc.)
NUM-FLT	Flight number
RT-OW	Round trip or one way trip
STOPS	Direct flight vs. flight with number of stops

resulting in a lower class tagging accuracy. The recall and precision rates are 89% and 87% for clean text, and 85% and 81% for ASR-transcribed text.

Examples (3)–(5) show how the training corpus affects the design of the class tagger and its performance.

(3) I want to book a one way ticket to Houston.

(4) Is there more than one way to get to Houston?

(5) Can I fly from Dallas to Boston by way of Baltimore?

Obviously, there are sufficient samples of *one way* in the corpus that are in a similar context as in (3), so the *one way* in (3) should be easily tagged as **RT-OW** by the statistical classer model. (4) is a legitimate sentence in the air travel domain. *One way* in (4) can refer to different travel routes or different means of transportation, such as by train, or bus, etc. If such expressions appear in the corpus often enough, for example, with more than ten occurrences for each case, we (as application designers) would have used a distinct tag for each case, and the classer would likely have learned the statistics from the corpus to perform the correct tagging. However, in our corpus (EngText3, 37,000 sentences), the use of *one way* in a similar fashion as the expression in (4) did not appear at all; thus the classer did not learn to perform such tagging. If (4) were used as a test sentence, the classer would either tag it as **RT-OW**, which is wrong, or not tag it at all, indicating that this phrase does not carry a relevant meaning in the task, which is debatable. *By way of* in (5) means 'a stopover in Baltimore'. The corpus includes sufficient samples of such an expression, and all of them are annotated as **STOPS**. Therefore the statistical classer model will tag *by way of* in (5) as **STOPS**.

4.4. STATISTICAL NLU PARSER

The semantic parser examines the class-tagged sentence and determines the meaning of the sentence by evaluating a large set of potential parse trees in a bottom-up left-to-right fashion. The parse hypothesis that scores the highest based on the statistical models is returned as the best parse hypothesis (Magerman, 1994). Figure 3 illustrates an example of the English parse tree. At the top of Figure 3, we show the original English sentence; in the second line (right below the original sentence) the classer output for the sentence is shown. This constitutes the actual input to the semantic parser. The leaf nodes (also call terminal nodes) starting with "%" symbol, such as "%loc-to," in the semantic parse tree are semantic variables; we call them "tags" for the semantic parser. "null" is a special tag used for all words that do not carry relevant semantic meaning for the task domain. We name them "tags" purely for internal use only; they do not have any relation with any other tags. A tag and the word associated with it is an attribute–value pair. For example, "%loc-to" is an attribute, its value is "LOC," which can be traced back from the classer as *San Francisco*. The symbols, such as DATE-DEP and SEGMENT, in the

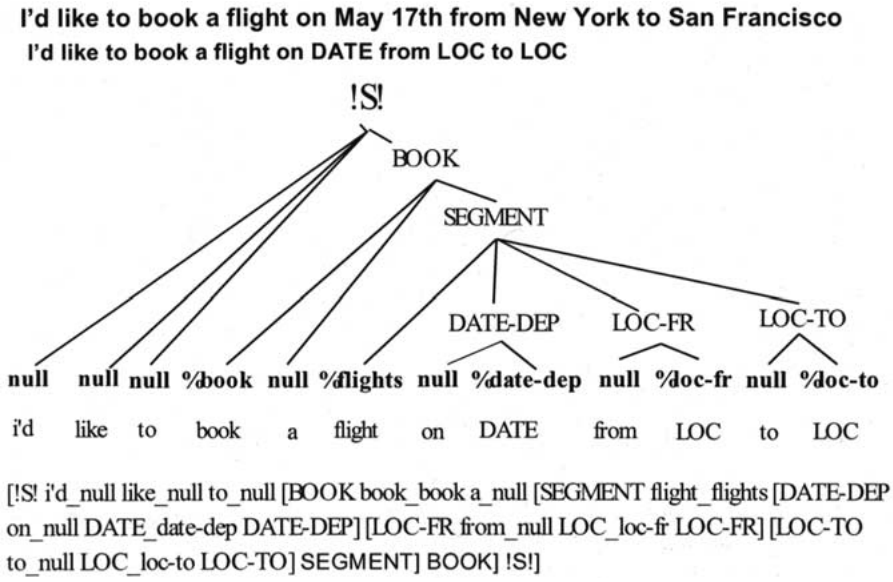


Figure 3. A parse tree example of an English sentence.

Table II. An example of tag-dependent English-to-Chinese dictionary

English	Tag	Chinese
First	%date	一号
First	%class	头等舱
First	%time	第一班, 最早的

nodes of the hierarchical tree are called "labels" (also called non-terminal nodes) internally. The bracketed text at the bottom (below the semantic tree) is a linearized form of the parse tree. Note the linearized form of a parse tree is unique. The number of distinct labels (non-terminal nodes) is about 70; the number of distinct tags (terminal nodes) is about 120.

Only the value part of an attribute–value pair in a semantic tree needs to be directly translated. Therefore tags are used as a semantic information source for the design of a semantic dependent dictionary for this specific task. For example, the word *first* in a date phrase *January first* should be translated differently from the *first* in a flight class phrase *first class seat*, and also differently from the *first* in a time phrase *the first flight in the morning*. The dictionaries used for this air travel translation are designed (see Section 5.3) to be dependent on the tags such as those as illustrated in Table II.

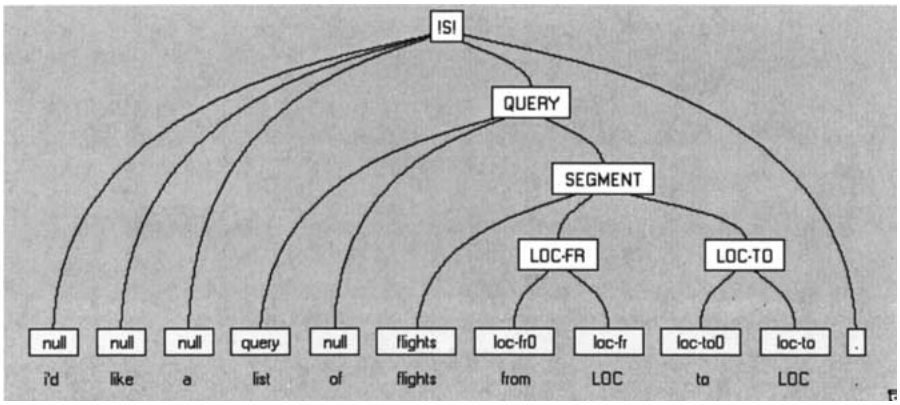


Figure 4. Semantic parse tree for: *I'd like a list of flights from Boston to Denver*.

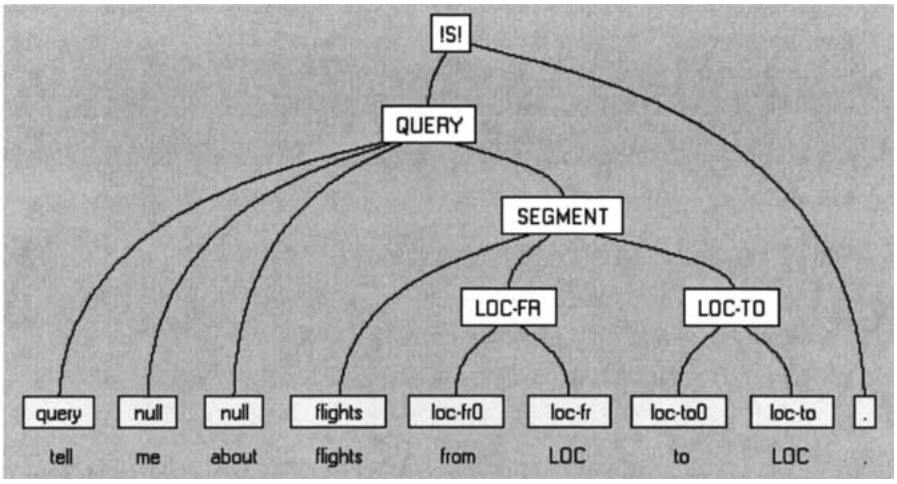


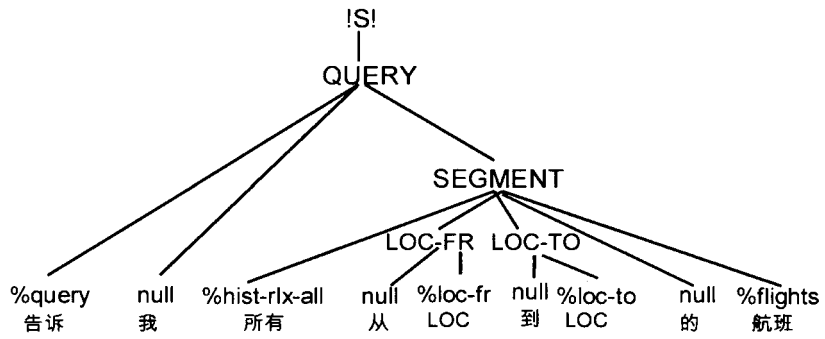
Figure 5. Semantic parse tree for: *Tell me about flights from Boston to Denver*.

The tags primarily supply semantic information for attribute–value pair translation, while the labels and the tree structure supply relations between concepts that are important for sentence generation, which is described on Section 4.8.

A training corpus, EngText4, of 15,000 sentences for the air travel reservation task is annotated, with an example shown in Figure 3, and is used to train the statistical models of the English semantic parser. We achieved a semantic recall rate of 87% and a precision rate of 88% for clean text, and 83% and 82% for ASR transcription.

In Figures 4 and 5, we show that different paraphrases of the same sentence end up with almost the same parse tree, due to the fact that their differences do not carry semantic information. The translation will ignore the differences between the two paraphrases and generate the same translation according to the semantically meaningful parts.

告诉 我 所有 从 达拉斯 到 波士顿 的 航班
Tell me all from Dallas go Boston of flights
告诉 我 所有 从 LOC 到 LOC 的 航班



[QUERY 告诉_query 我_null [SEGMENT 所有_hist-rlx-all [LOC-FR 从_null LOC_loc-fr LOC-FR] [LOC-TO 到_null LOC_loc-to LOC-TO] 的_null 航班_flights SEGMENT] QUERY]

Figure 6. An example of a Chinese sentence parse tree.

Figure 6 shows an example of a Chinese sentence parse tree. At the top of Figure 6, we show the original Chinese sentence. The spaces between words show the segmented transcription obtained from the recognizer. In the second line (right below the original sentence), we show a word-to-word literal English translation from Chinese. In the third line, we show the parser input, where the two words 达拉斯 'Dallas' and 波士顿 'Boston' are tagged as "LOC" by the classer.

The Chinese parser is trained from only a small corpus with 2,000 annotated sentences. As a result, the parser performance is relatively degraded compared with the English parser. The recall and precision rates are 82% and 83% for clean text, and 76% and 77% for ASR transcription.

4.5. SEMANTIC FEEDBACK: RESCORING THE N-BEST HYPOTHESES FROM ASR

While, in principle, using semantic parse tree information to rescore speech recognition results can be done for any language, we have only evaluated this for English because of limitations on available training data.

4.5.1. *A Dialog-State-Based Language Model for Rescoring*

EngText2 is a corpus of 11,000 sentences annotated with dialog state information as indicated in Table III. While the data in this corpus is clearly insufficient to build a complete language model, we believe it can be used to create interpolated models

Table III. Examples of dialog state information in the corpus and the LMs

Dialog State	# of words in training corpus	Dialog-State-Based LM
DATE-ARR	864	Arrival date
DATE-DEP	2527	Departure date
DATE	3585	Arrival or departure date
DONE	335	Confirmation
FROM	1568	Departure location
LIST	5181	Query
LOC	2770	Arrival or departure location
LOC-DATE-DEP	5297	Location and departure date
NONE	36148	Default
NUM-FLY	24	Flight number
TIME-ARR	22	Arrival time
TIME-DEP	3585	Departure time
TIME	4174	Arrival or departure time
TO	1202	Departure location

that will considerably improve recognition performance. Broadly, we consider two ways of using the dialog state information: (1) using the data that we have available for each state, we build a state specific model, which is then linearly interpolated with a general model built from a larger corpus, i.e., EngText1; (2) we also build ME language models using features that indicate dialog state information, which are then combined with a conventional trigram model.

The dialog-state-based language models used to improve speech recognition in MARS are different from the models with the same name described in Visweswariah (2001) and Gao (2001) used for a monolingual human–machine dialog system. In those works the dialog state is determined from a dialog management system and the system prompt, and the dialog-state-based language model is used only to predict the words in the next expected sentence from the user. In contrast, in MARS, the dialog state is obtained from the semantic parse tree and is used to select an appropriate language model for rescoring the N-best hypotheses for the current sentence. In addition, the number of distinct dialog states has increased from 6 to 20, because sentences that combine more than one application concept are now separated into distinct states. Using the dialog-state-based LM rescoring reduces the speech recognition error rate from 13.6% to 12.3% for the test data set, EngSpTest, as described in Section 5.2.

4.5.2. *Turn-Based Language Model for Rescoring*

A turn-based language model, as described in Sarikaya (2002), is also used in the rescoring process to refine the N-best hypotheses. The turn-based language model attempts to model the evolution of the conversation between two speakers. In order to train such models, the training corpus EngText1, is partitioned in terms of the speaking turns to build turn-specific language models. These models are then interpolated with the general models.

Turn-based language models have been found to reduce recognition error rate by 6% relative (Sarikaya, 2002). In MARS, the dialog-state-based language model and the turn-based language model rescoring procedures are combined, and the combined model reduces the error rate further from 12.3% to 12.1%.

After rescoring, the path with the highest score will be processed by the classer and the semantic parser again before being processed further.

4.6. INFORMATION EXTRACTOR

Information extraction is the task of analyzing both the class tagger and semantic parser outputs to extract appropriate information required for subsequent processing. Two associative arrays are used to store the classer and parser results respectively. The first array maps the class tags to the phrases in the original sentence for which they were substituted, such as locations and times. If we take Figure 5 as an example, the first array associates QUERY with 告诉 'tell', LOC with 波士顿, 'Boston' and 达拉斯 'Dallas' and FLIGHT with 航班 'flights', and so on. It should be noted that all phrases tagged with "null" are discarded. The second array maps the concepts contained in the semantic tree to the specific values that appeared in the sentence, including speech actions. In the example of Figure 5, the second array marks the speech action as "Query", maps the SEGMENT to "%hist-rlx-all LOC-FR LOC-TO %flights", and so on. The semantic concept representation, along with the class constituents, captures the information contained in the parsed sentence. This representation allows a direct phrase translation to be applied to the class constituents, which typically can be translated using the dictionary techniques described above. The semantic representation reserves translation of the abstract representation to be performed using a more general translation model. Thus this design should allow for the natural rearrangements that occur when concepts are represented in different languages.

4.7. CANONICALIZER

The canonicalizer is a component responsible for converting the simple classes of phrases, principally those that can be represented as digits, into values that are language independent. For example, the flight number *two eighteen* is converted to the numeric value *218* and the time *eight fifteen* is converted to the standard numeric value *8:15*. This is helpful, for example, because once the flight number

thirteen twenty is converted to *1320*, it can then easily be translated into Chinese as 一三二零. Due to spoken language conventions that allow for flexibility in the way that numbers are spoken, attempting direct literal translation would be very difficult. The canonicalizer subsystem is implemented using the familiar finite-state transducer architecture and requires hand crafting state machines for each language to extract the numerical representation from a chosen class of spoken phrases. Because dates, times, and numbers are typically used in a wide range of applications, the investment in the design of the canonicalization modules can be reused. In fact, in this application, the canonicalization module was derived from rules built for other speech recognition applications, specifically number and date formatting as used in dictation applications.

4.8. NATURAL LANGUAGE GENERATOR

As we briefly mentioned in Section 2.2, very little work has been done using a statistical learning approach to produce natural language text directly from a semantic representation, such as in our case. Ratnaparkhi (2000) introduced a statistical method to generate noun phrases from a simple semantic representation, attribute–value pairs, which is a special subclass of the semantic representation we want to deal with. We have developed our NLG component using a similar approach. The high-level semantic translation is accomplished by NLG in the target language from the semantic representation. More specifically, statistical NLG is used to discover the preferred concept ordering and to assign the lexical form of a grammatical sentence in the target language. The statistical models are directly learned from a training corpus, using no manually designed grammars or knowledge bases. In our speech translation system, the statistical NLG component has three kinds of inputs:

- A set of tree-structured language-independent semantic variables, as shown in Figures 3–6;
- a set of unordered translated attributes in the target language;
- a Probability model for language generation.

During translation, the source sentence is parsed, yielding the constituent structure of the semantic tree that is kept, while the concept ordering information is discarded. The word generation probability model is a maximum likelihood prediction based on maximum entropy modeling (Berger et al., 1996).

4.8.1. *Probability Model*

We use a maximum entropy probability model extended from the "NLG2" model described in Ratnaparkhi (2000). It describes a conditional distribution over $V \cup {}^{*}STOP^{*}$ for the current *symbol* to be generated, where V is the vocabulary of all possible symbols and ${}^{*}STOP^{*}$ is an artificial symbol to mark the end of an independent generation (as illustrated in following examples in model training). In the context of NLG in this paper, symbols refer to the introduction of semantic

concepts or individual target words into the output word sequence. Output begins with a particular sentence type,[1] as identified in the parsed input. Examples in the air travel domain include *Query*, *Action*, *Book*, *Define*, *Orders*, *Cue*, and *Cancel*. By including the sentence or phrase type in our generation scheme, we can narrow the probability space.

The sentence generation is further conditioned based on local *n*-grams and the set of concepts included in the sentence type that have not yet been included in the sentence being generated, using (6)

$$P(s_i \mid T_i, C_i, S_{i-2}, s_{i-1}) = \frac{\prod_{j=1}^{K} \alpha_j^{f_j(s_i, s_{i-1}, s_{i-2}, T_i, C_i}}{\sum_{s' \in V} \prod_{j=1}^{K} \alpha_j^{f_j(s', s_{i-1}, s_{i-2}, T_i, C_i}}, \tag{6}$$

where $\{s_{i-2}, s_{i-1}, s_i\}$ are the previous and current symbols in the generated sequence, T_i is the local sentence or phrase type in the corresponding portion of the semantic tree, C_i is the concept list that remains to be generated at s_i, and f_j is the binary feature that captures the co-occurrence evidence of the current symbol and its contexts (7):

$$f_j(s_i, s_{i-1}, s_{i-2}, T_i, C_i) = \begin{cases} 1 & \text{if } s_i \text{ is the current symbol and} \\ & s_{i-1}, s_{i-2}, T_i, C_i \text{ is true;} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The features are learned from training data to describe the relationship between generation contexts $\{T_i, C_i, s_{i-1}, s_{i-2}\}$ and the generation output s_i. K is the number of binary features observed in the training data, which is on the order of 10,000 for our limited domain. The feature weight α_j captures the influences of each feature, which is estimated from training data to maximize the training data likelihood.

4.8.2. *Model Training*

We employ a two-level training strategy: at the *macro* level the system learns the dominant structure and the connecting words of each sentence, and at the *micro* level the presentation orders of sub-concepts are learned.

For *macro* model training, the generation model is constructed from the training data that was annotated with semantic parser level annotations. So the corpus for macro model training for generating English is EngText4. For example, from the annotated parse tree example shown in Figure 3 for sentence (8), the *macro* training phrases shown in Table IV are derived, one for each level in the parse tree.

(8) I'd like to book a flight on May 17th from New York to San Francisco

Table IV. Training examples for macro model training

!S!	I'd like to BOOK *STOP*
BOOK	%book a SEGMENT *STOP*
SEGMENT	%flights DATE-DEP LOC-FR LOC-TO that %stop LOC-STP *STOP*
LOC-FR	from %loc-fr *STOP*
LOC-TO	to %loc-to *STOP*

The *micro* model is trained using the lower-level annotations in class tagging as described in Section 4.3, whose training corpus is EngText3. Thus from the Classer annotation shown in Figure 2 for sentence (9), the following training phrases for *micro* model training for English generation are extracted:

(9) I want to book a one way ticket to Houston Texas for March first morning.

- LOC: %city %state *STOP*
- DATE: %month %date *STOP*

In both macro and micro training, every training phrase starts with the sentence or phrase type to be generated, and lists the concepts and linking words in the *correct* order. In addition, to learn explicitly when the phrase is over, we have artificially added the symbol **STOP** to each training phrase.

The training procedures for the two sets of models are exactly identical; both include extracting features from training sentences and estimating feature weights. A vocabulary of 589 symbols, which include all the un-tagged words (e.g., *I'd*, *like*) all the tags (e.g., %book, %flights) and labels (e.g., LOC-FR, DATE-DEP) used in the parser annotation, is extracted for macro training from the annotated corpus. 33,287 features as defined in (7) are extracted from the same corpus.

The vocabulary of symbols for micro training includes only all un-tagged words and low-level class tags. The size is 408. The number of features is about 20,312.

For Chinese generation, the corpus ChText (2,000 annotated sentences) is used. There are 495 symbols and 10,522 features for macro training and 382 symbols and 8,932 features for micro training.

The feature weights are estimated through the *Improved Iterative Scaling* algorithm (Berger et al., 1996).

4.8.3. *Generation Search*

A recursive search based on the semantic tree structure of the input sentence is used to generate the word sequence in the target language. The generation procedure is described by the following pseudo code:

Traverse the semantic tree in a bottom-up fashion;
For each terminal concept that comprises more than two sub-concepts {
Search for the qualified symbol sequence according to the micro *model;*
}
For each non-terminal in the semantic tree {
Search for the qualified symbol sequence according to the macro *model;*
}
Link phrases to a sentence by another traversal of the semantic tree;
Substitute concept variables with their translated attributes.

Each single round of search in the loops of this procedure is similar to the one described in Ratnaparkhi (2000): "qualified" symbol sequences are obtained through a left-to-right breadth-first search in the symbol space. By "qualified", we mean those symbol sequences that survive the search and meet the following requirements:

- Mention all concepts in the semantic tree once and only once,
- end with $^*STOP^*$,
- have a total number of symbols less than an upper limit.

More specifically, at each index, for each active symbol s_n, we score its symbol path $s_1 s_2 \ldots s_{n-1}$ as in (10)

$$p(s_1 s_2 \ldots s_n \mid tree) \prod_{i=1}^{n} p(s_i \mid T_i, C_i, s_{i-1}, s_{i-2}). \tag{10}$$

During each round of search, as with N-best search in Viterbi decoding, only the top N best scored symbols are kept active at each time, with new symbols expanded from them. The use of N-best search allows for additional linguistic constraints to be applied. For example, one limitation of the use of "symbolic" trigrams is that the evaluation of the likelihood of a symbol sequence is done at the symbol level rather than at the word level. The generation scheme also assumes that symbols generated from different parents are independent. As a result, two speech acts "BOOK" and "SEGMENT" may be pieced together through the search as a candidate with high probability, with corresponding constituents *I want to book a SEGMENT* and *flights flying from Boston to Denver*. However, linking them together results in the ill-formed sentence (11).

(11) *I want to book **a flights** flying from Boston to Denver.

To address the singular *a* and the plural *flights*, we may re-score the N-best generated sentences with additional linguistic knowledge. One such simple re-scoring can be achieved by using pure word n-grams to re-rank the generated N-best sentences after attribute substitution. Alternatively, models that explicitly check for subject–verb agreement, plural–singular agreement and other such linguistic

properties may be used in future enhancements of this algorithm to penalize the N-best results that fail these tests.

4.8.4. *Robustness for OOT Questions*

As with many applications of statistical models, we will encounter Out-Of-Training (OOT) questions during the search stage due to the inevitable mismatch between the training set and the target semantic representations. OOT happens whenever the generation context, the 4-tuple $\{T_i, C_i, s_{i-1}, s_{i-2}\}$, has never appeared in the training data. This usually results when the concept list C_i has never been observed for phrase type T_i in the training data. If some remaining concepts are missing from T_i, some information may be lost in the generation result; if all remaining concepts are missing from the current T_i, then no phrase could be generated! In summary, OOT can occur for the following reasons:

1. Inconsistent annotations used in source and target language training data
2. Incorrect parser result
3. Radical source/target language difference
4. Scarce training data.

Among them, the first factor could be avoided by uniform annotation, but the last three could not be avoided entirely. We claim that some of the OOT questions can be handled within our NLG scheme by using back-off techniques. Our first strategy is to employ a bottom-up scheme to traverse the semantic tree and to generate lower-level phrases first. In this way, we can ensure local OOT will not affect the overall generation procedure. Our second strategy is to modify iteratively the semantic tree. During the bottom-up traversal of the semantic tree, we take the following steps whenever we encounter an OOT event:

1. If there is a single concept remaining, remove its parent (since this may indicate that the parser result is wrong) and promote this concept to a higher level
2. If a concept in a multi-concept list is OOT, promote this OOT concept to a higher level, and generate the local phrase after removing the OOT concept
3. If it reaches the root, a partial sentence is generated by the well-formed part of the semantic tree, and the remaining OOT concepts are translated into one or more fragments using a backup word-to-word translation procedure.

4.9. SPEECH SYNTHESIZER

In the final step, the translated Mandarin or English sentence is converted into speech. In order to generate high-quality synthetic speech for limited domains with minimal training data, a trainable, phrase-slicing and variable substitution synthesis system (Donovan et al., 1999) is used. This technology offers an intermediate form of automated speech production lying between the extremes of recorded utterance playback and full text-to-speech synthesis. The system incorporates a trainable

speech synthesizer and an application-specific set of pre-recorded phrases. The synthesis inventory is augmented with the synthesis information associated with the pre-recorded phrases used to construct the phone sequences. The synthesizer then performs a dynamic programming search over the augmented inventory to select a segment sequence to produce the output speech. The system enables the seamless splicing of pre-recorded phrases both with other phrases and with synthetic speech. It also enables very high quality speech within a limited domain.

It should be noted that sometimes the translation output is in the form of mixed source and target languages. This will be observed, for example, when untranslatable proper names of locations exist in source utterances. However, we stress that our text-to-speech system has the unusual ability to generate speech across language boundaries seamlessly.

5. Static Knowledge Sources, Training and Test Data

5.1. DATA USED FOR COMPONENT TRAINING

5.1.1. *Speech Data*

The acoustic models of speech recognition systems for English and Mandarin are trained from two large corpora, one for English, and another for Mandarin. Each includes over 200 hours of speech collected from approximately 2,000 speakers. The English corpus is mainly from US English speakers with a small portion of non-native speakers. The range of the speakers' ages is broad, from young teenagers to elders over 60 years old. The Mandarin speakers include those with strong provincial accents. Both of these corpora were collected for the ViaVoice product.

5.1.2. *Text Data*

For English, we have a corpus of over 100,000 sentences in the air travel domain collected over several years for the ATIS and DARPA Communicator projects (Luo et al., 2000; Visweswariah et al., 2001); we call the combined corpus EngText1. A portion of EngText1, about 15,000 sentences, is human transcription of live speech data recorded from an IBM telephone-based system built for the air travel task. This corpus is used to train the general language model and the turn-based language model for the air travel domain. A subset of EngText1, including about 11,000 sentences that we call EngText2, is marked with dialog state information, and is used to train dialog-state-based language models. Another subset of EngText1, about 37,000 sentences, including the transcription portion from speech data in EngText1, is called EngText3; this subset is annotated with class tags and used for classer training and micro level NLG model training. A subset of EngText3, about 15,000 sentences, is named EngText4 and is annotated with semantic parser level annotation as shown in Figures 3–5. EngText4 is used to train the statistical semantic parser and maximum entropy NLG model (macro model) for English.

The corresponding Chinese corpus, ChText, has only 2,000 sentences for the domain. It was annotated for both classer and parser annotation and used for training the Chinese classer, parser, and NLG models.

5.2. TEST DATA

A small amount of text and speech data is collected specifically for evaluating MARS. A script (EngTextTest) of 100 sentences for the air travel task has been collected in English and is then manually translated into Chinese (ChTextTest). These two text data sets, EngTextTest and ChTextTest, are annotated carefully according to the classer and the semantic parser annotation guidelines, as described in Sections 4.3 and 4.4, and are used to test the performance of the classer, the parser, and the text-to-text translation for both language directions (English → Chinese and Chinese → English).

For the purpose of automatically evaluating the translation performance, EngTextTest and ChTextTest are again translated back into their original languages by two separate human translators: ChRef1 and ChRef2, and EngRef1 and EngRef2.

Two English speakers (1 male, 1 female) and two Mandarin speakers (1 male, 1 female) are recorded reading the scripts EngTextTest and ChTextTest in a natural conversational manner. This speech test data, EngSpTest and ChSpTest, was used to test ASR performance when different language models are used. The ASR transcriptions of the speech are also used to test the classer and parser performance, and the overall speech-to-text translation performance.

5.3. ENGLISH–CHINESE DICTIONARY

In MARS, only the named attributes associated with the leaf nodes in the semantic parse tree need to be translated in the traditional sense of word-for-word translation. This is currently performed using language-to-language dictionaries. In cases where phrases or words may be ambiguous, a tag-specific translation dictionary is created. That is, a phrase or a word may have different translations when considered generally, but usually not within a specific semantic context. The two-way English–Chinese tag-dependent dictionaries are generated in three steps:

1. Extract all the words that are annotated with a specific tag in the corpus. Do this for all the tags and for both languages.
2. Extract the translations for all the words obtained in step (1) from a general electronic dictionary.
3. Remove all the irrelevant (out-of-domain) translations for each word and each tag.

The final result is a tag-dependent dictionary, where each word associated with a particular tag has a unique translation.

The first two steps can be done automatically. The third step currently requires manual checking.

6. Evaluation

Defining a useful performance metric for a translation system is a challenging problem in itself. One such metric, Bleu, recently proposed by our colleagues (Papineni et al., 2001) for use in text-to-text MT, was used for evaluating our MARS system. The Bleu metric requires human translated scripts as reference; in the examples shown below, the modified unigram precision (MUP) measure, the simplest version of the Bleu score, is used for illustration. Bleu scores based on higher order *n*-grams are used for evaluating the full test sets. The capability to handle the special issues related to spontaneous speech recognition errors and disfluencies is also examined.

We have compared the performance against the online BabelFish system (http://babelFish.altavista.com/tr).

In the following examples, for every Chinese sentence, a word segmented version with word-to-word English translations is printed on the next line.

Example 1 (from English to Chinese):
Original sentence (Perfect transcription from EngTextTest):
I would like to go to Boston tomorrow morning around 9 o'clock.

Human translation 1 (from ChRef 1):
我要明天大约上午九点去波士顿
我(I) 要(want) 明天(tomorrow) 大约(about) 上午(morning) 九点(nine o'clock) 去(go) 波士顿(Boston).

Human translation 2 (from ChRef 2):
我希望明天上午九点左右去波士顿.
我(I) 希望(wish) 明天(tomorrow) 上午(morning) 九点(nine o'clock) 左右(about) 去(go) 波士顿(Boston)

MARS translation:
我要订明天大约上午九点去波士顿的航班.
我(I) 要(want) 订(book) 明天(tomorrow) 大约(about) 上午(morning) 九点(nine o'clock) 去(go) 波士顿(Boston) 的(of) 航班(flight)

BabelFish translation:
我希望明早去到波士顿大约九时.
我(I) 希望(wish) 明早(tomorrow morning) 去(go) 到(to) 波士顿(Boston) 大约(about) 九时(nine hour)

MUP(MARS) = 8/10 (It is 8/10 when compared to Ref 1, and 6/10 compared to Ref 2. Take the maximum of them.)
MUP(BabelFish) = 5/8 (4/8 to Ref 1, 5/8 to Ref 2)

Compared to BabelFish, the MARS translation not only has higher MUP, but also has better word ordering and lexicon choice. The phrase *around 9 o'clock* has been moved to before *go to Boston*, which is the correct Chinese order. The word *o'clock* has been translated to 点 rather than 时 'hour' like in BabelFish. The former is a better choice than the latter for spoken language.

Another feature is that in the MARS translation, domain-specific words, 订 'book', and 航班 'flight', are added into the translation, which makes the MUP lower than it should be. However, it makes the translation better suited for the specific domain.

When we use ASR-transcribed text, which includes speech recognition errors and disfluencies as input to the translation systems, MARS outperforms BabelFish even more substantially. Below we show a transcribed text sentence, which includes two recognition errors and two disfluencies that are indicated by underlines. MARS completely ignores the errors and the disfluencies, and generates the same translation as if the actual spoken sentence was input, while BabelFish cannot handle disfluencies at all and also makes a lexical choice that is inappropriate for the domain.

Transcribed sentence (from EngSpTest1):
I *am delighted* to go *um* to Boston tomorrow morning *uh* 9 o'clock.

MARS translation: 我要订明天大约上午九点去波士顿的航班.
我(I) 要(want) 订(book) 明天(tomorrow) 大约(about) 上午(morning) 九点(nine o'clock) 去(go) 波士顿(Boston) 的(of) 航班(flight)

BabelFish translation:
我欢欣明早去 um 波士顿 uh 九时.
我(I) 欢欣(delighted) 明早(tomorrow morning) 去(go) um 波士顿(Boston) uh 九时 (nine hour)

MUP(MARS) = 7/10, MUP(BabelFish) = 10/5, when the same human translations are used as references.

Example 2 (from Chinese to English):

Original sentence (from **ChTextTest**)
星期六从波士顿飞往匹兹堡的航班都有哪些？
星期六(Saturday) 从(from) 波士顿(Boston) 飞往(fly to) 匹兹堡(Pittsburgh) 的(of) 航班 (flight) 都(all) 有(have) 哪些(which ones)

Human translation 1 (ChRef 1):
Which flights are flying from Boston to Pittsburgh on Saturday?

Human translation 2 (ChRef 2):
What are the flights from Boston to Pittsburgh on Saturday?

MARS translation:
Which ones flight from Boston to Pittsburgh on Saturday?

BabelFish translation:
Which Saturday flies to Pittsburgh from Boston the scheduled flight all has?

MUP(MARS) = 8/9
MUP(BabelFish) = 7/12

Example 3.

Original (from EngTextTest):
How much is a first class ticket from Baltimore to San Francisco

Human translation (from ChRef1)
从巴尔的摩去旧金山的头等舱机票是多少
从(from) 巴尔的摩(Baltimore) 去(go) 旧金山(San Francisco) 的(of) 头等舱(first class) 机票(ticket) 是(is) 多少(how much)

MARS translation:
从巴尔的摩出发去三藩市头等舱的机票多少钱?
从(from) 巴尔的摩(Baltimore) 出发(depart) 去(go) 三藩市(San Francisco) 头等舱(first class) 的(of) 机票(ticket) 多少(how much) 钱(money)

BabelFish translation:
多少是第一张类票从巴尔的摩向旧金山
多少(how much) 是(is) 第一(first) 张(piece) 类(class) 票(ticket) 从(from) 巴尔的摩(Baltimore) 向(towards) 旧金山(San Francisco)

MUP(MARS) = 7/9, MUP(BabelFish) = 6/9

Although the MUPs for MARS and BabelFish are very close, BabelFish makes wrong lexical choices for such words as *first class*, and *ticket*. The order for the translation for *how much* is also wrong.

Table V shows the summary of all the tests for MARS. For English, word recognition error rate (WER) is used to measure recognition performance, while for Mandarin Chinese, character recognition error rate (CER) is used to avoid the

Table V. Summary of test results for MARS

	ASR	Text-to-text (Bleu 4-gram)	Speech-to-text (Bleu 4-gram)
E → C	WER: 12.1%	0.52	0.37
C → E	CER: 17.4%	0.43	0.28

segmentation inconsistency. The CER is measured before the Chinese Segmenter is applied.

We did not run a systematic test on BabelFish to compare with MARS. However, from the three examples we showed above and many test examples we ran on-line with BabelFish, we believe MARS outperforms BabelFish for the specific task. In any case, BabelFish is a general translation system that works for a much broader domain, while MARS is domain specific, so a comparison between these two systems is clearly not significantly meaningful.

7. Conclusion and Future Work

Our system utilizes a semantic interpretation-based MT technique. With this approach, we apply statistical methods for many of the key components within the system. Hence our system should easily accommodate other domains and languages. Although our prototype system is still in the nascent stage, we are confident about its potential and will continue to work refining the individual components. We believe MARS provides a reasonable decomposition of the translation task that does not suffer from the limitations seen when ASR, MT and TTS systems are combined independently. Our experiences with dissimilar languages suggests that languages that are more closely related may perform even better within our framework.

Since this is a very preliminary prototype system, there are many research and implementation issues that need to be addressed in future. It is impossible to list all the possibilities to improve the current system, but we describe some areas below:

- Although the IBM ViaVoice Dictation system suffices as a dictation system, it still lacks some functions important for the SST task, such as utilizing prosody and intonation information. One apparently serious problem is that in spoken language, people often use exactly the same sentence to express different or even opposite meanings, just by using different tones and intonations. Better prosodic models will help distinguish such ambiguity.
- In the future, domain-specific acoustic and language models should be used to reduce speech recognition error rates.
- Other objective and subjective evaluation methods (Nießen et al., 2000) should be studied and compared with the Bleu metric.

- Better evaluation metrics suitable for domain-specific, SST should be developed. They should be able to measure the success rate in terms of concepts conveyed, robustness to speech recognition errors and disfluencies, and domain-dependent lexical choices, etc.
- More training data should be used for Chinese NLU and NLG components in order to achieve better performance.
- The NLU annotation system, which we inherited from our DARPA Communicator project, needs to be further modified to better fit the translation purpose.

Acknowledgements

The authors thank Drs Xiaoqiang Luo, Adwait Ratnaparkhi, Wei-Jing Zhu, Hakan Erdogan, Ruhi Sarikaya and other members in the speech-to-speech translation and NLU groups at IBM Research for their help in this work. The authors also thank the *Machine Translation* Special Issue editors and two reviewers for their careful review and useful suggestions. Special thanks to our colleague Jeff Kuo for proofreading our manuscript.

Notes

[1] We will use the terms "speech acts" and "sentence types" interchangeably in this paper.

References

Alshawi, H., S. Bangalore, and S. Douglas: 2000, 'Head-Transducer Models for Speech Translation and Their Automatic Acquisition from Bilingual Data', *Machine Translation* **15**, 105–124.

Axelrod, S.: 2000, 'Natural Language Generation in the IBM Flight Information system', in *ANLP/NAACL 2000 Workshop, Conversational Systems*, Seattle, Washington, pp. 21–26.

Bangalore, S. and Riccardi, G.: 2000, 'Stochastic Finite-State models for Spoken Language Machine Translation', in *ANLP/NAACL 2000 Workshop, Embedded Machine Translation Systems*, Seattle, Washington, pp. 52–59.

Berger, A. L., S. A. Della Pietra, and V. J. Della Pietra: 1996, 'A Maximum Entropy Approach to Natural Language Processing', *Computational Linguistics* **22**, 39–71.

Blanchon H. and C. Boitet: 2000, 'Speech Translation for French within the C-Star II Consortium and Future Perspectives', in *International Conference on Spoken Language Processing 2000 (ICSLP 2000)*, Beijing.

Brown, P. F., S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer: 1993, 'The Mathematics of Statistical Machine Translation: Parameter Estimation', *Computational Linguistics* **19**, 263–311.

Chen, C. J., R. A. Gopinath, M. D. Monkowski, M. A. Picheny, and K. Shen: 1997, 'New Methods in Continuous Mandarin Speech Recognition', in *EuroSpeech '97: 5th European Conference on Speech Communication and Technology*, Rhodes, Greece, pp. 1543–1546.

Davies, K.: 1999, 'The IBM Conversational Telephony System for Financial Applications', in *Proceedings of EuroSpeech'1999*, Budapest, pp. 275–278.

Donovan, R. E., M. Franz, J. S. Sorensen and S. Roukos: 1999, 'Phrase Splicing and Variable Substitution Using the IBM Trainable Speech Synthesis System', in *1999 IEEE International*

Conference on Acoustics, Speech, and Signal Processing, ICASSP99, Phoenix, Arizona, pp. 373–376.

Elhadad, M. and J. Robin: 1992, 'Controlling Content Realization with Functional Unification Grammars', in R. Dale, E. Hovy, D. Rosner and O. Stock (eds), *Aspects of Automated Natural Language Generation*, Springer, Berlin, pp. 89–104.

Gao, Y., B. Ramabhadran, and M. Picheny: 2000, 'New Adaptation Techniques for Large Vocabulary Continuous Speech Recognition', in *ISCA ITRW ASR2000 Automatic Speech Recognition: Challenges for the New Millenium*, Paris.

Gao, Y., H. Erdogan, Y. Li, V. Goel, and M. Picheny: 2001, 'Recent Advances in Speech Recognition System for IBM DARPA Communicator', in *Eurospeech 2001 – Scandinavia: 7th European Conference on Speech Communication and Technology*, Aalborg, Denmark, pp. 503–506.

García-Varea, I., A. Sanchis, and F. Casacuberta: 2000, 'A New Approach to Speech-Input Statistical Translation', in *ICPR 2000: 15th International Conference on Pattern Recognition*, Barcelona, Vol. 3, pp. 907–910.

Langkilde, I. and K. Knight: 1998, 'Generation that Explores Corpus-Based Statistical Knowledge', in *COLING-ACL '98: 36th Annual Meeting of the Association of Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Quebec, pp. 704–710.

Lavie, A., A. Waibel, L. Levin, M. Finke, D. Gates, M. Gavaldá, T. Zeppenfeld, and P. Zhan: 1997, 'JANUS-III: Speech-to-Speech Translation in Multiple Languages', in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)*, Munich, Germany, Vol. 1, pp. 99–102.

Lazzari, G.: 2000, 'Spoken Translation: Challenges and Opportunities', in *International Conference on Spoken Language Processing 2000 (ICSLP 2000)*, Beijing.

Levin, L. and S. Nirenburg: 1994, 'The Correct Place of Lexical Semantics in Interlingual MT', in *COLING 94: The 15th International Conference on Computational Linguistics*, Kyoto, pp. 349–355.

Luo, X. Q. and M. Franz: 2000, 'Semantic Tokenization of Verbalized Numbers in Language Modeling', in *International Conference on Spoken Language Processing 2000 (ICSLP 2000)*, Beijing.

Magerman, D. M.: 1994, 'Natural Language Parsing as Statistical Pattern Recognition', Ph.D. thesis, Stanford University.

Ney, H., S. Nießen, F. J. Och, H. Sawaf, C. Tillmann, and S. Vogel: 2000, 'Algorithms for Statistical Translation of Spoken Language', *IEEE Transactions on Speech and Audio Processing* **8**, 24–36.

Nießen, S., Och, F. J., Leusch, G., and Ney, H.: 2000, 'An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research', in *Second International Conference on Language Resources and Evaluation LREC-2000*, Athens, Greece, pp. 39–45.

Papineni, K., S. Roukos, T. Ward, and W-J. Zhu: 2001, 'BLEU: A Method for Automatic Evaluation of Machine Translation', IBM Research Report, RC22176, Thomas J. Watson Research Center, Yorktown Heights, NY, September 2001.

Ratnaparkhi, A.: 2000, 'Trainable Methods for Surface Natural Language Generation', in *1st Meeting of the North American Chapter of the Association for Computational Linguistics*, Seattle, Washington, pp. 194–201.

Visweswariah, K. and H. Printz: 2001, 'Language Models Conditioned on Dialog State', in *Eurospeech 2001 – Scandinavia: 7th European Conference on Speech Communication and Technology*, Aalborg, Denmark, pp. 251–254.

Wahlster, W. (ed.): 2000, *Verbmobil: Foundations of Speech-to-Speech Translation*, Springer, Berlin.

Yamamoto, S.: 2000, 'Toward Speech Communications Beyond Language Barrier – Research of Spoken Language Translation Technologies at ATR', in *International Conference on Spoken Language Processing 2000 (ICSLP 2000)*, Beijing.