Coling 2008

22nd International Conference on Computational Linguistics

Proceedings of the workshop on Speech Processing for Safety Critical Translation and Pervasive Applications

Workshop chairs:
Pierrette Bouillon, Farzad Ehsani, Robert Frederking, Michael McTear and Manny Rayner

23 August 2008

Order copies of this and other Coling proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-905593-52-1

Design by Chimney Design, Brighton, UK
Production and manufacture by One Digital, Brighton, UK

Introduction

Two ideas currently gaining popularity in spoken dialogue construction are safety critical translation and pervasive speech-enabled applications. Safety critical, and in particular, medical, applications have emerged as one of the most popular domains for speech translation. At the first workshop on medical speech translation, held at HLT 2006, a measure of consensus emerged on at least some points. The key issue that differentiates the medical domain from most other application areas for speech translation is its safety-critical nature; systems can realistically be field- deployed now or in the very near future; the basic communication model should be collaborative, and allow the client users to play an active role; and medical systems are often most useful when deployed on mobile devices. This last point offers a natural link to pervasive computing applications, where spoken language technologies provide an effective and natural interface for mobile devices in situations where traditional modes of communication are less appropriate.

However, there is so far little agreement on many central questions, including choices of architectures, component technologies, and evaluation methodologies. In this workshop we hope that people interested in these types of application will meet, exchange ideas and demo live systems.

Organizers:

Pierrette Bouillon, U Geneva (Switzerland)
Farzad Ehsani, Fluential (US)
Robert Frederking, CMU (US)
Michael McTear, U Ulster (Northern Ireland)
Manny Rayner, U Geneva (Switzerland)

Programme Committee:

Laurent Besacier, U Grenoble (France)
Pierrette Bouillon, U Geneva (Switzerland)
Mike Dillinger, SpokenTranslation (US)
Farzad Ehsani, Fluential (US)
Glenn Flores, U Texas (US)
Robert Frederking, CMU (US)
Hitoshi Isahara, NICT (Japan)
Michael McTear, U Ulster (Northern Ireland)
Shri Narayanan, USC (US)
Aarne Ranta, U Gothenburg (Sweden)
Manny Rayner, U Geneva (Switzerland)
Tanja Schultz, U Karlsruhe (Germany)
Harold Somers, U Manchester (UK) and Dublin City U (Ireland)
Bowen Zhou, IBM (US)

Table of Contents

Conference Programme

Saturday, August 23, 2008

9:30–9:40 Opening Remarks

Session 1: Architectures for Speech Translation

09:40–10:05 *Mitigation of Data Sparsity in Classifier-Based Translation*
Emil Ettelaie, Panayiotis G. Georgiou and Shrikanth S. Narayanan

10:05–10:30 *Speech Translation with Grammatical Framework*
Björn Bringert

10:30–11:00 Break

Session 2: Pervasive Speech Applications

11:00–11:25 *An Integrated Dialog Simulation Technique for Evaluating Spoken Dialog Systems*
Sangkeun Jung, Cheongjae Lee, Kyungduk Kim and Gary Geunbae Lee

11:25–11:50 *Economical Global Access to a VoiceXML Gateway Using Open Source Technologies*
Kulwinder Singh and Dong-Won Park

11:50–12:15 *Interoperability and Knowledge Representation in Distributed Health and Fitness Companion Dialogue System*
Jaakko Hakulinen and Markku Turunen

12:30–14:00 Lunch

Saturday, August 23, 2008 (continued)

Session 3: Speech Translation Demos

14:00–15:30 *The 2008 MedSLT System*
Manny Rayner, Pierrette Bouillon, Jane Brotanek, Glenn Flores, Sonia Halimi, Beth Ann Hockey, Hitoshi Isahara, Kyoko Kanzaki, Elisabeth Kron, Yukie Nakao, Marianne Santaholma, Marianne Starlander and Nikos Tsourakis

14:00–15:30 *Language Understanding in Maryland Virtual Patient*
Sergei Nirenburg, Stephen Beale, Marjorie McShane, Bruce Jarrell and George Fantry

14:00–15:30 *Rapid Portability among Domains in an Interactive Spoken Language Translation System*
Mark Seligman and Mike Dillinger

15:30–16:00 Break

Session 4: Speech Translation Systems

16:00–16:25 *Speech Translation for Triage of Emergency Phonecalls in Minority Languages*
Udhyakumar Nallasamy, Alan Black, Tanja Schultz, Robert Frederking and Jerry Weltman

16:25–16:50 *Speech to Speech Translation for Nurse Patient Interaction*
Farzad Ehsani, Jim Kimzey, Elaine Zuber, Demitrios Master and Karen Sudre

Session 5: A Shared Task for Medical Speech Translation?

16:50–17:05 *A Small-Vocabulary Shared Task for Medical Speech Translation*
Manny Rayner, Pierrette Bouillon, Glenn Flores, Farzad Ehsani, Marianne Starlander, Beth Ann Hockey, Jane Brotanek and Lukas Biewald

17:05–close Panel Discussion

Mitigation of data sparsity in classifier-based translation

Emil Ettelaie, Panayiotis G. Georgiou, Shrikanth S. Narayanan
Signal Analysis and Interpretation Laboratory
Ming Hsieh Department of Electrical Engineering
Viterbi School of Engineering
University of Southern California
ettelaie@usc.edu

Abstract

The concept classifier has been used as a translation unit in speech-to-speech translation systems. However, the sparsity of the training data is the bottle neck of its effectiveness. Here, a new method based on using a statistical machine translation system has been introduced to mitigate the effects of data sparsity for training classifiers. Also, the effects of the background model which is necessary to compensate the above problem, is investigated. Experimental evaluation in the context of cross-lingual doctor-patient interaction application show the superiority of the proposed method.

1 Introduction

Statistical machine translation (SMT) methods are well established in speech-to-speech translation systems as the main translation technique (Narayanan et al., 2003; Hsiao et al., 2006). Due to their flexibility these methods provide a good coverage of the dialog domain. The fluency of the translation, however, is not guaranteed. Disfluencies of spoken utterances plus the speech recognizer errors degrade the translation quality even more. All these ultimately affect the quality of the synthesized speech output in the target language, and the effectiveness of the concept transfer.

It is quite common, though, to use other means of translation in parallel to the SMT methods (Gao et al., 2006; Stallard et al., 2006). Concept classification, as an alternative translation method, has been successfully integrated in speech-to-speech translators (Narayanan et al., 2003; Ehsani et al., 2006). A well defined dialog domain, e.g. doctor-patient dialog, can be partly covered by a number of concept classes. Upon a successful classification of the input utterance, the translation task reduces to synthesizing a previously created translation of the concept, as a mere look up. Since the main goal in such applications is an accurate exchange of concepts, this method would serve the purpose as long as the input utterance falls within the coverage of the classifier. This process can be viewed as a quantization of a continuous "semantic" sub-space. The classifier is adequate when the quantization error is small (i.e. the derived concept and input utterance are good matches), and when the utterance falls in the same sub-space (domain) as the quantizer attempts to cover. Since it is not feasible to accurately cover the whole dialog domain (since a large number of quantization levels needed) the classifier should be accompanied by a translation system with a much wider range such as an SMT engine. A rejection mechanism can help identify the cases that the input utterance falls outside the classifier coverage (Ettelaie et al., 2006).

In spite of this short coming, the classifier-based translator is an attractive option for speech-to-speech applications because of its tolerance to "noisy" input and the fluency of its output, when it operates close to its design parameters. In practice this is attainable for structured dialog interactions with high levels of predictability. In addition, it can provide the users with both an accurate feedback and different translation options to choose from. The latter feature, specially, is useful for applications like doctor-patient dialog.

Building a concept classifier starts with identifying the desired concepts and representing them with canonical utterances that express these concepts. A good set of concepts should consist of the ones that are more frequent in a typical interaction in the domain. For instance in a doctor-patient dialog, the utterance "Where does it hurt?" is quite common and therefore its concept is a good choice. Phrase books, websites, and experts' judgment are some of the resources that can be used for concept selection. Other frequently used concepts include those that correspond to basic communicative and social aspects of the interaction such as greeting, acknowledgment and confirmation.

After forming the concept space, for each class,

Coling 2008: Proceedings of the workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, pages 1–4
Manchester, August 2008

utterances that convey its concept must be gathered. Hence, this training corpus would consist of a group of paraphrases for each class. This form of data are often very difficult to collect as the number of classes grow. Therefore, the available training data are usually sparse and cannot produce a classification accuracy to the degree possible. Since the classifier range is limited, high accuracy within that range is quite crucial for its effectiveness. One of the main issues is dealing with data sparsity. Other techniques have also been proposed to improve the classification rates. For example in (Ettelaie et al., 2006) the accuracy has been improved by introducing a dialog model. Also, a background model has been used to improve the discrimination ability of a given concept class model.

In this work a novel method for handling the sparsity is introduced. This method utilizes an SMT engine to map a single utterance to a group of them. Furthermore, the effect of the background model on classification accuracy is investigated.

Section 2 reviews the concept classification process and the background model. In Section 3 the sparsity handling method using an SMT is introduced. Data and experiments are described in Section 4. The results are discussed in Section 5.

2 Concept classifier and background model

The concept classifier based on the maximum likelihood criterion can be implemented as a language model (LM) scoring process. For each class a language model is built using data expressing the class concept. The classifier scores the input utterance using the class LM's and selects the class with highest score. In another word if C is the set of concept classes and $\mathbf{e}$ is the input utterance, the classification process is,

$$\hat{c} = \arg\max_{c \in C} \{P_c(\mathbf{e} \mid c)\} \quad (1)$$

where $P_c(\mathbf{e} \mid c)$ is the score of $\mathbf{e}$ from the LM of class c. The translation job is concluded by playing out a previously constructed prompt that expresses the concept $\hat{c}$ in the target language.

It is clear that a class with limited training data items will have an undertrained associated LM with poor coverage. In practice such a model fails to produce a usable LM score and leads to a poor classification accuracy. Interpolating the LM with a background language model results in a smoother model (Stolcke, 2002) and increases the overall accuracy of the classifier.

The background model should be built from a larger corpus that fairly covers the domain vocabulary. The interpolation level can be optimized for the best performance based on heldout set.

3 Handling sparsity by statistical machine translation

The goal is to employ techniques that limit the effects of data sparsity. What is proposed here is to generate multiple utterances – possibly with lower quality – from a single original one. One approach is to use an SMT to generate n-best lists of translation candidates for the original utterances. Such lists are ranked based on a combination of scores from different models (Ney et al., 2000). The hypothesis here is that for an SMT trained on a large corpus, the quality of the candidates would not degrade rapidly as one moves down the n-best list. Therefore a list with an appropriate length would consist of translations with acceptable quality without containing a lot of poor candidates. This process would result in more data, available for training, at the cost of using noisier data.

Although the source language of the SMT must be the same as the classifier's, its target language can be selected deliberately. It is clear that a language with large available resources (in the form of parallel corpora with the source language) must be selected. For simplicity this language is called the "intermediate language" here.

A classifier in the intermediate language can be built by first generating an n-best list for every source utterance in the classifier's training corpus. Then the n-best lists associated with each class are combined to form a new training set. The class LM's are now built from these training sets rather than the original sets of the source utterances.

To classify a source utterance $\mathbf{e}$, first the SMT is deployed to generate an n-best list (in the intermediate language) from it. The list will consist of candidates $\mathbf{f}_1$, $\mathbf{f}_2$,..., $\mathbf{f}_n$. The classification process can be reformulated as,

$$\hat{c} = \arg\max_{c \in C} \left\{ \prod_{i=1}^{n} \tilde{P}_c(\mathbf{f}_i \mid c) \right\} \quad (2)$$

Here, $\tilde{P}_c(\mathbf{f}_i \mid c)$ is the score of the i^{th} candidate $\mathbf{f}_i$ from the LM of class c. The scores are considered in the probability domain.

The new class LM's can also be smoothed by interpolation with a background model in the intermediate language.

4 Data and Experiments

4.1 Data

The data used in this work were originally collected for, and used in, the Transonics project (Narayanan et al., 2003) to develop an English/Farsi speech-to-speech translator in the doctor-patient interaction domain. For the doctor side, 1,269 concept classes were carefully chosen using experts' judgment and medical phrase books. Then, for each concept, English data were collected from a website, a web-based game, and multiple paraphrasing sessions at the Information Sciences Institute of the University

	Conventional (baseline)	n-best length			
		100	500	1,000	2,000
Accuracy [%]	74.9	77.4	77.5	76.8	76.4
Relative error reduction [%]	0.0	10.0	10.4	7.6	6.0
Accuracy in 4-best [%]	88.6	90.7	91.0	91.3	90.5
Relative error reduction [%]	0.0	18.4	21.1	23.7	16.7

Table 1: Classification accuracy for the conventional method and the proposed method with different lengths of n-best list

of Southern California. The total size of the data set consists of 9,893 English phrases.

As the test corpus for this work, 1,000 phrases were randomly drawn from the above set and the rest were used for training. To make sure that the training set covered every class, one phrase per class was excluded from the test set selection process.

To generate the n-best lists, a phrase based SMT (Koehn et al., 2003) was used. The intermediate language was Farsi and the SMT was trained on a parallel English/Farsi corpus with 148K lines (1.2M words) on the English side. This corpus was also used to build the classification background models in both languages. The SMT was optimized using a parallel development set with 915 lines (7.3K words) on the English side.

4.2 Classification Accuracy Measures

Classifier accuracy is often used as the the quality indicator of the classification task. However, it is common in the speech-to-speech translation systems to provide the user with a short list of potential translations to choose from. For example the user of system in (Narayanan et al., 2003) is provided with the top four classifier outputs. In such cases, it is practically useful to measure the accuracy of the classifier within its n-best outputs (e.g., $n = 4$ for the above system). In this work the classification accuracy was measured on both the single output and the 4-best outputs.

4.3 Experiments

To compare the proposed method with the conventional classification, a classifier based on each method was put to test. In the proposed method, it is expected that the accuracy is affected by the length of the n-best lists. To observe that, n-best lists of lengths 100, 500, 1000, and 2000 were used in the experiments. The results are shown in Table 1. In all of the above experiments the background interpolation factor was set to 0.9 which is close to the optimum value obtained in (Ettelaie et al., 2006).

To examine the effect of the background model, the conventional and proposed methods were tried with different values of the interpolation factor λ (the background model is weighted by $1 - \lambda$). For the conventional method the length of the n-best list was set to 500. Figure 1 shows the accuracy

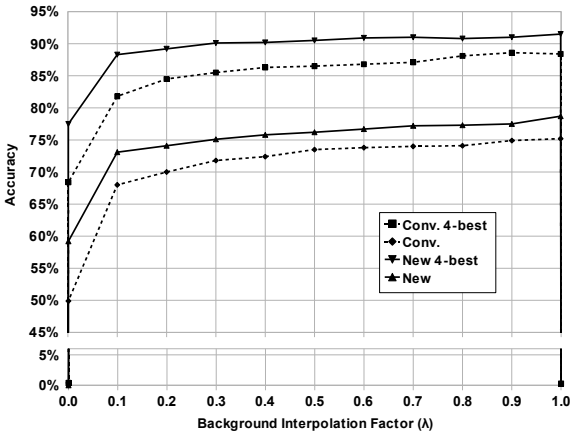


Figure 1: The effect of background model on classification accuracy

changes with respect to the interpolation factor for these two methods.

5 Discussion

Table 1 shows the advantage of the proposed method over the conventional classification with a relative error rate reduction up to 10.4% (achieved when the length of the SMT n-best list was 500). However, as expected, this number decreases with longer SMT n-best lists due to the increased noise present in lower ranked outputs of the SMT.

Table 1 also shows the accuracy within 4-best classifier outputs for each method. In that case the proposed method showed an error rate which was relatively 23.7% lower than the error rate of the conventional method. That was achieved at the peak of the accuracy within 4-best, when the length of the SMT n-best list was 1,000. In this case too, further increase in the length of the n-best list led to an accuracy degradation as the classifier models became noisier.

The effect of the background model on classifier accuracy is shown in Figure 1. The figure shows the one-best accuracy and the accuracy within 4-best outputs, versus the background interpolation factor (λ) for both conventional and proposed methods. As the curves indicate, with λ equal to zero the classifier has no discriminating feature since all the class scores are driven solely from the background model. However, a slight increase in λ, leads to a large jump in the accuracy. The reason is that the background model was built from a large general domain corpus and hence, had no bias toward any of the classes. With a small λ, the score from the background model dominates the overall class scores. In spite of that, the score differences caused by the class LM's are notable in improving the classifier performance.

As λ increases the role of the class LM's becomes more prominent. This makes the classifier models more discriminative and increases its accuracy as shown in Figure 1. When the factor is in the close vicinity of one, the smoothing effect of the background model diminishes and leaves the

classes with spiky models with very low vocabulary coverage (lots of zeros). This leads to a rapid drop in accuracy as λ reaches one.

Both the conventional and proposed methods follow the above trend as Figure 1 shows, although, the proposed method maintains its superiority throughout the range of λ that was examined. The maximum measured accuracies for conventional and proposed methods were 75.2% and 78.7% respectively and was measured at $\lambda = 0.999$ for both methods. Therefore, the error rate of the proposed method was relatively 14.1% lower than its counterpart from the conventional method.

Figure 1 also indicates that when the accuracy is measured within the 4-best outputs, again the proposed method outperforms the conventional one. The maximum 4-best accuracy for the conventional method was measured at the sample point $\lambda = 0.9$ and was equal to 88.6%. For the proposed method, that number was measured as 91.5% achieved at the sample point $\lambda = 0.999$. In another words, considering the 4-best classifier outputs, the error rate of the proposed method was relatively 25.4% lower.

6 Conclusion

The proposed language model based method can be used to improve the accuracy of the concept classifiers specially in the case of sparse training data. It outperformed the conventional classifier, trained on the original source language paraphrases, in the experiments. With this method, when the input utterance is within the classification domain, the classifier can be viewed as a filter that produces fluent translations (removes the "noise") from the SMT output.

The experiments also emphasized the importance of the background model, although indicated that the classification accuracy was not very sensitive to the value of the background interpolation factor. This relieves the developers from the fine tuning of that factor and eliminates the need for a development data set when a suboptimal solution is acceptable.

We believe that significant improvements to the technique can be made through the use of weighted n-best lists based on the SMT scores. In addition we believe that using a much richer SMT engine could provide significant gains through increased diversity in the output vocabulary. We intend to extend on this work through the use of enriched, multilingual SMT engines, and the creation of multiple classifiers (in several intermediate languages).

7 Acknowledgment

This work was supported in part by funds from DARPA.

References

Ehsani, F., J. Kinzey, D. Master, K. Sudre, D. Domingo, and H. Park. 2006. S-MINDS 2-way speech-to-speech translation system. In *Proc. of the Medical Speech Translation Workshop, Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*, pages 44–45, New York, NY, USA, June.

Ettelaie, E., P. G. Georgiou, and S. Narayanan. 2006. Cross-lingual dialog model for speech to speech translation. In *Proc. of the Ninth International Conference on Spoken Language Processing (ICLSP)*, pages 1173–1176, Pittsburgh, PA, USA, September.

Gao, Y., L. Gu, B. Zhou, R. Sarikaya, M. Afify, H. Kuo, W. Zhu, Y. Deng, C. Prosser, W. Zhang, and L. Besacier. 2006. IBM MASTOR SYSTEM: Multilingual automatic speech-to-speech translator. In *Proc. of the Medical Speech Translation Workshop, Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*, pages 53–56, New York, NY, USA, June.

Hsiao, R., A. Venugopal, T. Kohler, Y. Zhang, P. Charoenpornsawat, A. Zollmann, S. Vogel, A. W. Black, T. Schultz, and A. Waibel. 2006. Optimizing components for handheld two-way speech translation for an English-Iraqi Arabic system. In *Proc. of the Ninth International Conference on Spoken Language Processing (ICLSP)*, pages 765–768, Pittsburgh, PA, USA, September.

Koehn, P., F. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*, volume 1, pages 48–54, Edmonton, AB, Canada, May-June.

Narayanan, S., S. Ananthakrishnan, R. Belvin, E. Ettelaie, S. Ganjavi, P. Georgiou, C. Hein, S. Kadambe, K. Knight, D. Marcu, H. Neely, N. Srinivasamurthy, D. Traum, and D. Wang. 2003. Transonics: A speech to speech system for English-Persian interactions. In *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 670–675, St.Thomas, U.S. Virgin Islands, November-Decmeber.

Ney, H., S. Nießen, F. J. Och, C. Tillmann, H. Sawaf, and S. Vogel. 2000. Algorithms for statistical translation of spoken language. *IEEE Trans. on Speech and Audio Processing, Special Issue on Language Modeling and Dialogue Systems*, 8(1):24–36, January.

Stallard, D., F. Choi, K. Krstovski, P. Natarajan, R. Prasad, and S. Saleem. 2006. A hybrid phrase-based/statistical speech translation system. In *Proc. of the Ninth International Conference on Spoken Language Processing (ICLSP)*, pages 757–760, Pittsburgh, PA, USA, September.

Stolcke, A. 2002. SRILM - an extensible language modeling toolkit. In *Proc. of the International Conference on Spoken Language Processing (ICSLP)*, pages 901–904, Denver, CO, USA, September.

Speech Translation with Grammatical Framework

Björn Bringert
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
bringert@chalmers.se

Abstract

Grammatical Framework (GF) is a grammar formalism which supports interlingua-based translation, library-based grammar engineering, and compilation to speech recognition grammars. We show how these features can be used in the construction of portable high-precision domain-specific speech translators.

1 Introduction

Speech translators for safety-critical applications such as medicine need to offer high-precision translation. One way to achieve high precision is to limit the coverage of the translator to a specific domain. The development of such high-precision domain-specific translators can be resource intensive, and require rare combinations of developer skills. For example, consider developing a Russian–Swahili speech translator for the orthopedic domain using direct translation between the two languages. Developing such a system could require an orthopedist programmer and linguist who speaks Russian and Swahili. Such people may be hard to find. Furthermore, developing translators for all pairs of N languages requires $O(N^2)$ systems, developed by an equal number of bilingual domain experts.

The language pair explosion and the need for the same person to possess knowledge about the source and target languages can be avoided by using an interlingua-based approach. The requirement that developers be both domain experts and linguists can be addressed by the use of grammar libraries which implement the domain-independent linguistic details of each language.

Grammatical Framework (GF) (Ranta, 2004) is a type-theoretic grammar formalism which is well suited to high-precision domain-specific interlingua-based translation (Khegai, 2006), and library-based grammar engineering (Ranta, 2008). GF divides grammars into *abstract syntax* and *concrete syntax*. The abstract syntax defines *what* can be said in the grammar, and the concrete syntax defines *how* it is said in a particular language. If one abstract syntax syntax is given multiple concrete syntaxes, the abstract syntax can be used as an interlingua. Given an abstract and a concrete syntax, GF allows both parsing (text to abstract syntax) and linearization (abstract syntax to text). This means that interlingua-based translation is just a matter of parsing in one language and linearizing to another.

The GF resource grammar library (Ranta, 2008) implements the domain-independent morphological and syntactic details of eleven languages. A grammar writer can use functions from a resource grammar when defining the concrete syntax of an application grammar. This is made possible by GF's support for *grammar composition*, and frees the grammar writer from having to implement linguistic details such as agreement, word order etc.

In addition to parsing and linearization, the declarative nature of GF grammars allows them to be compiled to other grammar formats. The GF speech recognition grammar compiler (Bringert, 2007) can produce context-free grammars or finite-state models which can be used to guide speech recognizers.

These components, interlingua-based translation, grammar libraries, and speech recognition grammar compilation, can be used to develop

Coling 2008: Proceedings of the workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, pages 5–8
Manchester, August 2008

domain-specific speech translators based on GF grammars. Figure 1 shows an overview of a minimal unidirectional speech translator which uses these components. This is a proof-of-concept system that demonstrates how GF components can be used for speech translation, and as such it can hardly be compared to a more complete and mature system such as MedSLT (Bouillon et al., 2005). However, the system has some promising features compared to systems based on unification grammars: the expressive power of GF's concrete syntax allows us to use an application-specific interlingua without any transfer rules, and the wide language support of the GF Resource Grammar library makes it possible to quickly port applications to new languages.

In Section 2 we show a small example grammar for a medical speech translator. Section 3 briefly discusses how a speech translator can be implemented. Section 5 describes some possible extensions to the proof-of-concept system, and Section 6 offers some conclusions.

2 Example Grammar

We will show a fragment of a grammar for a medical speech translator. The example comes from Khegai's (2006) work on domain-specific translation with GF, and has been updated to use the current version of the GF resource library API.

The small abstract syntax (interlingua) shown in Figure 2 has three categories (**cat**): the start category Prop for complete utterances, Patient for identifying patients, and Medicine for identifying medicines. Each category contains a single function (**fun**). There are the nullary functions ShePatient and PainKiller, and the binary NeedMedicine, which takes a Patient and a Medicine as arguments, and produces a Prop. This simple abstract syntax only allows us to construct the term NeedMedicine ShePatient PainKiller. A larger version could for example include categories for body parts, symptoms and illnesses, and more functions in each category. An example of a term in such an extended grammar could be And (Injured TheyPatient Foot) (NeedMedicine HePatient Laxative).

For this abstract syntax we can use the English resource grammar to write an English concrete syntax, as shown in Figure 3. The resource grammar category NP is used as the linearization type (**lincat**) of the application grammar categories

```
abstract Health = {
flags startcat = Prop;
cat Patient; Medicine; Prop;
fun
  ShePatient : Patient;
  PainKiller  : Medicine;
  NeedMedicine : Patient → Medicine → Prop;
}
```

Figure 2: Example abstract syntax.

Patient and Medicine, and S is used for Prop. The linearizations (**lin**) of each abstract syntax function use overloaded functions from the resource grammar, such as $mkCl$ and mkN which create clauses and nouns, respectively.

```
concrete HealthEng of Health =
  open SyntaxEng, ParadigmsEng in {
lincat Patient, Medicine = NP; Prop = S;
lin
  ShePatient = mkNP she_Pron;
  PainKiller =
    mkNP indefSgDet (mkN "painkiller");
  NeedMedicine p m =
    mkS (mkCl p (mkV2 (mkV "need")) m);
}
```

Figure 3: English concrete syntax.

Figure 4 shows a Swedish concrete syntax created in the same way. Note that PainKiller in Swedish uses a mass noun construction rather than the indefinite article.

```
concrete HealthSwe of Health =
  open SyntaxSwe, ParadigmsSwe in {
lincat Patient, Medicine = NP; Prop = S;
lin
  ShePatient = mkNP she_Pron;
  PainKiller =
    mkNP massQuant
      (mkN "smärtstillande");
  NeedMed p m =
    mkS (mkCl p
      (mkV2 (mkV "behöver")) m);
}
```

Figure 4: Swedish concrete syntax.

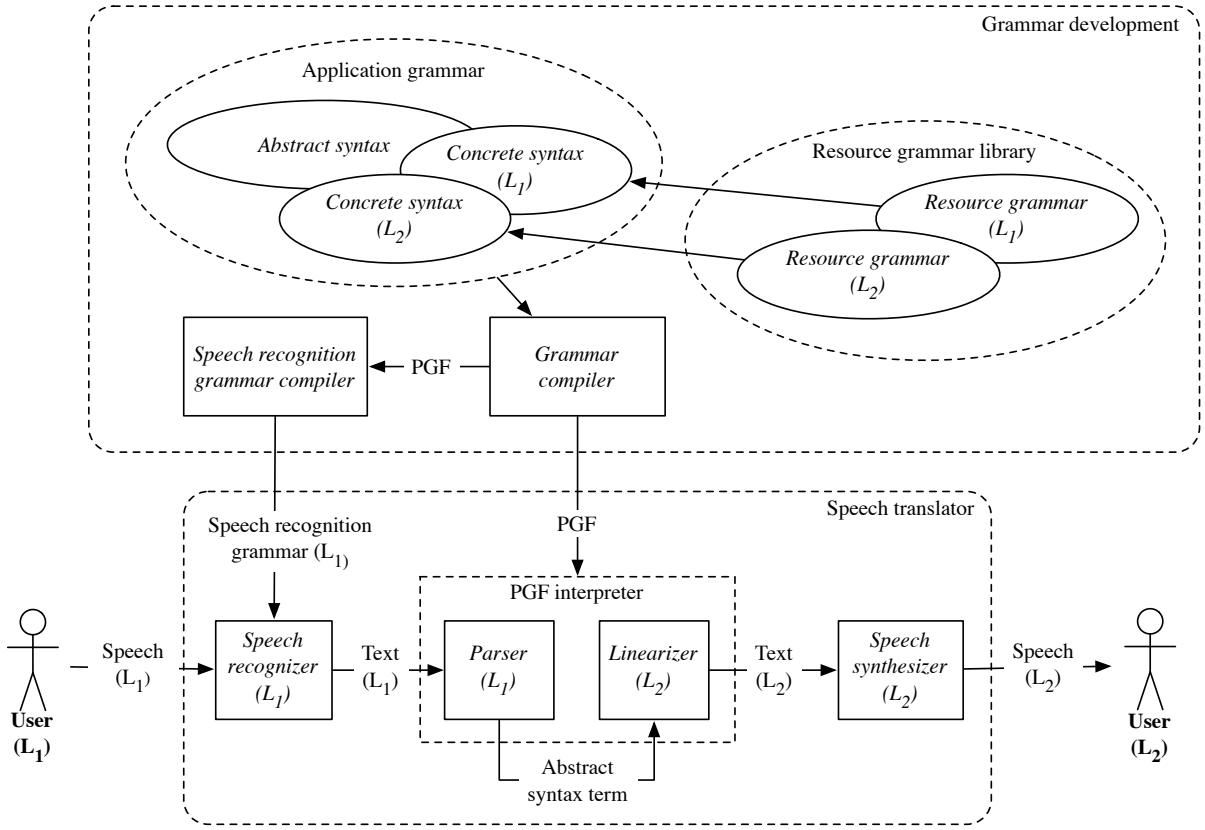


Figure 1: Overview of a GF-based speech translator. The developer writes a multilingual application grammar using the resource grammar library. This is compiled to a PGF (Portable Grammar Format) grammar used for parsing and linearization, and a speech recognition grammar. Off-the-shelf speech recognizers and speech synthesizers are used together with a PGF interpreter in the running system.

3 Speech Translator Implementation

The GF grammar compiler takes grammars in the GF source language used by programmers, and produces grammars in a low-level language (Portable Grammar Format, PGF (Angelov et al., 2008)) for which interpreters can be easily and efficiently implemented. There are currently PGF implementations in Haskell, Java and JavaScript. The GF speech recognition grammar compiler (Bringert, 2007) targets many different formats, including Nuance GSL, SRGS, JSGF and HTK SLF. This means that speech translators based on GF can easily be implemented on almost any platform for which there is a speech recognizer and speech synthesizer. We have run Java-based versions under Windows using Nuance Recognizer and RealSpeak or FreeTTS, Haskell-based versions under Linux using Nuance Recognizer and RealSpeak, and JavaScript-based prototypes in the Opera XHTML+Voice-enabled web browser on Zaurus PDAs and Windows desktops.

The speech translation system itself is domain-independent. All that is required to use it in a new domain is an application grammar for that domain.

4 Evaluation

Since we have presented a proof-of-concept system that demonstrates the use of GF for speech translation, rather than a complete system for any particular domain, quantitative translation performance evaluation would be out of place. Rather, we have evaluated the portability and speed of prototyping. Our basic speech translators written in Java and Haskell, using existing speech components and PGF interpreters, require less than 100 lines of code each. Developing a small domain for the translator can be done in under 10 minutes.

5 Extensions

5.1 Interactive Disambiguation

The concrete syntax for the source language may be ambiguous, i.e. there may be sentences for which parsing produces multiple abstract syntax

terms. The ambiguity can sometimes be preserved in the target language, if all the abstract syntax terms linearize to the same sentence.

In cases where the ambiguity cannot be preserved, or if we want to force disambiguation for safety reasons, we can use a *disambiguation grammar* to allow the user to choose an interpretation. This is a second concrete syntax which is completely unambiguous. When the user inputs an ambiguous sentence, the system linearizes each of the abstract syntax terms with the disambiguation grammar, and prompts the user to select the sentence with the intended meaning. If only some of the ambiguity can be preserved, the number of choices can be reduced by grouping the abstract syntax terms into equivalence classes based on whether they produce the same sentences in the target language. Since all terms in a class produce the same output, the user only needs to select the correct class of unambiguous sentences.

Another source of ambiguity is that two abstract syntax terms can have distinct linearizations in the source language, but identical target language linearizations. In this case, the output sentence will be ambiguous, even though the input was unambiguous. This could be addressed by using unambiguous linearizations for system output, though this may lead to the use of unnatural constructions.

5.2 Bidirectional Translation

Since GF uses the same grammar for parsing and linearization, the grammar for a translator from L_1 to L_2 can also be used in a translator from L_2 to L_1, provided that the appropriate speech components are available. Two unidirectional translators can be used as a bidirectional translator, something which is straightforwardly achieved using two computers. While PGF interpreters can already be used for bidirectional translation, a single-device bidirectional speech translator requires multiplexing or duplicating the sound hardware.

5.3 Larger Input Coverage

GF's *variants* feature allows an abstract syntax function to have multiple representations in a given concrete syntax. This permits some variation in the input, while producing the same interlingua term. For example, the linearization of PainKiller in the English concrete syntax in Figure 3 could be changed to:

$mkNP\ indefSgDet\ (\mathbf{variants}\{$
  mkN "painkiller"; mkN "analgesic"$\});$

6 Conclusions

Because it uses a domain-specific interlingua, a GF-based speech translator can achieve high precision translation and scale to support a large number of languages.

The GF resource grammar library reduces the development effort needed to implement a speech translator for a new domain, and the need for the developer to have detailed linguistic knowledge.

Systems created with GF are highly portable to new platforms, because of the wide speech recognition grammar format support, and the availability of PGF interpreters for many platforms.

With additional work, GF could be used to implement a full-scale speech translator. The existing GF components for grammar development, speech recognition grammar compilation, parsing, and linearization could also be used as parts of larger systems.

References

Angelov, Krasimir, Björn Bringert, and Aarne Ranta. 2008. PGF: A Portable Run-Time Format for Type-Theoretical Grammars. Manuscript, `http://www.cs.chalmers.se/~bringert/publ/pgf/pgf.pdf`.

Bouillon, P., M. Rayner, N. Chatzichrisafis, B. A. Hockey, M. Santaholma, M. Starlander, H. Isahara, K. Kanzaki, and Y. Nakao. 2005. A generic Multi-Lingual Open Source Platform for Limited-Domain Medical Speech Translation. pages 5–58, May.

Bringert, Björn. 2007. Speech Recognition Grammar Compilation in Grammatical Framework. In *Proceedings of the Workshop on Grammar-Based Approaches to Spoken Language Processing*, pages 1–8, Prague, Czech Republic.

Khegai, Janna. 2006. Grammatical Framework (GF) for MT in sublanguage domains. In *Proceedings of EAMT-2006, 11th Annual conference of the European Association for Machine Translation, Oslo, Norway*, pages 95–104, June.

Ranta, Aarne. 2004. Grammatical Framework: A Type-Theoretical Grammar Formalism. *Journal of Functional Programming*, 14(2):145–189, March.

Ranta, Aarne. 2008. Grammars as software libraries. In Bertot, Yves, Gérard Huet, Jean-Jacques Lévy, and Gordon Plotkin, editors, *From semantics to computer science: essays in honor of Gilles Kahn*. Cambridge University Press.

An Integrated Dialog Simulation Technique for Evaluating Spoken Dialog Systems

Sangkeun Jung, Cheongjae Lee, Kyungduk Kim, Gary Geunbae Lee
Department of Computer Science and Engineering
Pohang University of Computer Science and Technology(POSTECH)
San 31, Hyoja-Dong, Pohang, 790-784, Korea
{hugman, lcj80, getta, gblee}@postech.ac.kr

Abstract

This paper proposes a novel integrated dialog simulation technique for evaluating spoken dialog systems. Many techniques for simulating users and errors have been proposed for use in improving and evaluating spoken dialog systems, but most of them are not easily applied to various dialog systems or domains because some are limited to specific domains or others require heuristic rules. In this paper, we propose a highly-portable technique for simulating user intention, utterance and Automatic Speech Recognition (ASR) channels. This technique can be used to rapidly build a dialog simulation system for evaluating spoken dialog systems. We propose a novel user intention modeling and generating method that uses a linear-chain conditional random field, a data-driven domain specific user utterance simulation method, and a novel ASR channel simulation method with adjustable error recognition rates. Experiments using these techniques were carried out to evaluate the performance and behavior of previously developed dialog systems designed for navigation dialogs, and it turned out that our approach is easy to set up and shows the similar tendencies of real users.

1 Introduction

Evaluation of spoken dialog systems is essential for developing and improving the systems and for assessing their performance. Normally, humans are used to evaluate the systems, but training and employing human evaluators is expensive. Furthermore, qualified human users are not always immediately available. These inevitable difficulties of working with human users can cause huge delay in development and assessment of spoken dialog systems. To avoid the problems that result from using humans to evaluate spoken dialog systems, developers have widely used dialog simulation, in which a simulated user interacts with a spoken dialog system.

Many techniques for user intention, utterance and error simulation have been proposed. However, previously proposed simulation techniques cannot be easily applied to evaluate various dialog systems, because some of these techniques are specially designed to work with their own dialog systems, some require heuristic rules or flowcharts, and others try to build user side dialog management systems using specialized dialog managing methods. These problems motivated us to develop dialog simulation techniques which allow developers to build dialog simulation systems rapidly for use in evaluating various dialog systems.

To be successful, a simulation approach should not depend on specific domains or rules. Also it should not be coupled to a specific dialog management method. Furthermore, successful dialog simulation should fully support both user simulation and environment simulation. In user simulation, it must be capable of simulating both user intentions and user utterances, because user utterances are essential for testing the language understanding component of the dialog system. In addition to user simulation, environment simulation such as ASR channel simulation is desirable because it allows developers to test the dialog system in various acoustic environments.

In this paper, we propose novel dialog simulation techniques which satisfy these requirements. We introduce a new user intention simulation method based on the sequential graphical model, and a user utterance simulator which can generate diverse natural user utterances. The user intention and utterance simulators are both fully data-driven approaches; therefore they have high domain- and language portability. We also propose a novel Automatic Speech Recognizer (ASR) channel simulator which allows the developers to set the desired speech recognition performance level. Through a case study, we showed that our approach is feasible in successful dialog simulation to evaluate spoken dialog

Coling 2008: Proceedings of the workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, pages 9–16
Manchester, August 2008

systems.

This paper is structured as follows. We first provide a brief introduction of other dialog simulation techniques and their differences from our approach in Section 2. We then introduce the overall architecture and the detailed methods of intention, utterance and ASR channel simulation in Section 3. Experiments to test the simulation techniques, and a case study are described in Section 4. We conclude with a brief summary and suggest directions for future work in Section 5.

2 Related Works

Dialog simulation techniques can be classified according to the purpose of the simulation. One of the purposes is to support the refinement of dialog strategies. Some techniques use large amounts of simulated data for a systematic exploration of the dialog state space in the framework of reinforcement learning (Schatzmann et al., 2005; Schatzmann et al., 2007a). Other techniques use simulation techniques to investigate and improve the target dialog strategies by examining the results heuristically or automatically (Chung, 2004; Rieser and Lemon, 2006; Torres et al., 2008). A second purpose of dialog simulation techniques is to evaluate the dialog system itself qualitatively. Eckert et al., (1997) and López-Cózar et., (2003; 2006) used a dialog simulation to evaluate whole dialog systems.

Dialog simulation techniques can also be classified according to the layers of the simulation. Typically, dialog simulation can be divided into three layers: user intention, user surface (utterance) and error simulation.

Some studies have focused only on the intention level simulation (Rieser and Lemon, 2006; Schatzmann et al., 2007b; Cuayahuitl et al., 2005). The main purpose of those approaches was to collect and examine intention level dialog behavior for automatically learning dialog strategies. In this case, surface and error simulations were neglected or simply accessed normally.

Another approach is to simulate both user intention and surface. In this approach, user utterance generation is designed to express a given intention. Chung (2004) tried to use the natural language generation module of (Seneff, 2002) to generate this surface. He used a speech synthesizer to generate user utterances. López-Cózar et., (2003; 2006) collected real human utterances, and selected and played the voice to provide input for the spoken dialog system. Both Chung (2004) and López-Cózar et., (2003; 2006) used rule based intention simulation. They used real ASR to recognize the synthesized or played voice; hence, ASR channel simulation is not needed in their techniques. Scheffler and Young (2000; 2001) used the lattices which are derived from the grammars used by the recognition engine, but generated user utterances by associating the lattice edges with intentions. During utterance generation, they simulated errors in recognition and understanding by probabilistic substitution on the selection of the edge. Schatzmann et al., (2007a; 2007b) proposed a statistical model for user utterance generation and error simulation using agenda based intention simulation.

The existing rule-based techniques for simulating intentions or surfaces are not appropriate in the sense of portability criteria. In addition, specific dialog managing techniques based user simulators (e.g., (Torres et al., 2008)) are not desirable because it is not easy to implement these techniques for other developers. Another important criterion for evaluating dialog simulation techniques for use in evaluating spoken dialog systems is the range of simulation layers. Simulations that are restricted to only the intention level are not sufficient to evaluate the whole dialog system. Domain and language independent techniques for simulating both intentions and utterances are needed, and ASR channel simulation is desirable for evaluating the spoken dialog systems accurately because human-machine dialog is heavily influenced by speech recognition errors.

3 Dialog Simulation Architecture for Dialog System Evaluation

3.1 Overall Architecture

Typical spoken dialog systems deal with the dialog between a human user and a machine. Human users utter spoken language to express their intention, which is recognized, understood and managed by ASR, Spoken Language Understanding (SLU) and Dialog Manager (DM). Conventionally, ASR has been considered to be a component of dialog systems. However, in this research, we do not include a real ASR module in the dialog system component because a real ASR takes only fixed level of speech as an input. To use real voices, we must either collect real human speech or generate voices using a speech synthesizer. However, both approaches have limitations. When recording and playing real human voices, the cost of data collection is high and the simulator can simulate only the behavior of the humans who were recorded. When using a speech synthesizer, the synthesizer can usually generate the speech of one person, on a limited variety of speech behaviors; this means that the dialog system cannot be evaluated under various conditions. Also, in both approaches, freely adjusting the speech recognition performance level is difficult. In this research, instead of using real speech we simulate the ASR channel and add noises to a clean utterance from the user simulator to mimic the speech recognition result.

The overall architecture of our dialog simulation separates the user simulator into two levels: user intention simulator and utterance simulator (Fig. 1). The user intention simulator accepts the discourse circumstances with system intention as input and generates the next user intention. The user utterance simulator constructs a corresponding user sentence to express the given user intention. The simulated user sentence is fed to the ASR channel simulator, which then adds noises to the utterance. This noisy utterance is passed to a dialog sys-

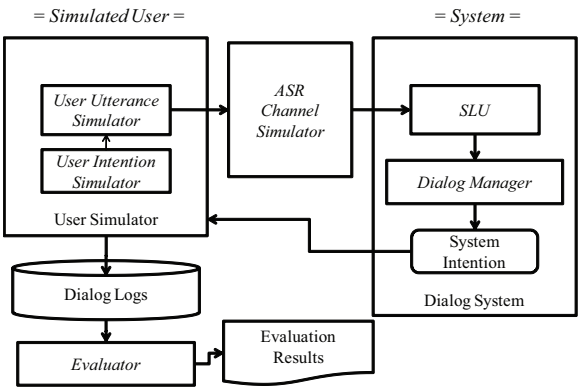


Figure 1: Overall architecture of dialog simulation

tem which consists of a SLU and a DM. The dialog system understands the user utterance, manages dialog and passes the system intention to the user simulator. User simulator, ASR channel simulator and dialog system repeat the conversation until the user simulator generates an end to the dialog.

After finishing simulating one dialog successfully, this dialog is stored in Dialog Logs. If the dialog logs contain enough dialogs, the evaluator uses the logs to evaluate the performance of the dialog system.

3.2 User Intention Simulation

The task of user intention simulation is to generate subsequent user intentions given current discourse circumstances. The intention is usually represented as abstracted user's goals and information on user's utterance (surface). In other words, generating the user's next semantic frame from the current discourse status constitutes the user intention simulation.

Dialog is basically sequential behavior in which participants use language to interact with each other. This means that intentions of the user or the system are naturally embedded in a sequential structure. Therefore, in intention modeling we must consider how to model this sequential property. Also, we must understand that the user's intention depends not only on previous n-gram user and system intentions, but also on diverse discourse circumstances, including dialog goal, the number of items, and the number of filled component slots. Sophisticated user intention modeling should be able to reflect the discourse information.

To satisfy the sequential property and use rich information for user intention modeling, we used linear-chain Conditional Random Field (CRF) model (Lafferty et al., 2001) for user intention modeling. Let Y, X be random vectors, $\Lambda = \{\lambda_k\} \in \mathrm{R}^K$ be a parameter vector, and $\{f_k(y, y', \mathbf{x_t})\}_{k=1}^K$ be a set of real-valued feature functions. Then a linear-chain CRF is a distribution of $p(\mathbf{y}|\mathbf{x})$ that takes the form

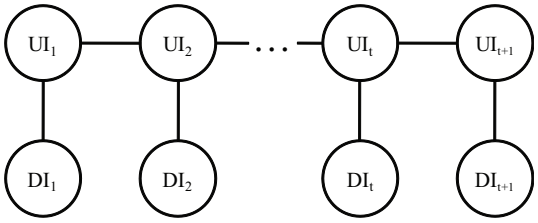


Figure 2: Conditional Random Fields for user intention modeling. UI_t: User Intention ; DI_t: Discourse Information for the tth user turn

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\Big\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x_t})\Big\} \quad (1)$$

where $Z(\mathbf{x})$ is an instance-specific normalization function.

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \exp\Big\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x_t})\Big\}$$

CRF is an undirected graphical model that defines a single log-linear distribution over the joint probability of an entire label sequence given a particular observation sequence. This single distribution removes the per-state normalization requirement and allows entire state sequences to be accounted for at once. This property is well suited to model the entire sequence of intentions in a dialog. Also, CRF is a conditional model, and not a joint model (such as the Hidden Markov Model). Arbitrary facts can be captured to describe the observation in the form of indicator functions. This means that CRF allows us to use rich discourse information to model intentions.

CRF has states and observations in each time line. We represent the user intention as state and discourse information as observations in CRF (Fig. 2). We represent the state as a semantic frame. For example in the semantic frame representing the user intention for the utterance 'I want to go to city hall' (Fig. 3), dialog_act is a domain-independent label of an utterance at the level of illocutionary force (e.g. statement, request, wh_question) and main_goal is the domain-specific user goal of an utterance (e.g. give_something, tell_purpose). Component slots represent named entities in the utterance. We use the cartesian product of each slot of semantic frame to represent the state of the utterance in our CRF model. In this example, the state symbol is 'request×search_loc×loc_name'.

For the observation, we can use various discourse events because CRF allows using rich information by interpreting each event as an indicator function. Because we pursue the portable dialog simulation technique, we separated the features of the discourse information into those that are domain independent and those that are domain dependent. Domain independent

Semantic Frame for User Inention Simulation

raw user utterance	I want to go to city hall.
dialog_act	*request*
main_goal	*search_loc*
component.[loc_name]	*cityhall*

Preprocessing Information for User Utterance Simulation

processed utterance	*I/PRP want/VB to/TO go/VB to/TO [loc_name]* /[loc_name]
Structure Tags	*PRP, VB, TO, VB, TO, [loc_name]*
Word Vocabulary	*I, want, to, go, to, [loc_name]*

Generation Target for User Utterance Simulation

Structure	*PRP → VB → TO → VB → TO → [loc_name]*
Word Sequence	*I → want → to → go → to → [loc_name]*

Figure 3: Example of semantic frame for user intention, and preprocessing and generation target for user utterance simulation.

Domain Independent Features

PREV_1_SYS_ACT	previous system action. Ex) PREV_1_SYS_ACT=confirm
PREV_1_SYS_ACT_ATTRIBUTES	previous system mentioned attributes. Ex) PREV_1_SYS_ACT_attributes=city_name
PREV_2_SYS_ACT	previous system action. Ex) PREV_2_SYS_ACT=confirm
PREV_2_SYS_ACT_ATTRIBUTES	previous system mentioned attributes. Ex) PREV_2_SYS_ACT_attributes=city_name
SYSTEM_HOLDING_COMP_SLOT	system recognized component slot. Ex) SYSTEM_HOLDING_COMP_SLOT=loc_name

Domain Dependent Features

OTHER_INFO	other useful domain dependent information Ex) OTHER_INFO(user_fav_rest)=gajokjung

Figure 4: Example feature design for navigation domain

UI_t ← user intention at t turn
S ← user intentions set ($UI_t \in S$)
$UI_1, UI_2, \ldots, UI_{t-1}$ ← already simulated user intention sequence
$DI_1, DI_2, \ldots, DI_t$ ← discourse information from 1 to t turn

For each UI_t in S
  Calculate P($UI_1, UI_2, \ldots, UI_t | DI_1, DI_2, \ldots, DI_t$)
UI_t ← random user intention from P($UI_1, UI_2, \ldots, UI_t | DI_1, DI_2, \ldots, DI_t$)

Figure 5: User intention generation algorithm

features include discourse information which is not relevant to the specific dialog domain and system. For example, previous system acts in Fig. 4 are not dependent on specific dialog domain. The actual values of previous system acts could be dependent on each dialog domain and system, but the label itself is independent because every dialog system has system parts and corresponding system acts. In contrast, domain specific discourse information exists for each dialog system. For example, in the navigation domain (Fig. 4), the current position of the user or the user's favorite restaurant could be very important for generating the user's intention. This information is dependent on the specific domain and system. We handle these features as 'OTHER_INFO'.

We trained the user intention model using dialog examples of human-machine. One training example consists of a sequence of user intentions and discourse information features in a given dialog. We collected training examples and trained the intention model using a typical CRF training method, a limited-memory quasi-Newton code for unconstrained optimization (L-BFGS) of (Liu and Nocedal, 1989).

To generate user intentions given specific discourse circumstances, we calculate the probability of a sequence of user intentions from the beginning of the dialog to the corresponding turn. For example, suppose that we need to generate user intention at the third turn (UI_3) (Fig. 2). We have previously simulated user intentions UI_1 and UI_2 using DI_1 and DI_2. In this case, we calculate the probability of $UI_1 \rightarrow UI_2 \rightarrow UI_3$ given DI_1, DI_2 and DI_3. Notice that DI_3 contains discourse information at the third turn: it includes previous system intention, attributes and other useful information. Using the algorithm (Fig. 5) we generate the user intention at turn t. The probability of $P(UI_1, UI_2, \ldots, UI_t | DI_1, DI_2, \ldots, DI_t)$ is calculated using the equation (1). In the generation of user intention at t turn, we do not select the UI_t which has higher probability. Instead, we select UI_t randomly based on the probability distribution $P(UI_1, UI_2, \ldots, UI_t | DI_1, DI_2, \ldots, DI_t)$ because we want to generate diverse user intention sequence given the same discourse context. If we select UI_t which has highest probability, user intention simulator always returns the same user intention sequence.

3.3 User Utterance Simulation

Utterance simulation generates surface level utterances which express a given user intention. For example, if users want to go somewhere and provide place name information, we need to generate corresponding utterances (e.g. 'I want to go to [place_name] or 'Let's go to [place_name]'). We approach the task of user utterance simulation by assuming that the types of structures and the vocabulary are limited when we make utterances to express certain context and intention in a specific domain, and that humans express their intentions by recombining and re-aligning these structures and vocabularies.

To model this process, we need to collect the types of structures and vocabularies. For this, we need to define the context space. We define the structure and vocabulary space as a production of dialog act and main goal. In an example of semantic frame for the utterance "I want to go to city hall" (Fig. 3), the structure and vocabulary (SV) space ID is 'request_#_search_loc', which is produced by the dialog act and the main goal. We collect structure tags, which consist of a part of speech tag, a component slot tag, and a vocabulary that corresponds to SV space. For example (Fig. 3), structure

First Phase – Generating Structures and Words given SV space

1. Repeat generate S_t based on $P_{SV}(S_{t+1}|S_t)$, until S_T = <*setence_end*> , where $S_t \in \boldsymbol{S}$, t=1,2,3,....T.
2. Generate W_t based on $P_{SV}(W_t|S_t)$, where t=1,2,3,..,T , $W_t \in \boldsymbol{V}$
3. The generation word sequence $\boldsymbol{W}$={W_1,W_2,..,W_T} is inserted into the set of generated utterance $\boldsymbol{U}$
4. Repeat *1* to *3* for *Max_Generation_Number* times, *Max_Generation_Number* is given by developers

Second Phase – Selection by measure

1.Rescore the utterance U_k in the set of $\boldsymbol{U}$ by the measure
2.Select top n-best

Figure 6: Algorithm of user utterance simulation

tags include PRP, VB, TO, VB as a part of speech tag and [loc_name] as a component slot tag. The vocabulary includes I, want, to, go, and [loc_name]. In the vocabulary, every named-entity word is replaced with its category name.

In this way, we can collect the structure tags and vocabulary for each SV space from the dialog logs. For the given SV space, we estimate probability distributions for statistical user utterance simulation using a training process. For each space, we estimate tag transition probability $P_{SV}(S_{t+1}|S_t)$ and collect structure tags set S_{SV} and vocabularies V_{SV}.

We devised a two-phase user utterance generation algorithm (Fig. 6). Symbols are as follows. The detail explanation of Fig. 6 will be followed in the next subsections.

- $\boldsymbol{S_{SV}}$: structure tag set for given SV
- $\boldsymbol{V_{SV}}$: vocabularies for given SV
- S_i : structure tag, $i = 0, ..., T$, $S_i \in \boldsymbol{S_{SV}}$
- W_i : word, $i = 0, ..., T$, $W_i \in \boldsymbol{V_{SV}}$
- W_{seq} : generated word sequence. $W_{seq} = (W_1, W_2, ..., W_T)$
- U_k : *k*-th sampled utterance, $k = 1, ...,$ *Max_Sampling_Number*, $U_k \in \boldsymbol{U}$

3.3.1 First Phase - Generating Structure and Word Sequence

We generate the structure tag S_1 based on the probability of $P_{SV}(S_1| < sentence\_start >)$ and then S_1 influences the generating of S_2 after $P_{SV}(S_2|S_1)$. In this way, a structure tag chain is generated sequentially based on the structure tag transition probability $P_{SV}(S_{t+1}|S_t)$ until the last generated structure tag S_T is $< sentence\_end >$. We assume that the current structure tag has a first order Markov property, which means that the structure tag is only influenced by the previous structure tag. After the structure tags are generated, the emission probability $P_{SV}(W_t|S_t)(w = 1, \ldots, T)$ is used to generate the word sequence given the tag sequence. We iterate the process of generating structures and word sequences sufficient times to generate many different structure tags and word sequences which may occur in real human expressions. Selecting natural utterances from the generated utterances requires an automatic evaluation metric.

3.3.2 Second Phase - Selection by the BLEU measure

To measure the naturalness of the generated utterances, we use the BLEU (Bilingual Evaluation Understudy) score (Papineni et al., 2001) which is widely used for automatic evaluation in Statistical Machine Translation (SMT). In SMT, translated candidate sentences are evaluated by comparing semantically equivalent reference sentences which have been translated by a human. Evaluation of the user utterance generation shares the same task of evaluation in SMT. We can evaluate the naturalness of generated utterances by comparing semantically equivalent reference utterances collected by humans. Therefore, the BLEU score can be adopted successfully to measure the naturalness of the utterances.

The BLEU score is the geometric mean of the n-gram precisions with a brevity penalty. The original BLEU metric is used to evaluate translated sentences by comparing them to several reference sentences. We modified the BLEU metric to compare one generated utterance with several reference utterances. To rescore the generated utterances, we used the Structure and Word interpolated BLEU score (SWB). After the first phase, we obtain generated utterances which have both structure and word sequence. To measure the naturalness of a generated utterance, we check both structural and lexical naturalness. We calculated Structure_Sequence_BLEU score using the generated structure tags sequences instead of words sequences with the reference structure tag sequences of the SV space in the BLEU calculation process. The Word_Sequence_BLEU is calculated by measuring BLEU score using the generated words sequence with the reference word sequences of the SV space. SWB is calculated as:

$$\begin{aligned} SWB \quad &= \alpha * \text{Structure\_Sequence\_BLEU} \\ &+ (1-\alpha) * \text{Word\_Sequence\_BLEU} \end{aligned}$$

In this study, we set $\alpha = 0.5$. Using SWB, we select the top 20-best generated utterances and return a corresponding generated utterance by selecting one of them randomly.

3.4 ASR channel Simulation

ASR channel simulation generates speech recognition errors which might occur in the real speech recognition process. In this study, we simulate the ASR channel and modify the generated clean utterance to a speech recognized erroneous utterance. Successful ASR channel simulation techniques should have the following properties: the developer should be able to set the simulated word error rate (WER) between 0% ˜ 100%; the simulated errors should be generated based on realistic

phone-level and word-level confusions; and the technique should be easily adapted to new tasks, at low cost.

Our ASR channel simulation approach is designed to satisfy these properties. The proposed ASR channel simulation method involved four steps: 1) Determining error position 2) Generating error types on error marked words. 3) Generating ASR errors such as substitution, deletion and insertion errors, and 4) Rescoring and selecting simulated erroneous utterances (Fig. 7 for Korean language example).

In the first step, we used the WER to determine the positions of erroneous words. For each word, we randomly generate a number between 0 and 1. If this number is between 0 and WER, we mark the word Error Word (1); otherwise we mark the word Clean Word (0). In the second step, we generate ASR error types for the error marked words based on the error type distribution. In the third step, we generate various types of ASR error. In the case of deletion error, we simply delete the error marked word from the utterance. In the case of insertion error, we select one word from the pronunciation dictionary randomly, and insert it before the error marked word. In the case of substitution error, we use a more complex process to select a substitutable word.

To select a substitutable word, we compare the marked error word with the words from pronunciation dictionary which are similar in syllable sequence and phoneme sequence. First, we convert the final word sequence from the user simulator into a phoneme sequence using a Grapheme-to-Phoneme (G2P) module (Lee et al., 2006). Then, we extract a part of the phoneme sequence which is similar to the error marked word from the entire phoneme sequence of the utterance. The reason for extracting a target phoneme sequence corresponding to one word from the entire phoneme sequence is that the G2P results vary between the boundaries of words. Then, we separate the marked word into syllables and compare their syllable-level similarity to other words in the pronunciation dictionary. We calculate a similarity score which interpolates syllable and phoneme level similarity using following equations.

$$\begin{aligned} Similarity \quad &= \beta * \text{Syllable\_Alignment\_Score} \\ &+ (1-\beta) * \text{Phone\_Alignment\_Score} \end{aligned}$$

We used the dynamic global alignment algorithm of (Needleman and Wunsch, 1970) for both syllable and phoneme sequence alignment. This alignment algorithm requires a weight matrix. As a weight matrix, we used a vowel confusion matrix which is based on the manner of articulation. We consider the position (back/front, high/mid/low) of the tongue and the shape (round/flat) of the lips. We select candidate words which have higher similarity than an arbitrary threshold θ and replace the error marked word with a random word from this set. We repeat steps 1 to 3 many times (usually 100) to collect error added utterances.

In the fourth step, we rescore the error added utterances using the language model (LM) score. This LM is trained using a domain corpus which is usually used in ASR. We select top n-best erroneous utterances (we set n=10) and choose one of them randomly. This utterance is the final result of ASR channel simulator, and is fed into the dialog system.

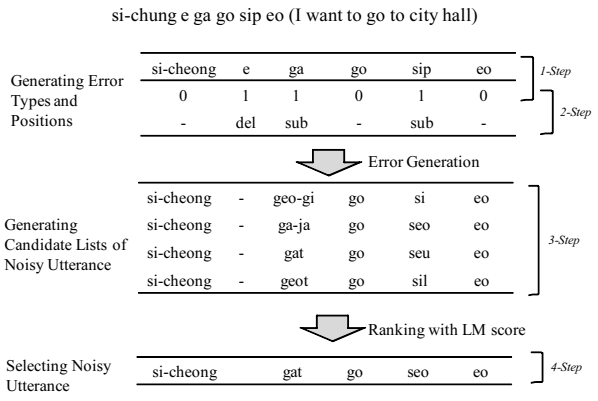


Figure 7: Example of ASR channel simulation

4 Experiments

We proposed a method that user intention, utterance and ASR channel simulation to rapidly assemble a simulation system to evaluate dialog systems. We conducted a case study for the navigation domain Korean spoken dialog system to test our simulation method and examine the dialog behaviors using the simulator. We used 100 dialog examples from real user and dialog system to train user intention and utterance simulator. We used the SLU method of (Jeong and Lee, 2006), and dialog management method of (Kim et al., 2008) to build the dialog system. After trained user simulator, we perform simulation to collect 5000 dialog samples for each WER settings (WER = 0 ˜ 40 %).

To verify the user intention and utterance simulation quality, we let two human judges to evaluate 200 randomly chosen dialogs and 1031 utterances from the simulated dialog examples (WER=0%). At first, they evaluate a dialog with three scale (1: Unnatural, 2: Possible, 3: Natural), then evaluate the utterances of a dialog with three scale (1: Unclear, 2: Understandable, 3: Natural).

The inter evaluator agreement (kappa) is 0.45 and 0.58 for dialog and utterance evaluation respectively, which show the moderate agreement (Fig. 8). Both judges show the positive reactions for the quality of user intention and utterance, the simulated dialogs can be possibly occurred, and the quality of utterance is close to natural human utterance.

We also did regression analysis with the results of human evaluation and the SWB score to find out the relationship between SWB and human judgment. Fig. 9 shows the result of polynomial regression (order 3) result. It shows that 'Unclear' utterance might have 0.5

	Human 1	Human 2	Average	Kappa
Dialog	2.38	2.22	2.30	0.45
Utterance	2.74	2.67	2.71	0.58

Figure 8: Human evaluation results on dialog and utterance

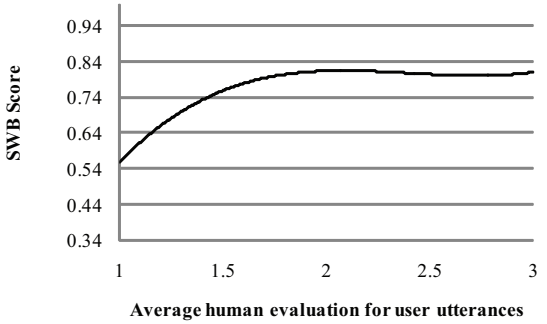
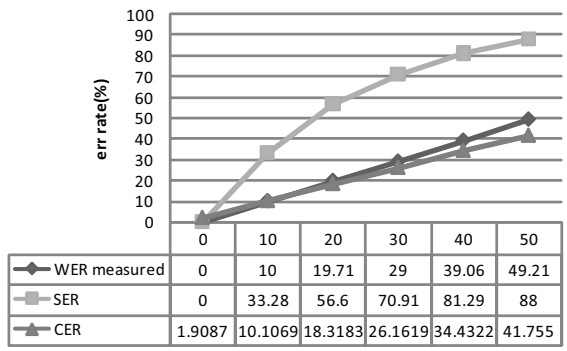


Figure 9: Relationship between SWB score and human judgment

˜ 0.7 SWB score, 'Possible' and 'Natural' simulated utterance might have over 0.75. It means that we can simulate good user utterance if we constrain the user simulator with the threshold around 0.75 SWB score.

To assess the ASR channel simulation quality, we compared how SLU of utterances was affected by WER. SLU was quantified according to sentence error rate (SER) and concept error rate (CER). Compared to WER set by the developer, measured WER was the same, SER increased more rapidly, and CER increased more slowly (Fig. 10). This means that our simulation framework models SLU errors effective as well as speech recognition errors.

Fig. 11 shows the overall dialog system behaviors using the user simulator and ASR channel simulator. As the WER rate increased, dialog system performance decreased and dialog length increased. This result is similar as observed to the dialog behaviors in real human-machine dialog.



	0	10	20	30	40	50
WER measured	0	10	19.71	29	39.06	49.21
SER	0	33.28	56.6	70.91	81.29	88
CER	1.9087	10.1069	18.3183	26.1619	34.4322	41.755

Figure 10: Relationship between given WER and measured other error rates. X-axis = WER fixed by ASR channel(%)

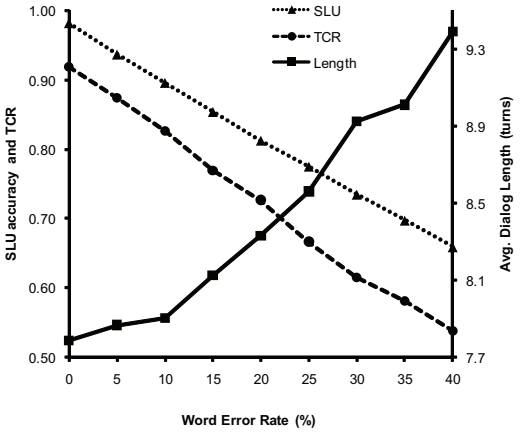


Figure 11: Dialog simulation result on navigation domain

5 Conclusion

This paper presented novel and easy to build dialog simulation methods for use in evaluation of spoken dialog systems. We proposed methods of simulating utterances and user intentions to replace real human users, and introduced an ASR channel simulation method that acts as a real speech recognizer. We introduce a method of simulating user intentions which is based on the CRF sequential graphical model, and an utterance simulator that generates user utterances. Both user intention and utterance simulators use a fully data-driven approach; therefore, they have high domain- and language portability. We also proposed a novel ASR channel simulator which allows the developers to set the speech recognition performance level. We applied our methods to evaluate a navigation domain dialog system; experimental results show that the simulators successfully evaluated the dialog system, and that simulated intention, utterance and errors closely match to those observed in real human-computer dialogs. We will apply our approach to other dialog systems and bootstrap new dialog system strategy for the future works.

6 Acknowledgement

This research was supported by the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Knowledge Economy of Korea.

References

Chung, G. 2004. Developing a flexible spoken dialog system using simulation. *Proc. ACL*, pages 63–70.

Cuayahuitl, H., S. Renals, O. Lemon, and H. Shimodaira. 2005. Human-Computer Dialogue Simulation Using Hidden Markov Models. *Automatic*

Speech Recognition and Understanding, 2005 IEEE Workshop on, pages 100–105.

Eckert, W., E. Levin, and R. Pieraccini. 1997. User modeling for spoken dialogue system evaluation. *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 80–87.

Jeong, M. and G. Lee. 2006. Jointly Predicting Dialog Act and Named Entity for Statistical Spoken Language Understanding. *Proceedings of the IEEE/ACL 2006 workshop on spoken language technology (SLT)*.

Kim, K., C. Lee, S Jung, and G. Lee. 2008. A frame-based probabilistic framework for spoken dialog management using dialog examples. In *the 9th sigdial workshop on discourse and dialog (sigdial 2008), To appear*.

Lafferty, J.D., A. McCallum, and F.C.N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the Eighteenth International Conference on Machine Learning table of contents*, pages 282–289.

Lee, J., S. Kim, and G.G. Lee. 2006. Grapheme-to-Phoneme Conversion Using Automatically Extracted Associative Rules for Korean TTS System. In *Ninth International Conference on Spoken Language Processing*. ISCA.

Liu, D.C. and J. Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528.

López-Cózar, R., A. De la Torre, JC Segura, and AJ Rubio. 2003. Assessment of dialogue systems by means of a new simulation technique. *Speech Communication*, 40(3):387–407.

López-Cózar, Ramón, Zoraida Callejas, and Michael Mctear. 2006. Testing the performance of spoken dialogue systems by means of an artificially simulated user. *Artif. Intell. Rev.*, 26(4):291–323.

Needleman, SB and CD Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–53.

Papineni, K., S. Roukos, T. Ward, and WJ Zhu. 2001. BLEU: a method for automatic evaluation of MT. *Research Report, Computer Science RC22176 (W0109-022), IBM Research Division, TJ Watson Research Center*, 17.

Rieser, V. and O. Lemon. 2006. Cluster-Based User Simulations for Learning Dialogue Strategies. In *Ninth International Conference on Spoken Language Processing*. ISCA.

Schatzmann, J., K. Georgila, and S. Young. 2005. Quantitative Evaluation of User Simulation Techniques for Spoken Dialogue Systems. In *6th SIGdial Workshop on Discourse and Dialogue*. ISCA.

Schatzmann, J., B. Thomson, and S. Young. 2007a. Error simulation for training statistical dialogue systems. *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 526–531.

Schatzmann, J., B. Thomson, and S. Young. 2007b. Statistical User Simulation with a Hidden Agenda. *Proc. SIGDial, Antwerp, Belgium*.

Scheffler, K. and S. Young. 2000. Probabilistic simulation of human-machine dialogues. *Proc. of ICASSP*, 2:1217–1220.

Scheffler, K. and S. Young. 2001. Corpus-based dialogue simulation for automatic strategy learning and evaluation. *Proc. NAACL Workshop on Adaptation in Dialogue Systems*, pages 64–70.

Seneff, S. 2002. Response planning and generation in the Mercury flight reservation system. *Computer Speech and Language*, 16(3):283–312.

Torres, Francisco, Emilio Sanchis, and Encarna Segarra. 2008. User simulation in a stochastic dialog system. *Comput. Speech Lang.*, 22(3):230–255.

Economical Global Access to a VoiceXML Gateway Using Open Source Technologies

Kulwinder Singh, Dong-Won Park
Dept of Information & Communications Engineering
PaiChai University, Daejeon, South Korea
{singh, dwpark}@pcu.ac.kr

Abstract

Voice over IP and the open source technologies are becoming popular choices for organizations. However, while accessing the VoiceXML gateways these systems fail to attract the global users economically. The objective of this paper is to demonstrate how an existing web application can be modified using VoiceXML to enable non-visual access from any phone. Moreover, we unleash a way for linking an existing PSTN-based phone line to a VoiceXML gateway even though the voice service provider (VSP) does not provide a local geographical number to global customers to access the application. In addition, we introduce an economical way for small sized businesses to overcome the high cost of setting up and using a commercial VoiceXML gateway. The method is based on Asterisk server. In order to elucidate the entire process, we present a sample Package Tracking System application, which is based on an existing website and provides the same functionality as the website does. We also present an online demonstration, which provides global access to commercial voice platforms (i.e. Voxeo, Tellme Studio, Bevocal and DemandVoice). This paper also discusses various scenarios in which spoken interaction can play a significant role.

1 Introduction

The end of the 20th century witnesses an explosive growth in Internet usage. We have seen an explosion in the number of browser-based visual applications, from the broad examples we use every day, such as e-commerce, movie or flight schedules, and financial information. The most common means for accessing information residing on many websites across the globe is still the dominating interface of point and click with a mouse using the graphical user interface (GUI). Additionally, telephone is also widely used to access information. Still, in densely populated countries it seems to be difficult to handle large amounts of calls simultaneously, which leads to long call queues and frustrated customers.
However, the challenge that is presented to the present Internet world is to make the enormous web content accessible to users who don't have the computers or maybe don't have the money to buy as well as visually impaired users. Since speech is the most natural means of communication, especially for these users, voice will be a dominating mode in newly designed multi-modal (Oviatt, S.L., 1999) user interfaces for future devices. This calls for a revolutionary design of a voice user interface (VUI) to supplement the conventional GUIs. Internet and telephony used to be two separate technologies to build applications accessible over the phone. VoiceXML bridges the gap; it leverages the existing web infrastructure and enables web developers to build voice-enabled web applications accessible from any telephone, by anyone, anywhere, anytime. A major advantage of VoiceXML is that it provides web content over a simple telephone device, making it possible to access an application even without a computer and an Internet connection. VoiceXML finds ready acceptance in the business world due to the following reasons.

Providing a voice-based interface with the web interface is an advantage to the visually challenged who are unable to use a visual interface. It is also possible to use the application for accessing a web-based interface even while on the move through a mobile phone, which is much easier to carry around than a personal computer. Phone applications are more spontaneous. Most people these days always have their phone on their hip. In many cases, the phone transaction can be completed before the PC even boots or you can log in. Lastly, there is no possibility of a virus from a phone call and it is typically much more secure.

The number of telephone users is far greater than the number of people who use personal computers or the Internet.

Thus, by using VoiceXML applications, we can reach out to more customers than is possible by using the Internet. Voice portals put all kinds of information at a consumer's fingertips anytime, anywhere. Customers just dial into the voice portal's 800 number and use simple voice commands to access whatever information they need. It's quick, easy, and effective, even from a car or the airport. However, it still fails to attract the huge global customers as they have to pay the long distance calling charge to access the information. Hence, this paper is an attempt to peep behind the curtain and analyze the market trends and thereby proposes a solution to resolve the current issues and satisfy the global customers by providing them a solution to access the VoiceXML gateway economically. The structure of this paper is as follows. In the

Coling 2008: Proceedings of the workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, pages 17–23
Manchester, August 2008

next section we present the voice hosting infrastructure. We then discuss our experimental results and finally conclude by presenting the scenario for using Voice User Interfacing followed by the summary of the outcome.

2 Voice Hosting Infrastructure

A voice hosting infrastructure requires many interlocking components such as telephony hardware, software: TTS (text to speech, ASR (automatic speech recognition), networking technology, monitoring and administrative services. We discuss all the essential elements bellow.

2.1 Linking

Most of the VoiceXML gateways (Ruiz, Q. Sanchez, M. 2003) can operate VoiceXML speech applications on any standard web server and can support both static and dynamic content, and provide a high degree of scalability and platform-independence. Also, voice applications can be seamlessly integrated into existing enterprise web and IT infrastructure. There are two ways to accomplish the task:

-Link your existing web server with VSP's voice gateways.

-Port your web applications to VSP's web server.

Linking an existing web application with VoiceXML gateways is fairly straightforward. As you see in figure 1, when a VoiceXML gateway receives a phone call, it looks at the number dialed to lookup the URL of the web server, then sends the HTTP request. You need to provide the URL of your web server to VSP. One VSP provides Web-based GUI for linking an application as shown in Figure 1.

There may be some changes required to your Web server before you connect with your VSP. Changes vary from VSP to VSP, or depending on your service provider and type of Web server. As an example, our application residing on an Apache HTTP Server, according to Bevocal, must modify the httpd.conf file to add the new MIME type in the following way.

 # AddType allows you to add to or override the MIME configuration.
 # file mime.types for specific file types.
 # MIME types for VoiceXML-related content.

AddType application/voicexml+xml	*.vxml*
AddType application/srgs	*.gram .srgs*
AddType application/srgs+xml	*.grxml*
AddType application/x-nuance-gsl	*.gsl .grammar*
AddType application/x-nuance-dynagram-binary	*.ngo.*

2.2 Mapping

Speech-enabled Internet portals, or voice portals, are quickly becoming the hottest trend in e-commerce-broadening access to Internet content to everyone with the most universal communications device of all, the telephone. Currently, voice hosting providers set up local toll free numbers or DID (direct inward dialing) numbers in order to access voice applications through their VoiceXML gateways. If the VSP is unable to provide the local DID numbers in the desired country, the users from that country have to pay international calling charges, which is sometimes quite expensive. We propose our idea to resolve this issue as follows.

SIP Mapping: It totally depends upon the telephony infrastructure of the VoiceXML gateway. If it is asterisk-based (Meggelen, J. V. Madsen, L. Smith J. 2007) then the job is fairly easy, otherwise it could be a tedious task to configure a VoiceXML gateway with a remote telephony network. Our proposed idea is independent of any kind of telephony infrastructure, provided it supports SIP signaling.

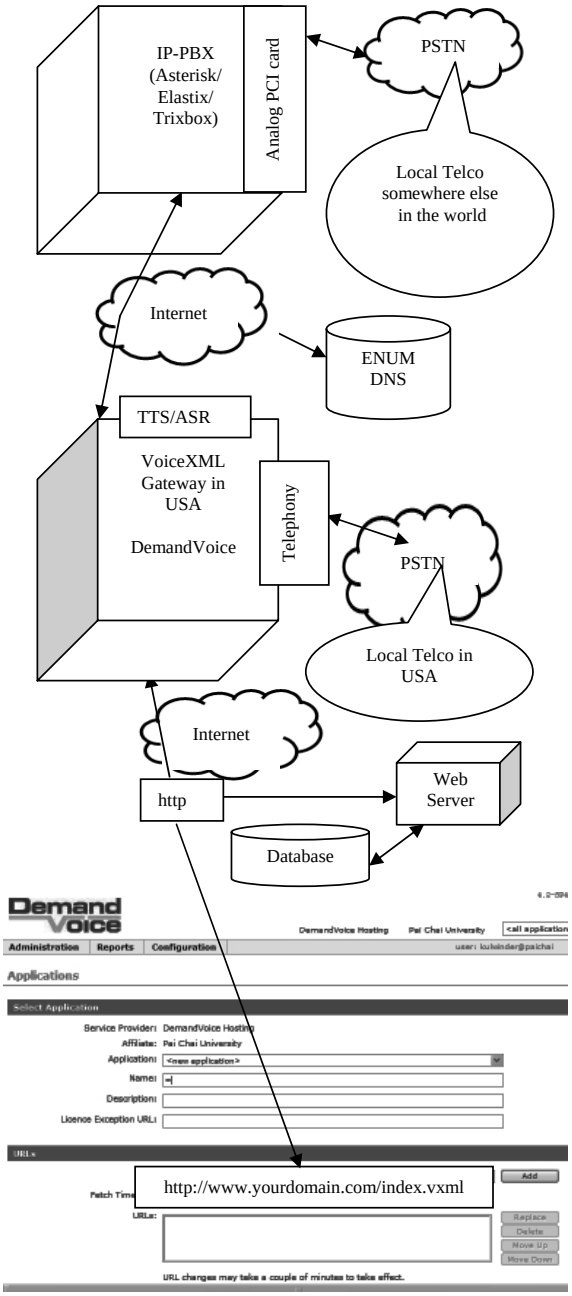


Figure 1. Linking a web server with a VoiceXML gateway

The most promising way to connect a VoiceXML gateway with a third party's Asterisk server (any IP-PBX) is to use the ENUM service. In order to use ENUM DNS efficiently, there are few steps needed to be followed. First of all, at e164.org (Ruiz, Q. Sanchez, M. 2003), in the ENUM database, we need to register the IP address and DID number, which is landing on your SIP extension of VoiceXML Gateway, as depicted in the figure 2.

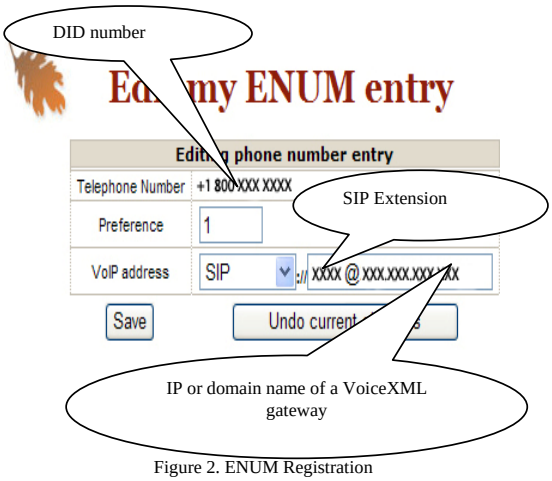


Figure 2. ENUM Registration

After editing the ENUM (tElephone NUmber Mapping) entry, we set up the ENUM trunk and outbound route on the remote IP-PBX machine.

We are running Elastix IP-PBX (elastix.org) on the remote side because it is easy to manage the configuration through GUI on Elastix. Moreover, it is an open source technology, and comes with a self-installing package that installs a complete operating system, Asterisk PBX, FreePBX etc.

9|.XXXXXXXXXXXX (X matches any digit from 0 to 9)

According to our dial plan shown above, let's assume that we need to dial the American DID number 641-543-6745, and dialing pattern would be like: 916415436745.

Our DID number 641-543-6745 is registered at e164.org. This means that when someone calls the DID, the call will land on the SIP number instead of DID number, as the *e164.org* DNS zone will provide the IP addressing and protocol information needed to connect to your VoiceXML gateway . In other words, the call will not go over the DID provider's network (see figure 3).

There would be a native or Packet2Packet SIP bridging between the VoiceXML gateway and remote IP-PBX. Ultimately, VSP and remote client will not pay any toll to PSTN operator or ITSP (Internet Telephony Service Provider) because the call bypasses their network. Moreover, the VSP does not need to open all the credentials of telephony setup of the VoiceXML gateway. So, most of the information will be isolated from the remote client. This is attractive to the VSP that does not want to register the sip number and IP address of the gateway in the ENUM (tElephone NUmber Mapping) database, (because some people are afraid to disclose their IP addresses to others). Moreover, they do not want to accept anonymous SIP calls, and want to run their own IP-PBX instead of using client's IP-PBX. In that case, we propose a very easy solution to set up the SIP extension on the VoiceXML

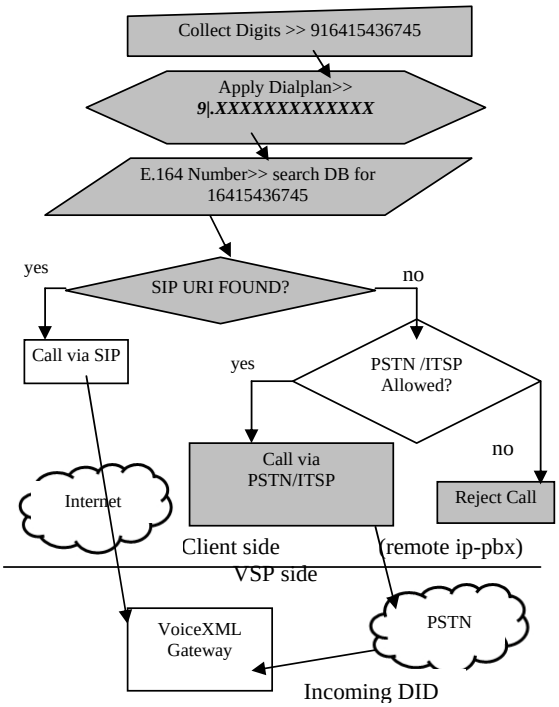


Figure 3. Flow chart of the call logic

gateway and configure it on the remote IP-PBX in the custom extension as shown in figure 4.

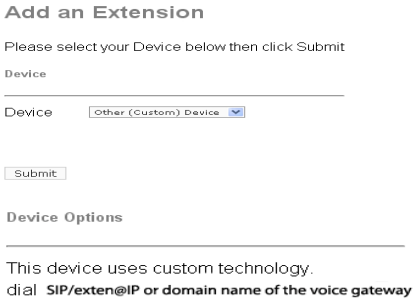


Figure 4. Custom extension settings

Our IP-PBX is connected with Bevocal, Tellme Studio, Voxeo and DemandVoice. So, our Device Options look like as follows

SIP/8773386225@voip.cafe.bevocal.com
SIP/8005558965@sip.studio.tellme.com

Both of the above mentioned methods are really good if VSP does not want to use a remote IP-PBX for outbound calls.

On the contrary, when VSP wants to setup outbound calls on the remote machine, we propose another idea to accomplish the task. Fortunately, this is very easy to configure the

machines on both sides, if a telephony infrastructure uses an asterisk-based PBX on both ends.
In this scenario, we can register the machines with each other using username and secret or we can use IP-based authentication without registering with each other. Actually, it is very easy on Elastix because it uses a Freepbx for configuring most of the tasks of Asterisk server.
In other words, it's becoming less and less common to have static IP addresses. So, if you have a dynamic IP address it is good to go with username and secret. Typically, we have to deal with sip.conf and extensions.conf on Asterisk, provided you use sip protocol. For a sample configuration code (Meggelen, J. V. Madsen, L. Smith J. 2007) see subsection *DID Mapping*.

DID Mapping: We have two scenarios to deal with: a) When a VoiceXML gateway does not support SIP signaling. b) When VSP wants to land the calls only on a DID number assigned for your application execution.

First, if it is a toll free DID number then the remote client can dial through ENUM in order to connect with a toll free gateway, and call will land on the toll free network, which is connected with a VoiceXML gateway (see figure 5). It means a toll free subscriber has to pay for it, and the call between a remote IP-PBX and the toll free gateway would be free, because it will go over the internet.

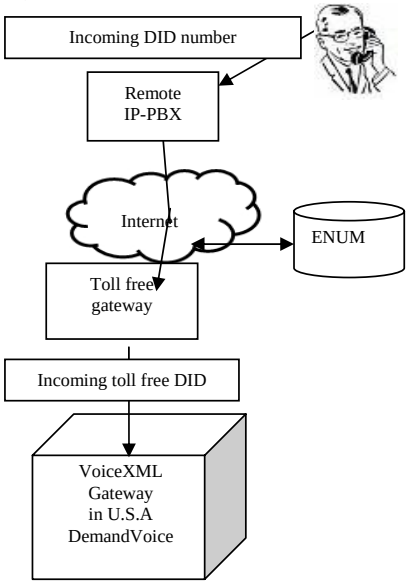


Figure 5. Remote toll free connectivity

For example, we connect DemandVoice's voice gateway using a toll free DID number remotely as follows:
Set up the custom extension as we discussed in subsection *SIP Mapping* , and it will directly connect with a toll free gateway (see figure 6).

SIP/8008042865@sip.tollfreegateway.com

Or you can dial through ENUM as we discussed in subsection *SIP Mapping*.
If it is a DID number and has no registration in the ENUM database then you need to originate the call using your ITSP, and the call will directly land on your DID assigned for your application by VSP. With the advent of VOIP technology,

```
Launched AGI Script /var/lib/asterisk/agi-bin/fixlocalprefix
AGI Script fixlocalprefix completed, returning 0
Executing [s@macro-dialout-enum:12] AGI("SIP/ remote-ip-pbx -", "enumlookup.
Launched AGI Script /var/lib/asterisk/agi-bin/enumlookup.agi
 enumlookup.agi: Looking up 18008042865 on e164.org via dns_get_record
 enumlookup.agi: Looking up 18008042865 on e164.arpa via dns_get_record
 enumlookup.agi: Looking up 18008042865 on e164.info via dns_get_record
 enumlookup.agi: Setting DIALARR to sip/164164180080428658sip.tollfreegat
AGI Script enumlookup.agi completed, returning 0

Called 164164180080428658sip.tollfreegateway.com
SIP/sip.tollfreegateway.com-09cdd8f8 is ringing
SIP/sip.tollfreegateway.com-09cdd8f8 is ringing
SIP/sip.tollfreegateway.com-09cdd8f8 answered SIP/
Packet2Packet bridging SIP/ remote-ip-pbx    and SIP/sip.tollfreegateway.com
```

Figure6. Asterisk CLI

there has been a flood of ITSP (Internet Telephony Service Provider) all over the world. It is really hard to choose one. We have tested the following configuration using our Static IP address on Elastix with VTWhite (Internet Telephony Service Provider) for VOIP termination and origination.

Peer Details:
allow=ulaw
canreinvite=no
context=from-pstn
disallow=all
dtmf=rfc2833
dtmfmode=rfc2833
host=sip.vtwhite.com
insecure=very
nat=yes
qualify=yes
sendrpid=yes
type=peer

Since our IP address is registered with VTWhite.com, there is no need for more typical authentication or registration parameters.

Inbound Routes:

DID number: 1XXXXXXXXX (11 digits)

Set destination for incoming calls landing on your DID. If you are dialing out through VTWhite you must set your outbound CID as follows:

"1XXX-XXX-XXXX"<1XXXXXXXXXX>

We have tested the following configuration with voiptalk.org (Internet Telephony Service Provider) using username and secret.

Peers Details:
host=voiptalk.org
insecure=very
secret=XXXX
type=peer
username=XXXX
username:secret@voiptalk.org/username

2.3 Porting

Many organizations have their existing toll free phone numbers, and they want to connect their existing numbers with a voice portal, and don't like to get a new phone number. Luckily, it is very easy in the United States to port the number from one carrier to another carrier. There are

many independent "RESPORG" (RESPonsible ORGanization) companies, which help for porting the numbers.

If there are issues for porting the existing number, we propose a very simple idea to install an asterisk-based IP-PBX at your premises and route the calls landing on your existing number to VoiceXML gateway using a sip or ITSP as we have discussed in section 2.2 *Mapping*.

2.4 Editing

Adding VoiceXML interface (Tsai, M.-J. 2005) (Kenneth, R. A. 2001) (Yankelovich, N., 2000) to web contents presents unique challenges to designers. Complexity depends upon the web application's architecture. In this section, we demonstrate how to modify an existing package tracking web site powered by a relational database. We use PHP, MySQL, Apache web server, and these tools are widely used in web applications development, because these are cross-platform and open source technologies. There are a couple of ways to add voice user interfacing (VUI). It is possible to add VoiceXML tags either on the fly when the VoiceXML interpreter extracts the contents from the web server or in other case tags can be embedded into an existing web page. However, we concentrate only on the latter case. First of all let's have a look on a web application (see figure 7) (Tracking number: 6754356786). This application is available on the following URL for demonstrating the task. http://biometrics.pcu.ac.kr/demo/search1.php

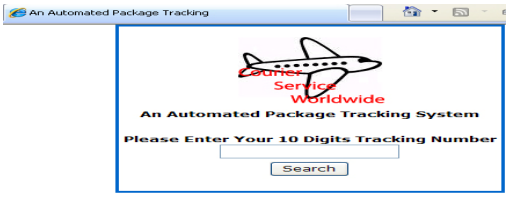


Figure 7. WEB-GUI for tracking the package

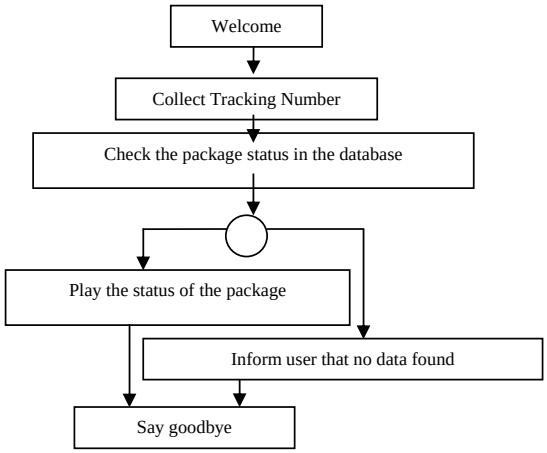


Figure 8. Call flow diagram for a VUI design

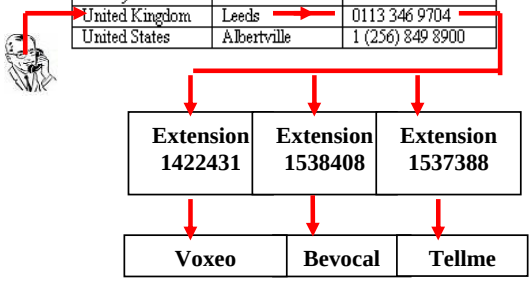
We design the call flow diagram (see figure 8) of the package tracking application as follows: According to our flow chart we need to make two VoiceXML documents. Before adding the VoiceXML tags into your webpage you must check with your VSP how to specify the document type definition (DTD) in your VoiceXML documents. Since our application is linked with Bevocal platform (BeVocal Café, 2007), we do the following way.

Collecting the Tracking number-Voicexml Document-1 (index.vxml) (see Appendix A).

Tracking Report-Voicexml Document-2 (track.php) (see Appendix B).

Table 1. Geographical Distribution of Phone Numbers for Accessing VoiceXML Gateways

Argentina	Buenos Aires	011 50314720
Australia	Sydney	02 90372744
Bahrain	National	0 16199009
Belgium	Brussels	02 7470254
Brazil	Campinas	019 31192787
Bulgaria	Sofia	02 4917134
Canada	Toronto	1 (647) 723 3640
Chile	Santiago	02 5821844
Cyprus	Nicosia	22 022630
Czech Republic	Prague	02 46019148
Denmark	National	77345753
Estonia	National	6681372
Finland	Helsinki	09 42597847
France	Paris	01 72898101
Germany	National	0180 35350033089
Guatemala	Guatemala City	02 3560986
Hungary	Budapest	01 9994901
Ireland	Dublin	01 6575613
Israel	Jerusalem	02 5695205
Italy	Rome	06 99268160
Japan	Tokyo	03 45903116
Latvia	Riga	7 661061
Lithuania	Vilnius	05 2111750
Luxembourg	National	2 0202381
Mexico	Mexico City	055 11689854
Netherlands	Amsterdam	020 8908243
New Zealand	Auckland	09 4427385
Norway	Oslo	02 1543295
Pakistan	Islamabad	0 51 8080931
Peru	Lima	01 7061950
Poland	Warsaw	022 3988047
Portugal	Porto	022 1451091
Romania	Bucharest	021 5398124
Spain	Barcelona	93 3905484
Sweden	Stockholm	08 52500225
Switzerland	Geneva	022 5330324
Thailand	Bangkok	02 1013109
Turkey	Istanbul	0212 4141710
United Kingdom	Leeds	0113 346 9704
United States	Albertville	1 (256) 849 8900



Now, it is time to call the application using a phone. We provide PSTN numbers from 40 courtiers to access the VoiceXML gateway of Bevocal, Tellme Studio, DemandVoice and Voxeo. In order to test the sample

package tracker you need to dial extension (1538408) for Bevocal after dialing the local number as depicted in Table 1. You need PIN: 1234 and Developer ID: 5369574 to access our application. User can also call our application from the following numbers without dialing any pin or extension numbers.

Direct Numbers:

Italy	Rome	06 916507970
United Kingdom	Leeds	0113 350 8176
United Kingdom	Manchester	0161 660 4556
United States	Bellevue	1 425 998 0503

We will try to keep alive these Geographical Distributed numbers for public use. Developers and researchers can test their applications by paying just local charges applied by the terminating PSTN operator.

3. Scenarios for Using Voice User Interfacing

Despite the availability of various media of communication utilized in human computer interaction, people tend to prefer the more social medium of communication such as voice. With the advent of the Internet, the PC has become the most preferred device which people turn to when they need to enquire for information. On the interaction side, the telephone seems to remain the best example for usability preferred by the various classes of users. So, to power of voice communication, with the richness of the Internet on one side, and the usability of the phone device on the other side, we present various situations in which VUI can be of great utility.

Situations:

- Driving
- No internet service
- Visually Impaired persons
- Replacement of human operators

VUI is the most time efficient modality for input, because voice input is nimbler than typing. VUI can be used to check and answer web emails while driving a vehicle. Another class of situations is when there is no Internet or PC available and the user needs to access web applications such internet banking, parcel tracking, directory assistance, online reservation, order status enquiry, instant messaging, electronic voting, dating/chat services, and information services. Moreover, visually impaired people can take advantage of the above mentioned services just over the regular phone. Furthermore, in many situations cost efficiency can be increased by replacing human operators in call centers and offices with a VoiceXML-based interactive voice response system.

4. Experimental Results

To verify the performance of our proposed idea, we implemented an IP-PBX, an automated package tracker and the business listing search using VoiceXML, PHP, and MySQL. Then, we linked remotely with various VoiceXML gateways, and tried to call the application using different codecs (ulaw, g729, gsm). We found that ulaw codec is much better for interacting with the ASR engine, and also it provides the best voice quality since it uses no compression. This means that it has the lowest latency and the highest MOS (Mean Opinion Score) because there are no additional processing delays. However, it requires high bandwidth, and this can be easily managed via proper network provisioning. The compression has very adverse affect on speech recognition when it comes to deal with the ASR engine. The more compression is used, the more characters will be lost. Fortunately, ulaw is immune to this effect. Table 2 shows the call volume according to bandwidth and codec. Table 3 shows the hardware and software specifications.

Table 2. VOIP codec and their utilization

Codec	Bandwidth used per Call	Calls per megabit
ulaw	79.7kbps	14
g729	29.0 kbps	100
gsm	34.2 kbps	67

Table 3. Hardware and software specifications

Component	Description
CPU	Intel(R) Xeon(TM) 2.80GHz
RAM	1 GB
Telephony Board	Sangoma A200/REMORA 4 port FXO/FXS
OS	Linux centos kernel 2.6.18
Web Server	Apache
VoiceXML platform	DemandVoice, Tellme, Bevocal and Voxeo

5. Conclusion

In this paper we have targeted the large number of international users who are deprived of taking the advantage of using the toll free number remotely, and have introduced an economical way to access VoiceXML gateways globally. Moreover, our globally distributed PSTN numbers are available to access VoiceXML platform for only research, test and educational purpose. We conclude that the call quality may differ depending upon the different feature sets (e.g., codecs) and network bandwidth available. In order to get a nice connectivity with a VoiceXML gateway, the call should pass through minimum VOIP gateways. Currently, we are developing a virtual user agent based on ATOM/RSS protocol, which can be accessed by phone globally for accessing information.

Appendix

A VoiceXML Document-1

```
<?xml version="1.0"?>
<!DOCTYPE vxml PUBLIC "-//BeVocal Inc//VoiceXML 2.0//EN"
 "http://cafe.bevocal.com/libraries/dtd/vxml2-0-bevocal.dtd">
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
   <form id="login">
      <field name="t_number" type="digits">
```

```
      <prompt>
        Welcome to <emphasis>Department of Information and
Communication Engineering, PaiChai
University, South Korea</emphasis>.This demo version of Parcel
tracking system is developed by
<emphasis>Mr. Singh </emphasis>.This research work is partially
sponsored by<emphasis>Demand voice dot com
</emphasis><break
size="medium"/>
Please enjoy the music while I connect you with a package tracking
system.
<audio src="http://biometrics.pcu.ac.kr/demo/m3.wav"></audio>
Welcome to an automated parcel tracking system. Please tell me the
10 digits tracking number of your
package.
      </prompt>
      <filled>
        <prompt>
        The tracking number you entered is
        <say-as type="number:digits"> <value
expr="t_number"/></say-as>
        Please wait while I'm checking this package's status.
       <audio
src="http://biometrics.pcu.ac.kr/demo/wait.wav"></audio>
       </prompt>
        <submit next="http://biometrics.pcu.ac.kr/demo/track.php"
method="post"
namelist="t_number"/>
      </filled>
      <noinput>
        I'm sorry, I am not familiar with your accent.  Now you can
just type the 10 digits tracking number from the key pad of your
phone.
        <reprompt/>
      </noinput>
      </field>
   </form>
</vxml>
```

B Voicexml Document-2

```
<?xml version="1.0"?>
<!DOCTYPE vxml PUBLIC "-//BeVocal Inc//VoiceXML 2.0//EN"
"http://cafe.bevocal.com/libraries/dtd/vxml2-0-bevocal.dtd">
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
<form><block> <prompt> <voice gender="male">
<?php
header("Content-type: application/voicexml+xml");
$number = trim($_POST['t_number']);
   $host = "hostname";
   $user = "db_user";
   $pass = "user_pass";
   $db = "db_name";
   $link = @mysql_connect($host, $user, $pass, $db) or die
("Unable to connect.");
   mysql_select_db($db) or die ("Unable to select database!");
$sql = "SELECT * from track WHERE t_number = '$number' ";
$result = mysql_query($sql);
if (!$result) {
  echo "Could not successfully run query ($sql) from DB: " .
mysql_error();
}
elseif(mysql_num_rows($result) == 0)
{
  echo "I could not find any information for that package. Thank
you for using the telephone package tracker.Good bye";
}
else
{
while ($Row = mysql_fetch_assoc($result))
     {
echo "The following events were reported for package number.";
?>
<say-as type="number:digits">
<?php
echo " $Row[t_number]";
?>
</say-as>
<break size="medium"/>
<?php
echo "$Row[t_status]";
?>
<break size="medium"/>
<?php
echo "$Row[t_address]";
?>
<break size="medium"/>
<say-as type="date:ymd">
<?php
echo "$Row[t_date]";
?>
</say-as>
<?php
echo "Thank you for using the telephone package tracker. Good
bye";
     }
}
mysql_free_result($result);
mysql_close($link);
?>
   </voice> </prompt></block>
   </form>
</vxml>
```


Acknowledgment

We would like to express our gratitude to Ashraf Alattar, PaiChai University, South Korea, and Mark Rayburn, Demandvoice.com, USA for their help in designing the network and for participating in many useful discussions.

References

Tsai, M.-J. 2005. *The VoiceXML Dialog System for the E-Commerce Ordering Service*, IEEE Proceedings of the Ninth International Conference.

Ruiz, Q. Sanchez, M. 2003. *Design of a VoiceXML Gateway*, Fourth Mexican International Conference on Computer Science p. 49.

Meggelen, J. V. Madsen, L. Smith J. 2007. *Asterisk: The Future of Telephony*, Second Edition. O'Reilly.

BeVocal Café, 2007. *VoiceXML development environment*

Kenneth, R. A. 2001.*Voice Enabling Web Applications: VoiceXML and Beyond*".Apress; 1 edition.

Yankelovich, N., 2000. *Designing Effective Speech Interfaces*, John Wiley & Sons, Inc.

Oviatt, S.L., 1999. *Ten myths of multimodal interaction* Communications of the ACM, 42 (11), November

Interoperability and Knowledge Representation in Distributed Health and Fitness Companion Dialogue System

Jaakko Hakulinen
Department of Computer Sciences
33014 University of Tampere, Finland
jaakko.hakulinen@cs.uta.fi

Markku Turunen
Department of Computer Sciences
33014 University of Tampere, Finland
markku.turunen@cs.uta.fi

Abstract

As spoken dialogue systems move beyond task oriented dialogues and become distributed in the pervasive computing environments, their growing complexity calls for more modular structures. When different aspects of a single system can be accessed with different interfaces, knowledge representation and separation of low level interaction modeling from high level reasoning on domain level becomes important. In this paper, a model utilizing a dialogue plan to communicate information from domain level planner to dialogue management and from there to a separate mobile interface is presented. The model enables each part of the system handle the same information from their own perspectives without containing overlapping logic.

1 Introduction

Most existing spoken dialogue systems provide a single interface to solve a well-defined task, such as booking tickets or providing timetable information. There are emerging areas that differ dramatically from task-oriented systems. In domain-oriented dialogues (Dybkjaer et al, 2004) the interaction with the system, typically modeled as a conversation with a virtual human-like character, can be the main motivation for the interaction. These systems are often multimodal, and may take place in pervasive computing environments where various mobile, robotic, and other untraditional interface are used to communicate with the system. For example, in the EU-funded COMPANIONS-project (Wilks, 2007) we are developing a conversational Health and Fitness Companion that develops long-lasting relationships with its users to support their healthy living and eating habits via mobile and physical agent interfaces. Such systems have different motivations for use compared to traditional task-based spoken dialogue systems. Instead of helping with a single, well defined task, the system aims at building a long-term relationship with its user and providing support on a daily basis.

1.1 Mobile and Physical Agent Interfaces

New kinds of interfaces are used increasingly often in conjunction with spoken dialogue technology. Speech suits mobile interfaces well because it can overcome the limited input and output modalities of the small devices and can also better support using during the moments when their hand or eyes are busy. Physical agent interfaces, on the other hand, have been used in systems, which try to make dialogue systems more part of people's life. In many cases, they include rich multimodal input and output while providing a physical outlook for the agent. While naturalistic human-like physical robots are under development, especially in Japan, there is room for a variety of different physical interface agents ranging from completely abstract (e.g., simple devices with lights and sound) to highly sophisticated anthropomorphic apparatus. For example, Marti and Schmandt (2005) used several toy animals, such as bunnies and squirrels, as physical embodied agents for a conversational system. Other example is an in-door guidance and receptionist application involving a physical interface agent that combines pointing gestures with conversational speech technology (Kainulainen et al., 2005). Some physical agent technology has also

Coling 2008: Proceedings of the workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, pages 24–31
Manchester, August 2008

been commercialized. For example, the wireless Nabaztag™/tag rabbits (http://www.nabaztag.com/) have been success

Both mobile use and physical agent interface can support the goal of making a spoken dialogue system part of users' everyday life and building a meaningful relationship between the system and the user. It has been found that mere existence of a physical interface changes users' attitude toward a system and having access to a system throughout the day via a mobile interface is likely to further support this relationship.

In this work, we have used the Nabaztag as a multimodal physical interface to create a conversational Health and Fitness Companion and a mobile version interface for outdoor usage has been implemented on Windows Mobile platform.

1.2 Inter-component Communication and Knowledge Representation Challenges

In systems, where multiple interfaces can be used to access parts of the same functionality and the system interacts with a user many times over a long time period, modeling the interaction and domain easily becomes complex. For example, the system should model interaction history on a longer timescale than a single session. With multiple interfaces, at least some such information could be useful if they can be shared between the interfaces. Furthermore, the system must include a model capable of reasoning about the domain, and learn from the user and his or her actions to provide meaningful interaction, such as to provide reasonable guidance on user's health and progress as the user's condition alters over time in our case with the Health and Fitness Companion. Such reasoning should be concentrated on one component, instead of duplicating the logic to keep the system maintainable. Still, the information must be communicated over different interfaces and the component inside them. Therefore, modularization of the system and appropriate knowledge representation become vital.

On dialogue management level, a common way to take some complexity away from the dialogue manager and limit its tasks more specifically to dialogue management is to separate domain specific processing, such as database queries, into a back-end component. Many researchers have worked with separating generic dialogue management processes from the domain specific processes. Example solutions include shells (Jönsson, 1991) and object oriented programming methods (Salonen, et al., 2004, O'Neill, et al., 2003). On the other hand, a simple back-end interface, e.g., SQL queries, can be included as configuration parameters (Pellon et al., 2000). Since dialogue management is usually based on state transition networks, form filling, or some other clearly defined model, separating domain specific processing to the back-end makes it possible to implemented dialogue management purely with the selected model.

Health and Fitness Companion, as discussed in the following, is based on a model where the domain specific module is more than just a simple interface and includes active processing of domain information, reasoning, learning, or other complex processes. We call such a component the cognitive model. While the task of a dialogue manager is to maintain and update dialogue state, the cognitive model reasons using the domain level knowledge. In our case, we have two dialogue managers, one for the home system with a physical interface agent and one for mobile system (yet another is in development, but not considered here). The two handle somewhat separate tasks but each provides input to another and the cognitive model. Separation of the task between the different parts is not trivial. For example, managing dialogue level phenomena, such as error handling and basic input processing, are tasks clearly in the areas of respective dialogue managers. However, cognitive modeling can help in error handling by spotting input that seems suspicious based on domain level information and input parsing by providing information on potential discussion topics. The solution we have devised is to have the cognitive model produce a dialogue plan for the dialogue management in home system. The dialogue management in the home system provides parsed user inputs to the cognitive model and to the mobile system. The mobile system provides similar input back to the home system, which communicates it back to the cognitive model.

In the following we describe the Health and Fitness dialogue system in general. Then we discuss the mobile interface, the dialogue manager of the home system and the cognitive model, before going into details on how the components have been separated. The solution, which provides great flexibility for each, is discussed before conclusions.

2 Health and Fitness Companion

The Health and Fitness Companion (H&F) is a conversational interface for supporting healthy lifestyle. The companion plans each day together with its user at home, and suggests healthy activities, such as walking to work, when possible. During the day, a mobile interface to the Companion can be used to record various physical activities, such as those walks to work. Afterwards, the user is able to report back to the companion on the day, and get more advice and support. At this point information recorded by the mobile system is automatically used by the system.

- Good morning, anything interesting organized for today?
- I'm going for a walk.
- Is that walk before dinner?
- No, I think I'll walk after I've eaten.
- OK, so you are going for a walk after dinner, is that correct?
- Yes.
- Great, why don't you cycle to work?
- Okay, I can do that.

Figure 1: Health and Fitness Companion Scenario.

As seen in Figure 1, H&F home system uses a Nabaztag/tag WLAN rabbit as a physical interface. Nabaztag provides audio output and push-to-talk input, and is able to move its ears and operate four colored lights to signal, for example, emotions. The mobile interface, as seen in figure 2, runs on a Window Mobile platform and uses push-to-talk speech input, speech output and a graphical interface with key and stylus input. The graphics include Nabaztag graphics and the same voice as in the home system is used for output to help users associate the two interfaces. The mobile Companion follows the user for physical activities, such as jogging, and collects data on the exercises and feeds this back into the main system. While it includes a multimodal speech interface, the main input modality for the mobile Companion can be considered to be GPS positioning. It is used to collect information on user's exercise and provide feedback during the exercise. It is also used as the detection for the completion of the exercises, which information is then forwarded to the home system and the cognitive model.

From technical viewpoint, H&F is a multimodal spoken dialogue system containing components for speech recognition (ASR), natural language understanding (NLU), dialogue management (DM), natural language generation (NLG), and speech synthesis (TTS). Furthermore, it includes a separate cognitive model (CM), which works in close co-operation with DM of the home system, as presented in the following sections. The dialogue system in the home system is implemented using Java and Jaspis framework (Turunen et al., 2005) with jNabServer (http://www.cs.uta.fi/hci/spi/jnabserver/) for Nabaztag connectivity. The cognitive model is implemented in Lisp and integrated into the Jaspis framework. The mobile interface is implemented in Java with native C++ code for speech technology components. It uses PART (http://part.sourceforge.net/) for persistent storage and HECL for scripting in dialogue manager (http://sourceforge.net/projects/hecl).

Figure 2: Mobile Companion Interface.

For speech recognition and synthesis, H&F uses Loquendo ASR and TTS. Current recognition grammars for the home system, derived from a WOZ data and extended using user test data, have a vocabulary of 1090 words and a total of 436 grammar rules. Recognition grammars are dynamically selected for each user input, based on the dialogue state. The mobile interface use mobile versions of Loquendo technology. Due to the technological limitations, more challenging acoustic environment, potential physical exhaustion of users, and more restricted domain, the recognition grammars in the mobile interface will remain significantly smaller than those of the home system.

NLU is based on SISR semantic tags (http://www.w3.org/TR/semantic-interpretation/) embedded in the recognition grammars. In the home system, where mixed initiative interaction is possible, the tags provide parameters compatible with predicates used to represent information on the dialogue management level. Input parsing unifies these parameters into full predicates based on the current dialogue state. In mobile system, more strict state based dialogue modeling can results in unambiguous output straight from the SISR tags.

Natural language generation is a mixture of canned strings and, in the home system, tree adjoining grammar based generation. In addition, control messages for Nabaztag ears and lights can be generated.

As discussed previously, distribution and coordination of the different tasks between different components can become rather complex in systems such as H&F without proper modeling of interaction, domain, and reasoning components. Next, we present a model which allows flexible interaction between the cognitive model and the dialogue management.

3 Dialogue Management and Cognitive Modeling

There is great consensus that components of a dialogue system can be split into at least three parts: an input module, which receives user input and parses it into a logical form, dialogue management, which maintains and updates dialogue state based on user input and generates output requests, and an output module, which generates natural language output to user based on the requests. In the case of H&F, we have also separated a cognitive model (CM) from dialogue manager (DM), as seen in Figure 3. We call this module the cognitive model, because it contains what can be considered higher level cognitive processes of the system. Next, we present DM of the home system, CM component, and the mobile interface, focusing on their interaction.

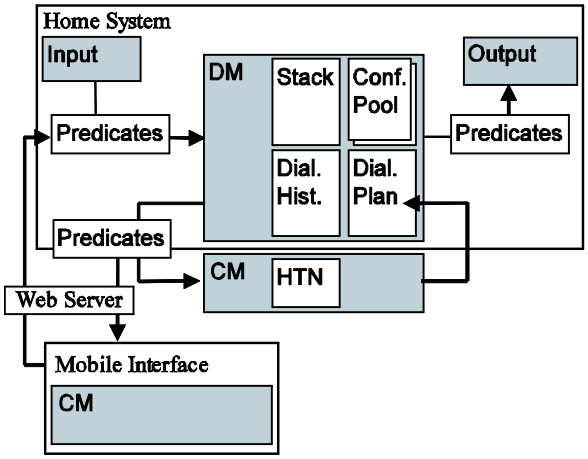


Figure 3: Information passed between the components.

3.1 Cognitive Model Responsibilities

The task of CM is to model the domain, i.e., know what to recommend to the user, what to ask from the user and what kind of feedback to provide. CM in H&F uses hierarchical task networks (HTNs) (Cavazza et al., 2008) as the method of planning healthy daily activity for the user. Part of a network can be seen in Figure 4. In the current H&F implementation, the planning domain included 16 axioms and 111 methods, 49 operators, 42 semantic tags, 113 evaluation rules and there are 17 different topics to be discussed with the user.

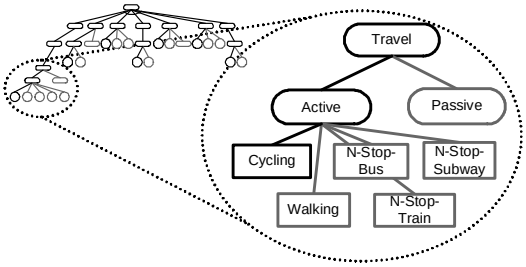


Figure 4: Hierachical Task Network.

CM is aware of the meaning of the concepts inside the system on a domain specific level. It generates and updates a dialogue plan according to the information received from the user. The plan is forwarded to DM. Interaction level issues are not directly visible to CM.

3.2 Dialogue Management in the Home System

The task of DM is to maintain and update a dialogue state. In the H&F system, the dialogue state includes a dialogue history tree (currently linear), a stack of active dialogue topics, and current user input, including ASR confidence scores and N-best lists. In addition, two pools of items that need to be confirmed are stored; one for items to be confirmed individually and another for those that can be confirmed together in one question.

DM receives user inputs as predicates parsed by the NLU component. If an utterance is successfully parsed and matches the current dialogue plan (see Section 3.3), DM does not need to know what the meaning of the input actually is. It just takes care of confirmations and provides the information to CM. When generating output requests based on the plan, DM can also be unaware of the specific meaning of the plan items. Overall, DM does not need to have the deep domain understanding CM specializes in.

DM, however, is aware of the relations of the predicates on the topics level, i.e., it knows, which predicates belong to each topic. This information is used primarily for parsing input. DM also has understanding of the semantics of the predicates which relates to interaction. Namely, relations such as question – answer pairs (suggestion – agreement, confirmation – acceptance/rejection, etc.) are modeled.

On implementation level, dialogue management is implemented as a collection of separate small dialogue agents, following the principles of the underlying Jaspis architecture. These agents are small software components, each taking care of a specific task and in each dialogue turn one or more agents are selected by DM. In the current H&F prototype, there are over 30 dialogue agents. There is a separate agent for each topic that can occur in the plan. In practice, one topic maps to a single plan item. These agents are all instances of a single class with specific configurations. Each agent handles all situations related to its topic; when the topic is the first item of an active plan, they produce related output and when the user provides input matching to the topic they forward that information back to the cognitive model. In addition, topic specific agents handle explicit topic switch requests from the user (e.g., "let's talk about lunch") and also take turn if the topic is found on top of the dialogue topic stack. A topic ends up in the stack when it has not been finished and a new topic is activated. The other agents found in the system include one that generates a confirmation if the ASR confidence score is too low, one that repeats the last system utterance when the user requests it ("please repeat the last one"), and an agent to handle ASR rejection errors.

3.3 Mobile System

Mobile system is designed mainly to support users' on their physical exercises and collected data on them fro the home system. The mobile system receives the day plan that the user has made with the home system and it is used as basis when users activates the system. This way, the user does not need to re-enter information such as the type of an exercise. This is possible, however, with simple spoken dialogue or by using the graphical user interface. During the exercise, GPS information is used by the system to provide feedback on pace to the user using speech output. For dialogue management, the mobile system uses a state based model, based on scripting. Since the mobile system focuses on the physical exercises, it is aware of the meaning of the predicates it receives on that level. It knows more about running and walking than any other component. At the same time, it ignores most of the day plan it receives. For example, eating related plan items are not relevant to the mobile system in any way and are ignored (however, in the future we could include the possibility to report on meals as well).

3.4 Dialogue Plan and Day Plan

The communication between the dialogue managers and CM is based on a dialogue plan and a day plan. Various kinds of dialogue plans (Larsson et al., 2000, Jullien and Marty, 1989) have been used inside dialogue managers in the past. A plan usually models what the system sees as the optimal route to task completion.

In H&F, CM provides a plan on how the current task (planning a day, reporting on a day) could proceed. The plan consists of items, which are basically expressions on domain specific propositional logic. Example 1 contains two items from the start of a plan for planning the day with the user in the morning. The first plan item (QUERY-PLANNED-ACTIVITY) can be realized as the question "Anything interesting planned for today?" by the system.

As new information becomes available (from the user), it forms a plan for the day or a report of the day. DM provides this information to CM,

piece by piece as it becomes available. At the same time, the information is uploaded into a web server, where the mobile interface can access it anytime.

As CM receives the information, it updates the dialogue plan as necessary. Query type items, whose information has been gathered, disappear from the plan and new items may appear.

The messages sent to CM can add new information (predicates) to CM state. DM can also remove information from CM if previously entered information is found to be untrue. Similarly, information uploaded to the web server for mobile use can be modified. The information includes statements on user's condition (tired), user's commitments to the system (will walk to work), user's preferences (does not like cafeterias) and user's reports on past activity (took a taxi to work), which can be accomplishments or failures of earlier commitments.

```
<plan>
 <plan-name>Generate-Task-
Model-Questions</plan-name>
 <plan-item>
   <action>QUERY-PLANNED-
ACTIVITY</action>
 </plan-item>
 <plan-item>
   <action>SUGGEST-TRAVEL-
METHOD</action>
   <param>CYCLING-
TRAVEL</param>
   <param>HOME</param>
   <param>WORK</param>
 </plan-item>
…
```

Example 1: Start of a plan.

DM in the home system can follow the dialogue plan produced by CM step by step. Each step usually maps to a single question, but can naturally result in a longer dialogue if the user's answer is ambiguous or error management is necessary, or if DM decides to split a single item to multiple questions. For example, the two dialogue turn pairs seen in example 2 are the result of a single plan item (QUERY-PLANNED-ACTIVITY). Since the first user utterance does not result in a complete, unambiguous predicate, DM asks a clarification question. A single user utterance can also result in multiple predicates (e.g., will not take bus, has preference to walking).

When the mobile interface is activated, it downloads the current day plan from the web server and uses it as a basis for the dialogue it has with the user. The exercise which will then take place can be linked to an item in the day plan, or it can be something new. As the exercise is completed (or aborted), information in this is uploaded to the web server. From there the DM of the home system can download it. This information is relevant to the DM when the user is reporting on a day. The home system downloads the information provided by the mobile system and reports it back to CM when the dialogue plan includes a related item. DM may also provide some feedback to the user based on the information. It is noteworthy, that CM does not need to differentiate in any way, whether the information on the exercise came from the mobile system or was gathered in a dialogue with the home system.

```
(  <plan-item>
  <action>QUERY-PLANNED-
ACTIVITY</action>
 </plan-item>)
S: Good morning. Anything in-
teresting organized for today?
U: I'm going jogging.
(<pred>
  <action>PLANNED-
ACTIVITY</action>
  <param>ACTIVITY-
JOGGING</param>
  <param>unknownTime</param>
</pred> )
S: Is that jogging exercise
before dinner?
U: No, it's after.
(  <pred>
  <action>PLANNED-
ACTIVITY</action>
  <param>ACTIVITY-
JOGGING</param>
  <param>AFTER-DINNER</param>
 </pred> )
```

Example 2: A dialogue fragment and a corresponding plan item and predicates, latter of which is forwarded to the cognitive model and the mobile interface.

Similarly, clarifications and confirmations are not directly visible to CM. DM can confirm items immediately (for example, when low confidence is reported by the NLG component) or it can delay confirmations to generate a single con-

firmation for multiple items at an appropriate moment.

Most importantly, when presenting questions and suggestions to the user, DM is free to choose any item in the plan, or even do something not included in the plan at all. When information from the mobile system is available, it can direct where we start the dialogue from. DM could also decide to do some small-talk to introduce sensitive topics, which can be useful in managing the user-system relationship (Bickmore and Picard, 2005). In the future, we see DM to have various kinds of knowledge on the dialogue topics: it can know how personal these topics are and how topics are related to each other. It may also have some topics of its own. The communication that is not related to the domain does not reach CM at any point.

CM can include additional annotation in the plan. One such example is the importance of the information. If information is marked important, it is likely, but not certain, that DM will explicitly confirm it. It is also possible for CM to explicitly request a confirmation by generating a separate plan item. For example, if a user reports on having run much more than they are likely to be capable of in their condition, CM can generate a confirmation plan item. It is worth noting, that DM cannot do reasoning on such level and therefore CM must participate in error handling in such cases.

3.5 Benefits of the Model

The presented model for interoperability between the mobile system, the DM of the home system and CM has provided great flexibility for each component. While the dialogue plan generated by CM provides a base for dialogue management, which, in most cases, is followed, DM can deviate from it. DM can handle confirmations as it pleases, add small talk, and process the plan items in any order. The model also supports mixed-initiative dialogues; while DM may follow the plan, the user may discuss any topic. In our current implementation, user input is parsed first against the previous system output, next to the current topic, and finally to the entire space of known predicates. If needed, we can also make parsing more detailed by parsing against dialogue history and the current plan. This way, the information produced by CM is used in input parsing. The dialogue plan can be used in dynamic construction of recognition grammars to support this on ASR grammar level.

Most importantly, all this is possible without including domain specific knowledge. All such information is kept exclusive in CM. Similarly, CM does not need to know the interaction level properties of the topics, such as recognition grammars and natural language generation details. These are internal to their specific components. The mobile system uses the same knowledge representation as CM, but CM does not need to be aware of its existence at all. Similarly, the mobile system can use any part of the information it receives, but is not forced to do anything specific. DM just feed all the information to it and lets it decide what to do with it. When the mobile system provides information back to the home system, DM handles this and CM can ignore completely the fact that different parts of the information it receives were generated using different systems. Similarly, the mobile system does not see any of the internals of the home system.

On an implementation level, the model is independent of the mechanics of either DM or CM. DM can be implemented using state transition networks (a network per plan item), forms (form per item), agent based model, like in the case of mobile system, or any other suitable method. Similarly, the plan does not tie CM to any specific implementation.

4 Conclusions

When dialogue systems move beyond limited task based domains and implement multimodal interfaces in pervasive computing environment, their complexity increases rapidly. Dialogue management, which in most cases is handled with well understood methods such as form filling or state transition networks, tends to grow more complex. Therefore, a model to modularize dialogue management and domain reasoning is needed. At the same time, distributed systems required various kinds of information to be communicated with components and systems.

While traditional spoken dialogue systems have been task-based, the Health and Fitness Companions are part of the users' life for a long time, months, or even years. This requires that they are part of life physically, i.e., interactions can take place on mobile setting and in home environment outside of traditional, task-based computing devices. With the physical presence of the interface agent and spoken, conversational dialogue we aim at building social, emotional relationships between the users and the Compan-

ion. Such relationships should help us in motivating the users towards healthier lifestyle. The mobility of the interface integrates the system into the physical activities they aim at supporting users in.

We have presented a model, which separates cognitive modeling from dialogue management and enables flexible interoperability between the two and also enables sharing the gathered knowledge to the mobile part of the system and back. This division, while similar to separation of a back-end from dialogue management, draws the line deeper into the area of interaction management. The cognitive model processes domain level information and generates dialogue plans. The dialogue manager focuses only on interaction level phenomena, such as initiative and error management, and other meta-communication. In order to enable flexible interaction, the plan provides a potential structure for the dialogue, but the dialogue manager is free to handle things in different order, and even add new topics. It can also include input from a mobile interface of the system without making this explicit to the cognitive model. One example of flexibility is error management; while the actual error correction is the task of the dialogue manager, domain level knowledge can reveal errors. Using the dialogue plan, the cognitive model can provide such information to the dialogue manager without knowledge on details of error management. The model also enables user initiative topic shifts, management of user-system relationship and other novel issues relevant in domain-oriented dialogue systems.

Overall, the model presented has enabled a clear division and interoperability of the different components handling separate parts of the interaction. The presented model has been implemented in the Health and Fitness Companion prototype, and it has enabled the cognitive model, the dialogue manager, and the mobile interface to be developed in parallel by different groups using various programming languages an integrated system.

5 Acknowledgements

This work is part of the EU-funded COMPANIONS-project (IST-34434). The Cognitive Model has been developed by University of Teesside, UK, while the mobile interface has been implemented in Swedish Institute of Computer Science.

References

Dybkjaer, L., Bernsen, N. O., Minker, W., Evaluation and usability of multimodal spoken language dialogue systems, Speech Communication, 43, 1-2, , June 2004, pp. 33-54.

Wilks, Y., Is There Progress on Talking Sensibly to Machines?, Science, 9 Nov 2007.

Marti, S. and Schmandt, C. Physical embodiments for mobile communication agents. Proceedings of the 18th annual ACM symposium on User interface software and technology: 231 – 240, 2005.

Kainulainen, A., Turunen, M., Hakulinen, J., Salonen, E.-P., Prusi, P., and Helin, L. A Speech-based and Auditory Ubiquitous Office Environment. Proceedings of 10th International Conference on Speech and Computer (SPECOM 2005): 231-234, 2005.

Jönsson, A. A Natural Language Shell and Tools for Customizing the Dialogue in Natural Language Interfaces, Research Report, LiTH-IDA-R-91-10, 1991.

Salonen, E.-P., Hartikainen, M., Turunen, M., Hakulinen J., Funk, J. A. Flexible Dialogue Management Using Distributed and Dynamic Dialogue Control. Proceedings of ICSLP 2004. pp. 197-200.

O'Neill, I. Hanna, P. Liu, X., McTear, M., The Queen's Communicator: An Object-Oriented Dialogue Manager, Eurospeech 2003, Geneva, Switzerland (2003), pp. 593–596.

Pellom, B. Ward, W. Pradhan, S., The CU Communicator: An Architecture for Dialogue Systems, Proceedings of ICSLP 2000, Beijing China, November 2000.

Turunen, M., Hakulinen, J., Räihä, K.-J., Salonen, E.-P., Kainulainen, A., and Prusi, P. An architecture and applications for speech-based accessibility systems. IBM Systems Journal, Vol. 44, No 3, 2005, pp. 485-504.

Cavazza, M., Smith, C., Charlton, D., Zhang, L., Turunen, M. and Hakulinen, J., A 'Companion' ECA with Planning and Activity Modelling, Proceedings of AAMAS08, 2008 (to appear).

Larsson, S. Ljunglöf, P. Cooper, R. Engdahl, E., Ericsson. S. GoDiS - an accommodating dialogue system. ANLP / NAACL '00 Workshop on Conversational Systems, May 2000.

Jullien C., Marty, J.-C. Plan revision in person-machine dialogue, Proceedings of ACL'89, Manchester, England, April 1989, pp.153-160.

Bickmore, T. W., Picard, R. W. Establishing and maintaining long-term human-computer relationships. ACM Trans. Computer-Human. Interaction Vol. 12, No. 2. (June 2005), pp. 293-327.

The 2008 MedSLT System

Manny Rayner[1], **Pierrette Bouillon**[1], **Jane Brotanek**[2], **Glenn Flores**[2]
Sonia Halimi[1], **Beth Ann Hockey**[3], **Hitoshi Isahara**[4], **Kyoko Kanzaki**[4]
Elisabeth Kron[5], **Yukie Nakao**[6], **Marianne Santaholma**[1]
Marianne Starlander[1], **Nikos Tsourakis**[1]

[1] University of Geneva, TIM/ISSCO, 40 bvd du Pont-d'Arve, CH-1211 Geneva 4, Switzerland
{Emmanuel.Rayner,Pierrette.Bouillon,Nikolaos.Tsourakis}@issco.unige.ch
{Sonia.Halimi,Marianne.Santaholma,Marianne.Starlander}@eti.unige.ch

[2] UT Southwestern Medical Center, Children's Medical Center of Dallas
{Glenn.Flores,Jane.Brotanek}@utsouthwestern.edu

[3] Mail Stop 19-26, UCSC UARC, NASA Ames Research Center, Moffett Field, CA 94035–1000
bahockey@ucsc.edu

[4] NICT, 3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan 619-0289
{isahara,kanzaki}@nict.go.jp

[5] 3 St Margarets Road, Cambridge CB3 0LT, England
elisabethkron@yahoo.co.uk

[6] University of Nantes, LINA, 2, rue de la Houssinière, BP 92208 44322 Nantes Cedex 03
yukie.nakao@univ-nantes.fr

Abstract

MedSLT is a grammar-based medical speech translation system intended for use in doctor-patient diagnosis dialogues, which provides coverage of several different subdomains and multiple language pairs. Vocabulary ranges from about 350 to 1000 surface words, depending on the language and subdomain. We will demo three different versions of the system: an any-to-any multilingual version involving the languages Japanese, English, French and Arabic, a bidirectional English ↔ Spanish version, and a mobile version running on a hand-held PDA. We will also demo the Regulus development environment, focussing on features which support rapid prototyping of grammar-based speech translation systems.

1 Introduction

MedSLT is a medium-vocabulary grammar-based medical speech translation system built on top of the Regulus platform (Rayner et al., 2006). It is intended for use in doctor-patient diagnosis dialogues, and provides coverage of several subdomains and a large number of different language-pairs. Coverage is based on standard examination questions obtained from physicians, and focusses primarily on yes/no questions, though there is also support for WH-questions and elliptical utterances.

Detailed descriptions of MedSLT can be found in earlier papers (Bouillon et al., 2005; Bouillon et al., 2008)[1]. In the rest of this note, we will briefly sketch several versions of the system that we intend to demo at the workshop, each of which displays new features developed over the last year. Section 2 describes an any-language-to-any-language multilingual version of the system; Section 3, a bidirectional English ↔ Spanish version; Section 4, a version running on a mobile PDA

[1] All MedSLT publications are available on-line at http://www.issco.unige.ch/projects/medslt/publications.shtml.

Coling 2008: Proceedings of the workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, pages 32–35
Manchester, August 2008

platform; and Section 5, the Regulus development environment.

2 A multilingual version

During the last few months, we have reorganised the MedSLT translation model in several ways[2]. In particular, we give a much more central role to the interlingua; we now treat this as a language in its own right, defined by a normal Regulus grammar, and using a syntax which essentially amounts to a greatly simplified form of English. Making the interlingua into another language has made it easy to enforce tight constraints on well-formedness of interlingual semantic expressions, since checking well-formedness now just amounts to performing generation using the interlingua grammar.

Another major advantage of the scheme is that it is also possible to systematise multilingual development, and only work with translation from source language to interlingua, and from interlingua to target language; here, the important point is that the human-readable interlingua surface syntax makes it feasible in practice to evaluate translation between normal languages and the interlingua. Development of rules for translation *to* interlingua is based on appropriate corpora for each source language. Development of rules for translating *from* interlingua uses a corpus which is formed by merging together the results of translating each of the individual source-language corpora into interlingua.

We will demonstrate our new capabilities in interlingua-based translation, using a version of the system which translates doctor questions in the headache domain from any language to any language in the set {English, French, Japanese, Arabic}. Table 1 gives examples of the coverage of the English-input headache-domain version, and Table 2 summarises recognition performance in this domain for the three input languages where we have so far performed serious evaluations. Differences in the sizes of the recognition vocabularies are primarily due to differences in use of inflection.

3 A bidirectional version

The system from the preceding section is unidirectional; all communication is in the doctor-to-patient direction, the expectation being that the patient will respond non-verbally. Our second demo, an early version of which is described in (Bouillon et al., 2007), supports bidirectional translation for the sore throat domain, in the English ↔ Spanish pair. Here, the English-speaking doctor typically asks WH-questions, and the Spanish-speaking patient responds with elliptical utterances, which are translated as full sentence responses. A short example dialogue is shown in Table 3.

[2]The ideas in the section are described at greater length in (Bouillon et al., 2008).

Language	Vocab	WER	SemER
English	447	6%	11%
French	1025	8%	10%
Japanese	422	3%	4%

Table 2: Recognition performance for English, French and Japanese headache-domain recognisers. "Vocab" = number of surface words in source language recogniser vocabulary; "WER" = Word Error Rate for source language recogniser, on in-coverage material; "SemER" = semantic error rate for source language recogniser, on in-coverage material.

Doctor:	Where is the pain? *¿Dónde le duele?*
Patient:	En la garganta. *I experience the pain in my throat.*
Doctor:	How long have you had a pain in your throat? *¿Desde cuándo le duele la garganta?*
Patient:	Más de tres días. *I have experienced the pain in my throat for more than three days.*

Table 3: Short dialogue with bidirectional English ↔ Spanish version. System translations are in italics.

4 A mobile platform version

When we have shown MedSLT to medical professionals, one of the most common complaints has been that a laptop is not an ideal platform for use in emergency medical situations. Our third demo shows an experimental version of the system using a client/server architecture. The client, which contains the user interface, runs on a Nokia Linux N800 Internet Tablet; most of the heavy processing, including in particular speech recognition, is hosted on the remote server, with the nodes communicating over a wireless network. A picture of

Where?	Is the pain above your eye?
When?	Have you had the pain for more than a month?
How long?	Does the pain typically last a few minutes?
How often?	Do you get headaches several times a week?
How?	Is it a stabbing pain?
Associated symptoms?	Do you vomit when you get the headaches?
Why?	Does bright light make the pain worse?
What helps?	Does sleep make the pain better?
Background?	Do you have a history of sinus disease?

Table 1: Examples of English MedSLT coverage

the tablet, showing the user interface, is presented in Figure 1. The sentences appearing under the back-translation at the top are produced by an online help component, and are intended to guide the user into the grammar's coverage (Chatzichrisafis et al., 2006).

The architecture is described further in (Tsourakis et al., 2008), which also gives performance results for another Regulus applications. These strongly suggest that recognition performance in the client/server environment is no worse than on a laptop, as long as a comparable microphone is used.

5 The development environment

Our final demo highlights the new Regulus development environment (Kron et al., 2007), which has over the last few months acquired a large amount of new functionality designed to facilitate rapid prototyping of spoken language applications[3]. The developer initially constructs and debugs her components (grammar, translation rules etc) in a text view. As soon as they are consistent, she is able to compile the source-language grammar into a recogniser, and combine this with other components to run a complete speech translation system within the development environment. Connections between components are defined by a simple config file. Figure 2 shows an example.

References

Bouillon, P., M. Rayner, N. Chatzichrisafis, B.A. Hockey, M. Santaholma, M. Starlander, Y. Nakao, K. Kanzaki, and H. Isahara. 2005. A generic multilingual open source platform for limited-domain medical speech translation. In *Proceedings of the 10th Conference of the European Association for Machine Translation (EAMT)*, pages 50–58, Budapest, Hungary.

Bouillon, P., G. Flores, M. Starlander, N. Chatzichrisafis, M. Santaholma, N. Tsourakis, M. Rayner, and B.A. Hockey. 2007. A bidirectional grammar-based medical speech translator. In *Proceedings of the ACL Workshop on Grammar-based Approaches to Spoken Language Processing*, pages 41–48, Prague, Czech Republic.

Bouillon, P., S. Halimi, Y. Nakao, K. Kanzaki, H. Isahara, N. Tsourakis, M. Starlander, B.A. Hockey, and M. Rayner. 2008. Developing non-european translation pairs in a medium-vocabulary medical speech translation system. In *Proceedings of LREC 2008*, Marrakesh, Morocco.

Chatzichrisafis, N., P. Bouillon, M. Rayner, M. Santaholma, M. Starlander, and B.A. Hockey. 2006. Evaluating task performance for a unidirectional controlled language medical speech translation system. In *Proceedings of the HLT-NAACL International Workshop on Medical Speech Translation*, pages 9–16, New York.

Kron, E., M. Rayner, P. Bouillon, and M. Santaholma. 2007. A development environment for building grammar-based speech-enabled applications. In *Proceedings of the ACL Workshop on Grammar-based Approaches to Spoken Language Processing*, pages 49–52, Prague, Czech Republic.

Rayner, M., B.A. Hockey, and P. Bouillon. 2006. *Putting Linguistics into Speech Recognition: The Regulus Grammar Compiler*. CSLI Press, Chicago.

Tsourakis, N., M. Georghescul, P. Bouillon, and M. Rayner. 2008. Building mobile spoken dialogue applications using regulus. In *Proceedings of LREC 2008*, Marrakesh, Morocco.

[3]This work is presented in a paper currently under review.

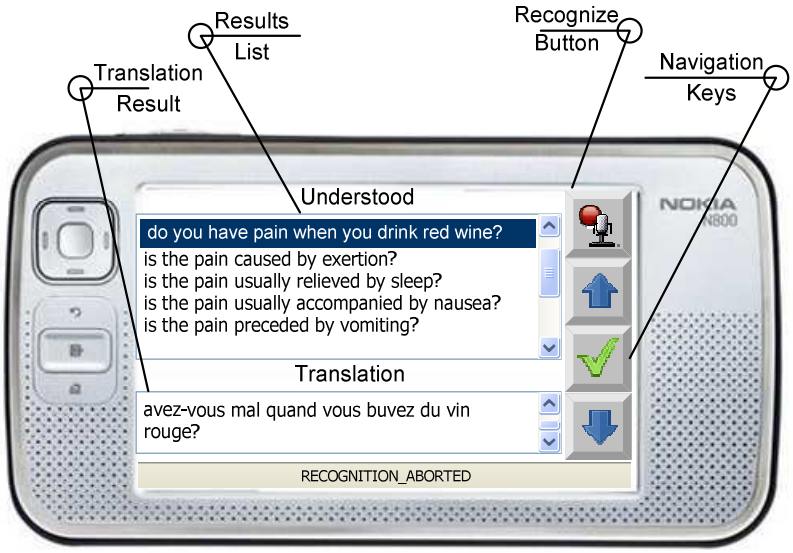


Figure 1: Mobile version of the MedSLT system, running on a Nokia tablet.

Figure 2: Speech to speech translation from the development environment, using a Japanese to Arabic translator built from MedSLT components. The user presses the Recognise button (top right), speaks in Japanese, and receives a spoken translation in Arabic together with screen display of various processing results. The application is defined by a config file which combines a Japanese recogniser and analysis grammar, Japanese to Interlingua and Interlingua to Arabic translation rules, an Arabic generation grammar, and recorded Arabic wavfiles used to construct a spoken result.

Language Understanding in Maryland Virtual Patient

Sergei Nirenburg
Stephen Beale
Marjorie McShane
University of Maryland Baltimore County
{sergei, sbeale, marge}@umbc.edu

Bruce Jarrell
George Fantry
University of Maryland School of Medicine
BJarrell@som.umaryland.edu
GFantry@medicine.umaryland.edu

Abstract

This paper discusses language understanding in the Maryland Virtual Patient environment. Language understanding is just one of many cognitive functions of the virtual patients in MVP, others including decision making about healthcare and lifestyle, and the experiencing and remembering of interoceptive events.

1 Introduction

Maryland Virtual Patient[2] (MVP) is an agent-oriented environment for automating certain facets of medical training. The environment contains a network of human and software agents, at whose core is a virtual patient – a knowledge-based model of a person with a disease. This model is implemented in a computer simulation. The virtual patient is a "double agent" that displays both physiological and cognitive function. Physiologically, it undergoes both normal and pathological processes in response to internal and external stimuli. Cognitively, it experiences symptoms, has lifestyle preferences, has memory (many of whose details fade with time), and communicates with the human user about its personal history and symptoms. Other software agents in the MVP environment include consulting physicians, lab technicians and a virtual mentor (tutor).

What makes virtual patient modeling feasible – considering that comprehensively modeling human physiology would be a boundless endeavor – is our task-oriented approach: we are not trying to recreate the human organism in all its details, we are modeling it to the extent necessary to support its realistic autonomous functioning in applications aimed at training the diagnostic and treatment skills of medical personnel.

Trainees can use MVP to interview a virtual patient; order lab tests; receive the results of lab tests from technician agents; receive interpretations of lab tests from consulting physician agents; posit hypotheses, clinical diagnoses and definitive diagnoses; prescribe treatments; follow-up after those treatments to judge their efficacy; follow a patient's condition over an extended period of time, with the trainee having control over the speed of simulation (i.e., the clock); and, if desired, receive mentoring from the automatic mentor.

The virtual patient (VP) simulation is grounded in an ontologically-defined model of human anatomy and physiology. Instances of virtual patients with particular diseases and particular physiological peculiarities are generated from core ontological knowledge about human physiology and anatomy by grafting a disease process onto a generic instance of a human. Disease processes themselves are described as complex events in the underlying ontology.

2 Reasoning by the Cognitive Agent

The cognitive side of the VP carries out reasoning in response to two types of input: interoception (the experiencing of physical stimuli, like symptoms) and language input. Specifically its functioning includes:

1. experiencing, interpreting and remembering symptoms
2. deciding to go see a doctor, initially and during treatment
3. understanding the doctor's language input as well as its intent

[2] Patent pending.

Coling 2008: Proceedings of the workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, pages 36–39
Manchester, August 2008

4. deciding whether to ask knowledge-seeking questions about a test or intervention suggested by the doctor
5. deciding whether to agree to a test or intervention suggested by the doctor.
6. deciding on what specifically to say in response to the doctor's questions, recommendations, etc.

In this paper we concentrate on point 3. We point readers to other works about MVP (e.g., McShane et al. 2007) for a discussion of other aspects of MVP.

Five types of subdialogs are supported in MVP.

1. Requests for information and responses. These include (a) the physician asking the patient questions about symptoms and lifestyle, and (b) the patient asking questions about features of suggested interventions as well as other options.
2. Requests for action and responses – primarily the physician suggesting that the patient agree to have an intervention.
3. Domain descriptions provided by the user, the key points of which must be understood and remembered ("learned") by the VP.
4. Scheduling follow-up appointments.
5. General dialog topics, like greetings, expressions of gratitude and other means for making the dialog more realistic in the user's eyes.

Our approach to treating dialog is unlike most other approaches in that all language-oriented reasoning is carried out on the basis of formal interpretations of text meaning. We call these interpretations ***text meaning representations*** or **TMR**s. Note that TMRs are written using the same ontologically grounded metalanguage as is used to represent interoception. In short, all knowledge and reasoning in our environment employs the same metalanguage, so whether a patient experiences new symptoms or learns information about its disease from the user, the new information will be stored the same way in the patient's memory.

There are several advantages to orienting an agent's language processing around TMRs rather than text strings. First, TMRs are unambiguous, since linguistic ambiguity is resolved as the TMRs are being produced. Second, TMRs reduce to a single representation many types of linguistic paraphrase, be it lexical (*esophagus ~ food pipe)*, syntactic (*I will administer it to you ~ It will be administered to you by me*) or even semantic (*Does the food get stuck when you swallow? ~ Do you have difficulty swallowing?*). Third, TMRs facilitate the detection of which aspects of meaning are central and which are of secondary importance. For example, the analyzer can determine which portions of input utterances merely convey politeness. To take an extreme example for illustration, the question "Do you have difficultly swallowing?" could be rendered by an overly polite physician as: "If you don't mind, I would really appreciate it if you would tell me whether you have any difficulty swallowing."

When the VP receives language input, it uses its lexicon, ontology and a reasoning-enabled analyzer to create a TMR corresponding to the input. Next, it determines the intent of that input – e.g., through the recognition of indirect speech acts. After that it plans its response then generates its response. Here we talk about the first two stages of text processing: understanding the dialog turn and understanding its intent.

3 Understanding a Dialog Turn

The input to understanding a dialog turn is text input by the user. Background knowledge that must be leveraged is the knowledge stored in the lexicon, ontology and the patient's long-term memory of assertions, also called its fact repository. The output is a TMR. TMR production actually comprises two stages: the first stage, production of the *basic TMR*, involves disambiguation and the determination of semantic dependencies; the second stage, production of the *extended TMR*, adds the results of procedural semantic routines, like the resolution of reference.

For example, the following questions are all synonyms at the level of extended TMR, at least at the grain-size of description needed for our current application: *Have you been coughing? Do you find yourself coughing*? *Do you experience any coughing? Do you ever experience coughing? Do you have a cough? Any coughing? Coughing?* etc. All of these questions ask whether or not the patient has the symptom ontologically described as the event called COUGH. The extended TMR for this set of questions is:

```
(REQUEST-INFO-1
  (THEME MODALITY-1.VALUE))
(MODALITY-1
  (TYPE EPISTEMIC)
  (SCOPE ASPECT-1))
```

```
(ASPECT-1
  (ITERATION MULTIPLE)
  (SCOPE COUGH-1))
(COUGH-1
  (EXPERIENCER HUMAN-1)
   (TIME
      (FIND-INTERVAL (FIND-ANCHOR-TIME)
      (FIND-INTERVAL-LENGTH) BEFORE)))
```

This TMR is read as follows. The input creates an instance of REQUEST-INFO. The instance is numbered, like all TMR instances, to distinguish it from other instances of that concept. The THEME of REQUEST-INFO-1 – i.e., what is being asked about – is whether or not COUGH-1 has occurred repetitively; this is shown in the ASPECT-1 frame. The COUGH event itself has the VP, HUMAN-1, as the EXPERIENCER. The time of the COUGH event is calculated using a procedural semantic routine that seeks a certain time interval in the past (we leave out details of *which* period of time in order to avoid a lengthy tangent). Although this example is a bit complex – involving both aspect and modality – it provides some insight into the format and content of TMRs in our environment.

The text analyzer can automatically create this same TMR for all of the different inputs in large part thanks to the lexicon. Syntactic knowledge in lexicon entries in OntoSem is formulated using an extended form of Lexical Functional Grammar, with variables used to link entities in the syntactic structure (syn-struc) zone of an entry with those in the semantic structure (sem-struc) zone. Lexicon entries can also contain calls to procedural semantic routines (meaning-procedures). The caret means "the meaning of" a given variable. $var0 is the head entry.

Have you been coughing? is a syntactic transformation of *Do you cough?*, which is understood directly by the analyzer as a question about *cough (v.)*, which is mapped to the concept COUGH in the respective lexicon entry.

```
(cough-v1
  (syn-struc
    ((subject ((root $var1) (cat n)))
     (root $var0) (cat v)))
  (sem-struc
    (COUGH (EXPERIENCER (value ^$var1)))))
```

For the other paraphrases, "superfluous" words must be attributed null semantics. For example, *to find oneself verb-ing* is semantically same as *to verb*, the only real difference being stylistic. There is a lexical sense of *find* that attributes null semantics to *find oneself* in the collocation *find oneself doing X*.

Examples in which question processing is folded into the lexicon entry are *Any + EVENT ? (Any coughing?)* and *EVENT? (Coughing?)*. The lexicon entry that covers these is keyed on the question mark, since it is the only element that is always available in these turns of phrase (since "any" is optional). The sem-struc is headed by the concept REQUEST-INFO, whose THEME is the value of epistemic modality scoping over the event in question.

This brief overview is intended only to give a taste of the process of language understanding by virtual patients in MVP. This process is exactly the same as language understanding in other applications of our text processor, called OntoSem (see Nirenburg and Raskin 2004).

The eventualities of text understanding by the cognitive agent of the VP are: (a) successful understanding, (b) the VP's belief that it understood, only to be corrected by the user, or (c) the failure of understanding, in which case the VP asks for clarification by the user.

4 Understanding the Intent of a Dialog Turn

The extended TMR is our most complete model of the meaning of an utterance, but it does not include what is called indirect speech act processing – i.e., understanding intentions of the speaker when they are not overtly mentioned in the utterance. Well-known examples of the dichotomy between expressed meaning and intended meaning include *It's cold in here* (which might be a statement/complaint or might be an indirect request for the interlocutor to do something about it, like close the window) and *Can you pass the salt?* (which might be a question about physical ability or an indirect request).

Our work on indirect speech acts includes long-term, fundamental theory building as well as short-term, immediately implementable solutions. At a fundamental level, speech act processing requires the speaker and the interlocutor to keep a full inventory of their beliefs about the other's knowledge, their understanding of their own and the other's plans and goals, both long-term and immediate, their understanding of what is and what is not within each person's or agent's power to do, and so on. More immediately, we have implemented a means of detecting indirect speech acts in the dialogs between VPs and us-

ers. Our approach, like all of our approaches to automatic reasoning, is grounded in TMRs.

There are three utterance types that the VP expects of the user, which correspond to three user plans: asking questions to learn information that will aid in diagnosis and treatment, explaining things to educate the VP, and giving advice to the VP about what it should do. At any point in the dialog when the user stops typing and expects a response from the VP, the VP must decide which of the plans the user is pursuing. Surface-level heuristics are not always definitive: e.g., *Would you agree to have a Heller myotomy*? is both a question and advice, and *I think that having a Heller myotomy is the best option* is both information and advice.

We prepare the VP to interpret indirect speech acts by creating TMR correspondences between the direct and the indirect meaning of certain types of utterances. Let us take as an example the doctor's offering advice on what to do. There are many ways the doctor can present advice, including the following, provided below with their respective TMRs. In all of these TMRs, HUMAN-1 is the doctor and HUMAN-2 is the patient (these TMRs are simplified for purposes of exposition; also note that all reference resolution has been carried out). INTERVENTION stands for any event that is ontologically an intervention – that is, a test or a medical procedure. Note that the lexicon directly supports the automatic generation of these TMRs.

1. I (would) advise/suggest/recommend (having) INTERVENTION

```
(ADVISE-1
   (THEME INTERVENTION-1)
   (AGENT HUMAN-1)
 (INTERVENTION-1
   (EXPERIENCER HUMAN-2))
```

2. I think you should have INTERVENTION

```
(MODALITY-1
   (TYPE BELIEF)
   (VALUE (> .7))
   (SCOPE MODALITY-2)
   (ATTRIBUTED-TO HUMAN-1))
(MODALITY-2
   (TYPE OBLIGATIVE)
   (VALUE .8)
   (SCOPE INTERVENTION-1)
   (ATTRIBUTED-TO HUMAN-1))
(INTERVENTION-1
    (EXPERIENCER HUMAN-2)))
```

3. I'd like to schedule you for <set you up for, set you up to have> INTERVENTION

```
 (MODALITY-1
    (TYPE VOLITIVE)
    (SCOPE EVENT-1)
    (VALUE .8)
    (ATTRIBUTED-TO HUMAN-1))
(SCHEDULE-EVENT-1
    (AGENT HUMAN-1)
    (THEME INTERVENTION-1)
    (BENEFICIARY HUMAN-2))
(INTERVENTION-1
    (EXPERIENCER HUMAN-2))
```

The "core" meaning that the VP must glean from any of these TMRs is the meaning shown in (1): that the doctor is advising that the patient have the intervention. The correlations between the TMRs in (2) and (3) and this core TMR are established using a TMR-to-TMR translation function. The efficacy of this translation process depends on (a) preparing for the full inventory of possible *types* of input TMRs that correspond to the given meaning, and (b) being able to extract from more complex TMRs these basic kernels of meaning. We have already implemented part (a) in our current system. Part (b) requires more long-term effort, the problem essentially being that one needs to teach the system to zero in on what is important and ignore what is unimportant. For example, negation is very important: *I advise you to have INTERVENTION* is very different from *I do **not** advise you to have INTERVENTION*. However, *I think I would choose to advise you to have INTERVENTION* includes aspects of meaning ('think', 'would choose') that are really not important and should be simplified to the main meaning of the proposition. We consider research on this aspect of agent reasoning to be a long-term endeavor

References

McShane, Marjorie, Sergei Nirenburg, Stephen Beale, Bruce Jarrell and George Fantry. 2007. Knowledge-based modeling and simulation of diseases with highly differentiated clinical manifestations. *11th Conference on Artificial Intelligence in Medicine (AIME 07)*, Amsterdam, The Netherlands, July 7-11, 2007.

Nirenburg, Sergei and Victor Raskin. 2004. *Ontological Semantics*. MIT Press.

Rapid Portability among Domains in an Interactive Spoken Language Translation System

Mark Seligman
Spoken Translation, Inc.
Berkeley, CA, USA 94705
mark.seligman
@spokentranslation.com

Mike Dillinger
Spoken Translation, Inc.
Berkeley, CA, USA 94705
mike.dillinger
@spokentranslation.com

Abstract

Spoken Language Translation systems have usually been produced for such specific domains as health care or military use. Ideally, such systems would be easily portable to other domains in which translation is mission critical, such as emergency response or law enforcement. However, porting has in practice proven difficult. This paper will comment on the sources of this difficulty and briefly present an approach to rapid inter-domain portability. Three aspects will be discussed: (1) large general-purpose lexicons for automatic speech recognition and machine translation, made reliable and usable through interactive facilities for monitoring and correcting errors; (2) easily modifiable facilities for instant translation of frequent phrases; and (3) quickly modifiable custom glossaries. As support for our approach, we apply our current SLT system, now optimized for the health care domain, to sample utterances from the military, emergency service, and law enforcement domains, with discussion of numerous specific sentences.

1 Introduction

Recent years have seen increasing research and commercial activity in the area of Spoken Language Translation (SLT) for mission-critical applications. In the health care area, for instance, such products as Converser (Dillinger & Seligman, 2006), S-MINDS (www.fluentialinc.com), and Med-SLT (Bouillon et al, 2005) are coming into use. For military applications, products like Phraselator (www.phraselator.com) and S-MINDS (www.fluentialinc.com) have been deployed. However, the demand for real-time translation is by no means restricted to these areas: it is clear in numerous other areas not yet extensively addressed – emergency services, law enforcement, and others.

Ideally, a system produced for one such domain (*e.g.*, health care) could be easily ported to other domains. However, porting has in practice proven difficult. This paper will comment on the sources of this difficulty and briefly present an approach to rapid inter-domain portability that we believe is promising. Three aspects of our approach will be discussed: (1) large general-purpose lexicons for automatic speech recognition (ASR) and machine translation (MT), made reliable and usable through interactive facilities for monitoring and correcting errors; (2) easily modifiable facilities for instant translation of frequent phrases; and (3) quickly modifiable custom glossaries.

As preliminary support for our approach, we apply our current SLT system, now optimized for the health care domain, to sample utterances from the military, emergency service, and law enforcement domains.

With respect to the principal source of the porting problems affecting most SLT systems to date: most systems have relied upon statistical approaches for both ASR and MT (Karat and Nahamoo, 2007; Koehn, 2008); so each new domain has required extensive and high-quality in-domain corpora for best results, and the difficulty of obtaining them has limited these systems' portability. The need for in-domain corpora can be eliminated through the use of a quite general corpus (or collection of corpora) for statistical training; but because large corpora give rise to quickly increasing perplexity and error rates, most SLT systems have been designed for specialized domains.

By contrast, breadth of coverage has been a central design goal of our SLT systems. Before any optimization for a specific domain, we "give our systems a liberal arts education" by incorporating very broad-coverage ASR and MT technology. (We presently employ rule-based rather than statistical MT components, but this choice is not essential.) For example, our MT lexicons for English<>Spanish translation in the health care area contain roughly 350,000 words in each direction, of which only a small percentage are specifically health care terms. Our translation grammars (presently licensed from a commercial source, and further developed with our collaboration) are similarly designed to cover the structures of wide-ranging general texts and spoken discourse.

To deal with the errors that inevitably follow as coverage grows, we provide a set of facilities that enable users from both sides of the language barrier to

Coling 2008: Proceedings of the workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, pages 40–47
Manchester, August 2008

interactively monitor and correct such errors. We have described these interactive techniques in (Dillinger and Seligman, 2004; Zong and Seligman, 2005; Dillinger and Seligman, 2006; and Seligman and Dillinger, 2006). With users thus integrated into the speech translation loop, automatically translated spoken conversations can range widely with acceptable accuracy (Seligman, 2000). Users can move among domains with relative freedom, even in advance of lexical or other domain specialization, because most domains are already covered to some degree. After a quick summary of our approach (in Section 2), we will demonstrate this flexibility (in Section 3).

While our system's facilities for monitoring and correction of ASR and MT are vital for accuracy and confidence in wide-ranging conversations, they can be time consuming. Further, interactivity demands a minimum degree of computer and print literacy, which some patients may lack. To address these issues, we have developed a facility called *Translation Shortcuts*™, through which prepared translations of frequent or especially useful phrases *in the current domain* can be instantly executed by searching or browsing. The facility is described in (Seligman and Dillinger, 2006). After a quick description of the Translation Shortcuts facility (Section 4), this paper will emphasize the contribution of the Translation Shortcuts facility to domain portability, showing how a domain-specific set of Shortcuts can be composed and integrated into the system very quickly (Section 5).

Finally, while the extensive lexical resources already built into the system provide the most significant boost to domain portability in our system, it will always be desirable to add specialized lexical items or specialized meanings of existing ones. Section 6 will briefly present our system's *glossary import* facility, through which lexical items can be added or updated very quickly. Our concluding remarks appear in Section 7.

2 Highly Interactive, Broad-coverage SLT

We now briefly summarize our group's approach to highly interactive, broad-coverage SLT. Our systems stress interactive monitoring and correction of both ASR and MT.

First, users can monitor and correct the speaker-dependent speech recognition system to ensure that the text which will be passed to the machine translation component is as correct as necessary. Voice commands (*e.g.*, **Scratch That** or **Correct <incorrect text>**) can be used to repair speech recognition errors. Thus, users of our SLT systems in effect serve to enhance the interface between ASR and MT.

Next, during the MT stage, users can monitor, and if necessary correct, translation errors.

As an initial safeguard against translation errors, we supply a *back-translation*, or re-translation of the translation. Using this paraphrase of the initial input, even a monolingual user can make an initial judgment concerning the quality of the preliminary machine translation output. If errors are seen, the user can modify specific parts of the input and retranslate. (Other systems, *e.g.* IBM's MASTOR (Gao et al, 2006), have also employed re-translation. Our implementations, however, exploit proprietary technologies to ensure that the lexical senses used during back-translation accurately reflect those used in forward translation. We also allow users to modify part or all of the input before regenerating the translation and back-translation.)

In addition, if uncertainty remains about the correctness of a given word sense, we supply a proprietary set of Meaning Cues™ – synonyms, definitions, examples, pictures, etc. – which have been drawn from various resources, collated in a database (called SELECT™), and aligned with the respective lexica of the relevant MT systems. (In the present English<>Spanish version of the system, this database contains some 140,000 entries, corresponding to more than 350,000 lexical entries. The cues are automatically grouped by meaning, and cue groups are automatically mapped to MT lexica using proprietary techniques – thus in effect retrofitting an MT system with the ability to explain to users the meanings of its pre-existing lexical items.) With these cues as guides, the user can monitor the current, proposed meaning and if necessary select a different, preferred meaning from among those available. Automatic updates of translation and back-translation then follow. (Our current MT vendor has modified its rule-based translation engine to allow specification of a desired sense when translating a word or expression; we provide guidelines for other vendors to do likewise. Comparable modifications for statistical MT engines will entail the setting of temporary weightings that will bias the selection of word or phrase translations for the current sentence only.) Future versions of the system will allow personal word-sense preferences thus specified in the current session to be optionally stored for reuse in future sessions, thus enabling a gradual tuning of word-sense preferences to individual needs. (However, such persistent personal preferences will still be applied sentence by sentence, rather than by permanently modifying lexica or phrase tables. Further, users will always be able to temporarily override, or permanently reset, their personal preferences.) Facilities will also be provided for sharing such preferences across a working group.

Given such interactive correction of both ASR and MT, wide-ranging, and even playful, exchanges become possible (Seligman, 2000). Such interactivity within a speech translation system enables increased accuracy and confidence, even for wide-ranging conversations.

3 Advantages of Very Broad Coverage for Domain Switching

This section discusses the advantages of very broad lexical coverage for rapid domain porting. Using our interactive SLT system in its present configuration, optimized for the health care domain but with a general-purpose foundation of over 60,000 lexical items for ASR and 350,000 lexical items for rule-based MT, we will test several input sentences from each of three distinct domains in which translation is mission-critical – military, emergency response, and law enforcement. The test sentences were invented by the authors; readers can judge their plausibility. They were pronounced by Seligman using the built-in microphone of a Motion Computing LE1600 tablet PC equipped with a push-to-talk button.

For each input, we will show (1) the English input, (2) the original Spanish translation, and (3) the English back-translation. We also comment on several factors:

- If ASR errors occurred, we describe their interactive resolution. (All inputs were corrected before proceeding with translation. All corrections were made by voice.)
- If our Meaning Cues facility indicated questionable meanings for any of the expressions in the input, we note the problems and describe the resolutions.
- Some problems in translation result from bugs or gaps in the translation component. These are marked for repair. (Because our MT components are presently rule-based, we can address such problems individually and manually. If a statistical MT component were used instead, the recorded errors could guide the selection of texts for further training.)

As mentioned, in our system, the back-translation is designed to function as the first line of defense against inadequate translation. If an unsatisfactory back-translation is obtained, we advise users to re-phrase the input and translate again until satisfied. (False negatives sometimes occur, though we work to eliminate them; however, it is best to err on the side of caution.) If the back-translation is satisfactory, we advise checking the Meaning Cues as a defense against false positives. These may result if an ambiguous English input word is translated into Spanish in the wrong sense (for instance, *bank* may be translated as *banco* ("financial institution") when *orilla del río* ("riverbank") is wanted), but is nevertheless re-translated as the same ambiguous English word (*bank*). We are experimenting with mechanisms to eliminate such cases by substituting non-ambiguous synonyms in the back-translation for ambiguous input words. In the current tests, if back-translations are judged insufficient to convey the intended meaning, paraphrases are substituted and any lexical translation errors are corrected until acceptable results are achieved. All such paraphrases are displayed below, whether they involve simple word substitutions or more extensive changes.

Military Domain

input	initial translation	back-translation	comments
Watch out for mines around here	tenga cuidado con minas por aquí	Be careful with mines around here	ASR: *mines* misrecognized as *minds* MT: *mines* mistranslated as *minas* ("reserve, mine"); meaning interactively changed to *minas* ("booby trap, ground-emplaced mine, land mine")
[retranslation]	tenga cuidado con minas por aquí	Be careful with mines around here	Spanish is unchanged, but correct meaning has been verified.
We confiscated their arms last week	confiscamos sus brazos la semana pasada	We confiscated their arms last week	MT: *arms* mistranslated as *brazos* ("brachia"); retranslated as *armas* ("arm, gun, weapon")
[retranslation]	confiscamos sus armas la semana pasada	We confiscated their weapons last week	
The operation is scheduled for oh 600	la operación es programada para oh 600	The operation is programmed for oh 600.	ASR: *The* misrecognized as *knee*. MT: Translation of *oh 600* is uncertain
The operation is scheduled for 6 a.m.	la operación es programada para 6 a.m.	The operation is programmed for 6 a.m.	MT: Translation of *6 a.m.* is still unclear.
The operation is scheduled for six o'clock in the morning	la operación es programada para las seis de la mañana	The operation is programmed for six in the morning	MT: Translation is now verified, given slight rewording (*six* instead of *six o'clock*).

We're training them as guerrillas	Los entrenamos como guerrillas	We train them like guerrillas	ASR: Correct spelling (*c.f. gorillas*) was produced. MT: Bug: tolerable back-translation error: *like* should be *as*.

NOTE: For the military domain and more generally, improved translation of *day times*, especially when expressed as *military time*, is clearly needed.

Emergency Response Domain

input	**initial translation**	**back-translation**	**comments**
Tell them to drop the food at headquarters	Dígales a ellos que dejen caer la comida en cuartel general	Tell them to them that they drop the food in headquarters	MT: Bug: tolerable Spanish>English mistranslation of pattern "*digales a ellos que* <action>" ("*tell them to* <action>"); *drop* mistranslated as "drop down, drop away, let fall, ...", but no suitable alternate meaning found; substituting *drop off*
... drop off ...	dígales a ellos que dejen caer la comida en cuartel general	Tell them to them that they drop the food in headquarters	MT: translation and back-translation unchanged; still no suitable meaning; substituting *leave*
... leave ...	Dígales a ellos que dejen la comida en cuartel general	Tell them to them that they leave the food at headquarters	MT: back-translation and Meaning Cues now okay
We need more shovels and crowbars right now	Necesitamos más palas y más palancas ahora mismo	we need more shovels and more levers right now	MT: back-translation *levers* is considered okay for *crowbars*
It's a matter of life and death	es cuestión de la vida y la muerte	it is issue of life and Death	MT: capitalization of *death* prompts uncertainty; rephrasing
It's absolutely critical.	Es absolutamente crítico.	it's absolutely critical	MT: meaning cues for *critical* are okay: "final, significant, key, crucial ..."
These people are desperately short of water	Estas personas andan desesperadamente escasas de agua.	These people are desperately scarce of water	MT: Spanish is okay, but poor back-translation of *escasas de* (should be "short of/low on") gives false negative, low confidence. Substituting *low on*.
.. low on ...	Estas personas andan desesperadamente de capa caída en agua.	These people incur in desperately on water.	MT: worse; rephrasing
These people are desperate for water	estas personas están desesperadas para agua.	These people are desperate for water.	MT: Preposition error in Spanish (*para* should be *por*) gives false positive, but meaning is clear

Law Enforcement Domain

input	**initial translation**	**back-translation**	**comments**
Step away from the car	Aléjese del coche	Get away from the car	MT: *get away* is acceptable for *step away*
May I see your license, please	Que pueda ver su licencia, por favor.	That I can see your license, please.	MT: Unacceptable mistranslation of pattern "*que pueda* <action>*, por favor*" ("*may I* <action>, *please*"); rephrasing
Show me your license, please	Muéstreme su licencia, por favor.	Show me your license, please	
Keep your hands where I can see them	Conserve sus manos donde las puedo ver.	Preserve your hands where I can see them.	MT: *keep* mistranslated as *conserve* ("take, hold, maintain, save, retain, preserve, ..."); retranslated as *mantenga* ("keep")

[retranslation]	Mantenga sus manos donde las puedo ver	Keep your hands where I can see them	
How long have you been living at this address?	Cuánto tiempo usted ha vivido en esta dirección?	How long have you been living in this address?	MT: minor but tolerable error with prepositions
Who's your insurer	Quién es su asegurador	Who is your insurer	

NOTE: General-purpose Spanish>English pattern "*que pueda* <action>, *por favor*" ("*may I* <action>, *please*") requires fix for all domains.

4 Translation Shortcuts

Having summarized our approach to highly interactive speech translation and discussed the advantages of very broad lexical and grammatical coverage for domain switching, we now turn to the use of Translation Shortcuts™ in domain ports. This section briefly describes the facility; and Section 5 explains the methods for quickly updating Shortcuts as an element of a rapid port.

A Translation Shortcut contains a short translation, typically of a sentence or two, which has been pre-verified, whether by a human translator or through the use of the system's interactive tools. Thus re-verification of the translation is unnecessary. In this respect, Translation Shortcuts provide a kind of translation memory. However, it is a handmade sort of memory (since Shortcuts are composed by linguists or explicitly saved by users) and a highly interactive sort as well (since users can browse or search for Shortcuts, can make and categorize their own Shortcuts, and are advised when the input matches a Shortcut). It is in the ease of composition or customization, as well as in the quality of the interaction, that innovation can be claimed.

We can consider the quality of interaction first. Access to stored Shortcuts is very quick, with little or no need for text entry. Several facilities contribute to meeting this design criterion:

- A *Shortcut Search* facility can retrieve a set of relevant Shortcuts given only keywords or the first few characters or words of a string. The desired Shortcut can then be executed with a single gesture (mouse click or stylus tap) or voice command.

  NOTE: If no Shortcut is found, the system automatically allows users access to the full power of broad-coverage, interactive speech translation. Thus, a seamless transition is provided between the Shortcuts facility and full, broad-coverage translation.
- A *Translation Shortcuts Browser* is provided, so that users can find needed Shortcuts by traversing a tree of Shortcut categories. Using this interface, users can execute Shortcuts by tapping or clicking alone.

Figure 1 below shows the Shortcut Search and Shortcuts Browser facilities in use.

- On the left, the Translation Shortcuts Panel contains the Translation Shortcuts Browser, split into two main areas, Shortcuts Categories (above) and Shortcuts List (below).

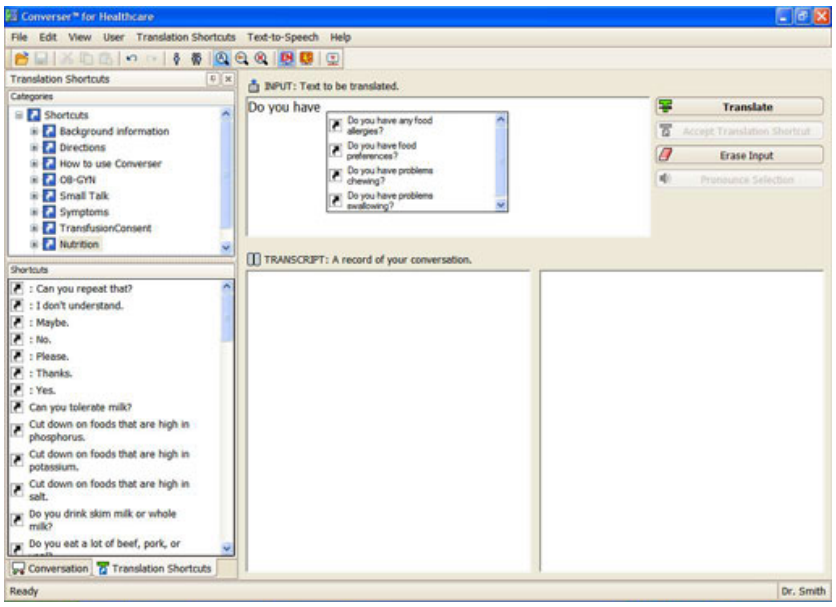


Figure 1: The Input Screen, showing the Translation Shortcuts Browser and Shortcut Search facilities. Note the new **Nutrition** category and the results of automatic Shortcut Search.

- The Categories section of the Panel shows current selection of the **Nutrition** category, containing frequently used questions and answers for a nutrition interview. This new category was created overnight, as described in Section 5, below. Currently hidden is its **Staff** subcategory, containing expressions most likely to be used by health care staff members. There is also a **Patients** subcategory, used for patient responses. Categories for **Background information**, **Directions**, etc. are also visible.
- Below the Categories section is the Shortcuts List section, containing a scrollable list of alphabetized Shortcuts. Double clicking on any visible Shortcut in the List will execute it. Clicking once will select and highlight a Shortcut. Typing **Enter** will execute any currently highlighted Shortcut.

We turn our attention now to the Input Window, which does double duty for Shortcut Search and arbitrary text entry for full translation. The search facility is also shown in Figure 1.

- Shortcuts Search begins automatically as soon as text is entered by any means – voice, handwriting, touch screen, or standard keyboard – into the Input Window.
- The **Shortcuts Drop-down Menu** appears just below the Input Window, as soon as there are results to be shown. The user has entered "Do you have". The drop-down menu shows the results of a search within the new **Nutrition** category based upon these initial characters.

If the user goes on to enter the exact text of any Shortcut in this category, *e.g*. "Do you have any food allergies?," the interface will show that this is in fact a Shortcut, so that verification of translation accuracy will not be necessary. However, final text not matching a Shortcut, *e.g*. "Do you have any siblings?" will be passed to the routines for full translation with verification.

A *Personal Translation Shortcuts*™ facility is in progress for future versions of the system: once a user has verified a translation via the interactive facilities described above, he or she can save it for future reuse by pressing a **Save as Shortcut** button. The new custom Shortcut will then be stored in a personal profile. Facilities for sharing Shortcuts will also be provided.

5 Rapid Customization of Translation Shortcuts for New Domains

Translation Shortcuts are stored and distributed as text-format XML files. Each file contains information about which categories (*e.g*. **Nutrition**) and subcategories (**Staff**, **Patient**, etc.) to which each phrase belongs. Since Shortcuts are stored as external data files, integration of new Shortcuts into the system is straightforward and highly scalable. Once we have built a database of frequently used expressions and their translations for a given domain (in which there may be thousands of expressions or just a few), we can automatically generate the associated files in XML format in minutes. Once this new file is added to the appropriate directory, the Shortcuts become usable in the next session for text- or voice-driven searching and browsing. The entire sequence can be completed overnight. In one case, the Nutrition Department of a major hospital submitted several pages of frequently asked questions, which were entered, translated, re-generated as an XML file, and integrated into the system for demonstration the next day.

```
<Category categoryName1= "Nutrition" categoryName2="Alimentación">
    <Categories>
     <Category categoryName1="Staff" categoryName2="Personal">
       <Shortcuts>
         <Shortcut categoryPath="Nutrition\\Staff">
           <Language1Text>Do you have any food allergies?</Language1Text>
           <Language2Text>¿Tiene alguna alergia a alguna comida?</Language2Text>
         </Shortcut>
         <Shortcut categoryPath="Nutrition\\Staff">
           <Language1Text>Can you tolerate milk?</Language1Text>
           <Language2Text>¿Tolera la leche?</Language2Text>
         </Shortcut>
         <Shortcut categoryPath="Nutrition\\Staff">
           <Language1Text>Do you follow a special diet at home?</Language1Text>
           <Language2Text>¿Sigue alguna dieta especial en casa?</Language2Text>
         </Shortcut>
       </Shortcuts>
     </Category>
     </Categories>
    </Category>
```

Figure 2: Sample fragment of an automatically formatted Translation Shortcuts file for the **Nutrition>Staff** category and subcategory.

6 Use of the Glossary Import for Quick Domain Switching

Similarly, our system includes a glossary import function which supports quick addition of domain-specific or other custom lexical information (*e.g.*, site-specific or client-specific vocabulary), once again in text format. This glossary file may provide additional terms or may stipulate preferred (and thus overriding) translations for existing terms. The glossary file is automatically generated from a simple, two-column text-format file in which each line contains the source-language and target-language terms. A system utility will then generate the necessary linguistic markup (in curly brackets in Figure 3) for each of the terms. (Markup can be elaborated as appropriate for the machine translation engine in use, *e.g.* to specify verb sub-categorization, semantic class, etc.) Like the XML file used for Translation Shortcuts, the resulting custom glossary file can simply be placed in the appropriate directory.

```
hemolítico { A, 11, 6, 0, } = hemolytic
hemolitopoyético { A, 11, 6, 0, } = hemolytopoietic
hemolizable { A, 11, 6, 0, } = hemolyzable
hemolización { N, 2, 2, 1, } = hemolyzation
hemolizar { V, 7, 0, 1, } = hemolyze
derecho { A, 11, 6, 0, } = right
```

Figure 3. Sample glossary-import entries for the health care domain.

Here, the entry for *right* establishes the "right-hand" sense as the system-wide default, overriding the current global default sense ("correct"). (The new global default can, however, be overridden in turn by a personally preferred sense as specified by a user's personal profile; and both kinds of preferences can be overridden interactively for any particular input sentence.) The other entries are domain-specific lexical additions for health care not in the general dictionary.

We make no claims for technical innovation in our Glossary Import facility, but simply point out its usefulness for rapid porting, in that new lexical items, or new preferred senses for old items, can be altered per user and from session to session.

7 Conclusion

The principal source of the porting problems affecting most SLT systems to date, we have observed, is that, given the general current reliance upon statistical approaches for both ASR and MT, each new domain has required an extensive and difficult-to-obtain new corpus for best results. One might consider the use of a single very large and quite general corpus (or collection of corpora) for statistical training; but large corpora engender quickly increasing perplexity and error rates, so this very-broad-coverage approach has generally been avoided.

Our approach, however, has been to adopt a broad-coverage design nevertheless, and to compensate for the inevitable increase in ASR and MT errors by furnishing users with interactive tools for monitoring and correcting these mistakes. (We have to date used rule-based rather than statistical MT components, but comparable interactive facilities could be supplied for the latter as well. Operational prototypes for English<>Japanese and English<>German suggest that the techniques can also be adapted for languages other than English<>Spanish.) Because such interactive tools demand some time and attention, we have also put into place easily modifiable facilities for instant translation of frequent phrases (Translation Shortcuts). And finally, since even systems with very large lexicons will require specialized lexical items or specialized meanings of existing ones, we have implemented a quick glossary import facility, so that lexical items can be added or updated very easily.

Our current SLT system, optimized for health care, is now in use at a medium-sized hospital in New Jersey, with more than twenty machines installed. For this paper, we have applied the same system, without modifications, to sample utterances from the military, emergency service, and law enforcement domains. While this exercise has yielded no quantitative results, readers can judge whether it demonstrates that users can convey mission-critical information with acceptable reliability in multiple domains, even in advance of any porting efforts. Users do pay a price for this flexibility, since time and attention are required for monitoring and correcting to achieve reliable results. However, when users judge that accuracy is not crucial, or when they are unable to monitor and correct, they can simply accept the first translation attempt as is. (A bilingual transcript of each conversation, soon to optionally include the back-translation, is always available for later inspection.) They can also gain considerable time through the use of Translation Shortcuts.

References

Bouillon, P., Rayner, M., et al. 2005. A Generic Multi-Lingual Open Source Platform for Limited-Domain Medical Speech Translation. Presented at *EAMT 2005*, Budapest, Hungary.

Dillinger, M. and Seligman, M. 2006. Converser™ : highly interactive speech-to-speech translation for health care. *HLT-NAACL 2006: Proceedings of the Workshop on Medical Speech Translation (*pp.40-43). New York, NY, USA.

Dillinger, M. and Seligman, M. 2004. System description: A highly interactive speech-to-speech translation system. In: Robert E. Frederking and Kathryn B. Taylor (Eds.),

*Machine translation: from real users to research: 6th conference of the Association for Machine Translation in the Americas -- AMTA 2004 (*pp. 58-63). Berlin: Springer Verlag.

Gao, Y., Liang, G., Zhou, B., Sarikaya, R., et al. (2006). IBM MASTOR system: multilingual automatic speech-to-speech translator. In: *HLT-NAACL 2006: Proceedings of the Workshop on Medical Speech Translation* (pp.57-60). New York, NY, USA.

Karat, C-M. and Nahamoo, D. 2007. Conversational interface technologies. In A. Sears & J. Jacko (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*. Mahwah, NJ: L. Erlbaum.

Koehn, P. 2008. *Statistical Machine Translation*. New York: Cambridge University Press.

Seligman, M.. 2000. Nine Issues in Speech Translation. *Machine Translation, 15*, 149-185.

Seligman, M. and Dillinger, M. 2006. Usability issues in an interactive speech-to-speech translation system for health care. *HLT-NAACL 2006: Proceedings of the Workshop on Medical Speech Translation* (pp. 1-8). New York, NY, USA.

Zong, C. and Seligman, M. 2005. Toward Practical Spoken Language Translation. *Machine Translation, 19*, 113-137.

Speech Translation for Triage of Emergency Phonecalls in Minority Languages

Udhyakumar Nallasamy, Alan W Black, Tanja Schultz, Robert Frederking
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
udhay@cmu.edu,
{awb,ref,tanja}@cs.cmu.edu

Jerry Weltman
Louisiana State University
Baton Rouge,
Louisiana 70802 USA
jweltm2@lsu.edu

Abstract

We describe Ayudame, a system designed to recognize and translate Spanish emergency calls for better dispatching. We analyze the research challenges in adapting speech translation technology to 9-1-1 domain. We report our initial research in 9-1-1 translation system design, ASR experiments, and utterance classification for translation.

1 Introduction

In the development of real-world-applicable language technologies, it is good to find an application with a significant need, and with a complexity that appears to be within the capabilities of current existing technology. Based on our experience in building speech-to-speech translation, we believe that some important potential uses of the technology do not require a full, complete speech-to-speech translation system; something much more lightweight can be sufficient to aid the end users (Gao et al, 2006).

A particular task of this kind is dealing with emergency call dispatch for police, ambulance, fire and other emergency services (in the US the emergency number is 9-1-1). A dispatcher must answer a large variety of calls and, due to the multilingual nature of American society, they may receive non-English calls and be unable to service them due to lack of knowledge of the caller language.

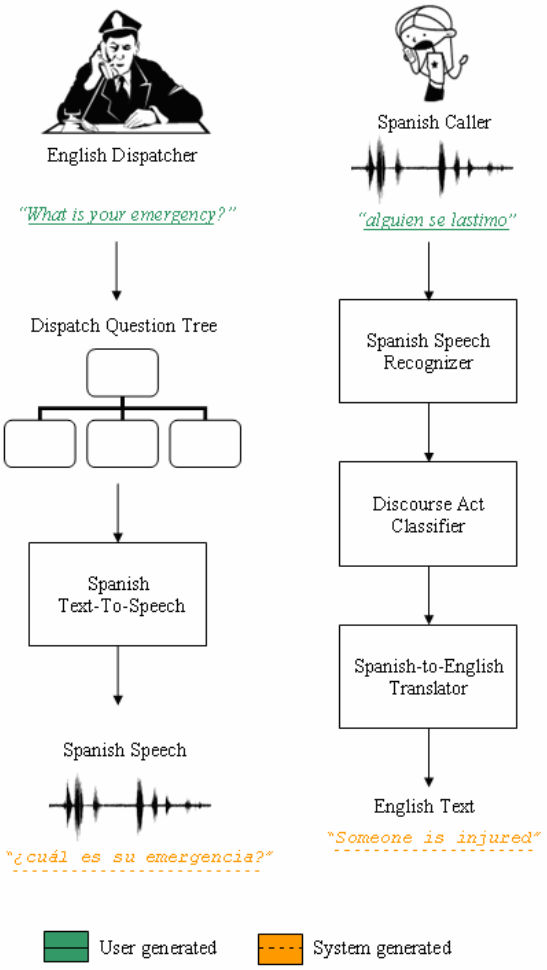


Figure 1. Ayudame system architecture

As a part of a pilot study into the feasibility of dealing with non-English calls by a mono-lingual English-speaking dispatcher, we have designed a translation system that will aid the dispatcher in communicating without understanding the caller's language.

Coling 2008: Proceedings of the workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, pages 48–53
Manchester, August 2008

The fundamental idea is to use utterance classification of the non-English input. The non-English is first recognized by a speech recognition system; then the output is classified into a small number of domain-specific classes called Domain Acts (DAs) that can indicate directly to the dispatcher the general intended meaning of the spoken phrase. Each DA may have a few important parameters to be translated, such as street addresses (Levin et al, 2003; Langley 2003). The dispatcher can then select from a limited number of canned responses to this through a simple menu system. We believe the reduction in complexity of such a system compared to a full speech-to-speech translation will be advantageous because it should be much cheaper to construct, easier to port to new languages, and, importantly, sufficient to do the job of processing emergency calls.

In the "NineOneOne" project, we have designed an initial prototype system, which we call "Ayudame" (Spanish word for "Help me"). Figure 1 gives an overview of the system architecture.

2 The NineOneOne Domain

Our initial interest in this domain was due to contact from the Cape Coral Police Department (CCPD) in Florida. They were interested in investigating how speech-to-speech translations could be used in emergency 9-1-1 dispatch systems. Most current emergency dispatching centers use some proprietary human translation service, such as Language Line (Language Line Services). Although this service provides human translation services for some 180 languages, it is far from ideal. Once the dispatcher notes that the caller cannot speak/understand English, they must initiate the call to Language Line, including identifying themselves to the Language Line operator, before the call can actually continue. This delay can be up to a minute, which is not ideal in an emergency situation.

After consulting with CCPD, and collecting a number of example calls, it was clear that full speech-to-speech translation was not necessary and that a limited form of translation through utterance classification (Lavie et al, 2001) might be sufficient to provide a rapid response to non-English calls. The language for our study is Spanish. Cape Coral is on the Gulf Coast of Florida and has fewer Spanish speakers than e.g. the Miami area, but still sufficient that a number of calls are made to their emergency service in Spanish, yet many of their operators are not sufficiently fluent in Spanish to deal with the calls.

There are a number of key pieces of information that a dispatcher tries to collect before passing on the information to the appropriate emergency service. This includes things like location, type of emergency, urgency, if anyone is hurt, if the situation is dangerous, etc. In fact many dispaching organizations have existing, well-defined policies on what information they should collect for different types of emergencies.

3 Initial system design

Based on the domain's characteristics, in addition to avoiding full-blown translation, we are following a highly asymmetrical design for the system (Frederking et al, 2000). The dispatcher is already seated at a workstation, and we intend to keep them "in the loop", for both technical and social reasons. So in the dispatcher-to-caller direction, we can work with text and menus, simplifying the technology and avoiding some cognitive complexity for the operator. So in the dispatcher-to-caller direction we require

- ***no*** English ASR,
- ***no*** true English-to-Spanish MT, and
- simple, domain-limited, Spanish speech synthesis.

The caller-to-dispatcher direction is much more interesting. In this direction we require

- Spanish ASR that can handle emotional spontaneous telephone speech in mixed dialects,
- Spanish-to-English MT, but
- ***no*** English Speech Synthesis.

We have begun to consider the user interfaces for Ayudame as well. For ease of integration with pre-existing dispatcher workstations, we have chosen to use a web-based graphical interface. For initial testing of the prototype, we plan to run in "shadow" mode, in parallel with live dispatching using the traditional approach. Thus Ayudame will have a listen-only connection to the telephone line, and will run a web server to interact with the dispatcher. Figure 2 shows an initial design of the web-based interface. There are sections for a transcript, the current caller utterance, the current dispatcher response choices, and a button to transfer the interaction to a human translator as a fall-back option. For each utterance, the DA classification is displayed in addition to the actual utterance (in case the dispatcher knows some Spanish).

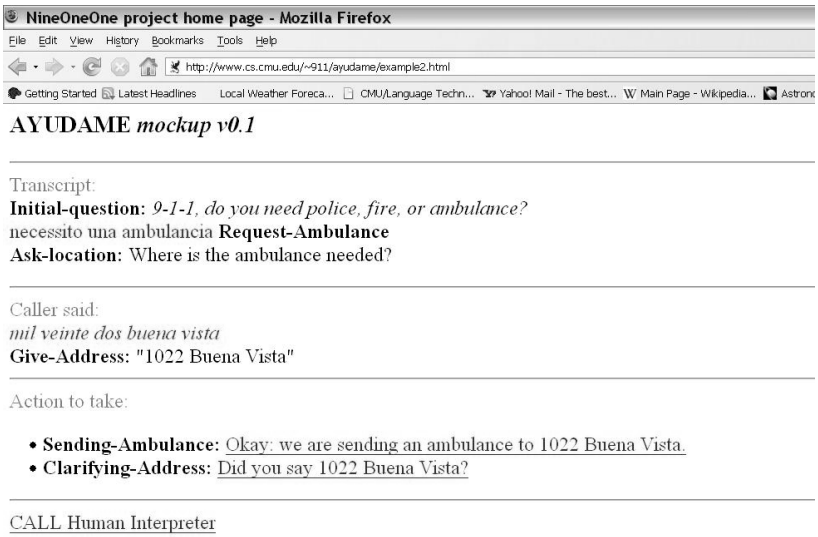


Figure 2. Example of initial GUI design

4 Automatic Speech Recognition

An important requirement for such a system is the ability to be able to recognize the incoming non-English speech with a word error rate sufficiently low for utterance classification and parameter translation to be possible. The issues in speech recognition for this particular domain include: telephone speech (which is through a limited bandwidth channel); background noise (the calls are often from outside or in noisy places); various dialects of Spanish, and potential stressed speech. Although initially we expected a substantial issue with recognizing stressed speakers, as one might expect in emergency situations, in the calls we have collected so far, although it is not a negligible issue, it is far less important that we first expected.

The Spanish ASR system is built using the Janus Recognition Toolkit (JRTk) (Finke et al, 1997) featuring the HMM-based IBIS decoder (Soltau et al, 2001). Our speech corpus consists of 75 transcribed 9-1-1 calls, with average call duration of 6.73 minutes (min: 2.31 minutes, max: 13.47 minutes). The average duration of Spanish speech (between interpreter and caller) amounts to 4.8 minutes per call. Each call has anywhere from 46 to 182 speaker turns with an average of 113 speaker turns per call. The turns that have significant overlap between speakers are omitted from the training and test set. The acoustic models are trained on 50 Spanish 9-1-1 calls, which amount to 4 hours of speech data.

The system uses three-state, left-to-right, subphonetically tied acoustic models with 400 context-dependent distributions with the same number of codebooks. Each codebook has 32 gaussians per state. The front-end feature extraction uses standard 39 dimensional Mel-scale cepstral coefficients and applies Linear Discriminant Analysis (LDA) calculated from the training data. The acoustic models are seeded with initial alignments from GlobalPhone Spanish acoustic models trained on 20 hours of speech recorded from native Spanish speakers (Schultz et al, 1997). The vocabulary size is 65K words. The language model consists of a trigram model trained on the manual transcriptions of 40 calls and interpolated with a background model trained on GlobalPhone Spanish text data consisting of 1.5 million words (Schultz et al, 1997). The interpolation weights are determined using the transcriptions of 10 calls (development set). The test data consists of 15 telephone calls from different speakers, which amounts to a total of 1 hour. Both development and test set calls consisted of manually segmented and transcribed speaker turns that do not have a significant overlap with other speakers. The perplexity of the test set according to the language model is 96.7.

The accuracy of the Spanish ASR on the test set is 76.5%. This is a good result for spontaneous telephone-quality speech by multiple unknown speakers, and compares favourably to the ASR accuracy of other spoken dialog systems. We had initially planned to investigate novel ASR techniques designed for stressed speech and multiple dialects, but to our surprise these do not

seem to be required for this application. Note that critical information such as addresses will be synthesized back to the caller for confirmation in the full system. So, for the time-being we will concentrate on the accuracy of the DA classification until we can show that improving ASR accuracy would significantly help.

5 Utterance Classification

As mentioned above, the translation approach we are using is based on utterance classification. The Spanish to English translation in the Ayudame system is a two-step process. The ASR hypothesis is first classified into domain-specific Domain Acts (DA). Each DA has a predetermined set of parameters. These parameters are identified and translated using a rule-based framework. For this approach to be accomplished with reasonable effort levels, the total number of types of parameters and their complexity must be fairly limited in the domain, such as addresses and injury types. This section explains our DA tagset and classification experiments.

5.1 Initial classification and results

The initial evaluation (Nallasamy et al, 2008) included a total of 845 manually labeled turns in our 9-1-1 corpus. We used a set of 10 tags to annotate the dialog turns. The distribution of the tags are listed below

Tag (Representation)	Frequency
Giving Name	80
Giving Address	118
Giving Phone number	29
Requesting Ambulance	8
Requesting Fire Service	11
Requesting Police	24
Reporting Injury/Urgency	61
Yes	119
No	24
Others	371

Table 1. Distribution of first-pass tags in the corpus.

We extracted bag-of-word features and trained a Support Vector Machine (SVM) classifier (Burges, 1998) using the above dataset. A 10-fold stratified cross-validation has produced an average accuracy of 60.12%. The accuracies of individual tags are listed below.

Tag	Accuracy (%)
Giving Name	57.50
Giving Address	38.98
Giving Phone number	48.28
Req. Ambulance	62.50
Req. Fire Service	54.55
Req. Police	41.67
Reporting Injury/Urgency	39.34
Yes	52.94
No	54.17
Others	75.74

Table 2. Classification accuracies of first-pass tags.

5.2 Tag-set improvements

We improved both the DA tagset and the classification framework in our second-pass classification, compared to our initial experiment. We had identified several issues in our first-pass classification:

- We had forced each dialog turn to have a single tag. However, the tags and the dialog turns don't conform to this assumption. For example, the dialog "Yes, my husband has breathing problem. We are at two sixty-one Oak Street"[1] should get 3 tags: "Yes", "Giving-Address", "Requesting-Ambulance".
- Our analysis of the dataset also showed that the initial set of tags are not exhaustive enough to cover the whole range of dialogs required to be translated and conveyed to the dispatcher.

We made several iterations over the tagset to ensure that it is both compact and achieves requisite coverage. The final tag set consists of 67 entries. We manually annotated 59 calls with our new tagset using a web interface. The distribution of the top 20 tags is listed below. The whole list of tags can be found in the NineOneOne project webpage: http://www.cs.cmu.edu/~911/

[1] The dialog is English Translation of *"sí, mi esposo le falta el aire. es acá en el dos sesenta y uno Oak Street"*. It is extracted from the transcription of a CCPD 9-1-1 emergency call, with address modified to protect privacy

Tag (Representation)	Frequency
Yes	227
Giving-Address	133
Giving-Location	113
Giving-Name	107
No	106
Other	94
OK	81
Thank-You	51
Reporting-Conflict	43
Describing-Vehicle	42
Giving-Telephone-Number	40
Hello	36
Reporting-Urgency-Or-Injury	34
Describing-Residence	28
Dont-Know	19
Dont-Understand	16
Giving-Age	15
Goodbye	15
Giving-Medical-Symptoms	14
Requesting-Police	12

Table 3. Distribution of top 20 second-pass tags

The new tagset is hierarchical, which allows us to evaluate the classifier at different levels of the hierarchy, and eventually select the best trade-off between the number of tags and classification accuracy. For example, the first level of tags for reporting incidents includes the five most common incidents, viz, Reporting-Conflict, Reporting-Robbery, Reporting-Traffic-accident, Reporting-Urgency-or-Injury and Reporting-Fire. The second level of tags are used to convey more detailed information about the above incidents (eg. Reporting-Weapons in the case of conflict) or rare incidents (eg. Reporting-Animal-Problem).

5.3 Second-pass classification and Results

We also improved our classification framework to allow multiple tags for a single turn and to easily accomodate any new tags in the future. Our earlier DA classification used a multi-class classifier, as each turn was restricted to have a single tag. To accomodate multiple tags for a single turn, we trained binary classifiers for each tag. All the utterances of the corresponding tag are marked positive examples and the rest are marked as negative examples. Our new data set has 1140 dialog turns and 1331 annotations. Note that the number of annotations is more than the number of labelled turns as each turn may have multiple tags. We report classification accuracies in the following table for each tag based on 10-fold cross-validation:

Tag (Representation)	Accuracy (%)
Yes	87.32
Giving-Address	42.71
Giving-Location	87.32
Giving-Name	42.71
No	37.63
Other	54.98
OK	72.5
Thank-You	41.14
Reporting-Conflict	79.33
Describing-Vehicle	96.82
Giving-Telephone-Number	39.37
Hello	38.79
Reporting-Urgency-Or-Injury	49.8
Describing-Residence	92.75
Dont-Know	41.67
Dont-Understand	36.03
Giving-Age	64.95
Goodbye	87.27
Giving-Medical-Symptoms	47.44
Requesting-Police	79.94

Table 4. Classification accuracies of individual second-pass tags

The average accuracy of the 20 tags is 58.42%. Although multiple classifiers increase the computational complexity during run-time, they are independent of each other, so we can run them in parallel. To ensure the consistency and clarity of the new tag set, we had a second annotator label 39 calls. The inter-coder agreement (Kappa coefficient) between the two annotators is 0.67. This is considered substantial agreement between the annotators, and confirms the consistency of the tag set.

6 Conclusion

The work reported here demonstrates that we can produce Spanish ASR for Spanish emergency calls with reasonable accuracy (76.5%), and classify manual transcriptions of these calls with reasonable accuracy (60.12% on the original tagset,

58.42% on the new, improved tagset). We believe these results are good enough to justify the next phase of research, in which we will develop, user-test, and evaluate a full pilot system. We are also investigating a number of additional techniques to improve the DA classification accuracies. Further we believe that we can design the overall dialog system to ameliorate the inevitable remaining misclassifications, based in part on the confusion matrix of actual errors (Nallasamy et al, 2008). But only actual user tests of a pilot system will allow us to know whether an eventual deployable system is really feasible.

Acknowledgements

This project is funded by NSF Grant No: IIS-0627957 "NineOneOne: Exploratory Research on Recognizing Non-English Speech for Emergency Triage in Disaster Response". Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of sponsors.

References

Burges C J C, A tutorial on support vector machines for pattern recognition, In Proc. *Data Mining and Knowledge Discovery*, pp 2(2):955-974, USA, 1998

Finke M, Geutner P, Hild H, Kemp T, Ries K and Westphal M, The Karlsruhe-Verbmobil Speech Recognition Engine, In Proc. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 83-86, Germany, 1997

Frederking R, Rudnicky A, Hogan C and Lenzo K, Interactive Speech Translation in the Diplomat Project, *Machine Translation Journal 15(1-2), Special issue on Spoken Language Translation*, pp. 61-66, USA, 2000

Gao Y, Zhou B, Sarikaya R, Afify M, Kuo H, Zhu W, Deng Y, Prosser C, Zhang W and Besacier L, IBM MASTOR SYSTEM: Multilingual Automatic Speech-to-Speech Translator, In Proc. *First International Workshop on Medical Speech Translation*, pp. 53-56, USA, 2006

Langley C, Domain Action Classification and Argument Parsing for Interlingua-based Spoken Language Translation. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2003

Language Line Services http://www.languageline.com

Lavie A, Balducci F, Coletti P, Langley C, Lazzari G, Pianesi F, Taddei L and Waibel A, Architecture and Design Considerations in NESPOLE!: a Speech Translation System for E-Commerce Applications,. In Proc. *Human Language Technologies (HLT)*, pp 31-34, USA, 2001

Levin L, Langley C, Lavie A, Gates D, Wallace D and Peterson K, Domain Specific Speech Acts for Spoken Language Translation, In Proc. *4th SIGdial Workshop on Discourse and Dialogue*, pp. 208-217, Japan, 2003

Nallasamy U, Black A, Schultz T and Frederking R, NineOneOne: Recognizing and Classifying Speech for Handling Minority Language Emergency Calls, In Proc. *6th International conference on Language Resources and Evaluation (LREC)*, Morocco, 2008

NineOneOne project webpage [www.cs.cmu.edu/~911]

Schultz T, Westphal M and Waibel A, The GlobalPhone Project: Multilingual LVCSR with JANUS-3, In Proc. *Multilingual Information Retrieval Dialogs: 2nd SQEL Workshop*, pp. 20-27, Czech Republic, 1997

Soltau H, Metze F, F¨ugen C and Waibel A, A One Pass-Decoder Based on Polymorphic Linguistic Context Assignment, In Proc. *IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*, Italy, 2001

Speech to Speech translation for Nurse Patient Interaction

Farzad Ehsani, Jim Kimzey, Elaine Zuber, Demitrios Master
Fluential Inc./ 1153 Bordeaux Dr.,
Sunnyvale, CA 94089
{farzad,jkimzey,elaine,dlm}
@fluentialinc.com

Karen Sudre
TeleNav, Inc./1130 Kifer Road,
Sunnyvale CA, 94086
karens@telenav.com

Abstract

S-MINDS is a speech translation system, which allows an English speaker to communicate with a limited English proficiency speaker easily within a question-and-answer, interview-style format. It can handle dialogs in specific settings such as nurse-patient interaction, or medical triage. We have built and tested an English-Spanish system for enabling nurse-patient interaction in a number of domains in Kaiser Permanente achieving a total translation accuracy of 92.8% (for both English and Spanish). We will give an overview of the system as well as the quantitative and qualitatively system performance.

1 Introduction

There has been a lot of work in the area of speech to speech translation by CMU, IBM, SRI, University of Geneva and others. In a health care setting, this technology has the potential to give nurses and other clinical staff immediate access to consistent, easy-to-use, and accurate medical interpretation for routine patient encounters. This could greatly improve safety and quality of care for patients who speak a different language from that of the healthcare provider.

This paper describes the building and testing of a speech translation system, S-MINDS (Speaking Multilingual Interactive Natural Dialog System), built in less than 3 months with Kaiser Permanente Hospital in San Francisco, CA. The system was able to gain fairly robust results for the domains that it was designed for, and we believe that it does demonstrate that building and deploying a successful speech translation system is becoming possible and even commercially viable.

2 Background

The number of people in the U.S. who speak a language other than English is large and growing, and Spanish is the most commonly spoken language next to English. According to the 2000 census, 18% of the U.S. population over age 5 (47 million people) did not speak English at home—a 48% increase from 1990. In 2000, 8% of the population (21 million) was LEP (Limited English Proficiency), with more than 65% of that population (almost 14 million people) speaking Spanish.

A body of research shows that language barriers impede access to care, compromise quality, and increase the risk of adverse outcomes. Although trained medical interpreters and bilingual healthcare providers are effective in overcoming such language barriers, the use of semi-fluent healthcare professionals and *ad hoc* interpreters (such as family members and friends) cause more interpreter errors and lower quality of care (Flores 2005).

When friends and family interpret, they are prone to omit, add, and substitute information. Often they inject their own opinions and observations, or impose their own values and judgments, rather than interpreting what the patient actually said. Frequently these *ad hoc* interpreters have limited English capabilities themselves and are unfamiliar with medical terminology. Furthermore, many patients are reluctant to disclose private or sensitive information in front of a family member, thus giving the doctor an incomplete picture; this sometimes prevents a

Coling 2008: Proceedings of the workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, pages 54–59
Manchester, August 2008

correct diagnosis. For example, a battered woman is unlikely to reveal the true cause of her injuries if her husband is being used as the interpreter.

The California Academy of Family Physicians Foundation conducted practice visits in 2006 and found that, "Although they realize the use of family members or friends as interpreters is probably not the best means of interpretation, all practice teams use them." (Chen et al 2007)

3 System Description

Fluential's speech translation system, S-MINDS[1], has a hybrid architecture (Figure 1) that combines multiple ASR engines and multiple translation engines. This approach only slightly increases the development cost of new translation applications, but it greatly improves the accuracy and the coverage of the system by leveraging the strengths of both statistical and grammar/rules-based systems. Furthermore, this hybrid approach enables rapid integration of new speech recognition and translation engines as they become available.

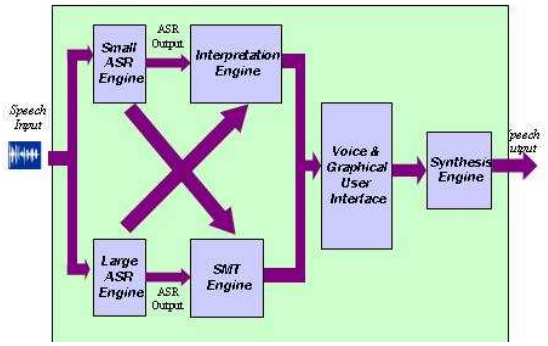


Figure 1. The hybrid system architecture of S-MINDS combines multiple ASR engines with an interpretation engine and an SMT engine. Note that this figure describes the interaction in English to-second language direction only. The 2nd language-to-English direction has only the small ASR engine and the interpretation engine.

3.1 Components of Speech Translation System

S-MINDS has a modular architecture with the components described below. All of these components already exist, so they will not need to be developed to conduct the research proposed in Phase I.

3.1.1 ASR Engine

S-MINDS employs multiple acoustic engines so the best engine can be chosen for each language. Within each language, two separate language models are active at the same time, telling the ASR engines which words and phrases to recognize. A smaller, more directed language model with higher accuracy is used to capture important and frequently used concepts. For less frequently used concepts, a larger language model that generally has broader coverage but somewhat lower accuracy is used. The combination of these two provides high accuracy for responses that can be anticipated and slightly lower accuracy but broader coverage for everything else. This method also allows development of new domains with very little data—for each domain, only a new domain-specific small language model needs to be built.

3.1.2 Interpretation Engine

Fluential has created an interpretation engine that is an alternative to an SMT engine. The S-MINDS interpretation engine uses information extracted from the output of the ASR engine and then performs a paraphrase translation in semantic space. This process is similar to what human interpreters do when they convey the essential meaning without providing a literal translation.

The advantage of an interpretation engine is that new domains can be added more quickly and with less data than is possible with an SMT engine. For high–volume, routine interactions, an interpretation engine can be extremely fast and highly accurate; however, the translation may lose some of the nuance. Again, this means that highly accurate target applications can be built with very little data—only a few examples of each concept are needed to train the interpretation engine.

3.1.3 Statistical Machine Translation Engine

For the S-MINDS SMT engine, Fluential is developing a novel approach that has generally improved the accuracy of speech translation systems.[2] This approach capitalizes on the intuition that language is broadly divided into two levels:

[1] Speaking Multilingual Interactive Natural Dialog System

[2] This effort is ongoing; it has not yet been fully implemented.

structure and vocabulary. Traditional statistical approaches force the system to learn both types of information simultaneously. However, if the acquisition of structural information is kept separate from the acquisition of vocabulary, the resulting system should learn both levels more efficiently. And by modifying the existing corpus to separate structure and vocabulary, we have been able to take full advantage of all the information in the bilingual corpus, producing higher-quality MT without requiring large bodies of training data. The most recent modification to this approach was the use of distance-based ordering (Zens and Ney, 2003) and lexicalized ordering (Tillmann and Zhang, 2005) to allow for multiple language models, including non-word models such as part-of-speech improved search algorithm, in order to improve its speed and efficiency.

3.1.4 VUI+GUI System

S-MINDS has a flexible user interface that can be configured to use VUI only or VUI+GUI for either the English speaker or the second-language speaker. Also, the English speaker can experience a different user interface than the second-language speaker. The system has the flexibility to use multiple types of microphones, including open microphones, headsets, and telephone headsets. Speech recognition can be confirmed by VUI, GUI, or both, and it can be configured to verify all utterances, no utterances, or just utterances that fall below a certain confidence level.

3.1.5 Synthesis Engine

S-MINDS can use text-to-speech (TTS) synthesis throughout the system; alternatively, it can use TTS in its SMT-based system and chunk-based recordings that are spliced together in its interpretation engine. Fluential licenses its TTS technology from Cepstral, and other vendors. In general we do not expect to be doing any research and development activities in this area, as Cepstral can easily create good synthesis models from the 10 hours of provided speech data (Schultz and Black, 2006, Peterson, 2007).

4 System Building

Fluential conducted fiver activities in order to build the system. They included: (1) Defining the task, (2) Collecting speech data to model nurse-patient interactions, (3) Building and testing a speech translation system in English and Spanish, (4) Using the system with patients and nurses and collecting data to measure system performance, and (5) Analyzing the results.

To define the task, Fluential conducted a two-hour focus group with six registered nurses from Med/Surg unit of Kaiser Medical Center in San Francisco. In this focus group, the nurses identified six nurse-patient encounter types that they wanted to use for the evaluation. These were: (1) Greeting/Goodbye, (2) Vital Signs, (3) Pain Assessment, (4) Respiratory Assessment, (5) Blood Sugar, (6) Placement of an I.V.

Fluential then collected speech data over a four-week period by recording nurse-patient interactions involving 11 nurses and 25 patients. Fluential also recruited 59 native Spanish speakers who provided speech data using an automated system that walked them through hypothetical scenarios and elicited their responses in Spanish.

The English recognizer had a vocabulary of 2,003 and it was trained with 9,683 utterances. The Spanish recognizer had a vocabulary of 822, and it was trained with 1,556 utterances. We suspect that the vocabulary size in Spanish would have been much bigger if we had more data.

5 System Evaluation

After building and testing the speech translation system, Fluential conducted a two-hour training session for each of the nurses before using the system with patients. A bilingual research assistant explained the study to patients, obtained their consent, and trained them for less than five minutes on the system. Nurses then used the system with Spanish-speaking patients for the six nurse-patient encounters that were built into the system. The system was used by three nurses with eleven patients for a total of 95 nurse-patient encounters creating a total of 500 conversation segments.[3]

To protect patients from a mistranslation, each encounter was remotely monitored by a bilingual interpreter, who immediately notified the nurse any time the system mistranslated. Each encounter was recorded, transcribed, and translated by a human.

3.1 Scoring Accuracy

[3] A conversation segment is a single continuous sequence of speech in a single language plus the translation of what was said.

The human translations were compared to the system's translations and given a score using the Laws Methodology of either Good, Fair, Poor, Mistranslated, or Not Translated. (Laws, 2004). If a translation were scored as Good or Fair, it was considered accurate. If the translation were scored as Poor, Mistranslated, or Not Translated, it was considered inaccurate.

Table 2 and 3 give examples of how we have used Law's method to grade actual interaction results from nurses and patients.

Table 2: Nurse Scoring Examples

What Nurse Said	S-MINDS™ Translation	Human Translation	S-MINDS Accuracy
I will give you an I.V.	Voy a colocarle un cateter para liquidos intravenosos.	Voy a colocarle un cateter de liquidos intravenosos.	Good
Let me check if I can give you medication for that.	Dejeme chequear si puedo darle algun medicamento.	Permitame reviso si puedo darle algun medicamento para eso.	Fair
I will check your…	Yo voy a revisarle los vendajes	Voy a revisar su …	Poor
Did someone take your vitals?	¿Le tomare sus signos vitals?	¿Alguien tomo sus signos vitals?	Mistranslated
Your heart rate is normal.	---	Su frecuencia cardiaca es normal.	Not Translated

Table 3: Patient Scoring Examples

What Patient Said	S-MINDS Translation	Human Translation	S-MINDS Accuracy
No, no tengo dificultad en respirar.	I don't have difficulty breathing.	No, I don't have difficulty breathing.	Good
En la parte baja del estomago.	The lower part of my stomach.	In the lower part of my stomach.	Fair
N/A	N/A	N/A	Poor
N/A	N/A	N/A	Mistranslated
Los huesos.	---	My bones.	Not Translated

6 Results

- Our internal milestones for Phase I was to achieve 80% accuracy using the Laws Methodology. Out of 500 conversation segments, the speech translation system had an overall accuracy rate of 93% combining both nurse- and patient-conversation segments,

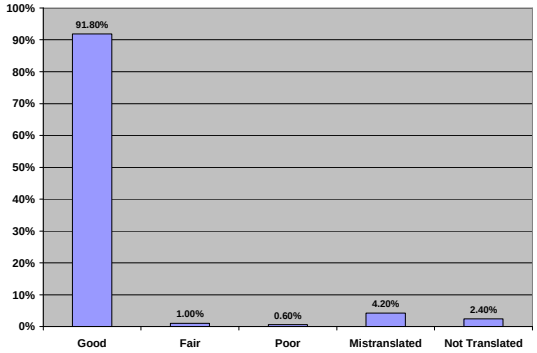


Figure 2: Total results for both nurses and patients.

6.1 Nurse Translation Results

Looking at just nurse conversation segments, the speech translation system had higher accuracy than for patient segments. Out of 404 nurse segments, the speech translation system had a 94% accuracy rate.

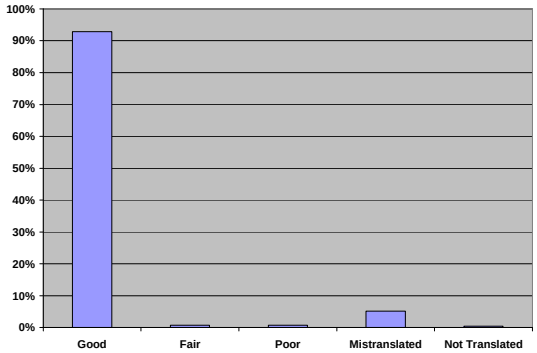


Figure 3: Accuracy for Nurse Conversational Segments

The biggest problem with system performance with nurses was with mistranslations. When nurses tried to say things that were not in the system, the system tried to map their utterances to something that was in the system. In each case of mistranslation, the system told the nurse what it was about to translate, gave the nurse a chance to stop the translation, and then translated the wrong thing when the nurse did not respond. We believe that system performance can be greatly improved in by collecting more speech data from patients and nurses, making changes to the user interface, and improving our training program.

6.2 Patient Translation Results

Looking at just patient conversation segments, the speech translation system had lower overall accuracy than for nurse segments. Out of 96 patient segments, the speech translation system had a 90% accuracy rate.

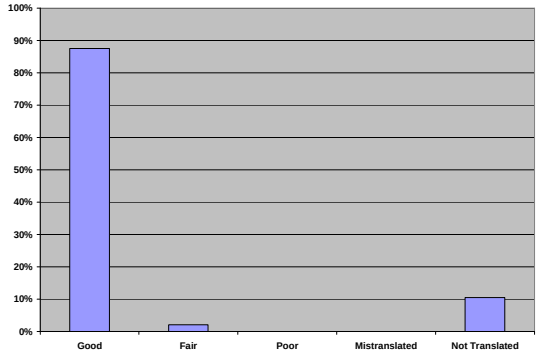


Figure 4: Results for Patients

All of the problems with system performance with patients were with responses that the system was not able to translate. The system never gave a Poor translation or Mistranslated. So there were times when the nurse knew that the patient tried to say something that the system could not translate, but there was never a time when the system gave the nurse false information. However, this percentage is quite high, and in a large context, it might cause additional problems.

6.3 Nurse Survey Results

After each time using the system, the nurses completed a user satisfaction survey that had five statements and asked them assign a 1-to-5 Likert score to each statement with 1 meaning "Strongly Disagree" and 5 meaning "Strongly Agree." Average scores for each question were:

4.7 The speech translator was easy to use.

4.5 The English voice was fluent and easy to understand.

4.4 I understood the patient better because of the speech translator.

4.5 I feel that I am providing better medical care because of the speech translator.

4.7 I would like to use the speech translator with my patients in the future.

6.4 Patient Survey Results

The patients also completed a similar user satisfaction survey, translated to Spanish, after using the system. Their average scores for each question were:

4.6 The speech translator was easy to use.

4.8 The Spanish voice was fluent and easy to understand.

4.7 I understood my nurse better because of the speech translator.

5.0 I feel that I am receiving better medical care because of the speech translator.

4.9 I would like to use the speech translator with my nurse in the future.

6.5 ANOVA Testing

We conducted Analysis of Variance (ANOVA) testing to evaluate whether there were any significant variations in translation accuracy by patient, nurse, or encounter type. There were no significant differences.

7 Discussion

We were able to build and evaluate a system in 3 months and show its utility by nurses and patients in clinical setting. The system seemed to work and was liked by both nurses and patients. The next question is whether such a system can scale and cover a much larger domain, and how much data and training is required to accomplish this.

References

Chen A., et al. (2007), *Addressing Language and Culture—A Practice Assessment for Health Care Professionals*, p3.

Flores Glenn, (2005), "The Impact of Medical Interpreter Services on the Quality of Health Care: A Systematic Review," Medical Care Research and Review, Vol. 62, No. 3, pp. 255-29

Laws, MB, Rachel Heckscher, Sandra Mayo, Wenjun. Li, Ira Wilson, (2004), "A New Method for Evaluating the Quality of Medical Interpretation**,"** *Medical Care.* 42(1):71-80, January 2004

Peterson Kay (2007). Senior Linguist, Cepstral LLC, Personal Communication.

Schultz, Tanja and A. W Black (2006), "Challenges with Rapid Adaptation of Speech Translation Systems to New Language Pairs"

Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-2006), Toulouse, France, May 15-19, 2006.

Tillmann, Christoph and T. Zhang, (2005), "A localized prediction model for statistical machine translation," in *Proceedings of the 43rd Annual Meeting of the ACL*, pp. 557-564, Ann Arbor, June 2005.

Zens, Richard, and H. Ney, (2003), "A comparative study of reordering constraints in statistical machine translation," in *Proceedings of the 41st Annual Meetings of the ACL*, pp. 144-151, Sapporo, Japan, July 2003Association for Computing Machinery. 1983. *Computing Reviews*, 24(11):503-512.

A Small-Vocabulary Shared Task for Medical Speech Translation

Manny Rayner[1], **Pierrette Bouillon**[1], **Glenn Flores**[2], **Farzad Ehsani**[3]
Marianne Starlander[1], **Beth Ann Hockey**[4], **Jane Brotanek**[2], **Lukas Biewald**[5]

[1] University of Geneva, TIM/ISSCO, 40 bvd du Pont-d'Arve, CH-1211 Geneva 4, Switzerland
{Emmanuel.Rayner,Pierrette.Bouillon}@issco.unige.ch
Marianne.Starlander@eti.unige.ch

[2] UT Southwestern Medical Center, Children's Medical Center of Dallas
{Glenn.Flores,Jane.Brotanek}@utsouthwestern.edu

[3] Fluential, Inc, 1153 Bordeaux Drive, Suite 211, Sunnyvale, CA 94089, USA
farzad@fluentialinc.com

[4] Mail Stop 19-26, UCSC UARC, NASA Ames Research Center, Moffett Field, CA 94035–1000
bahockey@ucsc.edu

[5] Dolores Labs
lukeab@gmail.com

Abstract

We outline a possible small-vocabulary shared task for the emerging medical speech translation community. Data would consist of about 2000 recorded and transcribed utterances collected during an evaluation of an English ↔ Spanish version of the Open Source MedSLT system; the vocabulary covered consisted of about 450 words in English, and 250 in Spanish. The key problem in defining the task is to agree on a scoring system which is acceptable both to medical professionals and to the speech and language community. We suggest a framework for defining and administering a scoring system of this kind.

1 Introduction

In computer science research, a "shared task" is a competition between interested teams, where the goal is to achieve as good performance as possible on a well-defined problem that everyone agrees to work on. The shared task has three main components: training data, test data, and an evaluation metric. Both test and training data are divided up into sets of items, which are to be processed. The evaluation metric defines a score for each processed item. Competitors are first given the training data, which they use to construct and/or train their systems. They are then evaluated on the test data, which they have not previously seen.

In many areas of speech and language processing, agreement on a shared task has been a major step forward. Often, it has in effect created a new subfield, since it allows objective comparison of results between different groups. For example, it is very common at speech conference to have special sessions devoted to recognition within a particular shared task database. In fact, a conference without at least a couple of such sessions would be an anomaly. A recent success story in language processing is the Recognizing Textual Entailment (RTE) task[1]. Since its inception in 2004, this has become extremely popular; the yearly RTE workshop now attracts around 40 submissions, and error rates on the task have more than halved.

Automatic medical speech translation would clearly benefit from a shared task. As was made apparent at the initial 2006 workshop in New York[2], nearly every group has both a unique architecture and a unique set of data, essentially making comparisons impossible. In this note, we will suggest an initial small-vocabulary medical

[1] http://www.pascal-network.org/Challenges/RTE/

[2] http://www.issco.unige.ch/pub/SLT_workshop_proceedings_book.pdf

Coling 2008: Proceedings of the workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, pages 60–63
Manchester, August 2008

shared task. The aspect of the task that is hardest to define is the evaluation metric, since there unfortunately appears to be considerable tension between the preferences of medical professionals and speech system implementers. Medical professionals would prefer to carry out a "deep" evaluation, in terms of possible clinical consequences following from a mistranslation. System evaluators will on the other hand prefer an evaluation method that can be carried out quickly, enabling frequent evaluations of evolving systems. The plan we will sketch out is intended to be a compromise between these two opposing positions.

The rest of the note is organised as follows. Section 2 describes the data we propose to use, and Section 3 discusses our approach to evaluation metrics. Section 4 concludes.

2 Data

The data we would use in the task is for the English ↔ Spanish language pair, and was collected using two different versions of the MedSLT system[3]. In each case, the scenario imagines an English-speaking doctor conducting a verbal examination of a Spanish-speaking patient, who was assumed to be have visited the doctor because they were displaying symptoms which included a sore throat. The doctor's task was to use the translation system to determine the likely reason for the patient's symptoms.

The two versions of the system differed in terms of the linguistic coverage offered. The more restricted version supported a minimal range of English questions (vocabulary size, about 200 words), and only allowed the patient to respond using short phrases (vocabulary size, 100 words). Thus for example the doctor could ask "How long have you had a sore throat?", and the patient would respond *Hace dos días* ("for two days"). The less restricted version supported a broader range of doctor questions (vocabulary size, about 450 words), and allowed the patient to respond using both short phrases and complete sentences (vocabulary size, about 225 words). Thus in response to "How long have you had a sore throat?", the patient could say either *Hace dos días* ("for two days") or *Tengo dolor en la garganta hace dos días* ("I have had a sore throat for two days").

Data was collected in 64 sessions, carried out over two days in February 2008 at the University of Texas Medical Center, Dallas. In each session, the part of the "doctor" was played by a real physician, and the part of the "patient" by a Spanish-speaking interpreter. This resulted in 1005 English utterances, and 967 Spanish utterances. All speech data is available in SPHERE-headed form, and totals about 90 MB. A master file, organised in spreadsheet form, lists metadata for each recorded file. This includes a transcription, a possible valid translation (verified by a bilingual translator), IDs for the "doctor", the "patient", the session and the system version, and the preceding context. Context is primarily required for short answers, and consists of the most recent preceding doctor question.

3 Evaluation metrics

The job of the evaluation component in the shared task is to assign a score to each translated utterance. Our basic model will be the usual one for shared tasks in speech and language. Each processed utterance will be assigned to a category; each category will be associated with a specified score; the score for a complete testset will the sum of the scores for all of its utterances. We thus have three sub-problems: deciding what the categories are, deciding how to assign a category to a processing utterance, and deciding what scores to associate with each category.

3.1 Defining categories

If the system attempts to translate an utterance, there are *a priori* three things that can happen: it can produce a correct translation, an incorrect translation, or no translation. Medical speech translation is a safety-critical problem; a mistranslation may have serious consequences, up to and including the death of the patient. This implies that the negative score for an incorrect translation should be high in comparison to the positive score for a correct translation. So a naive scoring function might be "1 point for a correct translation, 0 points for no translation, –1000 points for an incorrect translation."

However, since the high negative score for a mistranslation is justified by the possible serious consequences, not all mistranslations are equal; some are much more likely than others to result in clinical consequences. For example, consider the possible consequences of two different mistrans-

[3] http://www.issco.unige.ch/projects/medslt/

lations of the Spanish sentence *La penicilina me da alergias*. Ideally, we would like the system to translate this as "I am allergic to penicillin". If it instead says "I am allergic to *the* penicillin", the translation is slightly imperfect, but it is hard to see any important misunderstanding arising as a result. In contrast, the translation "I am *not* allergic to penicillin", which might be produced as the result of a mistake in speech recognition, could have very serious consequences indeed. (Note in passing that both errors are single-word insertions). Another type of result is a nonsensical translation, perhaps due to an internal system error. For instance, suppose the translation of our sample sentence were "The allergy penicillin does me". In this case, it is not clear what will happen. Most users will probably dismiss the output as meaningless; a few might be tempted to try and decipher it, with unpredictable results.

Examples like these show that it is important for the scoring metric to differentiate between different classes of mistranslations, with the differentiation based on possible clinical consequences of the error. For similar reasons, it is important to think about the clinical consequences when the system produces correct translations, or fails to produce a translation. For example, when the system correctly translates "Hello" as *Buenas días*, there are not likely to be any clinical consequences, so it is reasonable to reward it with a lower score than the one assigned to a clinically contentful utterance. When no translation is produced, it also seems correct to distinguish the case where the user was able recover by a suitably rephrasing the utterance from the one where they simply gave up. For example, if the system failed to translate "How long has this cough been troubling you?", but correctly handled the simpler formulation "How long have you had a cough?", we would give this a small positive score, rather than a simple zero.

Summarising, we propose to classify translations into the following seven categories:

1. Perfect translation, useful clinical consequences.
2. Perfect translation, no useful clinical consequences.
3. Imperfect translation, but not dangerous in terms of clinical consequences.
4. Imperfect translation, potentially dangerous.
5. Nonsense.
6. No translation produced, but later rephrased in a way the system handled adequately.
7. No translation produced, but not rephrased in a way the system handled adequately.

3.2 Assigning utterances to categories

At the moment, medical professionals will only accept the validity of category assignments made by trained physicians. In the worst case, it is clearly true that a layman, even one who has received some training, will not be able to determine whether or not a mistranslation has clinical significance.

Physician time is, however, a scarce and valuable resource, and, as usual, typical case and worst case may be very different. Particularly for routine testing during system development, it is clearly not possible to rely on expert physician assessments. We consequently suggest a compromise strategy. We will first carry out an evaluation using medical experts, in order to establish a gold standard. We will then repeat this evaluation using non-experts, and determine how large the differential is in practice.

We initially intend to experiment with two different groups of non-experts. At Geneva University, we will use students from the School of Translation. These students will be selected for competence in English and Spanish, and will receive a few hours of training on determination of clinical significance in translation, using guidelines developed in collaboration with Glenn Flores and his colleagues at the UT Southwestern Medical Center, Texas. Given that the corpus material is simple and sterotypical, we think that this approach should yield a useful approximation to expert judgements.

Although translation students are far cheaper than doctors, they are still quite expensive, and evaluation turn-around will be slow. For these reasons, we also propose to investigate the idea of performing evaluations using Amazon's Mechanical Turk[4]. This will be done by Dolores Labs, a new startup specialising in Turk-based crowdsourcing.

3.3 Scores for categories

We have not yet agreed on exact scores for the different categories, and this is something that is

[4] `http://www.mturk.com/mturk/welcome`

probably best decided after mutual discussion at the workshop. Some basic principles will be evident from the preceding discussion. The scale will be normalised so that failure to produce a translation is counted as zero; potentially dangerous mistranslations will be associated with a negative score large in comparison to the positive score for a useful correct translation. Inability to communicate can certainly be dangerous (this is the point of having a translation system in the first place), but mistakenly believing that one has communicated is usually much worse. As Mark Twain put it: "It ain't what you don't know that gets you into trouble. It's what you know for sure that just ain't so".

3.4 Discarding uncertain responses

Given that both speech recognition and machine translation are uncertain technologies, a high penalty for mistranslations means that systems which attempt to translate everything may easily end up with an average negative score - in other words, they would score worse than a system which did nothing! For the shared task to be interesting, we must address this problem, and in the doctor to patient direction there is a natural way to do so. Since the doctor can reasonably be assumed to be a trained professional who has had time to learn to operate the system, we can say that he has the option of aborting any translation where the machine does not appear to have understood correctly.

We thus relativise the task with respect to a "filter": for each utterance, we produce both a translation in the target language, and a "reference translation" in the source language, which in some way gives information about what the machine has understood. The simplest way to produce this "reference translation" is to show the words produced by speech recognition. When scoring, we evaluate both translations, and ignore all examples where the reference translation is evaluated as incorrect. To go back to the "penicillin" example, suppose that Spanish source-language speech recognition has incorrectly recognised *La penicilina me da alergias* as *La penicilina* no *me da alergias*. Even if this produces the seriously incorrect translation "I am not allergic to penicillin", we can score it as a zero rather than a negative, on the grounds that the speech recognition result already shows the Spanish-speaking doctor that something has gone wrong before any translation has happened. The reference translation may also be produced in a more elaborate way; a common approach is to translate back from the target language result into the source language.

Although the "filtered" version of the medical speech translation task makes good sense in the doctor to patient direction, it is less clear how meaningful it is in the patient to doctor direction. Most patients will not have used the system before, and may be distressed or in pain. It is consequently less reasonable to expect them to be able to pay attention to the reference translation when using the system.

4 Summary and conclusions

The preceding notes are intended to form a framework which will serve as a basis for discussion at the workshop. As already indicated, the key challenge here is to arrive at metrics which are acceptable to both the medical and the speech and language community. This will certainly require more negotiation. We are however encouraged by the fact that the proposal, as presented here, has been developed jointly by representatives of both communities, and that we appear to be fairly near agreement. Another important parameter which we have intentionally left blank is the duration of the task; we think it will be more productive to determine this based on the schedules of interested parties.

Realistically, the initial definition of the metric can hardly be more than a rough guess. Experimentation during the course of the shared task will probably show that some adjustment will be desirable, in order to make it conform more closely to the requirements of the medical community. If we do this, we will, in the interests of fairness, score competing systems using all versions of the metric.

Author Index