A Flexible Speech to Speech Phrasebook Translator

Manny Rayner
Netdecisions Technology Centre
Westbrook Centre
Milton Road
Cambridge CB4 1YG
United Kingdom

Manny.Rayner@netdecisions.com

Pierrette Bouillon
University of Geneva
TIM/ISSCO
40, bvd du Pont-d'Arve
CH-1211 Geneva 4
Switzerland

Pierrette.Bouillon@issco.unige.ch

Abstract

We describe a simple speech translation architecture intended for medical and other safety-critical applications, which is intended to represent a compromise between fixed-phrase translation on one hand and complex transfer-based translation on the other. Recognition is guided by an annotated CFG-based language model compiled from a unification grammar; transfer and generation use a minimal list-oriented semantic representation language. We present an evaluation of an initial prototype, which translates yes/no questions about hypoglycaemia from spoken French into spoken English using a vocabulary of about 200 words.

1 Introduction

Simplifying a little, there are right now four main types of speech translation system:

1. Elaborate research systems like VERBMOBIL (Wahlster, 2000) and Spoken Language Translator (Rayner et al., 2000). These systems are typically based on complex methods involving both rule-based and statistical processing, and deliver high-quality performance on unseen corpus data. They represent the result of many person-years of work and several million dollars of investment, and run on bespoke software and hardware platforms.

2. Trainable domain-specific systems, which induce domain-specific rules from moderate amounts of corpus data, e.g. (Frederking et al., 1997; Alshawi et al., 2000); speech recognition is typically performed using some kind of statistical language model. Systems of this kind are robust and can be constructed and deployed quickly.

3. Systems which combine domain-independent recognition and domain-independent translation, e.g. (FlexiPC, 2002). These systems are easy to construct, and appear to be useful for at least some real-world tasks. Given their basic architecture, however, their accuracy and robustness tend to be at the low end of the scale, and there is no straightforward way to adapt them to a specific domain.

4. Fixed-phrase translators, e.g. (Integrated-WaveTechnologies, 2002). These systems are by their nature completely accurate, and can be deployed on lightweight platforms, in particular on wearable computers. They are however by the same token completely inflexible, and are only suitable for specialised niche applications.

We do not wish to argue that any of the approaches above are inherently misguided; in a long term perspective we believe the first kind of approach is in fact probably the correct one. If, on the other hand, we are more interested in what can be done with today's technology, there are many kinds of important practical applications which do not appear to be well suited to

Proceedings of the Workshop on Speech-to-Speech Translation:
Algorithms and Systems, Philadelphia, July 2002, pp. 69-76.
Association for Computational Linguistics.

any of the paradigms we have mentioned. In this paper, we will be particularly interested in medical speech translation applications. There are good reasons to pay attention to medical domains; when we have talked to people about possible practical applications of speech translation technology, medical applications probably come up as frequently as all other application areas put together. The explanation is simple: there are no other areas where the potential payoff is as large, or as immediate. It requires little imagination to think of scenarios, not even particularly far-fetched ones, where access to a reliable medical speech translator could actually save someone's life. Anyone who has suddenly fallen ill in a country where they do not speak the local language will be aware that this is, for good reasons, an extremely frightening situation to be in. In most application areas, users are reluctant to use speech translation technology which is still far from perfect. With medical applications, many of these objections disappear; when their lives are at stake, people don't tend to be fussy.

Given the above, one might wonder why medical speech translation systems are not already in common use. The answer is again clear. For exactly the same reasons, a medical speech translation system must be totally dependable; particularly in the US, the likely consequences of an accident resulting from a mistranslation are enough to convince most medical professionals that they should only use systems which they can trust completely.

Unfortunately, this immediately rules out the first three architectures considered above. The completely domain-independent systems (3) are in no way reliable enough for this kind of work, and the inherent uncertainty involved in any kind of statistical approach makes it hard to believe that even a good domain-specific trainable system (2) would be regarded as acceptable. Elaborate linguistics-based systems (1) are better in this respect, but still not good enough; for example, the evaluation figures for the final version of the Spoken Language Translator (Carter et al., 2000) show that even correctly recognised utterances give bad or useless translations about 4% of the time. Having worked extensively with such systems, our strong impression is that a non-trivial proportion of bad translations is inescapable given today's methods. The key problem is that rule-based systems of this kind always permit fairly extensive ambiguity, which is in practice resolved using potentially fallible techniques of a statistical or heuristic nature. In addition to the question of reliability, it should be added that the extreme expense associated with developing and porting these systems would be a major practical obstacle to deploying them for the large number of sub-domains and language pairs required in practice.

Fixed phrase translators, naturally, do not suffer from the above problems, and can be quite successful in some safety-critical domains. For example the system described in (IntegratedWaveTechnologies, 2002), which translates about 500 fixed phrases, has apparently been used in real situations by the Oakland Police Force. For medical domains, however, a fixed phrase translator appears to be too restrictive. A doctor doesn't simply want to ask whether the patient experiences a certain symptom; they typically need to ask whether they experience it seldom or often, whether they experience it at certain times of day or in connection with specified other activities, and so on. This can in principle be done using a fixed phrase translator, but a little experimentation shows that the resulting dialogues tend to be unbearably slow and frustrating for both partners. ("Do you suffer from headaches?" ... "Often, sometimes, or only occasionally?" ... "After a meal?") Splitting up questions in this way can also introduce misunderstandings; for example, a patient may give a negative answer to "Do you suffer from headaches?", but a positive answer to "Do you *occasionally* suffer from headaches?"

None the less, it seems clear to us that the fixed-phrase approach is the one that comes closest to delivering what the users actually want. In the sequel, we will describe an architecture for what could be called a second-generation fixed-phrase translator; essentially, it is a phrasal translator which allows some variation in the in-

put language. This is close in spirit to the approach used in most normal phrase-books, which typically allow "slots" in at least some phrases ("How much does — cost?"; "How do I get to — ?"). To elaborate, our architecture is motivated by the following main considerations:

1. The system should run on standard platforms, and be easy to install and use.

2. The architecture should support rapid development of versions for new domains, subdomain and language-pairs; in particular, it should be easy to add new output languages.

3. It should be possible to develop a system *without* having a sizeable corpus — in practice, there never is a suitable corpus available.

4. Translation must be simple enough to be totally reliable. This will never be the case with statistical methods — hence the architecture must be based on rule-based linguistic methods, preferably simple ones. In particular, the architecture must as far as possible reduce both the complexity of internal representations, and their potential ambiguity.

The rest of the paper describes a concrete architecture motivated by the above considerations. Examples will be taken from our pilot application, a French-to-English phrasebook-style translator with a vocabulary of about 200 words, which allows a doctor to ask a patient questions relating to the symptoms of hypoglycaemia.

2 Overview of architecture

The architecture comprises three main modules. These are respectively responsible for source language speech recognition, including parsing and production of semantic representation; transfer and generation; and synthesis of target language speech. The speech processing modules (recognition and synthesis) are implemented on top of the standard Nuance Toolkit platform (Nuance, 2002). The language processing modules (transfer and generation) are a suite of simple routines written in SICStus Prolog.

Recognition is constrained by a CFG language model written in Nuance Grammar Specification Language (GSL), which also specifies the semantic representations produced. The grammar is not written by hand, but is rather compiled from a compact unification-grammar representation using the open source Regulus package (Rayner et al., 2001); the unification grammar, and the GSL representation it compiles into, are described in the next section. The speech and language processing modules communicate with each other through a minimal file-based protocol.

The semantic representations on both the source and target sides are expressed as attribute-value structures. Transfer rules map sets of attribute-value pairs to sets of attribute-value pairs; the great majority of the rules map single attribute-value pairs to single attribute-value pairs. Generation is handled by a small Definite Clause Grammar (DCG), which converts attribute-value structures into surface strings; its output is passed through a minimal post-transfer component, which applies a set of rules which map fixed strings to fixed strings. Speech synthesis is performed by the Nuance Vocalizer TTS engine.

3 Recognition and grammar

As described in the previous section, the recognition module is built on top of the Nuance Toolkit platform, using an annotated CFG language model consisting of a Nuance GSL grammar. This grammar is compiled from a unification-grammar representation using the Regulus tool. There are two important motivations for using unification grammar. Firstly, there is *efficiency*: the more compact nature of unification grammar, compared to CFG, substantially reduces the implementation effort required. The grammar for our pilot application currently contains only 28 unification-grammar rules, excluding lexical entries; these expand out into over 400 CFG rules.

An even more important advantage of using unification grammar is *uniformity*. Since the CFG rules are all derived automatically from the same compact underlying code-base, the implementor can be confident that related groups of CFG rules are always kept in step with each other. This means that it is practically feasible to construct a large CFG grammar in a short time, and keep it stable even if non-trivial changes are introduced during the development process.

In accordance with the generally minimalistic design philosophy of the project, semantic representations have been kept as simple as possible. The basic principle is that the representation of a clause is a flat list of attribute-value pairs: thus for example the representation of

"avez-vous souvent des maux d'estomac"
(lit: "have you often pains of stomach")
("do you often have stomach pains")

is the attribute-value list

```
[[state, feel],
 [frequency, souvent],
 [symptom, maux],
 [body_part, estomac]]]
```

In a broad domain, it is of course trivial to construct examples where this kind of representation runs into serious problems. In the very narrow domain of a phrasebook translator, it has many desirable properties. Grammar rules can in nearly all cases construct the semantic representation of the mother node by simple concatenation of the semantic representations of the daughters[1]. In general, the consequence is that operations on semantic representations typically manipulate lists rather than trees; the next section illustrates some of the advantages that follow from this fact. In a broad domain, we would pay a heavy price: the lack of structure in the semantic representations would often make them ambiguous. The very simple ontology of the phrasebook domain however means that ambiguity is not a problem; the components of a flat list representation can never be derived from more than one functional structure, so this structure does not need to be explicitly present.

[1] The only exception in our prototype grammar is the rule which allows a clause introduced by the subordinating conjunction *"quand"* ("when") to act as a modifier.

4 Transfer and generation

The minimal list-based representation language makes it possible to implement a simple but effective transfer and generation module. Transfer operates by applying rules which map lists of attribute-value pairs to lists of attribute-value pairs. Most rules are *transfer lexicon* (`t_lex`) entries, which map single attribute-value pairs to single attribute-value pairs associated with grammatical categories of the same kind. These pairs will often represent surface phrases consisting of more than one word. For example, the following two `t_lex` entries respectively map *"serrement à la poitrine"* into "tightness in the chest" and *"regarder la télé"* into "watch TV":

```
t_lex([symptom, serrement_poitrine],
      [symptom, tightness_in_the_chest]).
t_lex([hum_act, regarder_tv],
      [hum_act, watch_tv]).
```

A `t_lex` entry may equally well map an attribute-value pair to an attribute-value pair associated with a different grammatical category; for example, the following entry maps the representation of the adjective *"alcoolique"* to the representation of the noun phrase "an alcoholic":

```
t_lex([symptom, alcoolique],
      [symptom, alcoholic]).
```

It is also possible to write proper transfer rules (`t_rules`), which map a set of attribute-value pairs to a set of attribute-value pairs; for example, the following `t_rule` maps *"être d'humeur changeante"* (lit. "be of changing mood") to "suffer from mood swings":

```
t_rule([[state, etre],
        [symptom, d_humeur_changeante]],
       [[state, feel],
        [symptom, mood_swings]]).
```

The list-based representation language has allowed us to implement a simple but efficient transfer rule interpreter, which applies these

rules generally, irrespective of whether the pairs on the left-hand occur contiguously in the source-language input.

Many systems have shown that it is easy to generate from the kind of simple attribute-value representations used here. Our system performs generation using a small DCG grammar; typically, a rule absorbs one or more items from the transferred attribute-value list, and generates one or more output words. Once again, the list-based representation made it easy to implement the DCG in such a way that items can be absorbed in an arbitrary order. The result is that word-order differences between source and target pose no problems for translation.

Yet another advantage that follows from the minimal representation formalism is that it has been straightforward to write development tools that ensure internal consistency between the source-language, transfer, and target-language lexica. Early on in the project, we wrote a Prolog-based tool of this kind. If the source-language lexicon is extended or modified, the tool checks that each attribute-value pair in the source-language lexicon appears in the left-hand side of at least one transfer lexicon entry; if necessary, blank entries are added to the transfer lexicon for the implementor to complete. Similarly, the tool checks that every attribute-value pair appearing in the right-hand side of a transfer lexicon entry also appears in at least one generation lexicon entry. Use of the tool makes it possible to modify one part of the rule-base without manually having to keep track of the consequences, permitting a rapid development cycle.

5 A medical phrasebook translator

We have used the architecture outlined in the previous sections to construct a prototype French → English medical phrasebook translator. The basic scenario envisaged is that a French-speaking doctor suspects that an English-speaking patient may be suffering from some form of hypoglycaemia (low blood sugar). The symptoms of hypoglycaemia include anxiety, sweating, tachycardia, tremor, faintness, headache, confusion, convulsions, and coma; one of the reasons we have chosen hypoglycaemia as a domain is that these symptoms can coincide with those relating to many other conditions, which can often necessitate a lengthy verbal examination. In our initial prototype, we have limited ourselves to spoken yes/no questions, which we have based on those in a questionaire constructed by the Association of Hypoglycaemics of Quebec (Thériault, 2002). In terms of content, all questions are assumed to be of the basic form

"Do you
?(often/sometimes/ever/...)
(do something/experience symptom)
?(at time/when you do something)"

The grammar provides enough phrasal patterns that it is possible to ask about most domain concepts in a natural way. As described in the first section, however, we have intentionally constructed the system as a phrase-book rather than as a general translator. We only supply a minimal set of grammar rules, and assume that the user will be prepared to invest a little time in learning how to express themselves within these bounds.

The reason why it is not completely trivial to construct a system of this kind is that one cannot naturally ask about all domain symptoms using a single uniform phrasal pattern. The most common pattern is some version of

"avez-vous ?<freq> <symptom> ?<time>"
("do you suffer from ?<freq> <symptom> ?<time>")

so for example

"ressentez-vous des engourdissements ?" →
"do you suffer from numbness?" or
"éprouvez-vous souvent des maux de tête le matin ?" →
"do you often suffer from headache in the morning?"

However, there are many cases where some other pattern is required in order to express the question in a natural way. For example, we may need to use the verb *"être"* ("be"), e.g.

"êtes-vous émotive ?"→ "are you emotional?"

or to use an intransitive or transitive verb, e.g.

"urinez-vous fréquemment la nuit ?"→
"do you often urinate at night?"

"manquez-vous toujours d'énergie l'après-midi ?"→
(lit. "lack you always energy the afternoon?")
"do you always suffer from lack of energy in the afternoon?"

It may also be necessary to pose the question in the past tense, e.g.

"avez-vous déjà eu des convulsions ?" →
"have you ever suffered from convulsions?"

Finally, French has several different ways to form yes/no questions, of which the most common are subject/verb inversion, e.g.

"êtes-vous enceinte ?"
"are you pregnant?"

and fronting of *est-ce que*, e.g.

"est-ce que vous êtes enceinte ?"
"*est-ce que* you are pregnant?"

Although the *est-ce que* fronted construction is by default the preferred one in spoken French, there are many cases where inversion feels more natural, and it is in practice necessary to allow both constructions. Even when we try to keep the number of phrase types as low as we can, the choices along these different dimensions still multiply out to a non-trivial number of possibilities.

The current prototype has a vocabulary of about 200 words. The unification grammar used to create the recogniser contains 28 non-lexical rules and 179 lexical rules. The transfer lexicon contains 12 complex transfer rules and 104 transfer lexicon entries. The target language DCG contains 24 phrase-structure rules and 141 generation lexicon entries, and the post-transfer component contains 16 string-to-string rewriting rules. Creation of these linguistic resources required something between one and two person-weeks of expert effort; the most interesting aspect of this process was the focus we maintained thoughout on avoiding ambiguity in the analysis and generation grammars. This was primarily achieved by using sortal features consistently in both grammars, and maintaining a tight control of the domain ontology to ensure that the same sort of object can never occur in two different positions in a single clause. Occasionally this meant that the lexicon entries for a word had to be duplicated in two versions; for example, *"repas"* ("meal") can occur both as part of a temporal PP (*"manger entre repas"*, "eat between meals"), or as the object of some verbs (*"sauter un repas"*, "skip a meal"). In cases like these, we included separate entries in the lexicon for the two usages of the word, giving them distinct sortal categories in the ontology.

6 Evaluation

The real question we would like to answer when evaluating the prototype system is whether it is practically useful. Unfortunately, we are not yet in a position to do this, since the system is not mature enough in terms of coverage for it to be meaningful to subject it to a field trial. We thus have to content ourselves with more modest evaluation goals, and seek indirect evidence which suggests that an expanded version of the system could be useful. Some of this evidence consists of tests of the system's internal validity; in particular, we have carried out systematic checks that the analysis and generation grammars really are unambiguous, and that translation always produces an output. We do this by using the Nuance Toolkit's `generate` utility to create large sets of (text) utterances within the coverage of the analysis grammar, and then

processing them through the system with both grammars run in an all-solutions mode. A test of this kind on 10 000 randomly generated utterances showed that all 10 000 produced an output, and that for each utterance there was always exactly one possible semantic analysis, and one possible string generated from the transferred representation.

In terms of external evaluation criteria, our architecture is primarily aimed at providing adequate recognition and reliable translation; it consequently makes sense to start by testing these aspects of performance. Specifically, we investigated the following two measures:

1. If the user says something within the system's coverage, how often is it correctly recognised?

2. If what the user says is correctly recognised, how often is it correctly translated?

We investigated recognition quality on in-coverage utterances by again randomly producing a set of 500 such utterances using the `generate` utility. Since the recognition grammar overgenerates, some of these utterances are ungrammatical or nonsensical; a human judge manually filtered the set to leave 195 good-quality utterances, averaging 6.8 words in length.

The intention is that the translator would be normally used by experts who would have time to learn how to operate it. In order to simulate this pattern of use, we randomly divided the 195 good utterances into a "practice" set of 150 utterances and an "evaluation" set of 45 utterances. Five subjects (students who had not previously had exposure to the system) were each given twenty minutes to experiment with the system by reading out sentences from the practice set, and then competed on the task of reading out the evaluation utterances; each subject read each utterance once, and a small prize was given to the subject who got the best recognition result. Given the nature of the task, it seems more appropriate to evaluate in terms of sentence-level measures rather than word error rates. We consequently scored utterances as belonging to one of three possible categories: "WordsOK" (no word errors), "SemOK" (at least one word error, but the semantic representation was correct), and "Bad" (no recognition, or incorrect semantic representation). The results are presented in Table 1.

Subj	WordsOK	SemOK	Bad
Rec1	11	3	31
Rec2	32	5	8
Rec3	28	8	9
Rec4	18	6	21
Rec5	34	10	1
Av.	24.6 (55%)	6.4 (14%)	14.0 (31%)

Table 1: Recognition performance of 5 subjects reading 45 in-coverage utterances

Subj	Good	OK	Bad
Trans1	141	6	3
Trans2	98	51	1
Trans3	90	55	5
Av.	110 (73%)	37 (25%)	3 (2%)

Table 2: Quality of translation on 150 in-coverage utterances, as evaluated by three independent judges

In order to investigate the second question (translation quality), we randomly generated a new set of 150 in-coverage utterances, and process them through the system to produce text outputs. We then asked three independent bilingual judges to evaluate the source/target pairs as either fully correct ("Good"), acceptable ("OK") or incorrect/nonsensical ("Bad"). The results are presented in Table 2.

Although this evaluation makes no pretentions to being definitive, we find the results encouraging. Three of our five test users were able to adapt quickly to the system, and achieved high recognition accuracy after only a short practice period. We were initially concerned that as many as 2% of the utterances were mistranslated. Analysis of the results however

revealed that these utterances were problematic for reasons that had more to do with the evaluation methodology than the system. The judges did not appear to have strong intuitions about the correctness or otherwise of the critical translations; no translation was marked as bad by all three judges, and only one translation was marked as bad by two judges out of three. It was also noticeable that at least half of the source-language utterances which resulted in "bad" translations were dubious either syntactically or pragmatically, and should arguably have been filtered out when preparing the evaluation data. We intend soon to carry out a revised evaluation which will address these issues.

7 Conclusion and further directions

The main point we want to make in this paper is that today's technology makes it possible to build limited-domain speech to speech translators which represent an interesting compromise between trivial fixed-phase systems on the one hand and sophisticated VERBMOBIL-style systems on the other. These systems can offer sufficient coverage to allow a user to express themselves fairly freely after a little practice, but are still constrained enough that they appear to have the potential to reach levels of reliability appropriate for medical and other safety-critical applications. They can be quickly constructed on top of standard commercial platforms like the Nuance Toolkit, and run on ordinary PCs.

As already indicated, the critical question is whether a system of this kind can be expanded to the point where it becomes practically useful. In particular, it is still unclear how much more grammar and vocabulary are needed in order to achieve this goal, and how much performance will degrade if coverage is increased accordingly. In concrete terms, this amounts to asking whether it is feasible to construct controlled languages for at least some interesting domains which achieve a suitable balance between coverage and performance. We have now begun implementation of a second and more elaborate version of the system, and expect to be able to report on its performance by the time of the workshop.

Acknowledgements

The original idea of building a medical speech translator of this kind was suggested by Dr Vol Van Dalsem III, of El Camino Hospital, Mountain View, CA. Dr Van Dalsem has also provided many suggestions concerning subdomains and coverage, and is actively collaborating with us on further development of the system. Work on the initial prototype was carried out at TIM/ISSCO, Geneva University, under internal funding. The paper benefited greatly from a number of discussions with Ian Lewin.

References

H. Alshawi, S. Bangalore, and S. Douglas. 2000. Learning dependency translation models as collections of finite state head transducers. *Computational Linguistics*, 26(1).

D. Carter, M. Rayner, et al. 2000. Evaluation. In Rayner et al. (Rayner et al., 2000).

FlexiPC, 2002. http://www.flexipc.com/product/, then "translator". As of 15 Mar 2002.

R. Frederking, A. Rudnicky, and C. Hogan. 1997. Interactive speech translation in the diplomat project. In *Proceedings of the Spoken Language Translation workshop at the 35th Meeting of the Association for Computational Linguistics*, Madrid, Spain.

IntegratedWaveTechnologies, 2002. http://www.i-w-t.com/investor.html. As of 15 Mar 2002.

Nuance, 2002. http://www.nuance.com. As of 1 Feb 2002.

M. Rayner, D. Carter, P. Bouillon, V. Digalakis, and M. Wirén, editors. 2000. *The Spoken Language Translator*. Cambridge University Press.

M. Rayner, J. Dowding, and B.A. Hockey. 2001. A baseline method for compiling typed unification grammars into context free language models. In *Proceedings of Eurospeech 2001*, pages 729–732, Aalborg, Denmark.

M. Thériault, 2002. Questionnaire de dépistage pour adultes (in French). As of 15 Mar 2002.

W. Wahlster, editor. 2000. *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer.