NAACL-HLT 2007

Demonstrations

Bob Carpenter, Amanda Stent, and Jason D. Williams
Demonstrations Co-Chairs

23-25 April 2007
Rochester, New York, USA

Production and Manufacturing by
Omnipress Inc.
2600 Anderson Street
Madison, WI 53704

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
75 Paterson Street, Suite 9
New Brunswick, NJ 08901
USA
Tel: +1-732-342-9100
Fax: +1-732-342-9339
acl@aclweb.org

Table of Contents

Demonstration of PLOW: A Dialogue System for One-Shot Task Learning

James Allen, Nathanael Chambers, George Ferguson*, Lucian Galescu, Hyuckchul Jung, Mary Swift* and William Taysom

Florida Institute for Human and Machine Cognition, Pensacola, FL 32502

*Computer Science Department, University of Rochester, Rochester, NY 14627

Introduction

We describe a system that can learn new procedure models effectively from one demonstration by the user. Previous work to learn tasks through observing a demonstration (e.g., Lent & Laird, 2001) has required observing many examples of the same task. One-shot learning of tasks presents a significant challenge because the observed sequence is inherently incomplete – the user only performs the steps required for the current situation. Furthermore, their decision-making processes, which reflect the control structures in the procedure, are not revealed.

We will demonstrate a system called PLOW (Procedural Learning on the Web) that learns task knowledge through observation accompanied by a natural language "play-by-play". Natural language (NL) alleviates many task learning problems by identifying (i) a useful level of abstraction of observed actions; (ii) parameter dependencies; (iii) hierarchical structure; (iv) semantic relationships between the task and the items involved in the actions; and (v) control constructs not otherwise observable. Various specialized reasoning modules in the system communicate and collaborate with each other to interpret the user's intentions, build a task model based on the interpretation, and check consistency between the learned task and prior knowledge.

The play-by-play approach in NL enables our task learning system to build a task with high-level constructs that are not inferable from observed actions alone. In addition to the knowledge about task structure, NL also provides critical information to transform the observed actions into more robust and reliable executable forms. Our system learns how to find objects used in the task, unifying the linguistic information of the objects with the semantic representations of the user's NL descriptions about them. The objects can then be reliably found in dynamic and complex environments. See Jung et al (2006) and Chambers et al (2006) for more details on the PLOW system.

The PLOW System

PLOW learns tasks executable on the web involving actions such as navigation, information extraction and form filling, and can learn iterative steps that operate over lists of objects on pages. Figure 1 shows the system during learning a task to find publications for a specified author. Upper left is the Mozilla browser, in which the user can demonstrate action and the system can execute actions in a mixed-initiative fashion. The user may speak or type to the system (SR output is lower right), and PLOW combines knowledge from the language and the demonstrated actions to produce a parameterized procedure (described in generated natural language in the upper right corner). Figure 2 shows a complete training dialogue in which PLOW learns how to find article titles. To save space, simple acknowledgments by the system are not shown.

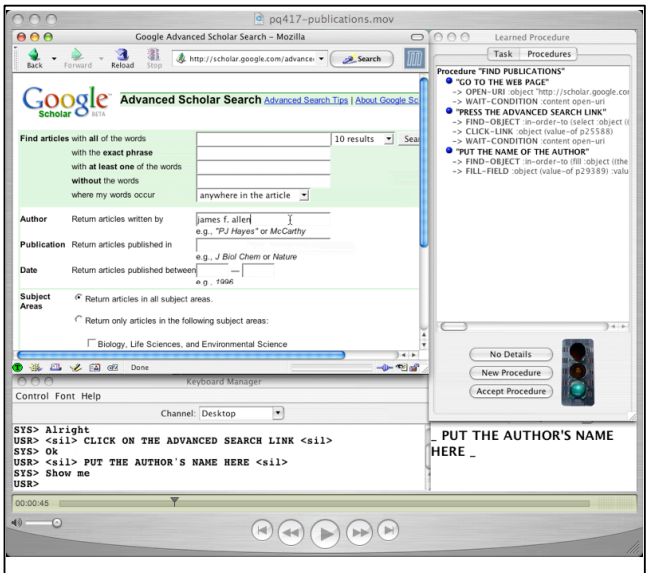


Figure 1: PLOW learning a task

U: Let me show you how to find someone's publications
U: Go to this webpage
`Navigates to http://scholar.google.com`
U: Click on the advanced search link
`Clicks Advanced Scholar Search`
U: Put the author's name here
`Types the name in the Author text field`
U: Click the search button
`Clicks on the Search Scholar button`
U: OK here is the list of results
`Selects the table on page`
PLOW: Can you check that I understand the data.
PLOW: puts up a new window with the results.
U: OK this is good.
U: this is the article title
`Highlights a title`
PLOW: attempts to find all titles and displays them
U: This is wrong.
PLOW OK
U: This is another article title
`Highlights another correct title`
PLOW: recomputes all the titles and displays them
`Switches back to Scholar window`
U: OK click next to get more results
`Clicks the Next link`
PLOW: When should I stop searching?
U: Get thirty results
PLOW: OK. I will do that
PLOW: iterates through pages until it gets 30 results
PLOW: I'm done.
U: We're done with the demonstration
U: Let's practice
PLOW: what is the name of the author?
Enters a new name
PLOW executes the procedure successfully!!

Figure 2: Learning to find titles of an author's publications (U = user)

Evaluation

The PLOW system was evaluated by independent evaluators who considered four task learning systems developed in the CALO project. There were 16 human subjects who received training on each of the systems and who worked through a number of successful scripted training sessions with each. They were then given ten new problems, ranging from slight variations to problems they had seen to problems that were substantially new. They were free to choose which problems to work on and which system to use and the resulting tasks learned were tested with different settings of the parameters and scored out of a total of 4 points based on a complex predefined evaluation criteria (not known to the developers). The PLOW system did well in the test, not only receiving the highest average score on tasks learned by a system (figure 3) but also was strongly preferred by the users and selected more than half the time (figure 4).

The Demonstration

If we are allowed a presentation we will demonstrate PLOW live on a task selected by the audience. In addition, we would like to have the system available for an extended period of time during the conference so that attendees can spend time using the system to teach it simple tasks. The system runs on a laptop and all that is needed for a demo is internet access.

Acknowledgements & References

This work was supported by DARPA grant NBCH-D-03-0010 under a subcontract from SRI International, ONR grant N000140510314, and NSF grant5-28096.

Chambers, N. et al. (2006). *Using Semantics to Identify Web Objects.* Proceedings AAAI.

Jung, H., J. Allen, et al. (2006). *One-Shot Proce-dure Learning from Instruction and Observation. FLAIRS,* Melbourne, FL.

Lent, M. and Laird, J. (2001) *Learning Procedural Knowledge through Observation*, Proc. of the Intl Conf. on Knowledge Capture.

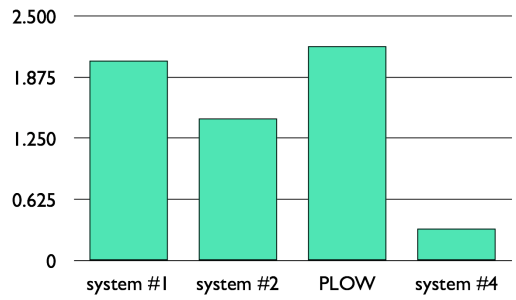


Figure 3: Average score (out of 4)

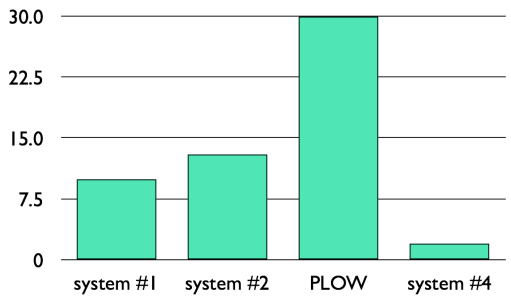


Figure 4: User preference for systems (55 trials)

The Automated Text Adaptation Tool

Jill Burstein, Jane Shore, John Sabatini, Yong-Won Lee & Matthew Ventura
Educational Testing Service
Rosedale Road MS 12R
Princeton, New Jersey 08541
{jburstein, jshore, jsabatini , ylee, mventura}@ets.org

1. Introduction

Text adaptation is a teacher practice used to help with reading comprehension and English language skills development for English language learners (ELLs) (Carlo, August, McLaughlin, Snow, Dressler, Lippman, Lively, & White, 2004; Echevarria, Vogt and Short, 2004; Yano, Long and Ross, 1994). The practice of text adaptation involves a teacher's modification of texts to make them more understandable, given a student's reading level. Teacher adaptations include text summaries, vocabulary support (e.g., providing synonyms), and translation. It is a time-consuming, but critical practice for K-12 teachers who teach ELLs, since reading-level appropriate texts are often hard to find. To this end, we have implemented the Automated Text Adaptation Tool v.1.0 (ATA v.1.0): an innovative, educational tool that automatically generates text adaptations similar to those teachers might create. We have also completed a teacher pilot study. Schwarm and Ostendorf (2005), and Heilman, Collins-Thompson, Callan, and Eskenazi (2006) describe related research addressing the development of NLP-based reading support tools.

During our interactive demonstration, conference participants can (a) login to the Internet-accessible tool, (b) import text files, and (c) experiment with adaptation features. We are currently interested in feedback from the computational linguistics community to inform tool development related to (a) feature enhancement, and (b) ideas for new NLP-based features. Until now, our primary source of feedback has been from teachers toward tool development from an educational perspective.

2. The Automated Text Adaptation Tool

NLP-based text adaptation capabilities in the tool are described in this section (also see Figure 1.) These adaptation features were selected for implementation since they resemble teacher-based adaptation methods.

2.1 English and Spanish Marginal Notes

Pedagogically, marginal notes are a kind of text summary. The *Rhext* automatic summarization tool (Marcu, 2000) is used to produce marginal notes in English. The amount of marginal notes generated can be increased or decreased based on students' needs. Using Language Weaver's[1] English-to-Spanish machine translation system, English marginal notes can be translated into Spanish.

2.2 Vocabulary Support

Synonyms for lower frequency (more difficult) words are output using a statistically-generated word similarity matrix (Lin, 1998). ATA v.1.0 generates *antonyms* for vocabulary in the text using WordNet®.[2] *Cognates* are words which have the same spelling and meaning in two languages (e.g., *animal* in English and Spanish). The tool generates these using an ETS English/Spanish cognate lexicon.

2.3 English and Spanish Text-to-Speech

The tool offers English and Spanish text-to-speech (TTS)[3]. English TTS may be useful for pronunciation support, while Spanish TTS provides access to the Spanish texts for Spanish-speaking ELLs who are not literate in Spanish.

[1] See http://www.languageweaver.com
[2] See http://wordnet.princeton.edu/
[3] See http://www.cstr.ed.ac.uk/projects/festival/ & http://cslu.cse.ogi.edu/tts/download/.

NAACL HLT Demonstration Program, pages 3–4,
Rochester, New York, USA, April 2007.

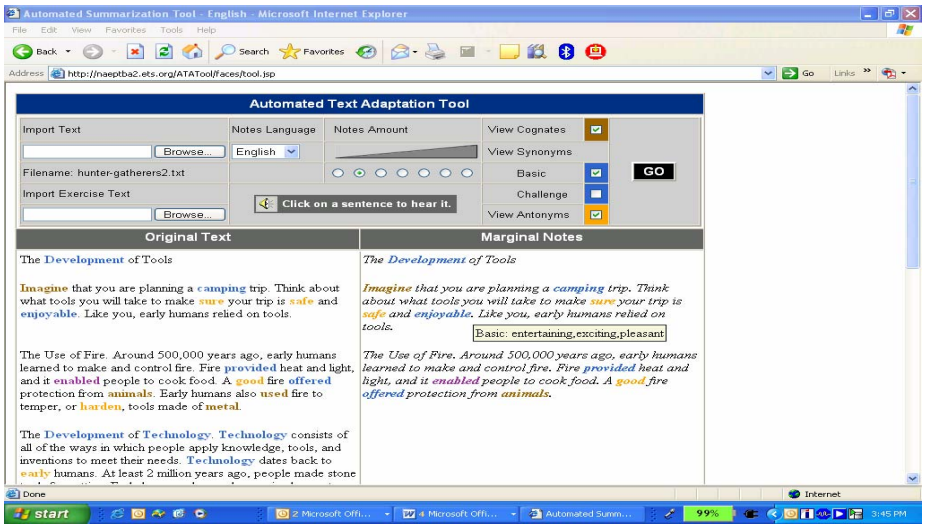


Figure 1. Example Main Interface Screen showing *English Marginal Notes* in the right column and *Synonyms* for "*enjoyable*" (*entertaining, enjoyable, pleasant*.)

3. Pilot Study with Teachers

The survey feedback indicated that the 12 teachers were positive about the tool's potential. Overall, the vocabulary and English marginal notes were the most favorite features, while the text-to-speech was the least favorite. Teachers commented that they would like to see an editing capability added that would allow them to make changes to the automatically generated outputs (i.e., vocabulary support, and English and Spanish marginal notes.) Teachers viewed the tool either as *lesson planning support*, or as a *student tool for independent work*.

4. Future Research

ATA v.1.0 is a young application that uses NLP methods to create text adaptations. The teacher pilot evaluation suggested that it produces adaptations with potentially effective support for ELLs. It could also save teachers lesson planning time. We are currently implementing teacher-suggested modifications, and planning a larger, school-based pilot. The pilot will evaluate the tool's effectiveness in terms of measurable learning gains in reading comprehension and English language skills.

References

Carlo, M.S., August, D., McLaughlin, B., Snow, C.E., Dressler, C., Lippman, D., Lively, T. & White, C. (2004). Closing the gap: Addressing the vocabulary needs of English language learners in bilingual and mainstream classrooms. Reading Research Quarterly, 39(2), 188-215.

Echevarria, J., Vogt, M., and Short, D. (2004). *Making Content Comprehensible for English Language Learners: the SIOP model.* New York: Pearson Education, Inc.

Heilman, M., Collins-Thompson, K., Callan, J., Eskenazi, M. (2006) Classroom Success of an Intelligent Tutoring System for Lexical Practice and Reading Comprehension. *In Proceedings of the Ninth International Conference on Spoken Language Processing*. Pittsburgh.

Lin, D. (1998). Automatic Retrieval and Clustering of Similar Words. *In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistic*s, Montreal, 898-904.

Marcu, D. (2000) *The Theory and Practice of Discourse Parsing and Summarization.* The MIT Press, Cambridge, Massachusetts.

Schwarm, S. and Ostendorf, M. *Reading Level Assessment Using Support Vector Machines and Statistical Language Models.* In Proceedings of the Association for Computational Linguistics, Ann Arbor, MI, 523-530.

Yano, Y., Long, M. & Ross, S. (1994). The effects of simplified and elaborated texts on foreign language reading comprehension. *Language Learning, 44*, 189-219.

Adaptive Tutorial Dialogue Systems Using Deep NLP Techniques

Myroslava O. Dzikovska, Charles B. Callaway, Elaine Farrow,
Manuel Marques-Pita, Colin Matheson and Johanna D. Moore
ICCS-HCRC, School of Informatics
University of Edinburgh
Edinburgh, EH8 9LW, United Kingdom
(mdzikovs,ccallawa,efarrow,mmpita,colin,jmoore)@inf.ed.ac.uk *

Abstract

We present tutorial dialogue systems in two different domains that demonstrate the use of dialogue management and deep natural language processing techniques. Generation techniques are used to produce natural sounding feedback adapted to student performance and the dialogue history, and context is used to interpret tentative answers phrased as questions.

1 Introduction

Intelligent tutoring systems help students improve learning compared to reading textbooks, though not quite as much as human tutors (Anderson et al., 1995). The specific properties of human-human dialogue that help students learn are still being studied, but the proposed features important for learning include allowing students to explain their actions (Chi et al., 1994), adapting tutorial feedback to the learner's level, and engagement/affect. Some tutorial dialogue systems use NLP techniques to analyze student responses to "why" questions. (Aleven et al., 2001; Jordan et al., 2006). However, for remediation they revert to scripted dialogue, relying on short-answer questions and canned feedback. The resulting dialogue may be redundant in ways detrimental to student understanding (Jordan et al., 2005) and allows for only limited adaptivity (Jordan, 2004).

We demonstrate two tutorial dialogue systems that use techniques from task-oriented dialogue systems to improve the interaction. The systems are built using the Information State Update approach (Larsson and Traum, 2000) for dialogue management and generic components for deep natural language understanding and generation. Tutorial feedback is generated adaptively based on the student model, and the interpretation is used to process explanations and to differentiate between student queries and hedged answers phrased as questions. The systems are intended for testing hypotheses about tutoring. By comparing student learning gains between versions of the same system using different tutoring strategies, as well as between the systems and human tutors, we can test hypotheses about the role of factors such as free natural language input, adaptivity and student affect.

2 The BEEDIFF Tutor

The BEEDIFF tutor helps students solve symbolic differentiation problems, a procedural task. Solution graphs generated by a domain reasoner are used to interpret student actions and to generate feedback.[1] Student input is relatively limited and consists mostly of mathematical formulas, but the system generates adaptive feedback based on the notion of student performance and on the dialogue history.

For example, if an average student asks for a hint on differentiating $sin(x^2)$, the first level of feedback may be "Think about which rule to apply", which

This work was supported under the 6th Framework Programme of the European Commission, Ref. IST-507826, and by a grant from The Office of Naval Research N000149910165.

[1] Solution graphs are generated automatically for arbitrary expressions, with no limit on the complexity of expressions except for possible efficiency considerations.

NAACL HLT Demonstration Program, pages 5–6,
Rochester, New York, USA, April 2007.

can then be specialized to "Use the chain rule" and then to giving away the complete answer. For students with low performance, more specific feedback can be given from the start. The same strategy (based on an initial corpus analysis) is used in producing feedback after incorrect answers, and we intend to use the system to evaluate its effectiveness.

The feedback is generated automatically from a single diagnosis and generation techniques are used to produce appropriate discourse cues. For example, when a student repeats the same mistake, the feedback may be "You've differentiated the inner layer correctly, but you're still missing the minus sign". The two clauses are joined by a contrast relationship, and the second indicates that an error was repeated by using the adverbial "still".

3 The BEETLE Tutor

The BEETLE tutor is designed to teach students basic electricity and electronics concepts. Unlike the BEEDIFF tutor, the BEETLE tutor is built around a pre-planned course where the students alternate reading with exercises involving answering "why" questions and interacting with a circuit simulator.

Since this is a conceptual domain, for most exercises there is no structured sequence of steps that the students should follow, but students need to name a correct set of objects and relationships in their response. We model the process of building an answer to an exercise as co-constructing a solution, where the student and tutor may contribute parts of the answer. For example, consider the question "For each circuit, which components are in a closed path". The solution can be built up gradually, with the student naming different components, and the system providing feedback until the list is complete. This generic process of gradually building up a solution is also applied to giving explanations. For example, in answer to the question "What is required for a light bulb to light" the student may say "The bulb must be in a closed path", which is correct but not complete. The system may then say "Correct, but is that everything?" to prompt the student towards mentioning the battery as well. The diagnosis of the student answer is represented as a set of correctly given objects or relationships, incorrect parts, and objects and relationships that have yet to be mentioned, and the system uses the same dialogue strategy of eliciting the missing parts for all types of questions.

Students often phrase their answers tentatively, for example "Is the bulb in a closed path?". In the context of a tutor question the interpretation process treats yes-no questions from the student as potentially hedged answers. The dialogue manager attempts to match the objects and relationships in the student input with those in the question. If a close match can be found, then the student utterance is interpreted as giving an answer rather than a true query. In contrast, if the student said "Is the bulb connected to the battery?", this would be interpreted as a proper query and the system would attempt to answer it.

Conclusion We demonstrate two tutorial dialogue systems in different domains built by adapting dialogue techniques from task-oriented dialogue systems. Improved interpretation and generation help support adaptivity and a wider range of inputs than possible in scripted dialogue.

References

V. Aleven, O. Popescu, and K. R. Koedinger. 2001. Towards tutorial dialog to support self-explanation: Adding natural language understanding to a cognitive tutor. In *Proc. AI-ED 2001*.

J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. 1995. Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4(2):167–207.

M. T. H. Chi, N. de Leeuw, M.-H. Chiu, and C. LaVancher. 1994. Eliciting self-explanations improves understanding. *Cognitive Science*, 18(3):439–477.

P. Jordan, P. Albacete, and K. VanLehn. 2005. Taking control of redundancy in scripted tutorial dialogue. In *Proc. of AIED2005*, pages 314–321.

P. Jordan, M. Makatchev, U. Pappuswamy, K. VanLehn, and P. Albacete. 2006. A natural language tutorial dialogue system for physics. In *Proc. of FLAIRS-06*.

P. W. Jordan. 2004. Using student explanations as models for adapting tutorial dialogue. In V. Barr and Z. Markov, editors, *FLAIRS Conference*. AAAI Press.

S. Larsson and D. Traum. 2000. Information state and dialogue management in the TRINDI Dialogue Move Engine Toolkit. *Natural Language Engineering*, 6(3-4):323–340.

POSSLT: A Korean to English Spoken Language Translation System

Donghyeon Lee, Jonghoon Lee, Gary Geunbae Lee

Department of Computer Science and Engineering
Pohang University of Science & Technology (POSTECH)
San 31, Hyoja-Dong, Pohang, 790-784, Republic of Korea
{semko, jh21983, gblee}@postech.ac.kr

Abstract

The POSSLT [1] is a Korean to English spoken language translation (SLT) system. Like most other SLT systems, automatic speech recognition (ASR), machine translation (MT), and text-to-speech (TTS) are coupled in a cascading manner in our POSSLT. However, several novel techniques are applied to improve overall translation quality and speed. Models used in POSSLT are trained on a travel domain conversational corpus.

1 Introduction

Spoken language translation (SLT) has become more important due to globalization. SLT systems consist of three major components: automatic speech recognition (ASR), statistical machine translation (SMT), text-to-speech (TTS). Currently, most of SLT systems are developed in a cascading method. Simple SLT systems translate a single best recognizer output, but, translation quality can be improved using the N-best hypotheses or lattice provided by the ASR (Zhang et. al., 2004; Saleem et. al., 2004).

In POSSLT, we used an N-best hypothesis re-ranking based on both ASR and SMT features, and divided the language model of the ASR according to the specific domain situation. To improve the Korean-English SMT quality, several new techniques can be applied (Lee et. al., 2006-b). The POSSLT applies most of these techniques using a preprocessor.

[1] POSSLT stands for POSTECH Spoken Language Translation system

2 System Description

The POSSLT was developed by integrating ASR, SMT, and TTS. The system has a pipelined architecture as shown in Fig. 1. LM loader, preprocessor and re-ranking module are newly developed to improve the translation quality and speed for POSSLT.

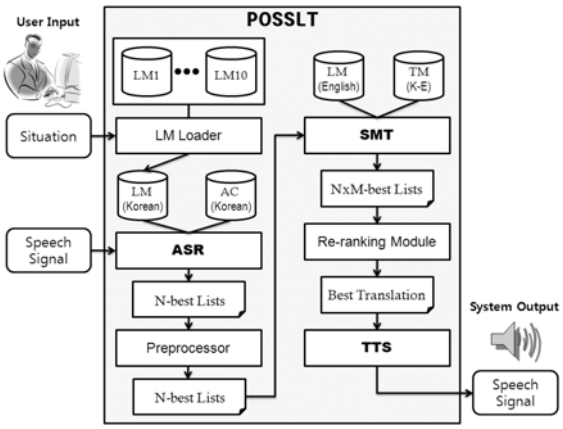


Figure 1: Overview of POSSLT

2.1 ASR

The system used HTK-based continuous speech recognition engine properly trained for Korean. The acoustic model, lexical model and language model of Korean are trained for conversational corpus. The phonetic set for Korean has 48 phoneme-like-units, and we used three-state tri-phone hidden Markov models and trigram language mod-

NAACL HLT Demonstration Program, pages 7–8,
Rochester, New York, USA, April 2007.

els. Pronunciation lexicons are automatically built by a Korean grapheme-to-phoneme (G2P) tool (Lee et. al., 2006-a). We used an *eojeol*[2] as a basic recognition unit for lexical and language models, because an *eojeol*-based recognition unit has the higher accuracy than the morpheme-based one. The ASR produces the N-best hypotheses determined through the decoding process, which are used as the input of SMT.

2.2 SMT

We implemented a Korean-English phrase-based SMT decoder based on Pharaoh (Koehn, 2004). The decoder needs a phrase translation model for the Korean-English pair and a language model for English. We used the Pharaoh training module and GIZA++ (Och and Ney, 2000) to construct the phrase translation table. For language modeling, SRILM toolkit (Stolcke, 2002) was used to build a trigram language model.

2.3 TTS

We used Microsoft SAPI 5.1 TTS engine for English TTS. The final best translation is pronounced using the engine.

2.4 LM Loader

In cascading SLT systems, SMT coverage depends on the used ASR. In order to increase the ASR coverage, our system loads and unloads the ASR language models dynamically. In our system which uses a travel corpus, language models are built for ten domain situation categories such as an airport, a hotel, a shopping, etc. Besides user utterances, user selection of the situation is needed as an input to decide which language model have to be loaded in advance. By using the divided language models, many benefits such as fast decoding, higher accuracy and more coverage can be obtained.

2.5 Preprocessor

In the Korean-English SMT task, there have been developed several techniques for improving the translation quality such as changing spacing units into morphemes, adding POS tag information, and deleting useless words (Lee et. al., 2006-b).

However, for these techniques, Part-Of-Speech (POS) tagger is needed. If the final analyzed form of an *eojeol* (in the form of a sequence of morphemes plus POS tags) is defined as a word in the ASR lexicon, the transformed sentences are directly generated by the ASR only, so POS tagger errors can be removed from the system. Preprocessor also removes useless words in SMT in the transformed sentences produced by the ASR.

2.6 Re-ranking Module

We implemented a re-ranking module to make a robust SLT system against the speech recognition errors. The re-ranking module uses several features: ASR acoustic model scores, ASR language model scores, and SMT translation scores. Finally, the re-ranking module sorts the N-best lists by comparing the total scores.

Acknowledgements

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment; IITA-2005-C1090-0501-0018)

References

A. Stolcke. 2002. *SRILM – An Extensible Language Modeling Toolkit*. Proc. of ICSLP.

F. J. Och and H. Ney. 2000. *Improved statistical alignment models*. Proc. of 38th Annual Meeting of the ACL, page 440-447, Hongkong, China, October 2000.

Jinsik Lee, Seungwon Kim, Gary Geunbae Lee. 2006-a. *Grapheme-to-Phoneme Conversion Using Automatically Extracted Associative Rules for Korean TTS System*. Proc. of Interspeech-ICSLP.

Jonghoon Lee, Donghyeon Lee, Gary Geunbae Lee. 2006-b. *Improving Phrase-based Korean-English Statistical Machine Translation*. Proc. of Interspeech-ICSLP.

P. Koehn. 2004. *Pharaoh: A Beam Search Decoder for Phrase-based Statistical Machine Translation Models*. Proc. of AMTA, Washington DC.

R. Zhang, G. Kikui, H. Yamamoto, T. Watanabe, F. Soong, and W. K. Lo. 2004. *A unified approach in speech-to-speech translation: Integrating features of speech recognition and machine translation*. Proc. of Coling 2004, Geveva.

S. Saleem, S. Chen Jou, S. Vogel, and T.Schultz. 2004. *Using word lattice information for a tighter coupling in speech translation systems*. Proc. of ICSLP 2004, Jeju, Korea.

[2] *Eojeol* is a spacing unit in Korean and typically consists of more than one morpheme.

Automatic Segmentation and Summarization of Meeting Speech

Gabriel Murray, Pei-Yun Hsueh, Simon Tucker
Jonathan Kilgour, Jean Carletta, Johanna Moore, Steve Renals
University of Edinburgh
Edinburgh, Scotland
{gabriel.murray,p.hsueh}@ed.ac.uk

1 Introduction

AMI Meeting Facilitator is a system that performs topic segmentation and extractive summarisation. It consists of three components: (1) a segmenter that divides a meeting into a number of locally coherent segments, (2) a summarizer that selects the most important utterances from the meeting transcripts. and (3) a compression component that removes the less important words from each utterance based on the degree of compression the user specified. The goal of the AMI Meeting Facilitator is two-fold: first, we want to provide sufficient visual aids for users to interpret what is going on in a recorded meeting; second, we want to support the development of downstream information retrieval and information extraction modules with the information about the topics and summaries in meeting segments.

2 Component Description

2.1 Segmentation

The AMI Meeting Segmenter is trained using a set of 50 meetings that are seperate from the input meeting. We first extract features from the audio and video recording of the input meeting in order to train the Maximum Entropy (MaxEnt) models for classifying topic boundaries and non-topic boundaries. Then we test each utterance in the input meeting on the Segmenter to see if it is a topic boundary or not. The features we use include the following five categories: (1) **Conversational Feature**: These include a set of seven conversational features, including the amount of overlapping speech, the amount of silence between speaker segments, the level of similarity of speaker activity, the number of cue words, and the predictions of LCSEG (i.e., the lexical cohesion statistics, the estimated posterior probability, the predicted class). (2) **Lexical Feature**: Each spurt is represented as a vector space of uni-grams, wherein a vector is 1 or 0 depending on whether the cue word appears in the spurt. (3) **Prosodic Feature**: These include dialogue-act (DA) rate-of-speech, maximum F0 of the DA, mean energy of the DA, amount of silence in the DA, precedent and subsequent pauses, and duration of the DA. (4) **Motion Feature**: These include the average magnitude of speaker movements, which is measured by the number of pixels changed, over the frames of 40 ms within the spurt. (5) **Contextual Feature**: These include the dialogue act types and the speaker role (e.g., project manager, marketing expert). In the dialogue act annotations, each dialogue act is classified as one of the 15 types.

2.2 Summarization

The AMI summarizer is trained using a set of 98 scenario meetings. We train a support vector machine (SVM) on these meetings, using 26 features relating to the following categories: (1) **Prosodic Features**: These include dialogue-act (DA) rate-of-speech, maximum F0 of the DA, mean energy of the DA, amount of silence in the DA, precedent and subsequent pauses,

NAACL HLT Demonstration Program, pages 9–10,
Rochester, New York, USA, April 2007.

and duration of the DA. (2) **Speaker Features**: These features relate to how dominant the speaker is in the meeting as a whole, and they include percentage of the total dialogue acts which each speaker utters, percentage of total words which speaker utters, and amount of time in meeting that each person is speaking. (3) **Structural Features**: These features include the DA position in the meeting, and the DA position in the speaker's turn. (4) **Term Weighting Features**: We use two types of term weighting: *tf.idf*, which is based on words that are frequent in the meeting but rare across a set of other meetings or documents, and a second weighting feature which relates to how word usage varies between the four meeting participants.

After training the SVM, we test on each meeting of the 20 meeting test set in turn, ranking the dialogue acts from most probable to least probable in terms of being extract-worthy. Such a ranking allows the user to create a summary of whatever length she desires.

2.3 Compression

Each dialogue act has its constituent words scored using *tf.idf*, and as the user compresses the meeting to a greater degree the browser gradually removes the less important words from each dialogue act, leaving only the most informative material of the meeting.

3 Related Work

Previous work has explored the effect of lexical cohesion and conversational features on characterizing topic boundaries, following Galley et al.(2003). In previous work, we have also studied the problem of predicting topic boundaries at different levels of granularity and showed that a supervised classification approach performs better on predicting a coarser level of topic segmentation (Hsueh et al., 2006).

The amount of work being done on speech summarization has accelerated in recent years. Maskey and Hirschberg(September 2005) have explored speech summarization in the domain of Broadcast News data, finding that combining prosodic, lexical and structural features yield the best results. On the ICSI meeting corpus, Murray et al.(September 2005) compared applying text summarization approaches to feature-based approaches including prosodic features, while Galley(2006) used skip-chain Conditional Random Fields to model pragmatic dependencies between meeting utterances, and ranked meeting dialogue acts using a combination or prosodic, lexical, discourse and structural features.

4 acknowledgement

This work was supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi- party Interaction, FP6-506811)

References

M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing. 2003. Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics.*

M. Galley. 2006. A skip-chain conditional random field for ranking meeting utterances by importance. In *Proceedings of EMNLP-06, Sydney, Australia.*

P. Hsueh, J. Moore, and S. Renals. 2006. Automatic segmentation of multiparty dialogue. In *the Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics.*

S. Maskey and J. Hirschberg. September 2005. Comparing lexial, acoustic/prosodic, discourse and structural features for speech summarization. In *Proceedings of the 9th European Conference on Speech Communication and Technology, Lisbon, Portugal.*

G. Murray, S. Renals, and J. Carletta. September 2005. Extractive summarization of meeting recordings. In *Proceedings of the 9th European Conference on Speech Communication and Technology, Lisbon, Portugal.*

Cedit – Semantic Networks Manual Annotation Tool

Václav Novák
Institute of Formal and Applied Linguistics
Charles University
Malostranské nám. 25, 11800 Praha, Czech Republic
novak@ufal.mff.cuni.cz

Abstract

We present a demonstration of an annotation tool designed to annotate texts into a semantic network formalism called MultiNet. The tool is based on a Java Swing GUI and allows the annotators to edit nodes and relations in the network, as well as links between the nodes in the network and the nodes from the previous layer of annotation. The data processed by the tool in this presentation are from the English version of the Wall Street Journal.

1 Introduction

Cedit is a part of a project to create a rich resource of manually annotated semantic structures (Novák, 2007) as a new layer of the Prague Dependency Treebank (Sgall et al., 2004). The new layer is based on the MultiNet paradigm described in (Helbig, 2006).

1.1 Prague Dependency Treebank

The Prague Dependency Treebank is a language resource containing a deep manual analysis of text (Sgall et al., 2004). PDT contains three layers of annotation, namely *morphological*, *analytical* (shallow dependency syntax) and *tectogrammatical* (deep dependency syntax). The units of each annotation level are linked to corresponding units from the shallower level. The morphological units are linked directly to the original text.

The theoretical basis of the treebank is described by the Functional Generative Description of language system (Sgall et al., 1986).

1.2 MultiNet

Multilayered Extended Semantic Networks (MultiNet), described in (Helbig, 2006), provide a universally applicable formalism for treatment of semantic phenomena of natural language. They offer distinct advantages over classical predicate calculus and its derivatives. Moreover, semantic networks are convenient for manual annotation because they are more intuitive.

MultiNet's semantic representation of natural language is independent of the language being annotated. However, syntax obviously varies across languages. To bridge the gap between different languages we can the deep syntactico-semantic representation available in the Functional Generative Description framework.

2 Project Goals

The main goals of the project are:

- Test the completeness and intuitiveness of MultiNet specification
- Measure differences in semantic networks of parallel texts
- Enrich the Prague Dependency Treebank with a new layer of annotation
- Provide data for supervised training of text-to-semantic-network transformation

NAACL HLT Demonstration Program, pages 11–12,
Rochester, New York, USA, April 2007.

- Test the extensibility of MultiNet to other languages than German

3 Cedit

The presented tool has two key components described in this section.

3.1 Input/Output processing

The input module of the tool loads XML files in Prague Markup Language (PML) and creates an internal representation of the semantic network, tectogrammatical layer, analytical layer, and the surface text (Pajas and Štěpánek, 2005). There is also an option to use files with named entity annotations. The sentences in this demo are all annotated with named entities.

The XML schema for the semantic network is an application of the Prague Markup Language.

3.2 Network GUI

The annotation GUI is implemented using Java Swing (Elliott et al., 2002). The key features of the tool presented in the demonstration are:

- Editing links between the semantic network and the tectogrammatical layer
- Adding and removing nodes
- Connecting nodes with directed edges
- Connecting edges with directed edges (i.e., creating relations on the metalevel)
- Editing attributes of both nodes and edges
- Undoing and redoing operations
- Reusing concepts from previous sentences

4 Related Work

There are various tools for annotation of the Prague Dependency Treebank. The Tred tool (Hajič et al., 2001), for example, allows users to edit many PML applications, even those that have never been seen before. This functionality is enabled by *roles* in PML specification (Pajas and Štěpánek, 2005). MultiNet structures can be edited using MWR tool (Gnrlich, 2000), but this tool is not primarily intended for annotation; it serves more as an interface to tools automatically transforming German sentences into MultiNet.

Acknowledgement

This work is supported by Czech Academy of Science grant 1ET201120505 and Czech Ministry of Education, Youth and Sports project LC536. The views expressed are not necessarily endorsed by the sponsors.

References

James Elliott, Robert Eckstein, Marc Loy, David Wood, and Brian Cole. 2002. *Java Swing*. O'Reilly.

Carsten Gnrlich. 2000. MultiNet/WR: A Knowledge Engineering Toolkit for Natural Language Information. Technical Report 278, University Hagen, Hagen, Germany.

Jan Hajič, Barbora Vidová-Hladká, and Petr Pajas. 2001. The Prague Dependency Treebank: Annotation Structure and Support. In *Proceedings of the IRCS Workshop on Linguistic Databases*, pages 105—114, Philadelphia, USA. University of Pennsylvania.

Hermann Helbig. 2006. *Knowledge Representation and the Semantics of Natural Language*. Springer-Verlag, Berlin Heidelberg.

Václav Novák. 2007. Large Semantic Network Manual Annotation. In *Proceedings of 7th International Workshop on Computational Semantics*, pages 355–358, Tilburg, Netherlands.

Petr Pajas and Jan Štěpánek. 2005. A Generic XML-Based Format for Structured Linguistic Annotation and Its Application to Prague Dependency Treebank 2.0. Technical Report 29, UFAL MFF UK, Praha.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. D. Reidel Publishing company, Dodrecht, Boston, London.

Petr Sgall, Jarmila Panevová, and Eva Hajičová. 2004. Deep Syntactic Annotation: Tectogrammatical Representation and Beyond. In A. Meyers, editor, *Proceedings of the HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 32–38, Boston, Massachusetts, USA. Association for Computational Linguistics.

Spoken Dialogue Systems for Language Learning*

Stephanie Seneff, Chao Wang, and Chih-yu Chao
Spoken Language Systems Group
MIT Computer Science and Artificial Intelligence Laboratory
The Stata Center, 32 Vassar Street, Cambridge, MA 02139, USA
{seneff,wangc,chihyu}@csail.mit.edu

Abstract

This demonstration will illustrate interactive computer games intended to help a native speaker of English learn Mandarin. These systems provide users with human-like conversational exercises with contextualized help mechanisms. Two distinctly different activities, a *translation game* and a *dialogue game* are illustrated. The level of difficulty can be manipulated, and the sentence variations covered by the systems familiarize users with different expressions of the same meaning. The systems preserve the qualities of a typical computer system, being infinitely patient and available any time of day. Students will be able to repeatedly practice conversation with no embarrassment.

1 Introduction

Mandarin Chinese is one of the most difficult languages for a native English speaker to learn. Chinese is substantially more difficult to master than the traditional European languages currently being taught in America – French, Spanish, German, etc., because of the lack of common roots in the vocabulary, the novel tonal and writing systems, and the distinctly different syntactic structure.

It is widely agreed among educators that the best way to learn to speak a foreign language is to engage in natural conversation with a native speaker of the language. Yet this is also one of the most costly ways to teach a language, due to the inherently one-to-one student-teacher ratio that it implies.

*This research is supported in part by the Industrial Technology Research Institute and the Cambridge MIT Initiative.

Recent research in the Spoken Language Systems group at MIT has focused on the idea of designing entertaining computer games as a device for teaching a foreign language, with initial emphasis on the language pair, English and Mandarin. The games are accessible at a Web page, and the student's speech is captured from a headset microphone to support natural spoken dialogue interaction. The system can also be installed to run completely stand-alone on the local laptop computer.

2 Demonstrated Systems

The demonstrated systems comprise two related activities, the *translation game* and the *dialogue game*. The translation game serves as preparation for the dialogue game: the user acquires expertise in speaking within the domain in the target language. The system randomly presents sentences in English and asks the student to speak a sentence of equivalent meaning in Mandarin. To imitate the competitive spirit of video games, the system offers ten *difficulty levels*, which are automatically adjusted depending on the student's monitored performance. After advancing to the highest difficulty level, they will subsequently be much better equipped to converse with the system within the dialogue game.

The *dialogue game* involves spoken conversational interaction to solve a particular scenario. The student and computer are tasked with jointly solving a specified goal. Differing difficulty levels are achieved via the device of a robotic tutor who assists the student in solving their side of the conversation.

2.1 Translation Game

The translation game is motivated by the learning approach advocated by Pimsleur (1967). By practicing translation repeatedly, language learners are

NAACL HLT Demonstration Program, pages 13–14,
Rochester, New York, USA, April 2007.

able to internalize the structures of the target language, and thus the vocabulary, grammar rules, and pronunciation are practiced concurrently. The user begins by translating isolated vocabulary items in Level 1, advancing to phrases and full sentences at higher levels. The most difficult level, Level 10, involves long and complicated sentences.

We have implemented this game in two domains: (1) flight reservations, and (2) hobbies and schedules. Details of the translation procedure can be found in (Wang and Seneff, 2006), and the algorithm for assessment is described in detail in (Wang and Seneff, 2006). The input utterance is processed through the speech recognizer and language understanding (Seneff, 1992) components, to achieve a simple encoding of its meaning. The system compares this meaning representation to one automatically derived from the targeted English equivalent. The system then speaks a paraphrase of the user's hypothesized utterance in both Chinese and English (Baptist and Seneff, 2000). If it has determined that the student was successful, it congratulates them and prompts them with the next English sentence for translation. At any time, the student can ask for assistance, in which case the system will provide them with a "correct" translation of the English utterance, which they can then attempt to imitate.

2.2 Dialogue Game

In the dialogue game (Seneff, 2006), the user is asked to solve a particular scenario, by role playing a specified persona, which changes dynamically every time the game is played. We will demonstrate the dialogue game in the hobbies and schedules domain. The student is provided with a specification of their preferences for participating in possible activities (swimming, dancing, watching movies, etc.) as well as a calendar specifying activities they are planning to do in the next few days. They are tasked with arranging with the computer to jointly participate in an activity that they both like, at a time when both are free. Another option is for either party to invite the other one to join them in an activity that is already on their schedule.

In addition to the robotic dialogue partner, the student is assisted in solving the task by a robotic *tutor*, who helps them plan what to say next. The tutor works with the same information that the student has, and independently plans the student's half of the conversation. At each dialogue turn, it provides a proposed response, based on the evolving dialogue context. Five different difficulty levels have been implemented, as follows:

1. *Eavesdropping:* The student can simply let the tutor carry out their side of the conversation by clicking a button to advance each dialogue turn.
2. *Parroting:* The system presents a proposed sentence in pinyin on the screen, and the student can just read it out loud well enough to be successfully understood.
3. *Translation:* The system presents an English sentence which the student needs to translate into Chinese.
4. *Characters:* The system presents the Chinese sentence in a character encoding.
5. *Solo:* The tutor stops being pro-active, but can be consulted if necessary.

Both the translation game and the dialogue game will be illustrated live in the demonstration. The systems can be evaluated by two types of basic performance measures: (1) for each system, the recognition accuracy and the translation accuracy serve as an index of quality; (2) calculating the success rate in the translation game and the number of turns taken to complete each dialogue will provide a quantitative view of interaction. Also a pre- & post-test design in the user study will further confirm the pedagogic value of the systems. Ongoing and future work involves expanding the domains supported and introducing the games to the classroom setting.

References

Baptist, L. and S. Seneff. 2000. "Genesis-II: A Versatile System for Language Generation in Conversational System Applications," *Proc. ICSLP*, III:271–274.

Pimsleur, P. 1967. "A Memory Schedule," *Modern Language Journal*, 51:73–75.

Seneff, S. 1992. "TINA: A Natural Language System for Spoken Language Applications," *Computational Linguistics*, 18(1):61–86.

Seneff, S. 2006. "Interactive Computer Aids for Acquiring Proficiency in Mandarin," Keynote Speech, *Proc. ISCSLP*, pp. 1–11.

Wang, C and S. Seneff. 2006. "High-quality Speech Translation in the Flight Domain," *Proc. INTERSPEECH*.

Wang, C. and S. Seneff 2007. "Automatic Assessment of Student Translations for Foreign Language Tutoring," *Proc. NAACL-HLT*.

RavenCalendar: A Multimodal Dialog System for Managing a Personal Calendar*

Svetlana Stenchikova and **Basia Mucha** and **Sarah Hoffman** and **Amanda Stent**
Department of Computer Science
Stony Brook University
Stony Brook, NY 11794-4400
sveta,basia,shoffman,stent@cs.sunysb.edu

1 Introduction

Dialog applications for managing calendars have been developed for every generation of dialog systems research (Heidorn, 1978; Yankelovich, 1994; Constantinides and others, 1998; Horvitz and Paek, 2000; Vo and Wood, 1996; Huang and others, 2001). Today, Web-based calendar applications are widely used. A spoken dialog interface to a Web-based calendar application permits convenient use of the system on a handheld device or over the telephone.

In this demo, we present *RavenCalendar*, a multimodal dialog system built around the Google Calendar and Google Maps Web applications. *RavenCalendar* allows the user to create, modify and remove calendar events, query for events, and hear descriptions of events. In our demonstration we will focus on two aspects of *RavenCalendar*: its flexible approach to language understanding and dialog management, and its multimodal interface.

Flexible dialog management In *RavenCalendar*, during event management or event querying a user can provide event information either in a single utterance (e.g. *"Add a meeting for tomorrow from two to three in the computer science lounge"* or by engaging in a subdialog with the system (Table 1). This functionality is made possible because we use the Ravenclaw dialog manager (Bohus and Rudnicky, 2003).

*We would like to thank Dan Bohus, Antoine Raux, and Thomas Harris for providing Olympus, and for answering many questions about Ravenclaw. This research is based on work supported by the National Science Foundation under grant no. 0325188.

S: Welcome to RavenCalendar, the automated dialog calendar system.
S: What can I do for you today?
U: add a meeting for Monday
S: For what time do you want to add the meeting?
U: from ten until eleven
S: The event was added type meeting for tomorrow starting at ten a.m.
U: List all events for the next week

Table 1: Sample dialog with *RavenCalendar*

Multimodality A *RavenCalendar* user may interact with the calendar directly using the Google Calendar interface, or may interact through *RavenCalendar* using text, speech, map gestures or a combination of these media. A user may use the Google Maps interface to specify the location of an event; the system uses Google Maps to display the locations of events.

2 System Description

RavenCalendar, whose architecture is shown in Figure 1, is developed using Ravenclaw and Olympus (Bohus and others, 2007). Olympus is a dialog system shell; Ravenclaw is the Olympus dialog manager. In developing *RavenCalendar*, we chose to use an existing dialog shell to save time on system development. (We are gradually replacing the Olympus components for speech recognition, generation and TTS.) *RavenCalendar* is one of the first dialog systems based on Olympus to be developed outside of CMU. Other Olympus-based systems developed at CMU include the Let's Go (Raux and others, 2005), Room Line, and LARRI (Bohus and Rudnicky, 2002) systems.

Flexible dialog management The Ravenclaw dialog manager (Bohus and Rudnicky, 2003) allows "object-oriented" specification of a

NAACL HLT Demonstration Program, pages 15–16,
Rochester, New York, USA, April 2007.

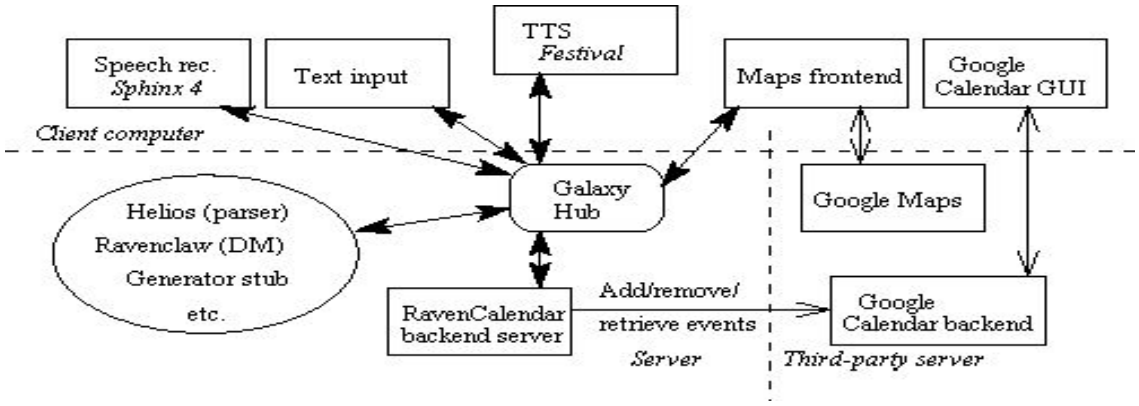


Figure 1: RavenCalendar Design

dialog structure. In *RavenCalendar*, we define the dialog as a graph. Each node in the graph is a minimal dialog component that performs a specific action and has pre- and post-conditions. The dialog flow is determined by edges between nodes. With this structure, we maximize the reuse of minimal dialog components. Ravenclaw gives a natural way to define a dialog, but fine-tuning the dialog manager was the most challenging part of system development.

Multimodality In *RavenCalendar*, a back-end server integrates with Google Calendar for storing event data. Also, a *maps front end* server integrates with Google Maps. In addition to the locations recognized by Google Maps, an XML file with pre-selected location-name mappings helps the user specify locations.

3 Current and Future Work

We are currently modifying *RavenCalendar* to use grammar-based speech recognition for tighter integration of speech recognition and parsing, to automatically modify its parsing grammar to accommodate the words in the user's calendar, to permit trainable, adaptable response generation, and to connect to additional Web services and Web-based data resources. This last topic is particularly interesting to us. *RavenCalendar* already uses several Web-based applications, but there are many other Web services of potential utility to mobile users. We are now building a component for RavenClaw that searches a list of URLs for event types of interest to the user (e.g. sports events, music events), and automatically notifies the user of events of interest. In the future, we plan to incorporate additional Web-based functionality, with the ultimate goal of creating a general-purpose dialog interface to Web applications and services.

References

D. Bohus et al. 2007. Olympus: an open-source framework for conversational spoken language interface research. In *Proceedings of the Workshop "Bridging the Gap" at HLT/NAACL 2007*.

D. Bohus and A. Rudnicky. 2002. LARRI: A language-based maintenance and repair assistant. In *Proceedings of IDS*.

D. Bohus and A. Rudnicky. 2003. Ravenclaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *Proceedings of Eurospeech*.

P. Constantinides et al. 1998. A schema based approach to dialog control. In *Proceedings of ICSLP*.

G. Heidorn. 1978. Natural language dialogue for managing an on-line calendar. In *Proceedings of ACM/CSCER*.

E. Horvitz and T. Paek. 2000. DeepListener: Harnessing expected utility to guide clarification dialog in spoken language systems. In *Proceedings of ICSLP*.

X. Huang et al. 2001. MIPAD: A next generation PDA prototype. In *Proceedings of ICSLP*.

A. Raux et al. 2005. Let's go public! Taking a spoken dialog system to the real world. In *Proceedings of Interspeech*.

M. Tue Vo and C. Wood. 1996. Building an application framework for speech and pen input integration in multimodal learning interfaces. In *Proceedings of ICASSP*.

N. Yankelovich. 1994. Talking vs taking: Speech access to remote computers. In *Proceedings of the Conference on Human Factors in Computing Systems*.

The CALO Meeting Assistant

L. Lynn Voss
Engineering and Systems Division
SRI International
Menlo Park, CA 94025
loren.voss@sri.com

Patrick Ehlen
CSLI
Stanford University
Stanford, CA 94305
ehlen@stanford.edu

and

The DARPA[†] CALO Meeting Assistant Project Team*

Abstract

The CALO Meeting Assistant is an integrated, multimodal meeting assistant technology that captures speech, gestures, and multimodal data from multiparty interactions during meetings, and uses machine learning and robust discourse processing to provide a rich, browsable record of a meeting.

1 Introduction

Technologies that assist in making meetings more productive have a long history. The latest chapter in that history involves projects that integrate recent advances in speech, natural language understanding, vision, and multimodal interaction technologies in an effort to produce tools that can perceive what happens at a meeting, extract salient events and interactions, and produce a record of the meeting that people can later consult or analyze.

Research projects such as the ICSI Meeting Project (Janin et al. 2004) have sought to produce automated and segmented transcripts from natural, multiparty speech as it occurs in meetings. Others, like the ISL Smart Meeting Room Task (Waibel et al. 2003), and the M4 and AMI projects (Nijholt, op den Akker, & Heylen 2005), employ instrumented meeting rooms to collect multiple streams of behavior data and analyze the interactions of meeting participants to produce a rich and flexible record of their meeting activities, while also providing a supportive environment for collaboration.

The CALO Meeting Assistant is similar to the latter in that it collects multiple streams of information about the behaviors of people in meetings, and assimilates speech, movement, and note-taking behavior to create a rich representation of the meeting that can be analyzed and reviewed at many levels. However, a primary aim of the CALO Meeting Assistant is to integrate its observations with those of a larger system of agents, which can assess the meeting data it collects in the context of the ongoing projects and workflow in the work lives of each of the meeting participants. Thus, our meeting assistant aims to reach beyond an intelligent room that understands only the activities of people in meetings, and attempts to understand their overarching concerns and interpret their behaviors from the perspective of what their meetings mean to them.

That overarching system of agents is being developed under the DARPA CALO (Cognitive Assistant that Learns and Organizes) Program, which seeks to produce machine learning technology in the form of personalized agents that support high-level knowledge workers in carrying out their professional activities. The CALO system handles a broad range of interrelated decision-making tasks that are traditionally resistant to automation; doing so partly by interacting with, being advised by, and learning from its users. The CALO system can take initiative on completing routine tasks, and on assisting when the unexpected happens.

CALO is designed from the ground up as a cognitive system. Whereas conventional, hand-coded software excels at a narrow set of capabilities in a

[†] This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or the Department of Interior-National Business Center.

[*] The DARPA CALO MA Project is a collaborative effort among researchers at Adapx, CMU, Georgia Tech, MIT, SRI, Stanford University, UC Berkeley, and UC Santa Cruz.

NAACL HLT Demonstration Program, pages 17–18,
Rochester, New York, USA, April 2007.

particular domain, cognitive systems maintain explicit, declarative models of their capabilities, ongoing activities, and operating environments. These models enable CALO to extend and improve its capabilities through learning and adaptation. Cognitive systems are better equipped to cope with unexpected developments, learn to improve over time, and adapt to the contexts and requirements of different situations. CALO also uses natural interfaces that enable simple, effective interactions with humans and other cognitive systems.

The CALO Meeting Assistance Project is developing capabilities to enable CALO to participate in group discussions and meetings. Unlike instrumented "intelligent room" meeting projects, this system is designed for users in an office environment with access to the Internet, a laptop, and some small, off-the-shelf peripheral devices (such as headsets, webcams, and digital writing devices) to capture speech, gestures, and handwriting. It aims to be unobtrusive by leveraging cross-training, unsupervised learning, and lightweight supervision captured from normal user interaction (e.g., users reviewing and editing notes, or adding detected action items to a to-do list).

These data are transparently processed at a central server location and redistributed, so the meeting assistant interacts seamlessly with other CALO desktop functionalities, using a common ontology.

2 What it does

The CALO Meeting Assistant helps its owners by capturing and interpreting meeting conversations and activities and, as appropriate, retrieving relevant information. Information gleaned from a meeting can be incorporated in the respective owner's CALO knowledge stores to, for example, track commitments and remember references to projects, people, places, and dates. An archive of each meeting provides a searchable record for users, as well as a history of training data for CALO's learning components. Learning areas include the following:

Speech processing—Automatic transcriptions are produced from conversational speech among multiple speakers while adapting to speaker and background noise; recognizing prosodic cues; learning new vocabulary; and constructing person, role, and topic-specific language models.

Visual recognition—Faces, gaze direction, gestures, and activities are detected, and detection is improved through lightly-supervised learning and unsupervised cross-training.

Discourse understanding—Dialog moves are recognized, topics are segmented and grouped through supervised and unsupervised generative models, action items are detected, and discussions can be threaded across documents and email.

Multimodal reinforcement—Pen, speech, and text inputs combine to offer natural communications.

Meeting activity—Speech and note-taking activities combine to provide cross-training for recognizing meeting phases, and for tracking agendas and document usage.

3 Demo

We demonstrate how the CALO Meeting Assistant captures speech, pen, and other meeting data using an ordinary laptop; produces an automated transcript; segments by topic; and performs shallow discourse understanding to produce a list of probable action items arising from a single, pre-recorded meeting. We then demonstrate a Meeting Rapporteur that provides a meeting summary and allows participants to review and organize the meeting transcript, audio, notes, action items, and topics—all while providing actions in a feedback loop that supports the meeting assistant's semi-supervised learning process. Finally, we discuss the potential and current development of real-time capabilities that allow users to interact with the meeting assistant during an ongoing meeting.

References

Janin, A., Ang, J., Bhagat, S., Dhillon, R., Edwards, J., Marcias-Guarasa, J., Morgan, N., Peskin, B., Shriberg, E., Stolcke, A., Wooters, C., and Wrede, B. 2004. The ICSI meeting project: Resources and research. In *Proceedings of the 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing* (*ICASSP '04) Meeting Recognition Workshop (NIST RT-04)*.

Nijholt, A., op den Akker, R., and Heylen, D. 2005. Meetings and meeting modeling in smart environments. *AI & Society*, 20(2):202-220.

Waibel, A., Schultz, T., Bett, M., Denecke, M., Malkin, R., Rogina, I., Stiefelhagen, R., and Yang, J. 2003. SMaRT: The smart meeting room task at ISL. In *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, pp 752-755.

OMS-J: An Opinion Mining System for Japanese Weblog Reviews Using a Combination of Supervised and Unsupervised Approaches

Guangwei Wang
Graduate School of Information
Science and Technology
Hokkaido University
Sapporo, Japan 060-0814
wgw@media.eng.hokudai.ac.jp

Kenji Araki
Graduate School of Information
Science and Technology
Hokkaido University
Sapporo, Japan 060-0814
araki@media.eng.hokudai.ac.jp

Abstract

We introduce a simple opinion mining system for analyzing Japanese Weblog reviews called OMS-J. OMS-J is designed to provide an intuitive visual GUI of opinion mining graphs for a comparison of different products of the same type to help a user make a quick purchase decision. We first use an opinion mining method using a combination of supervised (a Naive Bayes Classifier) and unsupervised (an improved SO-PMI: Semantic Orientation Using Pointwise Mutual Information) learning.

1 Introduction

Nowadays, there are numerous Web sites containing personal opinions, e.g. customer reviews of products, forums, discussion groups, and blogs. Here, we use the term Weblog for these sites. How to extract and analyze these opinions automatically, i.e. "Opinion Mining", has seen increasing attention in recent years.

This paper presents a simple opinion mining system (OMS-J) for analyzing Japanese Weblog reviews automatically. The novelty of OMS-J is twofold: First, it provides a GUI using intuitive visual mining graphs aimed at inexperienced users who want to check opinions on the Weblog before purchasing something. These graphs can help the user to make a quick decision on which product is suitable. Secondly, this system combines a supervised and an unsupervised approach to perform opinion mining. In related work (Chaovalit, 2005; Turney, 2002), both supervised and unsupervised approaches have been shown to have their pros and cons. Based on the merits of these approaches and the characteristics of Japanese (Kobayashi, 2003), we proposed an opinion mining method using a Naive Bayes Classifier (supervised approach) and an improved SO-PMI method (unsupervised approach) to perform different parts of the classification task (Wang, 2006).

OMS-J implements Weblog opinion mining by the steps shown in Figure 1. In the next section, we describe the proposed system in detail.

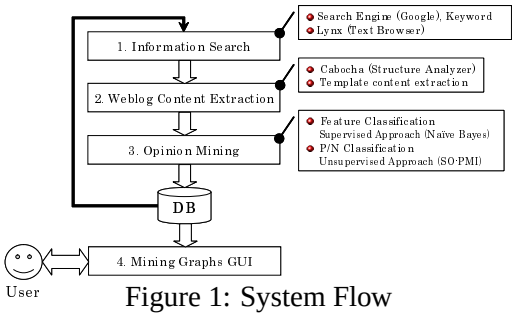


Figure 1: System Flow

2 Proposed System

2.1 Information Search

The first step is information search. We used the Google search engine[1] to get all the information on one product category or one specific product in the Japanese weblog on the Internet. The search keyword is the product category name or the product name. The URL range of the search is restricted by the URL type (e.g. blog, bbs, review).

2.2 Weblog Content Extraction

The Content Extraction step first analyzes the Weblog content using a dependency structure analyzer for Japanese, Cabocha[2]. Based on the syntactic characteristics of Japanese reviews and the results

[1] http://www.google.co.jp/

[2] http://www.chasen.org/~taku/software/cabocha/

NAACL HLT Demonstration Program, pages 19–20,
Rochester, New York, USA, April 2007.

of related work (Kobayashi, 2003; Taku, 2002), we designed the following templates to extract opinion phrases:

< noun + auxiliary word + adj / verb / noun >
< adj + noun / undefined / verb >
< noun + verb >
< noun + symbol + adj / verb / noun >
Except the above < adj >

2.3 Opinion Mining

Opinion mining methods can usually be divided into two types: supervised and unsupervised approaches. Supervised approaches are likely to provide more accurate classification results but need a training corpus. Unsupervised approaches on the other hand require no training data but tend to produce weaker results.

We propose a combined opinion mining method by performing feature classification and P/N classification (Wang, 2006). The purpose of these classifications is to know what the opinion expresses about a certain product's features. Feature means a product's attribute, i.e. price, design, function or battery feature. Based on our previous study, it is easy to create a feature corpus. Therefore feature classification is performed by a supervised approach, a Naive Bayes Classifier. P/N classification classifies reputation expressions into positive or negative meaning using an unsupervised approach, SO-PMI. The SO-PMI approach measures the similarity of pairs of words or phrases based on the mutual information theory, in our case the closeness of an opinion and words for "good" or "bad".

No human effort is required when mining a new product or category. Only inputting the name of the product or category is needed. It does however require quite a lot of processing time, since the SO-PMI approach using a search engine is very time consuming. Adding new features requires manual work, since a small hand labeled training corpus is used. Similar categories of products, for instance cameras and mp3 players, use the same features though, so this is not done very often.

2.4 Mining Graphs GUI

Finally, OMS-J provides a GUI with mining graphs showing the opinion mining data in the database, as shown in Figure 2. These graphs show the distribution of positive and negative opinions of each feature type such as "design", and for each product. The distribution of positive opinions among the different product choices are shown in a pie chart, as is the same for negative opinions. This GUI can also show graphs for a single product's mining results, showing the positive/negative opinion distribution of each feature.

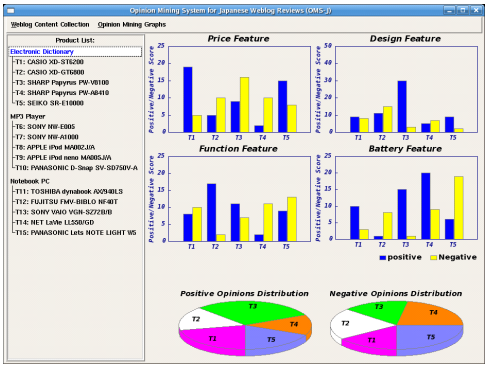


Figure 2: OMS-J's GUI Screenshot for One Product Category

3 Demonstration

During the demonstration, we will show that OMS-J is an intuitive opinion mining system that can help people to make a quick decision on purchasing some product. OMS-J's trial version has been developed and tested with three kinds of products: Electronic Dictionaries, MP3 Players and Notebook PCs. The experiment results were positive. We will show how the system works when a user wants to buy a good MP3 player or wants to get a feel for the general opinions on a specific Notebook PC etc.

References

Pimwadee Chaovalit and Lina Zhou. 2005. *Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches*. Proceedings of the 38th Annual HICSS.

Peter D. Turney. 2002. *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*. Proceedings 40th Annual Meeting of the ACL, pp. 417-424.

Nozomi Kobayashi, Kentaro Inui, Yuji Matsumoto, Kenji Tateishi and Toshikazu Fukushima. 2003. *Collecting evaluative expressions by a text mining technique*. IPSJ SIG NOTE, Vol.154, No.12.

Guangwei Wang and Kenji Araki. 2006. *A Decision Support System Using Text Mining Technology*. IEICE SIG Notes WI2-2006-6, pp. 55-56.

Taku Kudoh and Yuji Matsumoto. 2002. *Applying Cascaded Chunking to Japanese Dependency Structure Analysis*. Information Processing Society of Japan (IPSJ)Academic Journals, Vol 43, No 6, pp. 1834-1842.

Learning to find transliteration on the Web

Chien-Cheng Wu
Department of Computer Science
National Tsing Hua University
101 Kuang Fu Road, Hsin chu, Taiwan
d9283228@cs.nthu.edu.tw

Jason S. Chang
Department of Computer Science
National Tsing Hua University
101 Kuang Fu Road, Hsin chu, Taiwan
jschang@cs.nthu.edu.tw

This prototype demonstrate a novel method for learning to find transliterations of proper nouns on the Web based on query expansion aimed at maximizing the probability of retrieving transliterations from existing search engines. Since the method we used involves learning the morphological relationships between names and their transliterations, we refer to this IR-based approach as *morphological query expansion for machine transliteration*. The *morphological query expansion* approach is general in scope and can be applied to translation and transliteration, but we focus on transliteration in this paper.

Many texts containing proper names (e.g., "The cities of Mesopotamia prospered under Parthian and Sassanian rule.") are submitted to machine translation services on the Web every day, and there are also service on the Web specifically target transliteration of proper names, including *CHINET* (Kwok et al. 2005) ad *Livetrans* (Lu, Chien, and Lee 2004).

Machine translation systems on the Web such as *Yahoo Translate* (babelfish.yahoo.com) and *Google Translate* (translate.google.com/translate_t.g) typically use a bilingual dictionary that is either manually compiled or learned from a parallel corpus. However, such dictionaries often have insufficient coverage of proper names and technical terms, leading to poor translation due to out of vocabulary problem. The OOV problems of machine translation or cross language information retrieval can be handled more effectively by learning to find transliteration on the Web.

Consider Sentence 1 containing three place names.

1. *The cities of Mesopotamia prospered under Parthian and Sassanian rule.*
2. *城市繁榮下parthian 達米亞、sassanian統治。*
3. 美索不達米亞城市在巴底亞和薩珊統治下繁榮起來。

Google Translate produce Sentence 2, leaving "Parthian" and "Sassanian" not translated. A good response might be a translation like Sentence 3 where all place names have appropriate transliterations (underlined). These transliterations can be more effectively retrieved from mixed code Web pages by extend each of the place names into a query (e.g., "Parthian NEAR 巴"). Intuitively by requiring one of likely prefix transliteration morphemes (e.g., "巴" or "帕" for "par-" names), we can bias the search engine towards retrieving the correct transliterations (e.g., "巴底亞" and "帕提亞") in snippets of many top-ranked documents.

The method involves pairing up the prefixing morphemes between name and transliteration in a set of train data, calculating the statistical association for these pair, and selecting pairs with a high degree of statistical association. The results of this training stage are morphological relationships between prefixes and postfixes of names and transliterations. At run time, a given name is automatically extended into a query with relevant prefixing morphemes, then the query is submit to some search engine. After retrieving snippets from a search engine, the system extract transliterations from the snippets based on redundancy, proximity between name and transliteration, and cross language morphological relationships of prefix and postfix.

We present a new machine transliteration system based on information retrieval and morphological query expansion. The system automatically

NAACL HLT Demonstration Program, pages 21–22,
Rochester, New York, USA, April 2007.

learns to extend the proper names into a query expected to retrieve and extract transliterations of the proper names. Consider the case of transliteration of "Parthian." The system looks at possible prefixes of the given name, including *p-*, *pa-*, *par-*, and *part-*, and determine determines the best *n* query expansions (e.g., "Parthian 巴," "Parthian 帕"). These effective expansions automatically during training by analyzing a collection of 23,615 place names and transliterations pairs.

We evaluated the prototype system using a list of 500 proper names. The results show that 60% of the time there are sufficient relevant data on the Web to carry out effective machine transliteration based on IR and morphological query expansion. Of many results returned by the system, the top 1, two and three results are 0.88, 0.93, and 0.94. By performing query expansion, the system improves the recall rate from 0.48 to 0.60.

The results indicate that most names and transliteration counterparts can often be found on the Web and the proposed method are very effective in retrieving and extracting transliterations based on a statistical machine transliteration model trained on a bilingual name list. Our demonstration prototype shows alternative transliterations in use on the Web and snippets of such usage, so that the user can easily validate these transliterations.

The prototype supports:

- Searching and extracting transliterations of a given term
- Listing alternative transliterations on the Web
- Listing alternative transliteration in a local dictionary
- Browsing of snippets containing for each alterative transliteration
- Saving transliterations in a local dictionary
- Selecting and saving transliteration in snippets to a local dictionary

The method explored here can be extended as an alterative way to support such MT subtasks as back transliteration (Knight and Graehl 1998) and noun phrase translation (Koehn and Knight 2003). Finally, for more challenging tasks, such as handling sentences, the improvement of translation quality probably will also be achieved by combining this IR-based approach and statistical machine translation. For example, a preprocessing unit may replace the proper names in a sentence with transliterations (e.g., mixed code text such as Sentence 4) *on the fly* or by looking up a local dictionary before sending it off to MT for finally translation.

4. *The cities of* 美索不達米亞 *prospered under* 巴底亞 *and* 薩珊 *rule*.

Morphological query expansion represents an innovative way to capture cross-linguistic relations in name transliteration. The method is independent of the bilingual lexicon content making it easy to adopt to other proper names such person, product, or organization names. This approach is useful in a number of machine translation subtasks, including name transliteration, back transliteration, named entity translation, and terminology translation.

References

Y. Cao and H. Li. (2002). *Base Noun Phrase Translation Using Web Data and the EM Algorithm*, In Proc. of COLING 2002, pp.127-133.

K. Knight, J. Graehl. (1998). *Machine Transliteration*. In Journal of Computational Linguistics 24(4), pp.599-612.

P. Koehn, K. Knight. (2003). *Feature-Rich Statistical Translation of Noun Phrases*. In Proc. of ACL 2003, pp.311-318.

KL Kwok, P Deng, N Dinstl, HL Sun, W Xu, P Peng, *CHINET: a Chinese name finder system for document triage*. Proceedings of 2005 International Conference on Intelligence, 2005.

T. Lin, J.C. Wu, and J. S. Chang. (2004). *Extraction of Name and Transliteration in Monolingual and Parallel Corpora*. In Proc. of AMTA 2004, pp.177-186.

WH Lu, LF Chien, HJ Lee. *Anchor text mining for translation of Web queries: A transitive translation approach*. ACM Transactions on Information Systems (TOIS), 2004.

M. Nagata, T. Saito, and K. Suzuki. (2001). *Using the Web as a bilingual dictionary*. In Proc. of ACL 2001 DD-MT Workshop, pp.95-102.

A Conversational In-car Dialog System

Baoshi Yan[1] **Fuliang Weng**[1] **Zhe Feng**[1] **Florin Ratiu**[2] **Madhuri Raya**[1] **Yao Meng**[1]
Sebastian Varges[2] **Matthew Purver**[2] **Annie Lien**[1] **Tobias Scheideck**[1] **Badri Raghunathan**[1]
Feng Lin[1] **Rohit Mishra**[4] **Brian Lathrop**[4] **Zhaoxia Zhang**[4] **Harry Bratt**[3] **Stanley Peters**[2]
Research and Technology Center, Robert Bosch LLC, Palo Alto, California[1]
Center for the Study of Language and Information, Stanford University, Stanford, California[2]
Speech Technology and Research Lab, SRI International, Menlo Park, California[3]
Electronics Research Lab, Volkswagen of America, Palo Alto, California[4]

Abstract

In this demonstration we present a conversational dialog system for automobile drivers. The system provides a voice-based interface to playing music, finding restaurants, and navigating while driving. The design of the system as well as the new technologies developed will be presented. Our evaluation showed that the system is promising, achieving high task completion rate and good user satisfation.

1 Introduction

As a constant stream of electronic gadgets such as navigation systems and digital music players enters cars, it threatens driving safety by increasing driver distraction. According to a 2005 report by the National Highway Traffic Safety Administration (NHTSA) (NHTSA, 2005), driver distraction and inattention from all sources contributed to 20-25% of police reported crashes. It is therefore important to design user interfaces to devices that minimize driver distraction, to which voice-based interfaces have been a promising approach as they keep a driver's hands on the wheel and eyes on the road.

In this demonstration we present a conversational dialog system, CHAT, that supports music selection, restaurant selection, and driving navigation (Weng et al., 2006). The system is a joint research effort from Bosch RTC, VW ERL, Stanford CSLI, and SRI STAR Lab funded by NIST ATP. It has reached a promising level, achieving a task completion rate of 98%, 94%, 97% on playing music, finding restaurants, and driving navigation respectively.

Specifically, we plan to present a number of features in the CHAT system, including end-pointing with prosodic cues, robust natural language understanding, error identification and recovery strategies, content optimization, full-fledged reponse generation, flexible multi-threaded, multi-device dialog management, and support for random events, dynamic information, and domain switching.

2 System Descriptions

The spoken dialog system consists of a number of components (see the figure on the next page). Instead of the hub architecture employed by Communicator projects (Seneff et al., 1998), it is developed in Java and uses flexible event-based, message-oriented middleware. This allows for dynamic registration of new components. Among the component modules in the figure, we use the Nuance speech recognition engine with class-based n-grams and dynamic grammars, and the Nuance Vocalizer as the TTS engine. The Speech Enhancer removes noises and echo. The Prosody module will provide additional features to the Natural Language Understanding (NLU) and Dialog Manager (DM) modules to improve their performance.

The NLU module takes a sequence of recognized words and tags, performs a deep linguistic analysis with probabilistic models, and produces an XML-based semantic feature structure representation. Parallel to the deep analysis, a topic classifier assigns n-best topics to the utterance, which are used in the cases where the dialog manager cannot make any sense of the parsed structure. The NLU module also supports dynamic updates of the knowledge base.

The DM module mediates and manages interac-

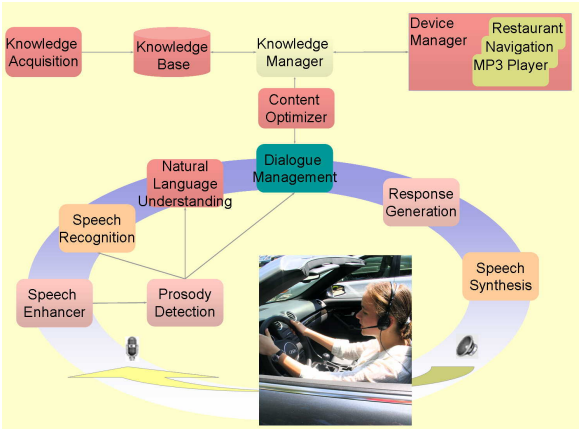
NAACL HLT Demonstration Program, pages 23–24,
Rochester, New York, USA, April 2007.

tion. It uses an information-state-update approach to maintain dialog context, which is then used to interpret incoming utterances (including fragments and revisions), resolve NPs, construct salient responses, track issues, etc. Dialog states can also be used to bias SR expectation and improve SR performance, as has been performed in previous applications of the DM. Detailed descriptions of the DM can be found in (Lemon et al., 2002) (Mirkovic and Cavedon, 2005).

The Knowledge Manager (KM) controls access to knowledge base sources (such as domain knowledge and device information) and their updates. Domain knowledge is structured according to domain-dependent ontologies. The current KM makes use of OWL, a W3C standard, to represent the ontological relationships between domain entities.

The Content Optimization module acts as an intermediary between the dialog management module and the knowledge management module and controls the amount of content and provides recommendations to user. It receives queries in the form of semantic frames from the DM, resolves possible ambiguities, and queries the KM. Depending on the items in the query result as well as configurable properties, the module selects and performs an appropriate optimization strategy (Pon-Barry et al., 2006).

The Response Generation module takes query results from the KM or Content Optimizer and generates natural language sentences as system responses to user utterances. The query results are converted into natural language sentences via a bottom-up approach using a production system. An alignment-based ranking algorithm is used to select the best generated sentence.



The system supports random events and dynamic external information, for example, the system prompts users for the next turn when they drive close to an intersection and dialogs can be carried out in terms of the current dynamic situation. The user can also switch among the three different applications easily by explicitly instructing the system which domain to operate in.

3 Acknowledgement

This work is partially supported by the NIST Advanced Technology Program.

References

Oliver Lemon, Alex Gruenstein, and Stanley Peters. 2002. Collaborative activities and multi-tasking in dialogue systems. In *Traitement Automatique des Langues (TAL)*, page 43(2).

Danilo Mirkovic and Lawrence Cavedon. 2005. Practical Plug-and-Play Dialogue Management. In *Proceedings of the 6th Meeting of the Pacific Association for Computational Linguistics (PACLING)*, page 43(2), Tokyo, Japan.

National Highway Traffic Safety Administration NHTSA. 2005. *NHTSA Vehicle Safety Rulemaking and Supporting Research Priorities: Calendar Years 2005-2009*. January.

Heather Pon-Barry, Fuliang Weng, and Sebastian Varges. 2006. Evaluation of content presentation strategies for an in-car spoken dialogue system. In *Proceedings of the 9th International Conference on Spoken Language Processing (Interspeech/ICSLP)*, pages 1930–1933, Pittsburgh, PA, September.

Stephanie Seneff, Ed Hurley, Raymond Lau, Christine Pao, Philipp Schmid, and Victor Zue. 1998. GALAXY-II: A Reference Architecture for Conversational System Development. In *International Conference on Spoken Language Processing (ICSLP)*, page 43(2), Sydney, Australia, December.

Fuliang Weng, Sebastian Varges, Badri Raghunathan, Florin Ratiu, Heather Pon-Barry, Brian Lathrop, Qi Zhang, Tobias Scheideck, Harry Bratt, Kui Xu, Matthew Purver, Rohit Mishra, Annie Lien, Madhuri Raya, Stanley Peters, Yao Meng, Jeff Russel, Lawrence Cavedon, Liz Shriberg, and Hauke Schmidt. 2006. CHAT: A conversational helper for automotive tasks. In *Proceedings of the 9th International Conference on Spoken Language Processing (Interspeech/ICSLP)*, pages 1061–1064, Pittsburgh, PA, September.

TextRunner: Open Information Extraction on the Web

Alexander Yates **Michele Banko** **Matthew Broadhead**
Michael Cafarella **Oren Etzioni** **Stephen Soderland**
University of Washington
Computer Science and Engineering
Box 352350
Seattle, WA 98195-2350
{ayates,banko,hastur,mjc,etzioni,soderlan}@cs.washington.edu

1 Introduction

Traditional information extraction systems have focused on satisfying precise, narrow, pre-specified requests from small, homogeneous corpora. In contrast, the TextRunner system demonstrates a new kind of information extraction, called Open Information Extraction (OIE), in which the system makes a single, data-driven pass over the entire corpus and extracts a large set of relational tuples, without requiring *any* human input. (Banko et al., 2007) TextRunner is a fully-implemented, highly scalable example of OIE. TextRunner's extractions are indexed, allowing a fast query mechanism.

Our first public demonstration of the TextRunner system shows the results of performing OIE on a set of 117 million web pages. It demonstrates the power of TextRunner in terms of the raw number of facts it has extracted, as well as its precision using our novel assessment mechanism. And it shows the ability to automatically determine synonymous relations and objects using large sets of extractions. We have built a fast user interface for querying the results.

2 Previous Work

The bulk of previous information extraction work uses hand-labeled data or hand-crafted patterns to enable relation-specific extraction (e.g., (Culotta et al., 2006)). OIE seeks to avoid these requirements for human input.

Shinyama and Sekine (Shinyama and Sekine, 2006) describe an approach to "unrestricted relation discovery" that does away with many of the requirements for human input. However, it requires clustering of the documents used for extraction, and thus scales in quadratic time in the number of documents. It does not scale to the size of the Web.

For a full discussion of previous work, please see (Banko et al., 2007), or see (Yates and Etzioni, 2007) for work relating to synonym resolution.

3 Open IE in TextRunner

OIE presents significant new challenges for information extraction systems, including

Automation of relation extraction, which in traditional information extraction uses hand-labeled inputs.

Corpus Heterogeneity on the Web, which makes tools like parsers and named-entity taggers less accurate because the corpus is different from the data used to train the tools.

Scalability and efficiency of the system. Open IE systems are effectively restricted to a single, fast pass over the data so that they can scale to huge document collections.

In response to these challenges, TextRunner includes several novel components, which we now summarize (see (Banko et al., 2007) for details).

1. Single Pass Extractor

The TextRunner extractor makes a single pass over all documents, tagging sentences with part-of-speech tags and noun-phrase chunks as it goes. For each pair of noun phrases that are not too far apart, and subject to several other constraints, it applies a classifier described below to determine whether or not to extract a relationship. If the classifier

NAACL HLT Demonstration Program, pages 25–26,
Rochester, New York, USA, April 2007.

deems the relationship trustworthy, a tuple of the form $t = (e_i, r_j, e_k)$ is extracted, where e_i, e_k are entities and r_j is the relation between them. For example, TEXTRUNNER might extract the tuple *(Edison, invented, light bulbs)*. On our test corpus (a 9 million document subset of our full corpus), it took less than 68 CPU hours to process the 133 million sentences. The process is easily parallelized, and took only 4 hours to run on our cluster.

2. Self-Supervised Classifier

While full parsing is too expensive to apply to the Web, we use a parser to generate training examples for extraction. Using several heuristic constraints, we automatically label a set of parsed sentences as trustworthy or untrustworthy extractions (positive and negative examples, respectively). The classifier is trained on these examples, using features such as the part of speech tags on the words in the relation. The classifier is then able to decide whether a sequence of POS-tagged words is a correct extraction with high accuracy.

3. Synonym Resolution

Because TEXTRUNNER has no pre-defined relations, it may extract many different strings representing the same relation. Also, as with all information extraction systems, it can extract multiple names for the same object. The RESOLVER system performs an unsupervised clustering of TEXTRUNNER's extractions to create sets of synonymous entities and relations. RESOLVER uses a novel, unsupervised probabilistic model to determine the probability that any pair of strings is co-referential, given the tuples that each string was extracted with. (Yates and Etzioni, 2007)

4. Query Interface

TEXTRUNNER builds an inverted index of the extracted tuples, and spreads it across a cluster of machines. This architecture supports fast, interactive, and powerful relational queries. Users may enter words in a relation or entity, and TEXTRUNNER quickly returns the entire set of extractions matching the query. For example, a query for "Newton" will return tuples like *(Newton, invented, calculus)*. Users may opt to query for all tuples matching synonyms of the keyword input, and may also opt to merge all tuples returned by a query into sets of tuples that are deemed synonymous.

4 Experimental Results

On our test corpus of 9 million Web documents, TEXTRUNNER extracted 7.8 million well-formed tuples. On a randomly selected subset of 400 tuples, 80.4% were deemed correct by human reviewers.

We performed a head-to-head comparison with a state-of-the-art traditional information extraction system, called KNOWITALL. (Etzioni et al., 2005) On a set of ten high-frequency relations, TEXTRUNNER found nearly as many correct extractions as KNOWITALL (11,631 to 11,476), while reducing the error rate of KNOWITALL by 33% (18% to 12%).

Acknowledgements

This research was supported in part by NSF grants IIS-0535284 and IIS-0312988, DARPA contract NBCHD030010, ONR grant N00014-05-1-0185 as well as gifts from Google, and carried out at the University of Washington's Turing Center.

References

M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. 2007. Open Information Extraction from the Web. In *IJCAI*.

A. Culotta, A. McCallum, and J. Betz. 2006. Integrating Probabilistic Extraction Models and Relational Data Mining to Discover Relations and Patterns in Text. In *HLT-NAACL*.

O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. 2005. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. *Artificial Intelligence*, 165(1):91–134.

Y. Shinyama and S. Sekine. 2006. Preemptive Information Extraction Using Unrestricted Relation Discovery. In *HLT-NAACL*.

A. Yates and O. Etzioni. 2007. Unsupervised Resolution of Objects and Relations on the Web. In *NAACL-HLT*.

The Hidden Information State Dialogue Manager: A Real-World POMDP-Based System

Steve Young, Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye
Cambridge University Engineering Department
Trumpington Street, Cambridge, CB21PZ, United Kingdom
{sjy, js532, brmt2, kw278, hy216}@eng.cam.ac.uk

Abstract

The Hidden Information State (HIS) Dialogue System is the first trainable and scalable implementation of a spoken dialog system based on the Partially-Observable Markov-Decision-Process (POMDP) model of dialogue. The system responds to n-best output from the speech recogniser, maintains multiple concurrent dialogue state hypotheses, and provides a visual display showing how competing hypotheses are ranked. The demo is a prototype application for the Tourist Information Domain and achieved a task completion rate of over 90% in a recent user study.

1 Partially Observable Markov Decision Processes for Dialogue Systems

Recent work on statistical models for spoken dialogue systems has argued that Partially Observable Markov Decision Processes (POMDPs) provide a principled mathematical framework for modeling the uncertainty inherent in human-machine dialogue (Williams, 2006; Young, 2006; Williams and Young, 2007). Briefly speaking, POMDPs extend the traditional fully-observable Markov Decision Process (MDP) framework by maintaining a *belief state*, ie. a probability distribution over dialogue states. This enables the dialogue manager to avoid and recover from recognition errors by sharing and shifting probability mass between multiple hypotheses of the current dialogue state. The framework also naturally incorporates n-best lists of multiple recognition hypotheses coming from the speech recogniser.

Due to the vast number of possible dialogue states and policies, the use of POMDPs in practical dialogue systems is far from straightforward. The size of the belief state scales linearly with the number of dialogue states and belief state updates at every turn during a dialogue require all state probabilities to be recomputed. This is too computationally intensive to be practical with current technology. Worse than that, the complexity involved in policy optimisation grows exponentially with the number of states and system actions and neither exact nor approximate algorithms exist that provide a tractable solution for systems with thousands of states.

2 The Hidden Information State (HIS) Dialogue Manager

The Hidden Information State (HIS) dialogue manager presented in this demonstration is the first trainable and scalable dialogue system based on the POMDP model. As described in (Young, 2006; Young et al., 2007) it partitions the state space using a tree-based representation of user goals so that only a small set of partition beliefs needs to be updated at every turn. In order to make policy optimisation tractable, a much reduced summary space is maintained in addition to the master state space. Policies are optimised in summary space and the selected summary actions are then mapped back to master space to form system actions. Apart from some very simple ontology definitions, the dialog manager has no application dependent heuristics.

The system uses a grid-based discretisation of the

NAACL HLT Demonstration Program, pages 27–28,
Rochester, New York, USA, April 2007.

Figure 1: The HIS Demo System is a Tourist Information application for a fictitious town

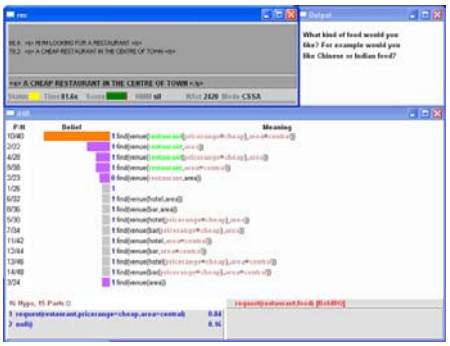


Figure 2: A system screenshot showing the ranking of competing dialogue state hypotheses

state space and online ϵ-greedy policy optimisation. While this offers the potential for online adaptation with real users at a later stage, a simulated user is needed to bootstrap the training process. A novel agenda-based simulation technique was used for this step, as described in (Schatzmann et al., 2007).

3 The HIS Demo System

The HIS demo system is a prototype application for the Tourist Information domain. Users are assumed to be visiting a fictitious town called "Jasonville" (see Fig. 1) and need to find a suitable hotel, bar or restaurant subject to certain constraints. Examples of task scenarios are *"finding a cheap Chinese restaurant near the post office in the centre of town"* or *"a wine bar with Jazz music on the riverside"*. Once a venue is found, users may request further information such as the phone number or the address.

At run-time, the system provides a visual display (see Fig. 2) which shows how competing dialogue state hypotheses are being ranked. This allows developers to gain a better understanding of the internal operation of the system.

4 Demo System Performance

In a recent user study the demo system was evaluated by 40 human subjects. In total, 160 dialogues were recorded with an average Word-Error-Rate of 29.8%. The performance of the system was measured based on the recommendation of a correct venue and achieved a task completion rate of 90.6% with an average number of 5.59 dialogue turns to completion (Thomson et al., 2007).

The results demonstrate that POMDPs facilitate design and implementation of spoken dialogue systems, and that the implementation used in the HIS dialogue manager can be scaled to handle real world tasks. The user study results also show that a simulated user can be successfully used to train a POMDP dialogue policy that performs well in experiments with real users.

5 Accompanying materials

The demo system and related materials are accessible online at our website
`http://mi.eng.cam.ac.uk/research/dialogue/`.

References

J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young. 2007. Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In *Proceedings of HLT/NAACL, Rochester, NY*.

B. Thomson, J. Schatzmann, K. Weilhammer, H. Ye, and S. Young. 2007. Training a real-world POMDP-based Dialogue System. In *Proceedings of Bridging the Gap: Academic and Industrial Research in Dialog Technology, Workshop at HLT/NAACL, Rochester, NY*.

J. D. Williams and S. Young. 2007. Partially Observable Markov Decision Processes for Spoken Dialog Systems. *Computer Speech and Language*, 21(2):231–422.

J. D. Williams. 2006. *Partially Observable Markov Decision Processes for Spoken Dialogue Management*. Ph.D. thesis, University of Cambridge.

S. Young, J. Schatzmann, K. Weilhammer, and H. Ye. 2007. The Hidden Information State Approach to Dialog Management. In *Proc. of ICASSP (forthcoming)*, Honolulu, Hawaii.

S. Young. 2006. Using POMDPs for Dialog Management. In *Proc. of IEEE/ACL SLT*, Palm Beach, Aruba.

Text Comparison Using Machine-Generated Nuggets

Liang Zhou
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
liangz@isi.edu

Abstract

This paper describes a novel text comparison environment that facilities text comparison administered through assessing and aggregating information nuggets automatically created and extracted from the texts in question. Our goal in designing such a tool is to enable and improve automatic nugget creation and present its application for evaluations of various natural language processing tasks. During our demonstration at HLT, new users will able to experience first hand text analysis can be fun, enjoyable, and interesting using system-created nuggets.

1 Introduction

In many natural language processing (NLP) tasks, such as question answering (QA), summarization, etc., we are faced with the problem of determining the appropriate granularity level for information units in order to conduct appropriate and effective evaluations. Most commonly, we use sentences to model individual pieces of information. However, more and more NLP applications require us to define text units smaller than sentences, essentially decomposing sentences into a collection of phrases. Each phrase carries an independent piece of information that can be used as a standalone unit. These finer-grained information units are usually referred to as *nuggets*.

Previous work shows that humans can create nuggets in a relatively straightforward fashion. A serious problem in manual nugget creation is the inconsistency in human decisions (Lin and Hovy, 2003). The same nugget will not be marked consistently with the same words when sentences containing multiple instances of it are presented to human annotators. And if the annotation is performed over an extended period of time, the consistency is even lower.

Given concerns over these issues, we have set out to design an evaluation toolkit to address three tasks in particular: 1) provide a consistent definition of what a nugget is; 2) automate the nugget extraction process systematically; and 3) utilize automatically extracted nuggets for text comparison and aggregation.

The idea of using semantic equivalent nuggets to compare texts is not new. QA and summarization evaluations (Lin and Demner-Fushman, 2005; Nenkova and Passonneau, 2004) have been carried out by using a set of manually created nuggets and the comparison procedure itself is either automatic using n-gram overlap counting or manually performed. We envisage the nuggetization process being automated and nugget comparison and aggregation being performed by humans. It's crucial to still involve humans in the process because recognizing semantic equivalent text units is not a trivial task. In addition, since nuggets are system-produced and can be imperfect, annotators are allowed to reject and re-create them. We provide easy-to-use editing functionalities that allow manual overrides. Record keeping on edits over erroneous nuggets is conducted in the background so that further improvements can be made for nugget extraction.

NAACL HLT Demonstration Program, pages 29–30,
Rochester, New York, USA, April 2007.

2 Nugget Definition

Based on our manual analysis and computational modeling of nuggets, we define them as follows:

Definition:

- A nugget is predicated on either an *event* or an *entity*.
- Each nugget consists of two parts: the anchor and the content.

The anchor is either:

- the head noun of the entity, or
- the head verb of the event, plus the head noun of its associated entity (if more than one entity is attached to the verb, then its subject).

The content is a coherent single piece of information associated with the anchor. Each anchor may have several separate contents. When the nugget contains nested sentences, this definition is applied recursively.

3 Nugget Extraction

We use syntactic parse trees produced by the Collins parser (Collins, 1999) to obtain the structural representation of sentences. Nuggets are extracted by identifying subtrees that are descriptions for entities and events. For entities, we examine subtrees headed by "NP"; for events, subtrees headed by "VP" are examined and their corresponding subjects (siblings headed by "NP") are investigated as possible entity attachments for the verb phrases. Figure 1 shows an example where words in brackets represent corresponding nuggets' anchors.

4 Comparing Texts

When comparing multiple texts, we present the annotator with each text's sentences along with nuggets extracted from individual sentences (see Appendix A). Annotators can select multiple nuggets from sentences across texts to indicate their semantic equivalence. Equivalent nuggets are grouped into nugget groups. There is a frequency score, the number of texts it appeared in, for each nugget group. We allow annotators to modify the nugget groups' contents, thus creating a new label (or can be viewed as a super-nugget) for each nugget group. Record keeping is conducted in the background automatically each time a nugget group is created. When the annotator changes the content of a nugget group, it indicates that either the system-extracted nuggets are not perfect or a super-nugget is created for the group (see Appendix B and C). These editing changes are recorded. The recorded information affords us the opportunity to improve the nuggetizer and perform subsequence study phrase-level paraphrasing, text entailment, etc.

Sentence:
The girl working at the bookstore in Hollywood talked to the diplomat living in Britain.

Nuggets are:
[girl] working at the bookstore in Hollywood
[girl] working at the bookstore
[bookstore] in Hollywood
girl [talked] to the diplomat living in Britain
girl [talked] to the diplomat
[diplomat] living in Britian

Figure 1. Nugget example. (words in brackets are the anchors).

5 Hardware Requirement

Our toolkit is written in Java and can be run on any machine with the latest Java installed.

References

Collins, M. 1999. Head-driven statistical models for natural language processing. *PhD Dissertation*, University of Pennsylvania.

Lin, C.Y. and E. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In Proceedings of *NAACL-HLT 2003*.

Lin, J. and D. Demner-Fushman. 2005. Automatically evaluating answers to definition questions. In *Proceedings of HLT-EMNLP 2005*.

Nenkova, A. and R. Passonneau. 2004. Evaluating content selection in summarization: the pyramid method. In Proceedings of NAACL-HLT 2004.

Voice-Rate: A Dialog System for Consumer Ratings

Geoffrey Zweig, Y.C. Ju, Patrick Nguyen, Dong Yu,
Ye-Yi Wang and Alex Acero
Speech Research Group
Microsoft Corp.
Redmond, WA 98052
{gzweig,yuncj,panguyen,dongyu, yeyi-wang,alexac}@microsoft.com

Abstract

Voice-Rate is an automated dialog system which provides access to over one million ratings of products and businesses. By calling a toll-free number, consumers can access ratings for products, national businesses such as airlines, and local businesses such as restaurants. Voice-Rate also has a facility for recording and analyzing ratings that are given over the phone. The service has been primed with ratings taken from a variety of web sources, and we are augmenting these with user ratings. Voice-Rate can be accessed by dialing 1-877-456-DATA.

1 Overview

Voice-Rate is an automated dialog system designed to help consumers while they are shopping. The target user is a consumer who is considering making an impulse purchase and would like to get more information. He or she can take out a cellphone, call Voice-Rate, and get rating information to help decide whether to buy the item. Here are three sample scenarios:

- Sally has gone to Home Depot to buy some paint to touch-up scratches on the wall at home. She'll use exactly the same color and brand as when she first painted the wall, so she knows what she wants. While at Home Depot, however, Sally sees some hand-held vacuum cleaners and decides it might be nice to have one. But, she is unsure whether which of the available models is better: The "Black & Decker CHV1400 Cyclonic DustBuster," the "Shark SV736" or the "Eureka 71A." Sally calls Voice-Rate and gets the ratings and makes an informed purchase.
- John is on vacation with his family in Seattle. After going up in the Space Needle, they walk by "Abbondanza Pizzeria" and are considering lunch there. While it looks good, there are almost no diners inside, and John is suspicious. He calls Voice-Rate and discovers that in fact the restaurant is highly rated, and decides to go there.
- Returning from his vacation, John drops his rental car off at the airport. The rental company incorrectly asserts that he has scratched the car, and causes a big hassle, until they finally realize that they already charged the last customer for the same scratch. Unhappy with the surly service, John calls Voice-Rate and leaves a warning for others.

Currently, Voice-Rate can deliver ratings for over one million products, two hundred thousand restaurants in over sixteen hundred cities; and about three thousand national businesses.

2 Technical Challenges

To make Voice-Rate operational, it was necessary to solve the key challenges of name resolution and disambiguation. Users rarely make an exactly correct specification of a product or business, and it is necessary both to utilize a "fuzzy-match" for name lookup, and to deploy a carefully designed disambiguation strategy.

NAACL HLT Demonstration Program, pages 31–32,
Rochester, New York, USA, April 2007.

Voice-Rate solves the fuzzy-matching process by treating spoken queries as well as business and product names as documents, and then performing TF-IDF based lookup. For a review of name matching methods, see e.g. Cohen et al., 2003. In the ideal case, after a user asks for a particular product or business, the best-matching item as measured by TF-IDF would be the one intended by the user. In reality, of course, this is often not the case, and further dialog is necessary to determine the user's intent. For concreteness, we will illustrate the disambiguation process in the context of product identification.

When a user calls Voice-Rate and asks for a product review, the system solicits the user for the product name, does TF-IDF lookup, and presents the highest-scoring match for user confirmation. If the user does not accept the retrieved item, Voice-Rate initiates a disambiguation dialog.

Aside from inadequate product coverage, which cannot be fixed at runtime, there are two possible sources for error: automatic speech recognition (ASR) errors, and TF-IDF lookup errors. The disambiguation process begins by eliminating the first. To do this, it asks the user if his or her exact words were the recognized text, and if not to repeat the request. This loop iterates twice, and if the user's exact words still have not been identified, Voice-Rate apologizes and hangs up.

Once the user's exact words have been validated, Voice-Rate gets a positive identification on the product category. From the set of high-scoring TF-IDF items, a list of possible categories is compiled. For example, for "The Lord of the Rings The Two Towers," there are items in Video Games, DVDs, Music, VHS, Software, Books, Websites, and Toys and Games. These categories are read to the user, who is asked to select one. All the close-matching product names in the selected category are then read to the user, until one is selected or the list is exhausted.

3 Related Work

To our knowledge, Voice-Rate is the first large scale ratings dialog system. However, the technology behind it is closely related to previous dialog systems, especially directory assistance or "411" systems (e.g. Kamm et al., 1994, Natarajan et al., 2002, Levin et al., 2005, Jan et al., 2003). A general discussion of name-matching techniques such as TF-IDF can be found in (Cohen et al., 2003, Bilenko et al., 2003).

The second area of related research has to do with web rating systems. Interesting work on extracting information from such ratings can be found in, e.g. (Linden et al., 2003, Hu et al., 2004, Gammon et al., 2005). Work has also been done using text-based input to determine relevant products (Chai et al., 2002). Our own work differs from this in that it focuses on *spoken* input, and in its *breadth* – covering both products and businesses.

References

M. Bilenko, R. Mooney, W. W. Cohen, P. Ravikumar and S. Fienberg. 2003. Adaptive Name-Matching in Information Integration. *IEEE Intelligent Systems* 18(5): 16-23 (2003).

J. Chai, V. Horvath, N. Nicolov, M. Stys, N. Kambhatla, W. Zadrozny and P. Melville. 2002. Natural Language Assistant- A Dialog System for Online Product Recommendation. *AI Magazine (23)*, 2002

W. W. Cohen, P Ravikumar and S. E. Fienberg . 2003. A comparison of string distance metrics for name-matching tasks. *Proceedings of the IJCAI-2003 Workshop on Information*, 2003

M. Gamon, A. Aue, S. Corston-Oliver and E. Ringger. 2005. Pulse: Mining Customer Opinions from Free Text. In *Lecture Notes in Computer Science. Vol. 3646. Springer Verlag. (IDA 2005).*, pages 121-132.

M. Hu and B. Liu. 2004. Mining and summarizing customer reviews. *Proceedings of the 2004 ACM SIGKDD international conference.*

E. E. Jan, B. Maison, L. Mangu and G. Zweig. 2003. Automatic construction of Unique Signatures and Confusable sets for Natural Language Directory Assistance Application. *Eurospeech 2003*

C. A. Kamm, K. M. Yang, C. R. Shamieh and S. Singhal. 1994. Speech recognition issues for directory assistance applications. *Second IEEE Workshop on Interactive Voice Technology for Telecommunications Applications.*

E. Levin and A. M. Manš. 2005. Voice User Interface Design for Automated Directory Assistance *Eurospeech 2005.*

G. Linden, B. Smith and J. York. Amazon.com recommendations: item-to-item collaborative filtering. 2003. *Internet Computing, IEEE* , vol.7, no.1pp. 76- 80.

P. Natarajan, R. Prasad, R. Schwartz and J. Makhoul. 2002. A Scalable Architecture for Directory Assistance Automation, *ICASSP 2002*, Orlando, Florida.

Author Index