IraqComm and FlexTrans: A Speech Translation System and Flexible Framework

Michael W. Frandsen, Susanne Z. Riehemann, and Kristin Precoda
SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025
michael.frandsen@sri.com, susanne.riehemann@sri.com, precoda@speech.sri.com

***Abstract*-SRI International's IraqComm™ system performs bidirectional speech-to-speech machine translation between English and Iraqi Arabic in the domains of force protection, municipal and medical services, and training. The system was developed primarily under DARPA's TRANSTAC Program and includes: speech recognition components using SRI's Dynaspeak® engine; MT components using SRI's Gemini™ and SRInterp; and speech synthesis from Cepstral, LLC. The communication between these components is coordinated by SRI's Flexible Translation (FlexTrans) Framework, which has an intuitive easy-to-use graphical user interface and an eyes-free hands-free mode, and is highly configurable and adaptable to user needs. It runs on a variety of standard portable hardware platforms and was designed to make it as easy as possible to build systems for other languages, as shown by the rapid development of an analogous system in English/Malay.**

I. Overview

The IraqComm™ system translates conversations between speakers of English and Iraqi Arabic. The speech recognition components are speaker-independent and noise-robust. The system has a vocabulary of tens of thousands of English and Iraqi Arabic words taken from the domains of force protection, municipal services, basic medical services, and personnel recruiting and training. Performance on other topics is related to the degree of similarity with the training domains.

SRI's Flexible Translation (FlexTrans) Framework is highly adaptable to the user's needs. It has an intuitive graphical interface, an eyes-free/hands-free mode, and many practical features. It was designed to facilitate the Spoken Language Communication and Translation System for Tactical Use (TRANSTAC) program's goal of rapidly developing and fielding robust, reliable systems in new languages.

The system has been running on ruggedized laptops in Iraq since early 2006. It currently runs on standard Windows computers and can be adapted to various specialized hardware platforms (see Section 5).

The purpose of this paper is to describe the architecture of a speech-to-speech translation framework that was designed to be flexible, configurable, and adaptable both to user needs and to the characteristics of different languages. We do give some details about the speech recognition and translation components of a particular version of the IraqComm™ system, and provide an example and some discussion of translation quality. But the main focus is on system architecture, user interface, system configurability, adaptability to different hardware and other languages, and lessons learned in the course of system development.

II. System Architecture

In this paper we will refer to the component that integrates and controls the Automatic Speech Recognition (ASR), Machine Translation (MT), and Text-to-Speech (TTS) components as the Flexible Translation (FlexTrans) Framework, because the name 'IraqComm' usually refers to FlexTrans together with the current ASR, MT, and TTS components, as well as, on occasion, the hardware platform.

At a high level, information flow through the system is simple: speech recognition, then translation, then speech synthesis in the other language. At a more detailed level the system is more complicated. Inter-component filters are applied at various stages, the timing of the various processes needs to be coordinated, certain properties of inputs and component outputs trigger messages to the user (e.g., 'speech too loud' or 'unable to recognize speech'), the exact flow varies with system settings such as 'autotranslate', and the user can abort ASR, MT, TTS, and other audio playback at any point. The flow chart in Figure 1 shows a simplified view of part of the FlexTrans system translating a single utterance.

The complete system is far more complex than the flow chart shows, and handles a variety of user interactions, some of which are described in more detail in Section 4. It is also robust enough to cope with any combinations of GUI elements being activated simultaneously or in quick succession, as can happen with a touchscreen interface.

A. Speech Recognition

The ASR components use Dynaspeak® [1], SRI's high-accuracy, embeddable, speaker-independent, real-time recognition engine. The IraqComm system uses a 16 kHz sampling rate, a 10 ms frame advance rate, and Mel frequency cepstral coefficients.

During the development of the components for each language, a decision-tree state-clustered triphone model was discriminatively trained using Minimum Phone Frame Error (MPFE) training and compiled into a state graph together with the pronunciation dictionary and a heavily pruned n-gram language model. For more information on the training and use of acoustic models, see [2]. For more in-depth information on our particular approach, see [3].

During recognition, a time-synchronous Viterbi search of the state graph generates a lattice, using Gaussian shortlists to speed up the computation. A second pass of rescoring based on the SRI Language Modeling Toolkit (SRILM) [4] uses a much larger higher-order language model with several million n-grams.

The acoustic models are trained with added noise of types that can be expected in the target environment. For signal-to-noise (SNR) ratios between 5 and 15 dB, this can reduce errors by about 25%. In these low SNR cases, additional Probabilistic Optimum Filtering (POF) compensation is applied, which can reduce errors even more significantly [5].

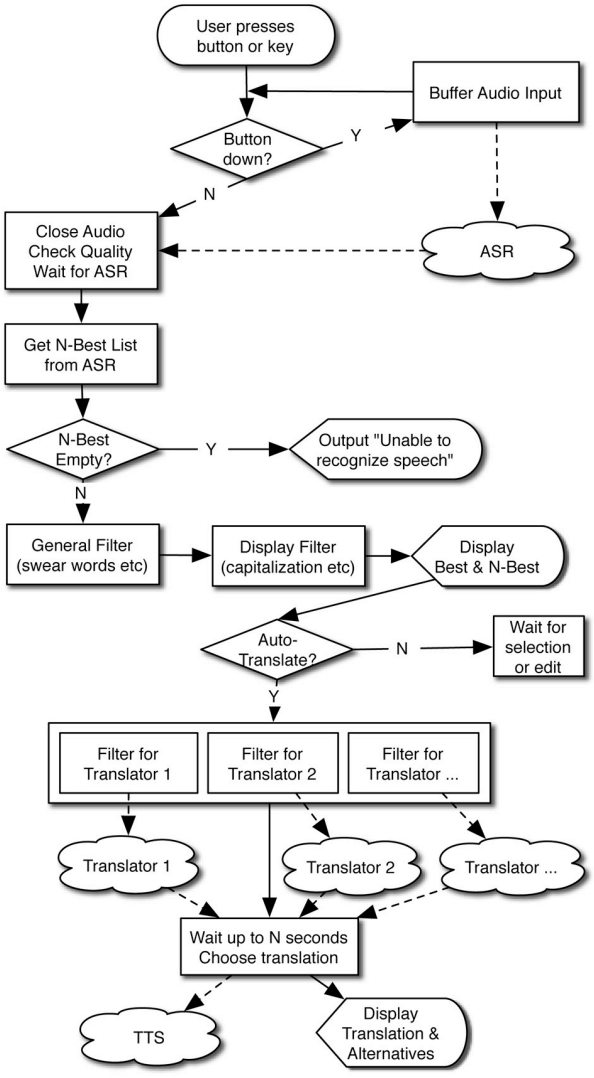

Fig. 1. Flow chart of part of the FlexTrans system

The system accommodates lists of swear words and phrases to remove before translating, in order to avoid the possibility of insulting someone because of a recognition error. This approach was chosen instead of removing these words from the ASR vocabulary, so as to avoid affecting recognition quality. In order to indicate to the user that this was done on purpose, asterisks are displayed on the screen in place of the swear words.

B. Translation

In the current IraqComm system, the translation from English to Iraqi Arabic is provided by Gemini if parsing and generation succeed within a specified time period. Otherwise, a statistical translation is used. It is often the case that the rule-based translation is more accurate and more easily understandable than the statistical translation. For the translation from Iraqi Arabic to English, SRI's SRInterp statistical MT engine is applied.

The Gemini [6] based translator uses two context-free unification grammars connected by an interlingua. It parses an English utterance and converts it to a (quasi-) logical form, from which it generates an Iraqi Arabic translation (see [7] for a similar approach to Pashto translation).

In Iraqi Arabic, yes-no questions have the same word order as statements, which, depending on the intonation of the TTS, could lead to misunderstandings. One interesting property of Gemini is that it knows when an English input is a yes-no question if inverted word order is used, and can therefore mark the translation as a question. Examples of this can be found in Figure 2, questions (1), (3), and (5). The way Gemini marks these utterances as questions – by inserting *do you* – is perfectly understandable but not frequently used, so not likely to be part of the training data for a statistical system.

The SRInterp component is statistical and phrase-based. It has a probabilistic bilingual phrase table and an n-gram language model for the target language. During translation, a dynamic programming algorithm uses these resources to find the translation that is the optimal compromise between adequacy and fluency. For background on this type of approach to machine translation see [8] and [9].

C. Speech Synthesis

The English and Iraqi Arabic TTS components, provided by Cepstral [10], are based on concatenative synthesis and use the Swift™ engine. Both voices are male and can be used at varying playback speeds. The voices can be customized for any desired lexical items via a user dictionary.

III. EXAMPLE

The quality of speech recognition and translation is good enough to exchange information and achieve successful communication when the topic is common conversational speech or matches the training data domain well. The sample dialog in Figure 2 shows that the English translations are comprehensible although some are not perfect. For example, in (4) the *that* clause *that a person was running* is not complete. In (8), the pronoun is wrong due to a speech recognition error (*I* instead of *he*). Similar mistakes are present in the other direction. The word order of the Iraqi Arabic translation in (9) is not standard, and in (11) there is a problem with gender agreement.

(1)	was anyone injured in the explosion هل أي واحد كان مجروح بالإنفجار
(2)	لا ما كان أكو أحد بالبناية no there was no one in the building
(3)	do you know who put the explosives there هل تعرف منو يحط المتفجرات هناك
(4)	لا بس أنا شفت فد شخص كان ديركض قبل الإنفجار no but I saw a person was running before the explosion
(5)	can you describe that person هل تقدر توصف ذاك الشخص
(6)	عمره كان تقريباً أربعين سنة هو طويل كان عنده لحية he was about forty years old and he's tall and he has a beard
(7)	which direction did he go وين راح
(8)	اتجاه المدرسة وبعدين صعدت سيارة تويوتا حمرا بإتجاه الجنوب in the direction of the school and then I caught a Toyota red towards the south
(9)	when was this شوقت هذا كان
(10)	تقريباً قبل عشر دقايق about ten minutes ago
(11)	do you remember anything else that might be helpful تتذكر أي شيء ثاني ممكن تكون مفيدة
(12)	لا ما أعتقد no I don't think so
(13)	thank you very much شكراً جزيلاً

Fig. 2. Sample Dialog

It is possible that these types of mistakes can lead to misunderstandings, but they do not usually cause problems because the meaning is clear in context. This is particularly true for users who have a strong incentive to try to communicate. They can take care to make sure the recognition result is correct before translating; make use of gestures and facial expressions in addition to speech; use the frequent phrases to communicate about problems (e.g., ask the Iraqi speaker to use shorter, simpler sentences when a particular long utterance turns out to have a confusing translation); look at the translation alternatives; use the 'do you understand' button or simple yes/no questions for confirmation at important points in the conversation; and know from intuition or experience what types of utterances are difficult, such as figurative speech or translating names.

It is difficult to give meaningful numbers about the accuracy of the system out of context, because accuracy varies significantly depending on the topic and type of conversation and the individual speakers. It is hard to know whether any given test is harder or easier than actual field use. Evaluating speech-to-speech translation systems is a research area in itself, with many possible approaches and tradeoffs, and no clear ideal solution. Some of the evaluation approaches currently in use in the DARPA TRANSTAC program are described in [11]. It should be noted that as development of the system continues, any evaluation is only a snapshot in time and rapidly obsolete as research progresses.

A detailed analysis of the performance of a particular set of components is not the focus of this paper, but to give at least some indication of system performance, we can mention that during the NIST evaluation in June 2008, new users – both English speaking subject matter experts and foreign language speakers – agreed that the system was usable and made it easy to have the interaction. Over 80% of the translations were considered adequate by human judges. Most of the inadequate translations were due to recognition errors.

For English, most of the recognition errors involved the unstressed reduced forms of short function words like articles, prepositions, pronouns, and auxiliary verbs, and thus could be avoided by making minor corrections to the ASR results before translating. In 40% of these cases, the correct recognition was one of the alternatives provided.

For Iraqi Arabic, most of the recognition errors involved prefixes and suffixes of otherwise correctly recognized words. Correcting these by typing is possible, but in some settings it may not be practical to have the Iraqi person do this. However, for about 20% of the utterances with Iraqi recognition errors the correct recognition was in the list of alternatives, so it is helpful to at least have the Iraqi speaker point to items in that list if possible.

The FlexTrans system provides an additional feature that can help improve the user's confidence in the adequacy of the translation. It is possible to turn on 'backtranslation' so that the English speaker can see a translation back into English of what the Iraqi speaker heard. If this backtranslation is comprehensible, the English speaker has reason to be quite confident that the original translation was correct and comprehensible to the Iraqi speaker.

The converse is not true: if the backtranslation is wrong, it is more likely that the translation into Arabic is fine and the problem is with the translation back into English. But the user cannot be sure which is the case. However, with some experience the user can learn what types of backtranslation mistakes are to be expected due to ambiguities in Iraqi Arabic (e.g., confusing *he* and *you*) or due to Gemini's marking of questions, which can result in backtranslations like *do you can describe that person* for (5).

If backtranslation becomes an important feature, it is possible to improve it by taking into account what types of problems Gemini output can pose as input for statistical MT, and for example stripping question markers before doing the backtranslation and coordinating the vocabularies carefully.

IV. Meeting User Needs

A. User Interface

The user interface was designed to be intuitive and easy for new users while providing advanced features for more experienced users. A screenshot of the IraqComm GUI can be seen in Figure 3.

Because the system needs to function in noisy environments, endpointing can be difficult, so instead it uses 'hold to talk', which is very familiar to military users of walkie talkies. The user can choose to hold down the button using the mouse, the touchscreen, or the 'E' and 'I' keys on the keyboard. The system buffers some audio in case the user starts speaking before hitting the button, and strips initial silence using start of speech (SOS) detection.

The main recognition result and translation are displayed prominently in large type in the two large text boxes; alternative recognition results and alternative translations are displayed on the right.

The user can select alternative recognition results from the n-best list, or edit the recognition result to correct minor mistakes using a regular keyboard or an on-screen keyboard.

Fig. 3. IraqComm Screenshot

V. Configurability

There are various menu items that allow users to adapt the behavior of the system to their needs. One example is "extra politeness". When consulting with a higher or equally ranked Iraqi official or elder, one may want to use more polite language than one might use in other situations.

In some situations it is desirable to be able to operate the system without needing to look at the screen or touch the computer. In eyes-free/hands-free mode, a two-button input device is used to indicate what language is recognized. There are cueing beeps for both languages, and the user can use commands like 'computer repeat'. A hierarchical dynamic grammar was developed for this purpose and merged with the regular English ASR grammar, and this command-and-control mode can be activated separately. It is also possible to play the ASR result for confirmation before translation.

The FlexTrans system can handle stereo input with one language per channel, so if a stereo audio card is available, it is possible to plug in two headsets and avoid having to pass a microphone back and forth. The audio does not get picked up from both headsets at the same time, so one speaker's breathing cannot interfere with the recognition for the other speaker, as would be the case without stereo. It is also possible to play the audio through the headsets and control the output channels for each language.

A. Keeping the User Informed

The FlexTrans system was designed to be responsive to the users and keep them informed about what is happening. It is possible to abort the current operation at almost any stage. There is a status message in the bottom left corner of the screen, a working bar indicating when the system is busy, and corresponding sounds in eyes-free mode. There are also visual and/or auditory notifications when a button click is received.

B. Other Practical Features

A menu item shows the conversation history, including all system inputs and outputs and associated information. The user can archive the files for future review in html format, which is is more human readable than the detailed system logs.

To speed up communication, a shortcut list of frequent phrases is provided that can be translated with one simple click, including short instructions in Iraqi Arabic explaining the basics of the system. It is easy for the user to add frequent phrases from a system translation, or by asking a trusted native speaker to record the translation.

It is also possible for the user to add custom words or phrases to the speech recognition components and the statistical translation components.

VI. System Generality

Much of the FlexTrans system is generic and can be adapted to work on different hardware or for different languages.

A. Different Hardware

The software is adaptable to a variety of hardware.

The FlexTrans system is easy to use on different screen sizes and resolutions. It is easy to include settings for new resolutions because it is using font multipliers instead of setting each font size manually. If necessary, some of the widgets can be hidden, such as the 'speech alternatives', 'translation alternatives', or 'frequent phrases', and can be made accessible from a button or menu and/or in a separate window instead. The lowest resolution we have tried so far is 640x480.

The software can be adapted to work on slower computers with less memory. It is easy to drop backup translation components. Many of the components can be adapted to be less resource-intensive, though there may be accuracy or speed

tradeoffs. But it is possible to reduce vocabulary size without noticeable effect on translation quality, by removing low-frequency items.

The smallest computer we have run the system on is the Sony VAIO VGN-UX280P. The dimensions of this device are approximately 6 x 4 x 1.5 inches, with a weight of 1.2 pounds.

The FlexTrans software is written in C++ and uses Trolltech's Qt libraries. These provide both the GUI framework and other common libraries, which should minimize issues with porting the application to other platforms. The UTF-8 encoding for Unicode was used to be more compatible with old libraries and applications.

A. New Languages

Where possible, the FlexTrans system was designed to facilitate supporting a new language. For example, it would be relatively easy to integrate our previous Pashto [7] work into the more generic FlexTrans architecture, and recently we have built a system to translate between English and Malay.

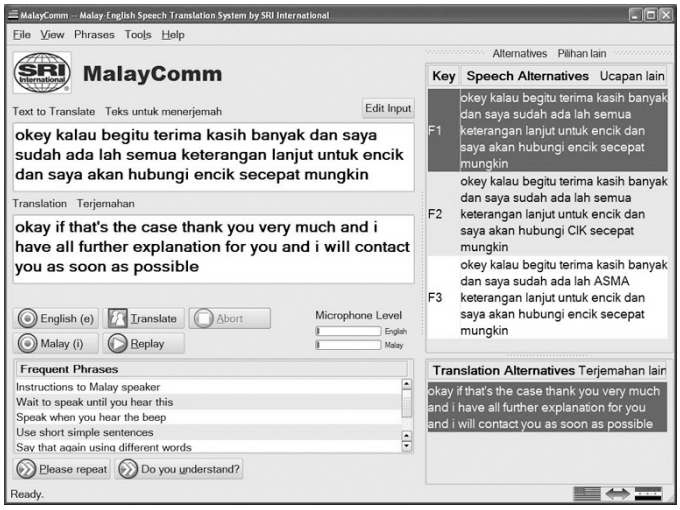

Fig. 4. Screenshot of English/Malay system

This was facilitated by several features of the FlexTrans framework.

Properties like language name, translation processes, GUI labels, etc., are set in a configuration file and can be changed easily.

Even though the current systems are intended to be primarily controlled by the English speaker, most of the FlexTrans system was designed to be symmetric, and many differences are achieved via parameters. For example, the fact that the foreign language speaker is cued with a beep is a configuration parameter rather than a hard-coded system behavior.

The FlexTrans system is built to allow for any number of translation components in each direction, and it is possible to send the input to different components based on characteristics of the input, or select which of several parallel outputs to use.

Other writing systems can be accommodated easily because Qt is Unicode-based and has bidirectional language support. Arabic right-to-left text works seamlessly.

The FlexTrans system has a notion of input and output filters so text can be appropriately prepared for a particular translator, post-processed after a particular ASR engine, preprocessed for a particular TTS engine, formatted for writing on the screen or to a log file, etc. For the IraqComm system this includes filters for converting the character set, special treatment of hyphenated words, possessive words, acronyms, and capitalization. In addition to the set of built-in filters, new ones can be created by specifying sets of regular expressions and calling other fiters, without having to recompile the software. This means that new components can be swapped in simply by installing them and specifying in the configuration file which filters to apply at various points. Some of the points where filters are applied can be seen in the flow chart in Figure 1.

VII. LESSONS LEARNED

This fairly complex system was developed in quite a short period of time – just half a year before the first test system was sent to Iraq – and the fact that it needs to be fieldable places certain demands on its usability and reliability. Given these circumstances, we found particular value in the following procedures.

It was very helpful to have easily retrievable records of project-internal interactions, including discussions of problems and their solutions. The fact that the QA engineer was involved in all stages of the process helped avoid time-consuming loops and delays, and allowed for earlier and more focused testing. Real-time collaboration between the software and QA engineers helped constrain the search space for tracking down problems and accelerated the process of identifying their sources by ruling out potential explanations quickly.

It became clear that this type of audio-based software is subject to potential hardware and timing issues that make it particularly important to have different people test on a variety of hardware and with various configurations. In addition, the users may interact with the software in idiosyncratic ways, again emphasizing the need for a broad base of testers.

It turned out to be useful to engineer the system such that multiple versions can coexist on the same computer. This not only affected FlexTrans but also required changes to the default behavior of some of the other components, e.g. changing the default installation location for the TTS voices.

Designing the system to be easily extensible took more initial effort, but paid off very quickly.

VIII. CONCLUSIONS

It seems clear that spoken language translation is now good enough to be useful in many situations when it is not possible to find a human translator, or to triage and help decide where the available human translators are most urgently needed. The remaining challenges in developing actual systems for a variety of languages on small portable devices all seem

solvable in principle. It will be interesting to try to integrate ASR, MT, and UI more closely in a way that enables the system to benefit from feedback and to learn. It also remains to be seen what kinds of communication problems may arise as more people unfamiliar with these technologies start to use them. But miscommunication occurs even without these devices, and because they encourage more direct communication and raise awareness of the potential of misunderstandings, they might inherently counteract most of the additional sources of potential problems.

IX. Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) and the Department of Interior-National Business Center (DOI-NBC) under Contract Number NBCHD040058. Approved for Public Release, Distribution Unlimited. The PI for the TRANSTAC project at SRI is Kristin Precoda. The FlexTrans framework was designed and developed primarily by Michael Frandsen, with some initial design and code contributions by Shane Mason and design input by Kristin Precoda and Susanne Riehemann. Huda Jameel is an Iraqi Arabic language expert, and we would particularly like to thank her for her help with the examples in this paper. The speech recognition components were developed primarily by Dimitra Vergyri, Sachin Kajarekar, Wen Wang, Murat Akbacak, Ramana Rao Gadde, Martin Graciarena, and Arindam Mandal. Gemini translation was provided by Andreas Kathol, and SRInterp translation by Jing Zheng. Other contributions were made by Horacio Franco, Donald Kintzing, Josh Kuhn, Xin Lei, Carl Madson, Sarah Nowlin, Colleen Richey, Talia Shaham, and Julie Wong. The system also contains TTS from Cepstral, LLC.

References

[1] H. Franco, J. Zheng, J. Butzberger, F. Cesari, M. Frandsen, J. Arnold, V.R.R. Gadde, A. Stolcke, and V. Abrash, "Dynaspeak: SRI's scalable speech recognizer for embedded and mobile systems," in *Human Language Technology*, 2002.

[2] B.H. Huang and L.R. Rabiner, "Hidden Markov Models for Speech Recognition", *Technometrics* (publ. by American Statistical Association), Vol. 33, No. 3, 1991, pp. 251–272.

[3] K. Precoda, J. Zheng, D. Vergyri, H. Franco, C. Richey, A. Kathol, and S. Kajarekar, "Iraqcomm: A next generation translation system," in *Interspeech*, 2007.

[4] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *International Conference on Spoken Language Processing*, 2002, pp. 901–904.

[5] M. Graciarena, H. Franco, G. Myers, and V. Abrash, "Robust feature compensation in nonstationary and multiple noise environments," in *Eurospeech*, 2005.

[6] J. Dowding, J.M. Gawron, D. Appelt, J. Bear, L. Cherny, R. Moore, and D. Moran, "Gemini: A natural language system for spoken-language understanding," in *Human Language Technology*, 1993, pp. 43–48.

[7] A. Kathol, K. Precoda, D. Vergyri, W. Wang, and S. Riehemann, "Speech translation for low-resource languages: The case of Pashto," in *Eurospeech*, 2005.

[8] P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer, "The mathematics of statistical machine translation: parameter estimation," in *Computational Linguistics*, 19(2), 1993, pp, 263–311.

[9] P. Koehn, F.J. Och, and D. Marcu, "Statistical phrase based translation," In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics* (HLT/NAACL), 2003.

[10] L. Tomokiyo, K. Peterson, A. Black, and K. Lenzo, "Intelligibility of machine translation output in speech synthesis," in *Interspeech*, 2006, pp. 2434–2437.

[11] B.A. Weiss, C. Schlenoff, G. Sanders, M.P. Steves, S. Condon, J. Phillips, and D. Parvaz, "Performance Evaluation of Speech Translation Systems," in *Proceedings of the Sixth International Language Resources and Evaluation*, 2008.