HLT-NAACL 2006

# Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics

## Demonstrations

Alex Rudnicky, John Dowding and Natasa Milic-Frayling
Demonstrations Chairs

June 4-9, 2006
New York City, USA

Published by the Association for Computational Linguistics
http://www.aclweb.org

# Table of Contents

# InfoMagnets: Making Sense of Corpus Data

**Jaime Arguello**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15216
jarguell@andrew.cmu.edu

**Carolyn Rosé**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15216
cprose@cs.cmu.edu

## Abstract

We introduce a new interactive corpus exploration tool called InfoMagnets. InfoMagnets aims at making exploratory corpus analysis accessible to researchers who are not experts in text mining. As evidence of its usefulness and usability, it has been used successfully in a research context to uncover relationships between language and behavioral patterns in two distinct domains: tutorial dialogue (Kumar et al., submitted) and on-line communities (Arguello et al., 2006). As an educational tool, it has been used as part of a unit on protocol analysis in an Educational Research Methods course.

## 1 Introduction

Exploring large text corpora can be a daunting prospect. This is especially the case for behavioral researchers who have a vested interest in the latent patterns present in text, but are less interested in computational models of text-representation (e.g. *the vector-space model*) or unsupervised pattern-learning (e.g. *clustering)*. Our goal is to provide this technology to the broader community of learning scientists and other behavioral researchers who collect and code corpus data as an important part of their research. To date none of the tools that are commonly used in the behavioral research community, such as HyperResearch, MacShapa, or Nvivo, which are used to support their corpus analysis efforts, make use of technology more advanced than simplistic word counting approaches. With InfoMagnets, we are working towards bridging the gap between the text-mining community and the corpus-based behavioral research community. The purpose of our demonstration is to make the language technologies community more aware of opportunities for applications of language technologies to support corpus oriented behavioral research.

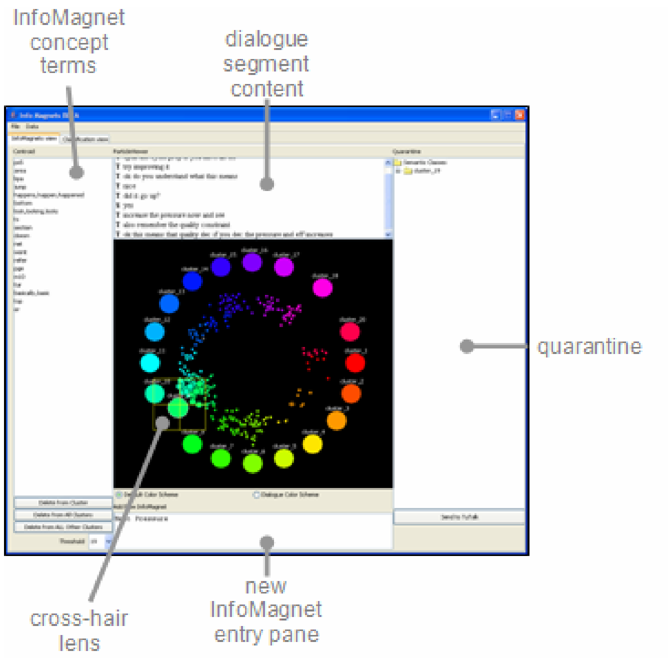


Figure 1: InfoMagnets Screenshot

InfoMagnet's novelty is two-fold: First, it provides an intuitive visual metaphor that allows the user to get a sense of their data and organize it for easy retrieval later. This is important during the sense making stage of corpus analysis work just before formal coding scheme development begins. Secondly, it allows the user to interact with clustering technology, and thus influence its behavior, in effect introducing human knowledge into the clustering process. Because of this give and take between the clustering technology and the human influence, the tool is able to achieve an organization of textual units that is not just optimal from an algorithmic stand-point, but also optimal for the user's unique purpose, which non-interactive clustering algorithms are not in general capable of achieving.

Using visual metaphors to convey to the user proximity and relations between documents and automatically generated clusters is not a new technique (Chalmers and Chitson, 1992; Dubin, 1995; Wise et al., 1995; Leuski and Allan, 2000; Rasmussen and Karypis, 2004). InfoMagnet's novelty comes from giving the user more control over the ultimate clustering organization. The user is able to incrementally influence the formation and reorganization of cluster centroids and *immediately* see the effect on the text-to-cluster assignment. Thus, the user can explore the corpus in more effective and meaningful ways.

In what follows, we more concretely elaborate on InfoMagnet's functionality and technical details. We then motivate its usability and usefulness with a real case study.

## 2 Functionality

Exploring a textual corpus in search of interesting topical patterns that correlate with externally observable variables is a non-trivial task. Take as an example the task of characterizing the process by which students and tutors negotiate with one another over a chat interface as they navigate instructional materials together in an on-line exploratory learning environment. A sensible approach is to segment all dialogue transcripts into topic-oriented segments and then group the segments by topic similarity. If done manually, this is a challenging task in two respects. First, to segment each dialogue the analyst must rely on their knowledge of the domain to locate where the focus of the dialogue shifts from one topic to the next. This, of course, requires the analyst to know what to look for and to remain consistent throughout the whole set of dialogues. More importantly, it introduces into the topic analysis a primacy bias. The analyst may miss important dialogue digressions simply because they are not expected based on observations from the first few dialogues viewed in detail. InfoMagnets addresses these issues by offering users a constant bird's eye view of their data. See Figure 1.

As input, InfoMagnets accepts a corpus of textual documents. As an option to the user, the documents can be automatically fragmented into topically-coherent segments (referred to also as *documents* from here on), which then become the atomic textual unit[1]. The documents (or topic segments) are automatically clustered into an initial organization that the user then incrementally adjusts through the interface. Figure 1 shows the initial document-to-topic assignment that InfoMagnets produces as a starting point for the user. The large circles represent InfoMagnets, or topic oriented cluster centroids, and the smaller circles represent documents. An InfoMagnet can be thought of as a set of words representative of a topic concept. The similarity between the vector representation of the words in a document and that of the words in an InfoMagnet translate into attraction in the two-dimensional InfoMagnet space. This semantic similarity is computed using Latent Semantic Analysis (LSA) (Landauer et al., 1998). Thus, a document appears closest to the InfoMagnet that best represents its topic.

A document that appears equidistant to two InfoMagnets shares its content equally between the two represented topics. Topics with lots of documents nearby are popular topics. InfoMagnets with only a few documents nearby represent infrequent topics. Should the user decide to remove an InfoMagnet, any document with some level of attraction to that InfoMagnet will animate and reposition itself based on the topics still represented by the remaining InfoMagnets. At all times, the InfoMagnets interface offers the analyst a bird's eye view of the entire corpus as it is being analyzed and organized.

Given the automatically-generated initial topic representation, the user typically starts by browsing the different InfoMagnets and documents. Using a magnifying cross-hair lens, the user can view the contents of a document on the top pane. As noted above, each InfoMagnet represents a topic concept through a collection of words (from the corpus) that convey that concept. Selecting the InfoMagnet displays this list of words on the left pane. The list is shown in descending order of importance with respect to that topic. By browsing each InfoMagnet's list of words and browsing

[1] Due to lack of space, we do not focus on our topic-segmentation algorithm. We intend to discuss this in the demo.

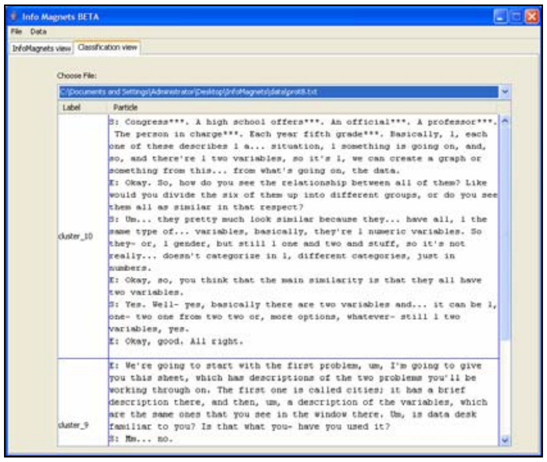
nearby documents, the user can start recognizing topics represented in the InfoMagnet space and can start labeling those InfoMagnets.

InfoMagnets with only a few neighboring documents can be removed. Likewise, InfoMagnets attracting too many topically-unrelated documents can be split into multiple topics. The user can do this semi-automatically (by requesting a split, and allowing the algorithm to determine where the best split is) or by manually selecting a set of terms from the InfoMagnet's word list and creating a new InfoMagnet using those words to represent the new InfoMagnet's topic. If the user finds words in an InfoMagnet's word list that lack topical relevance, the user can remove them from InfoMagnet's word list or from all the InfoMagnets' word lists at once.

Users may also choose to manually assign a segment to a topic by "snapping" that document to an InfoMagnet. "Snapping" is a way of overriding the attraction between the document and other InfoMagnets. By "snapping" a document to an InfoMagnet, the relationship between the "snapped" document and the associated InfoMagnet remains constant, regardless of any changes made to the InfoMagnet space subsequently.

If a user would like to remove the influence of a subset of the corpus from the behavior of the tool, the user may select an InfoMagnet and all the documents close to it and place them in the "quarantine" area of the interface. When placed in the quarantine, as when "snapped", a document's assignment remains unchanged. This feature is used to free screen space for the user.

If the user opts for segmenting each input discourse and working with topic segments rather than whole documents, an alternative interface allows the user to quickly browse through the corpus sequentially (Figure 2). By switching between this view and the bird's eye view, the user is able to see where each segment fits sequentially into the larger context of the discourse it was extracted from. The user can also use the sequential interface for making minor adjustments to topic segment boundaries and topic assignments where necessary. Once the user is satisfied with the topic representation in the space and the assignments of all documents to those topics, the tool can automatically generate an XML file, where all documents are tagged with their corresponding topic labels.



C:\Documents and Settings\Administrator\Desktop\InfoMagnets\data\prot8.txt

| Label | Particle |
|---|---|
| cluster_10 | S: Congress***. A high school offers***. An official***. A professor***. The person in charge***. Each year fifth grade***. Basically, 1, each one of these describes 1 a... situation, 1 something is going on, and, so, and there're 1 two variables, so it's 1, we can create a graph or something from this... from what's going on, the data. I: Okay. So, how do you see the relationship between all of them? Like would you divide the six of them up into different groups, or do you see them all as similar in that respect? S: Um... they pretty much look similar because they... have all, 1 the same type of... variables, basically, they're 1 numeric variables. So they- or, 1 gender, but still 1 one and two and stuff, so it's not really... doesn't categorize in 1, different categories, just in numbers. I: Okay, so, you think that the main similarity is that they all have two variables. S: Yes. Well- yes, basically there are two variables and... it can be 1, one- two one from two two or, more options, whatever- still 1 two variables, yes. I: Okay, good. All right. |
| cluster_9 | I: We're going to start with the first problem, um, I'm going to give you this sheet, which has descriptions of the two problems you'll be working through on. The first one is called cities; it has a brief description there, and then, um, a description of the variables, which are the same ones that you see in the window there. Um, is data desk familiar to you? Is that what you- have you used it? S: Mm... no. |

Figure 2. InfoMagnet's alternative sequential view

## 3 Implementation

As mentioned previously, InfoMagnets uses Latent Semantic Analysis (LSA) to relate documents to InfoMagnets. LSA is a dimensionality reduction technique that can be used to compute the semantic similarity between text spans of arbitrary size. For a more technical overview of LSA, we direct the reader to (Landauer et al., 1998).

The LSA space is constructed using the corpus that the user desires to organize, possibly augmented with some general purpose text (such as newsgroup data) to introduce more domain-general term associations. The parameters used in building the space are set by the user during pre-processing, so that the space is consistent with the semantic granularity the user is interested in capturing.

Because documents (or topic-segments) tend to cover more than one relevant topic, our clustering approach is based on what are determined heuristically to be the most important terms in the corpus, and not on whole documents. This higher granularity allows us to more precisely capture the topics discussed in the corpus by not imposing the assumption that documents are about a single topic. First, all terms that occur less than $n$ times and in less than $m$ documents are removed from consideration[2]. Then, the remaining terms are clustered via average-link clustering, using their LSA-based vector representations and using cosine-correlation as a vector similarity measure. Our clustering algorithm combines top-down clustering (Bisecting K-Means) and bottom-up clustering (Agglomerative Clustering) (Steinbach et al., 2000). This hybrid

[2] $n$ and $m$ are parameters set by the user.

clustering approach leverages the speed of bisecting K-means and the greedy search of agglomerative clustering, thus achieving a nice effectiveness versus efficiency balance.

Cluster centroids (InfoMagnets) and documents (or topic segments) are all treated as bag-of-words. Their vector-space representation is the sum of the LSA vectors of their constituent terms. When the user changes the topic-representation by removing or adding a term to an InfoMagnet, a new LSA vector is obtained by projecting the new bag-of-words onto the LSA space and re-computing the cosine correlation between all documents and the new topic.

## 4 An Example of Use

InfoMagnets was designed for easy usability by both computational linguistics and non-technical users. It has been successfully used by social psychologists working on on-line communities research as well as learning science researchers studying tutorial dialogue interactions (which we discuss in some detail here).

Using InfoMagnets, a thermodynamics domain expert constructed a topic analysis of a corpus of human tutoring dialogues collected during classroom study focusing on thermodynamics instruction (Rosé et al., 2005). Altogether each student's protocol was divided into between 10 and 25 segments such that the entire corpus was divided into approximately 379 topic segments altogether. Using InfoMagnets, the domain expert identified 15 distinct topics such that each student covered between 4 and 11 of these topics either once or multiple times throughout their interaction.

The topic analysis of the corpus gives us a way of quickly getting a sense of how tutors divided their instructional time between different topics of conversation. Based on this topic analysis of the human-tutoring corpus, the domain expert designed 12 dialogues, which were then implemented using a dialogue authoring environment called TuTalk (Gweon et al., 2005). In a recent very successful classroom evaluation, we observed the instructional effectiveness of these implemented tutorial dialogue agents, as measured by pre and post tests.

## Acknowledgments

This work was funded by Office of Naval Research, Cognitive and Neural Science Division, grant number N00014-05-1-0043.

## References

Jaime Arguello, Brian S. Butler, Lisa Joyce, Robert Kraut, Kimberly S. Ling, Carolyn Rose and Xiaoqing Wang (2006). Talk to Me: Foundations of Successful Individual-Group Interactions in Online Communities. *To appear in Proceedings of CHI: Human Factors in Computing*.

Matthew Chalmers and Paul Chitson (1992). Bead: Explorations in Information Visualization. *In Proceedings of ACM SIGIR,* 330-337

David Dubin (1995). Document Analysis for Visualization. *In Proceedings of ACM SIGIR*, 199-204.

Gahgene Gweon, Jaime Arguello, Carol Pai, Regan Carey, Zachary Zaiss, and Carolyn Rosé (2005). Towards a Prototyping Tool for Behavior Oriented Authoring of Conversational Interfaces, *Proceedings of the ACL Workshop on Educational Applications of NLP*.

Rohit Kumar, Carolyn Rosé, Vincent Aleven, Ana Iglesias, Allen Robinson (submitted). Evaluating the Effectiveness of Tutorial Dialogue Instruction in an Exploratory Learning Context, *Submitted to ITS '06*

Thomas Landauer, Peter W. Foltz, and Darrell Laham (1998). *Introduction to Latent Semantic Analysis*. Discourse Processes, 25, 259-284.

Anton Leuski and James Allan (2002). *Lighthouse: Showing the Way to Relevant Information*. In Proceedings of the IEEE InfoVis 2000

Matt Rasmussen and George Karypis (2004). *gCLUTO: An Interactive Clustering, Visualization, and Analysis System*. Technical Report # 04-021

Carolyn Rosé, Vincent Aleven, Regan Carey, Allen Robinson, and Chih Wu (2005). A First Evaluation of the Instructional Value of Negotiable Problem Solving Goals on the Exploratory Learning Continuum, *Proceedings of AI in Education '05*

Michael Steinbach, George Karypis, and Vipin Kuma (2000). A comparison of document clustering techniques. *In KDD Workshop on Text Mining*.

James A. Wise, James J. Thomas, Kelly Pennock, David Lantrip, Marc Pottier, and Anne Schur (1995). Visualizing the Non-Visual: Spatial Analysis and Interaction with Information from Text Documents. In Proceedings of IEEE InfoVis '95, 51-58.

# From Pipedreams to Products, and Promise!

**Janet M. Baker**
Saras Institute / Dibner Institute
MIT – Bldg E56-100
38 Memorial Drive
Cambridge, MA 02139 USA
HistSpch@mit.edu

**Patri J. Pugliese**
Saras Institute / Dibner Institute
MIT – Bldg E56-100
38 Memorial Drive
Cambridge, MA 02139 USA
HistSpch@mit.edu

## Abstract

This demonstration provides a historical perspective of a number of research and commercial systems in Spoken Language Technology over the past 20+ years. A series of chronologically ordered video clips from many sources will be presented to illustrate the many steps and the tremendous progress that has been achieved over the years. The clips themselves are drawn from diverse academic and commercial research labs, product presentations, and user applications. All show systems being demonstrated or in actual use. Over 20 different laboratory systems, products, and companies are represented in this collection of video materials. Each of the clips has previously been shown publicly. The present selection primarily focuses on speech and natural language systems for speech recognition and synthesis. Additional contributions to this collection are welcome.

## 1. Project Description

Preparations of these materials are being done in conjunction with the History of Speech and Language Technology Project being conducted by Saras Institute, in affiliation with the Dibner Institute for the History of Science and Technology, at MIT (Cambridge, MA). The overall mission for this project is to collect, preserve, and make readily available information about significant research discoveries, technical achievements, and business developments in speech and language technology. For further information on this project, please go to www.SarasInstitute.org.

Work on this project is on-going. Additional contributions of relevant materials are welcome in the area of Spoken Language Technology, including speech and natural language systems and applications incorporating speech recognition, speech synthesis, interactive dialogue, information retrieval, machine translation, multimodal interfaces, etc. Please contact us at HistSpch@mit.edu if you have materials you would like to contribute or with any inquiries, updates, corrections, and suggestions.

## 2. Introduction

This demonstration is intended to increase awareness and understanding of the Spoken Language Technology field, and an appreciation of the evolutionary and revolutionary steps which have turned pipedreams of the past into present products, and future promise. The progression of many stages from research and commercial laboratories into working systems are well illustrated by this collection of video clips. Each video clip (typically 1-5 minutes duration) represents technology at the cutting edge. There are 3 categories of video clips:

1) *Laboratory and Prototype Systems*
In some cases, pioneers and major contributors of the field are personally demonstrating their systems.

2) *Commercial Product Demonstrations*
These run the gamut of televised interviews/demonstrations (including many live demos) to instructional and commercial videos.

3) *User Applications*
Here users are demonstrating how they use their systems on a routine basis in a variety of diverse applications.

## 3. Hardware

With the advent of ever more powerful inexpensive silicon and the integration of computers with audio interfaces, the computing platforms on which spoken language technology resides have undergone dramatic changes, progressively reaching many more users, ever more conveniently. Over the past 30+ years, we have witnessed the transition from monolithic room-filling computers to personal, even hand-held devices supporting state-of-the-art speech technology. The type and scope of applications have concomitantly multiplied with the ready access of affordable useful technology. Like other initially expensive centralized hardware and even biological systems (!), as the cost curves come down, evolution progresses, and more processing can be conducted relatively or wholly autonomously, the processing itself can be far more distributed. So while there continue to be operations or services (e.g. call centers) which are still best done in a centralized fashion, the distribution and proliferation of stand-alone systems (e.g. PCs, and cell phones) become progressively more feasible and popular.

## 4. Speech Synthesis

The primary focus of the present set of demonstrations is on speech recognition and synthesis. Starting with Homer Dudley's Voder (a manually-controlled speech synthesizer) at the 1939 World's Fair, audio examples of historical speech synthesis approaches and techniques clearly demonstrate extensive progress (see Resources). Video clips illustrate users interacting with different types of synthesis systems and applications.

Mechanical speaking machines in the 1700's eventually gave way to electrical devices in the early 1920's, which in turn gave way to computer generation of speech by the 1960's. The invention of the speech spectrogram in the 1940's spurred in-depth speech research, and significantly facilitated speech signal and waveform analyses, which blossomed in the 1960's.

Speech generation has taken two basic forms. With analysis-resynthesis the speech waveform is first parameterized and then regenerated or played back, as in the Voder. With the advent of powerful inexpensive microprocessors and DSP chips in the 1970's, a multitude of diverse sound-producing consumer products hit the market. The popular Speak 'n Spell learning toy introduced in 1978, was the most notable early entry.

Speak and Spell, model 2
(Owned by J. M. Baker)

*Articulatory synthesis*, first demonstrated in 1958, models the components and the characteristics of the physical production system - the articulators, their movements and their trajectories, as well as the vocal tract, its resonances, excitations, etc. A clear understanding of such a system is highly desirable, and may eventually be achieved. Unfortunately, the underlying complexity of the speech production system still confounds understanding and application utility. Consequently, while studies in articulatory synthesis are still ongoing, they were largely superceded by formant synthesizers.

*Formant synthesis*, also referred to as synthesis-by-rule, characterizes speech in terms of a source-filter model. In this model, one or more sound sources, representing the vibrating vocal cords and noise dynamically produced at articulator constrictions, excite one or more filters, representing the vocal tract and side-tube (e.g. nasal branch, etc.) resonances. A catalog of sounds (corresponding to (sub) phones, diphones, or other units) can be constructed and then reassembled (and smoothed) in accordance with a dictionary of word or phrase pronunciations. With careful selection of materials, and the careful tuning and adjustments of parameters, synthetic speech can be made to sound very natural. Although computationally efficient, automatically achieving high quality output is neither easy nor consistently achievable.

*Concatenative synthesis* refers to the process of sequentially combining prerecorded exemplars of speech or other waveforms to produce the desired output. A large database (including many phonetic elements, allophones, words, etc.) of well-concatenated sound elements can easily produce synthetic speech indistinguishable from natural speech. This process is very memory-intensive, but typically produces the highest quality speech synthesis available. It is widely deployed in applications requiring natural-sounding speech output from a given voice.

Although three major approaches are outlined here, a number of hybrid synthesizers, HMM synthesizers, and other synthesis methodologies are utilized to address different requirements.

## 5. Speech Recognition

In the past century, speech recognition has progressed from recognizing small vocabularies to transcribing general purpose dictation in real time, recognizing commands in noisy environments, and reliably extracting words and information from telephone conversations and television broadcasts. In 1922, the toy dog "Rex", would spring from his doghouse when he was called by name! Early digit recognizers were demonstrated in the 1950's and 1960's, when the predominant approach was to recognize whole word templates. This approach continued up to the beginning of the 1970's when it started being gradually replaced by Hidden Markov Model (HMM) systems using stochastic models to more accurately characterize the naturally highly variable speech signal.

Radio Rex in his house (Photo by Hy Murveit, Rex owned by Michael Cohen)

Spurred on by government funders for "speech understanding" in Europe, Japan, and the USA, many university and commercial laboratories commenced to advance the technology. Proceeding first through limited vocabulary systems and highly constrained grammars, systems gradually expanded the number of words they could simultaneously distinguish, despite greater variability in speakers, languages, and progressively more challenging acoustic/channel and natural language environments. Starting in the 1970's, hefty special-purpose commercial hardware systems were deployed for limited vocabulary industrial applications (e.g. for hands-free command/control, data entry, quality control, etc.), speaker verification, simple telephone data input and query systems, etc. Inexpensive PC sound board enabling discrete recognition started appearing in the market by the late 1970's to mid 1980's.

In the latter 1980's, vocabulary expanded to several thousand words, and then in 1990, suddenly exploded to full general-purpose dictation capabilities, though still limited to *discrete* "one word at a time" input. Meanwhile in the early 1990's, the first speech audio-mining and audio information retrieval capabilities were successfully proven to work on prerecorded *conversational continuous* speech, on telephone and broadcast data. Real-time *continuous* speech dictation "software only" products became available and sold to millions of customers by the end of the 1990's. The availability of large corpora of recorded speech and text dramatically improved modeling capabilities and system performance for both dictation and transcription.

Meantime telephone query systems allowed for users to engage in dialogue to get stock quotes, weather updates, and even make train and plane reservations. By the year 2000, commercial telephone directory assistance systems started appearing as well. Today call centers routinely employ speech technology to elicit and supply customer information through interactive spoken dialogue, in large part replacing expensive human operators. Though directed dialogues predominate, some mixed initiative dialogues ("How can I help you?") are becoming available. Spoken language translation programs, coupling speech recognition to machine translation software, started appearing first in the laboratory in the late 1990's, and then progressed to the marketplace on PCs and handheld devices, by about 2000. Major government-funded research initiatives are presently focusing on speech-to-speech systems with different language inputs and outputs.

With embedded systems, speech input and output are now available on a growing number of consumer products including automotive navigation systems, PDAs and toys; hands-free voicedialing is shipping on millions of cell phones. Despite funding cuts and other setbacks about year 2000, the steady stream of ever improving speech technology is gradually becoming an integral part of systems and services, large and small. The issues we face in improving speech technology continue to be very challenging. The retrospective afforded by this demonstration reflects on the great progress that we have made!

## 6. Resources

The American Association for Artificial Intelligence (AAAI), founded in 1979, includes on their website an variety of materials on the history of speech technology. http://www.aaai.org/AITopics/html/speech.html#readon

Janet M. Baker, "Milestones in Speech Technology – Past and Future!", Speech Technology Magazine, September/October 2005.

The web site "comp.speech Frequently Asked Questions" provides a myriad of links to sites dealing with the various aspects of speech technology. http://www.speech.cs.cmu.edu/comp.speech/

Eurovamp (Voice Adapted Multipurpose Peripherals) maintains a web site which includes tutorials on the basic principals and history of speech recognition and synthesis. From their Home page, click on "Training" to get to the tutorial menu. http://www.eurovamp.com/

IEEE History Center has a website on Automatic Speech Synthesis & Recognition which includes interviews with seminal contributors (under "Archives") to speech technology an other relevant materials. http://www.ieee.org/organizations/history_center/sloan/ASSR/assr_index.html

Saras Institute has a website for the History of Speech and Language Technology with information on historical artifacts, institutions, major contributors, resources, etc. http://www.SarasInstitute.org

The Smithsonian Speech Synthesis History Project (SSSHP) provides a collection of tape recordings, technical records and artifacts of speech synthesis technology from 1922 to the mid-1980s. http://www.mindspring.com/~ssshp/ssshp_cd/ss_home.htm

Special Workshop in Maui (SWIM): Lectures by Masters in Speech Processing, January 11-14, 2004, included a series of papers by senior researchers on various aspects of the history of speech technology. A "Program Guide" with summaries of these "Peer Lectures" is available at: http://dspincars.sdsu.edu/swim/WorkshopGuide.pdf

"The Typewriter one hundred years hence": cartoon from *The Illustrated Phonographic World* (June, 1984).

# SmartNotes: Implicit Labeling of Meeting Data through User Note–Taking and Browsing

**Satanjeev Banerjee**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
banerjee@cs.cmu.edu

**Alexander I. Rudnicky**
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
air@cs.cmu.edu

## Abstract

We have implemented SmartNotes, a system that automatically acquires labeled meeting data as users take notes during meetings and browse the notes afterwards. Such data can enable meeting understanding components such as topic and action item detectors to automatically improve their performance over a sequence of meetings. The SmartNotes system consists of a laptop based note taking application, and a web based note retrieval system. We shall demonstrate the functionalities of this system, and will also demonstrate the labeled data obtained during typical meetings and browsing sessions.

## 1 Goals of the *SmartNotes* System

Most institutions hold a large number of meetings every day. Several of these meetings are important, and meeting participants need to recall the details of the discussions at a future date. In a previous survey (Banerjee et al., 2005) of busy professors at Carnegie Mellon University we showed that meeting participants needed to recall details of past meetings on average about twice a month. Performing such retrieval is not an easy task. It is time consuming; in our study participants took on average between 15 minutes to an hour to recall the information they were seeking. Further, the quality of the retrieval is dependent on whether or not the participants had access to the notes at the meeting. On a scale of 0 to 5, with 5 denoting complete satisfaction with retrieval results, participants reported a satisfaction of 3.4 when they did not have notes, and 4.0 when they did.

Despite the prevalence of important meetings and the importance of notes, there is a relative paucity of technology to help meeting participants take notes easily at meetings. Some commercial applications allow users to take notes (e.g. OneNote[1]) and even record audio/video (e.g. Quindi[2]), but no product attempts to automatically take notes. Our **long term goal** is to create a system that makes note–taking easier by performing tasks such as automatically highlighting portions of the meeting that are likely to be important to the user, automatically detecting "note–worthy" phrases spoken during the meeting, etc.

To perform such note taking, the system needs to form an understanding of the meeting. Our **short term goal** is to create a system that can detect the topics of discussion, the action items being discussed, and the roles of the meeting participants. Additionally, these components must adapt to specific users and groups of users since different people will likely take different notes at the same meeting. Thus we wish to implement the note taking system in such a way that the user's interactions with the system result in *labeled meeting data* that can then be used to adapt and improve the meeting understanding components.

Towards these goals, we have built *SmartNotes* which helps users easily record and retrieve notes.

[1]http://office.microsoft.com/onenote
[2]http://www.quindi.com

The system also records the user interactions to form labeled meeting data that can later be used to automatically improve the meeting understanding components. In the next section we describe the meeting understanding components in more detail. Next we describe SmartNotes itself, and show how it is currently helping users take and retrieve notes, while acquiring labeled data to aid each of the meeting understanding components. Finally we end with a discussion of what functionality we plan to demonstrate at the conference.

## 2 Automatic Meeting Understanding

**Topic detection and segmentation:** We are attempting to automatically detect the topics being discussed at meetings. This task consists of two subtasks: discovering the points in a meeting when the topic changes, and then associating a descriptive *label* to the segment between two topic shifts. Our current strategy for topic shift detection (Banerjee and Rudnicky, 2006a) is to perform an edge detection using such features as speech activity (who spoke when and for how long), the words that each person spoke, etc. For labeling, we are currently simply associating the agenda item names recorded in the notes with the segments they are most relevant to, as decided by a tf.idf matching technique. Topic detection is particularly useful during meeting information retrieval; (Banerjee et al., 2005) showed that when users wish to retrieve information from past meetings, they are typically interested in a specific discussion topic, as opposed to an entire meeting.

**Action item detection:** An obvious application of meeting understanding is the automatic discovery and recording of action items as they are discussed during a meeting. Arguably one of the most important outcomes of a meeting are the action items decided upon, and automatically recording them could be a huge benefit especially to those participants that are likely to not note them down and consequently forget about them later on.

**Meeting participant role detection**: Each meeting participant plays a variety of roles in an institution. These roles can be based on their function in the institution (managers, assistants, professors, students, etc), or based on their expertise (speech recognition experts, facilities experts, etc). Our current strategy for role detection (Banerjee and Rudnicky, 2006b) is to train detectors on hand labeled data. Our next step is to perform discovery of new roles through clustering techniques. Detecting such roles has several benefits. First, it allows us to build prior expectations of a meeting between a group of participants. For example, if we know person A is a speech recognition expert and person B a speech synthesis expert, a reasonable expectation is that when they meet they are likely to talk about technologies related speech processing. Consequently, we can use this expectation to aid the action item detection and the topic detection in that meeting.

## 3 SmartNotes: System Description

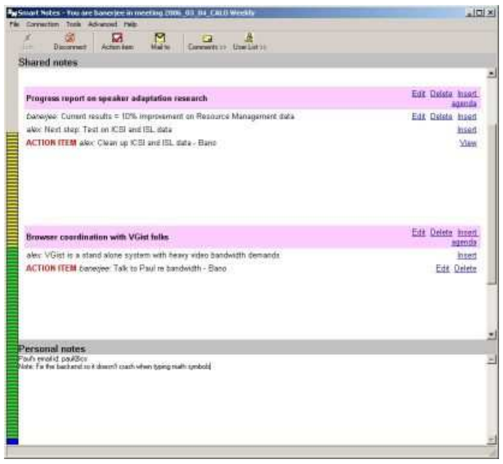
We have implemented SmartNotes to help users take *multi–media notes* during meetings, and retrieve them later on. SmartNotes consists of two major components: The note taking application which meeting participants use to take notes during the meeting, and the note retrieval application which users use to retrieve notes at a later point.

### 3.1 SmartNotes Note Taking Application

The note taking application is a stand–alone system, that runs on each meeting participant's laptop, and allows him to take notes during the meeting. In addition to recording the text notes, it also records the participant's speech, and video, if a video camera is connected to the laptop. This system is an extension of the Carnegie Mellon Meeting Recorder (Banerjee et al., 2004).

Figure 1 shows a screen–shot of this application. It is a server–client application, and each participant logs into a central server at the beginning of each meeting. Thus, the system knows the precise identity of each note taker as well as each speaker in the meeting. This allows us to avoid the onerous problem of automatically detecting who is speaking at any time during the meeting. Further, after logging on, each client automatically synchronizes itself with a central NTP time server. Thus the time stamps that each client associates with its recordings are all synchronized, to facilitate merging and play back of audio/video during browsing (described in the next sub–section).

Once logged in, each participant's note taking


Smart Notes - You are banerjee in meeting 2006_03_04_CALO Weekly
File Connection Tools Advanced Help
Disconnect
Action item
Mail to
Comments >>
User List >>
Shared notes
Edit Delete Insert agenda
Edit Delete Insert
Insert
View
Edit Delete Insert agenda
Insert
Edit Delete
Personal notes

Progress report on speaker adaptation research

banerjee: Current results = 10% improvement on Resource Management data

alex: Next step: Test on ICSI and ISL data

ACTION ITEM alex: Clean up ICSI and ISL data - Banо

Browser coordination with VGist folks

alex: VGist is a stand alone system with heavy video bandwidth demands

ACTION ITEM banerjee: Talk to Paul re bandwidth - Banо

Paul's email id: paul@cs

Note: Fix the backend so it doesn't crash when typing math symbols

Figure 1: *Screen shot of the SmartNotes note–taking client*

area is split into two sections: a *shared* note taking area, and a *private* note taking area. Notes written in the shared area are viewable by all meeting participants. This allows meeting participants to share the task of taking notes during a meeting: As long as one participant has recorded an important point during a meeting, the other participants do not need to, thus making the note taking task easier for the group as a whole. Private notes that a participant does not wish to share with all participants can be taken in the private note taking area.

The interface has a mechanism to allow meeting participants to insert an agenda into the shared area. Once inserted, the shared area is split into as many boxes as there are agenda items. Participants can then take notes during the discussion of an agenda item in the corresponding agenda item box. This is useful to the participants because it organizes the notes as they are being taken, and, additionally, the notes can later be retrieved agenda item by agenda item. Thus, the user can access all notes he has taken in different meetings regarding "buying a printer", without having to see the notes taken for the other agenda items in each such meeting.

In addition to being useful to the user, this act of inserting an agenda and then taking notes within the relevant agenda item box results in generating (unbeknownst to the participant) **labeled data for the topic detection component**. Specifically, if we define each agenda item as being a separate "topic", and make the assumption that notes are taken approximately concurrent with the discussion of the contents of the notes, then we can conclude that there is a shift in the topic of discussion at some point between the time stamp on the last note in an agenda item box, and the time stamp on the first note of the next agenda item box. This information can then be used to improve the performance of the topic shift detector. The accuracy of the topic shift data thus acquired depends on the length of time between the two time points. Since this length is easy to calculate automatically, this information can be factored into the topic detector trainer.

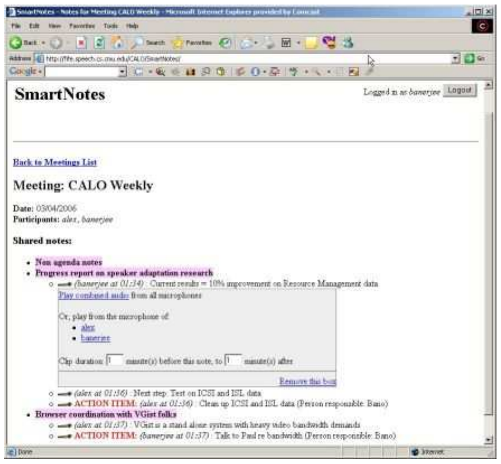
The interface also allows participants to enter action items through a dedicated action item form. Again the advantage of such a form to the participants is that the action items (and thus the notes) are better organized: After the meeting, they can perform retrieval on specific fields of the action items. For example, they can ask to retrieve all the action items assigned to a particular participant, or that are due a particular day, etc.

In addition to being beneficial to the participant, the action item form filling action results in generating **labeled data for the action item detector**. Specifically, if we make the assumption that an action item form filling action is preceded by a discussion of the action item, then the system can couple the contents of the form with all the speech within a window of time before the form filling action, and use this pair as a data point to retrain its action item detector.

## 3.2 SmartNotes Note Retrieval Website

As notes and audio/video are recorded on each individual participant's laptop, they also get transferred over the internet to a central meeting server. This transfer occurs in the background without any intervention from the user, utilizes only the left–over bandwidth beyond the user's current bandwidth usage, and is robust to system shut–downs, crashes, etc. This process is described in more detail in (Banerjee et al., 2004).

Once the meeting is over and all the data has been transferred to the central server, meeting participants can use the SmartNotes multi–media notes retrieval system to view the notes and access the recorded audio/video. This is a web–based application that uses the same login process as the stand–along note



SmartNotes

Logged in as *banerjee* Logout

Back to Meetings List

Meeting: CALO Weekly

**Date:** 03/04/2006
**Participants:** *alex*, *banerjee*

**Shared notes:**

- **Non agenda notes**
- **Progress report on speaker adaptation research**
  - *(banerjee at 01:34)* : Current results = 10% improvement on Resource Management data
    Play combined audio from all microphones
    Or, play from the microphone of:
    - alex
    - banerjee
    Clip duration: 1 minute(s) before this note, to 1 minute(s) after
    Remove this box
  - *(alex at 01:36)* : Next step: Test on ICSI and ISL data
  - ACTION ITEM: *(alex at 01:36)* : Clean up ICSI and ISL data (Person responsible: Banerjee)
- **Browser coordination with VGist folks**
  - *(alex at 01:37)* : VGist is a stand alone system with heavy video bandwidth demands
  - ACTION ITEM: *(banerjee at 01:37)* : Talk to Paul re bandwidth (Person responsible: Banerjee)

Figure 2: *Screen shot of the SmartNotes website*

taking system. Users can view a list of meetings they have recorded using the SmartNotes application in the past, and then for each meeting, they can view the shared notes taken at the meeting. Figure 2 shows a screen shot of such a notes browsing session. Additionally, participants can view their own private notes taken during the meeting.

In addition to viewing the notes, they can also access all recorded audio/video, indexed by the notes. That is, they can access the audio/video recorded around the time that the note was entered. Further they can specify how many minutes before and after the note they wish to access. Since the server has the audio from each meeting participant's audio channel, the viewer of the notes can choose to listen to any one person's channel, or a combination of the audio channels. The merging of channels is done in real time and is achievable because their time stamps have been synchronized during recording.

In the immediate future we plan to implement a simple key–word based search on the notes recorded in all the recorded meetings (or in one specific meeting). This search will return notes that match the search using a standard tf.idf approach. The user will also be provided the option of rating the quality of the search retrieval on a one bit satisfied/not–satisfied scale. If the user chooses to provide this rating, it can be used as a feedback to improve the search. Additionally, which parts of the meeting the user chooses to access the audio/video from can be used to form a model of the parts of the meetings most relevant to the user. This information can help the system tailor its retrieval to individual preferences.

## 4 The Demonstration

We shall demonstrate both the SmartNotes note taking client as well as the SmartNotes note–retrieval website. Specifically we will perform 2 minute long mock meetings between 2 or 3 demonstrators. We will show how notes can be taken, how agendas can be created and action items noted. We will then show how the notes and the audio/video from the 2 minute meeting can be accessed through the SmartNotes note retrieval website. We shall also show the automatically labeled data that gets created both during the mock meeting, as well as during the browsing session. Finally, if time permits, we shall show results on how much we can improve the meeting understanding components' capabilities through labeled meeting data automatically acquired through participants' use of SmartNotes at CMU and other institutions that are currently using the system.

## References

S. Banerjee and A. I. Rudnicky. 2006a. A texttiling based approach to topic boundary detection in multi–participant conversations. Submitted for publication.

S. Banerjee and A. I. Rudnicky. 2006b. You are what you say: Using meeting participants' speech to detect their roles and expertise. In *Analyzing Conversations in Text and Speech Workshop at HLT–NAACL 2006*, New York City, USA, June.

S. Banerjee, J. Cohen, T. Quisel, A. Chan, Y. Patodia, Z. Al-Bawab, R. Zhang, P. Rybski, M. Veloso, A. Black, R. Stern, R. Rosenfeld, and A. I. Rudnicky. 2004. Creating multi-modal, user–centric records of meetings with the Carnegie Mellon meeting recorder architecture. In *Proceedings of the ICASSP Meeting Recognition Workshop*, Montreal, Canada.

S. Banerjee, C. Rose, and A. I. Rudnicky. 2005. The necessity of a meeting recording and playback system, and the benefit of topic–level annotations to meeting browsing. In *Proceedings of the Tenth International Conference on Human-Computer Interaction*, Rome, Italy, September.

# MTTK: An Alignment Toolkit for Statistical Machine Translation

**Yonggang Deng** [1]
Center for Language and Speech Processing[1]
Johns Hopkins University
Baltimore, MD 21218
dengyg@jhu.edu

**William Byrne**[1,2]
Machine Intelligence Lab [2]
Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, UK
wjb31@cam.ac.uk

## Abstract

The MTTK alignment toolkit for statistical machine translation can be used for word, phrase, and sentence alignment of parallel documents. It is designed mainly for building statistical machine translation systems, but can be exploited in other multi-lingual applications. It provides computationally efficient alignment and estimation procedures that can be used for the unsupervised alignment of parallel text collections in a language independent fashion. MTTK Version 1.0 is available under the Open Source Educational Community License.

## 1 Introduction

Parallel text alignment procedures attempt to identify translation equivalences within collections of translated documents. This can be be done at various levels. At the finest level, this involves the alignment of words and phrases within two sentences that are known to be translations (Brown et al., 1993; Och and Ney, 2003; Vogel et al., 1996; Deng and Byrne, 2005). Another task is the identification and alignment of sentence-level segments within document pairs that are known to be translations (Gale and Church, 1991); this is referred to as sentence-level alignment, although it may also involve the alignment of sub-sentential segments (Deng et al., ) as well as the identification of long segments in either document which are not translations. There is also document level translation which involves the identification of translated document pairs in a collection of documents in multiple languages. As an example, Figure 1 shows parallel Chinese/English text that is aligned at the sentence, word, and phrase levels.

Parallel text plays a crucial role in multi-lingual natural language processing research. In particular, statistical machine translation systems require collections of sentence pairs (or sentence fragment pairs) as the basic ingredients for building statistical word and phrase alignment models. However, with the increasing availability of parallel text, human-created alignments are expensive and often unaffordable for practical systems, even at a small scale. High quality automatic alignment of parallel text has therefore become indispensable. In addition to good alignment quality, several other properties are also desirable in automatic alignment systems. Ideally, these should be general-purpose and language independent, capable of aligning very different languages, such as English, French, Chinese, German and Arabic, to give a few examples of current interest. If the alignment system is based on statistical models, the model parameters should be estimated from scratch, in an unsupervised manner from whatever parallel text is available. To process millions of sentence pairs, these models need to be capable of generalization and the alignment and estimation algorithms should be computationally efficient. Finally, since noisy mismatched text is often found in real data, such as parallel text mined from web pages, automatic alignment needs to be robust. There are systems available for these purposes, notably the GIZA++ (Och and Ney, 2003) toolkit and

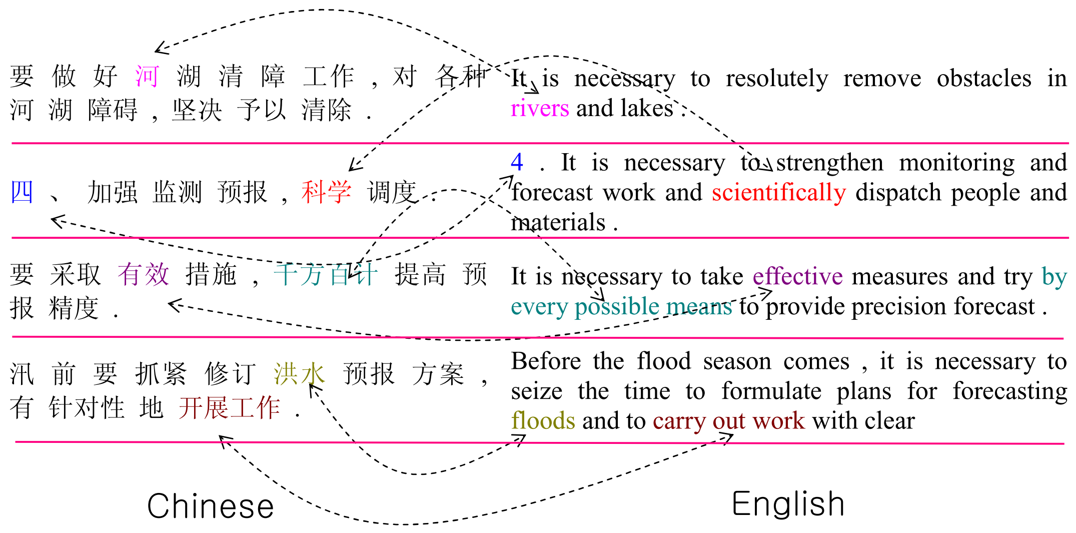

要 做 好 河 湖 清 障 工作 ， 对 各种 河 湖 障碍 ， 坚决 予以 清除 .
It is necessary to resolutely remove obstacles in rivers and lakes .
四 、 加强 监测 预报 ， 科学 调度 .
4 . It is necessary to strengthen monitoring and forecast work and scientifically dispatch people and materials .
要 采取 有效 措施 ， 千方百计 提高 预报 精度 .
It is necessary to take effective measures and try by every possible means to provide precision forecast .
汛 前 要 抓紧 修订 洪水 预报 方案 ， 有 针对性 地 开展工作 .
Before the flood season comes , it is necessary to seize the time to formulate plans for forecasting floods and to carry out work with clear
Chinese
English

Figure 1: Chinese/English Parallel Corpus Aligned at the Sentence, Word, and Phrase Levels: horizontal lines denote the segmentations of a sentence alignment and arrows denote a word-level mapping.

the Champollion Toolkit (Ma et al., 2004).

This demo introduces MTTK, the Machine Translation Toolkit. The toolkit can be used to train statistical models and perform parallel text alignment at different levels. Target applications include not only machine translation, but also bilingual lexicon induction, cross lingual information retrieval and other multi-lingual applications.

## 2 MTTK Components

MTTK is a collection of C++ programs and Perl and shell scripts that can be used to build statistical alignment models from parallel text. Respective of the text to be aligned, MTTK's functions are categorized into the following two main parts.

### 2.1 Chunk Alignment

Chunk alignment aims to extract sentence or sub-sentence pairs from parallel corpora. A chunk can be multiple sentences, a sentence or a sub-sentence, as required by the application. Two alignment procedures are implemented: one is the widely used dynamic programming procedure that derives monotone alignment of sentence segments (Gale and Church, 1991); the other is divisive clustering procedure that begins by finding coarse alignments that are then iteratively refined by successive binary splitting (Deng et al., ). These two types of alignment procedures complement each other. They can be used together to improve the overall sentence alignment quality.

When translation lexicons are not available, chunk alignment can be performed using length-based statistics. This usually can serve as a starting point of sentence alignment. Alignment quality can be further improved when the chunking procedure is based on translation lexicons from IBM Model-1 alignment model (Brown et al., 1993). The MTTK toolkit also generates alignment score for each chunk pair, that can be utilized in post processing, for example in filtering out aligned segments of dubious quality.

### 2.2 Word and Phrase Alignment

After a collection of sentence or sub-sentence pairs are extracted via chunk alignment procedures, statistical word and phrase alignment models can be estimated with EM algorithms. MTTK provides implementations of various alignment, models including IBM Model-1, Model-2 (Brown et al., 1993), HMM-based word-to-word alignment model (Vogel et al., 1996; Och and Ney, 2003) and HMM-based word-to-phrase alignment model (Deng and Byrne, 2005). After model parameters are estimated, the Viterbi word alignments can be derived. A novel computation performed by MTTK is the generation of model-based phrase pair posterior distributions (Deng and Byrne, 2005), which plays an important role in extracting a phrase-to-phrase translation probabilities.

## 3 MTTK Features

MTTK is designed to process huge amounts of parallel text. Model parameter estimation can be carried out parallel during EM training using multiple CPUs. The entire parallel text is split into parts. During each E-step , statistics are collected parallel over each part, while in the M-steps, these statistics are merged together to update model parameters for next iteration. This parallel implementation not only reduces model training time significantly, it also avoids memory usage issues that arise in processing millions of sentence pairs, since each E-Step need only save and process co-occurrence that appears in its part of the parallel text. This enables building a single model from many millions of sentence pairs.

Another feature of MTTK is language independence. Linguistic knowledge is not required during model training, although when it is available, performance can be improved. Statistical parameters are estimated and learned automatically from data in an unsupervised way. To accommodate language diversity, there are several parameters in MTTK that can be tuned for individual applications to optimize performance.

## 4 A Typical Application of MTTK in Parallel Text Alignment

A typical example of using MTTK is give in Figure 2. It starts with a collection of document pairs. During pre-processing, documents are normalized and tokenized into token sequences. This preprocessing is carried out before using the MTTK, and is usually language dependent, requiring, for example, segmenting Chinese characters into words or applying morphological analyzing to Arabic word sequences.

Statistical models are then built from scratch. Chunk alignment begins with length statistics that can be simply obtained by counting the number of tokens on in each language. The chunk aligning procedure then applies dynamic programming to derive a sentence alignment. After sorting the generated sentence pairs by their probabilities, high quality sentence pairs are then selected and used to train a translation lexicon. As an input for next round chunk alignment, more and better sentence pairs can be extracted and serve as training material for a better translation lexicon. This bootstrapping procedure identifies high quality sentence pairs in an iterative fashion.

To maximize the number of training words for building word and phrase alignment models, long sentence pairs are then processed further using a divisive clustering chunk procedure that derives chunk pairs at the sub-sentence level. This provides additional translation training pairs that would otherwise be discarded as being overly long.

Once all usable chunk pairs are identified in the chunk alignment procedure, word alignment model training starts with IBM Model-1. Model complexity increases gradually to Model-2, and then HMM-based word-to-word alignment model, and finally to HMM-based word-to-phrase alignment model (Deng and Byrne, 2005). With these models, word alignments can be obtained using the Viterbi algorithm, and phrase pair posterior distributions can be computed in building a phrase translation table.

In published experiments we have found that MTTK generates alignments of quality comparable to those generated by GIZA++, where alignment quality is measured both directly in terms of Alignment Error Rate relative to human word alignments and indirectly through the translation performance of systems constructed from the alignments (Deng and Byrne, 2005). We have used MTTK as the basis of translation systems entered into the recent NIST Arabic-English and Chinese-English MT Evaluations as well as the TC-STAR Chinese-English MT evaluation (NIST, 2005; TC-STAR, 2005).

## 5 Availability

MTTK Version 1.0 is released under the Open Source Educational Community License[1]. The tools and documentation are available at http://mi.eng.cam.ac.uk/∼wjb31/distrib/mttkv1/ .

[1]http://www.opensource.org/licenses/ecl1.php

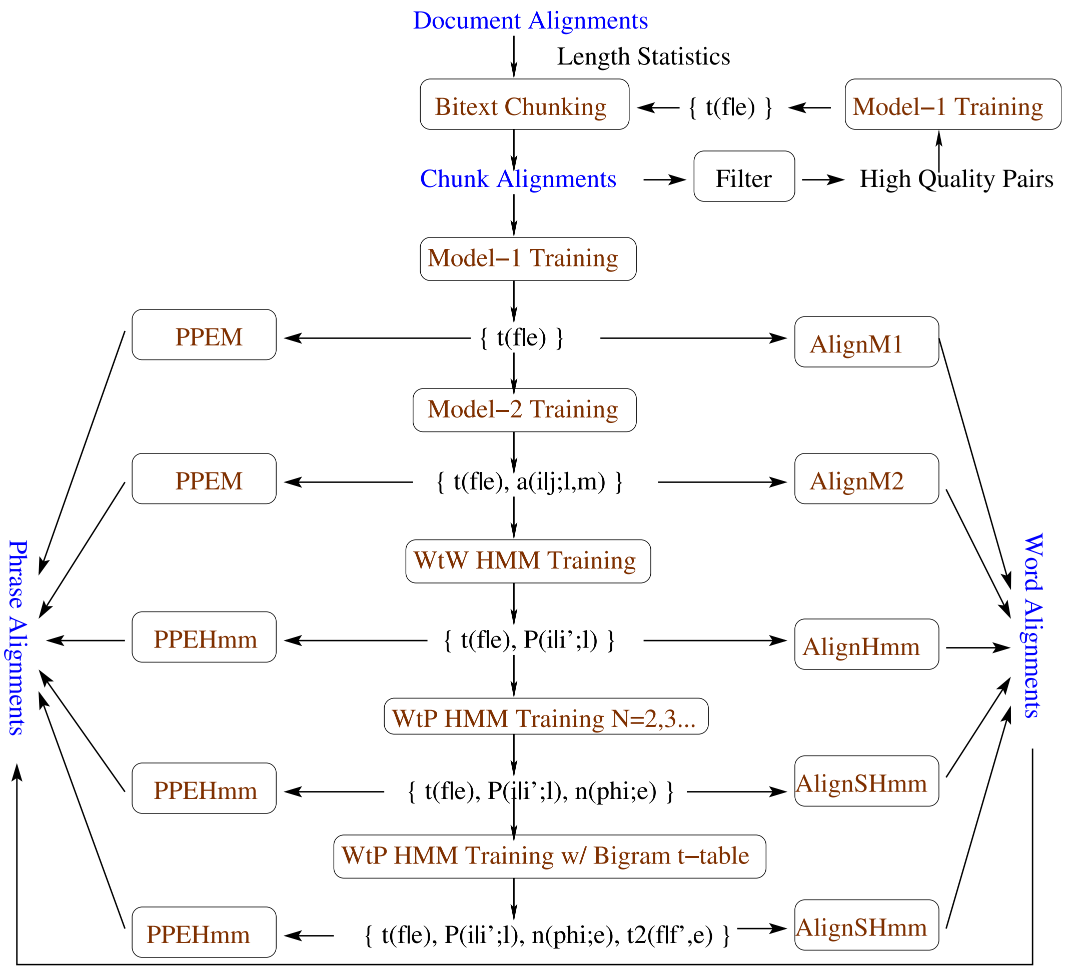


Figure 2: A Typical Unsupervised Translation Alignment Procedure with MTTK.

## 6 Acknowledgements

Funded by ONR MURI Grant N00014-01-1-0685.

## References

P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. 1993. The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 19:263–312.

Y. Deng and W. Byrne. 2005. Hmm word and phrase alignment for statistical machine translation. In *Proc. of HLT-EMNLP*.

Y. Deng, S. Kumar, and W. Byrne. Segmentation and alignment of parallel text for statistical machine translation. *Journal of Natural Language Engineering*. to appear.

W. A. Gale and K. W. Church. 1991. A program for aligning sentences in bilingual corpora. In *Meeting of the Association for Computational Linguistics*, pages 177–184.

X. Ma, C. Cieri, and D. Miller. 2004. Corpora & tools for machine translation. In *Machine Translation Evaluation Workshop*, Alexandria, VA. NIST.

NIST, 2005. *The NIST Machine Translation Evaluations Workshop*. North Bethesda, MD, June. http://www.nist.gov/speech/tests/summaries/2005/mt05.htm.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

TC-STAR, 2005. *TC-STAR Speech-to-Speech Translation Evaluation Meeting*. Trento, Italy, April. http://www.tc-star.org/.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM based word alignment in statistical translation. In *Proc. of the COLING*.

# AquaLog: An ontology-driven Question Answering System to interface the Semantic Web

**Vanessa Lopez Garcia**
Knowledge Media Institute
The Open University.
Walton Hall, Milton Keynes,
MK7 6AA United Kingdom.
v.lopez@open.ac.uk

**Enrico Motta**
Knowledge Media Institute
The Open University.
Walton Hall, Milton Keynes,
MK7 6AA United Kingdom.
e.motta@open.ac.uk

**Victoria Uren**
Knowledge Media Institute
The Open University.
Walton Hall, Milton Keynes,
MK7 6AA United Kingdom.
v.s.uren@open.ac.uk

## Abstract

The *semantic web* (SW) vision is one in which rich, *ontology-based semantic markup* will become widely available. The availability of semantic markup on the web opens the way to novel, sophisticated forms of question answering. AquaLog is a portable question-answering system which takes queries expressed in natural language (NL) and an ontology as input, and returns answers drawn from one or more knowledge bases (KB). AquaLog presents an elegant solution in which different strategies are combined together in a novel way. AquaLog novel ontology-based *relation similarity service* makes sense of user queries.

## 1 Introduction

AquaLog (Lopez, 2005) is a fully implemented ontology-driven Question Answering (QA) system[1], which takes an ontology and a NL query as an input and returns answers drawn from semantic markup (viewed as a KB) compliant with the input ontology. In contrast with much existing work on ontology-driven QA, which tends to focus on the use of ontologies to support query expansion in information retrieval (Mc Guinness, 2004), AquaLog exploits the availability of semantic statements to provide precise answers to complex queries expressed in NL.

AquaLog makes use of the GATE NLP platform, string distance metric, generic lexical resources, such as WordNet, as well as the structure of the input ontology, to make sense of the terms and relations expressed in the input query with respect to the target KB. Naturally, these terms and relations match the terminology familiar to the user rather than those used in the ontology.

We say that AquaLog is portable because the configuration time required to customize the system for a particular ontology is negligible. We believe that in the SW scenario it makes sense to provide a NL query interface *portable with respect to ontologies*, our AquaLog system allows to choose an ontology and then ask queries with respect to its universe of discourse. The reason for this is that the architecture of the system and the reasoning methods are domain-independent, relying on an understanding of general-purpose knowledge representation languages, such as OWL[2], and the use of generic lexical resources, such as WordNet.

Moreover, AquaLog learning mechanism ensures that, for a given ontology and a particular community jargon used by end users, its performance improves over time, as the users can easily correct mistakes and allow AquaLog to learn novel associations between the NL relations used by users and the ontology structure.

**Approach.** AquaLog uses a sequential process model (see Fig. 1), in which NL input is first translated into a set of intermediate representations – called *Query Triples*, by the Linguistic Component. The Linguistic Component uses the GATE infrastructure and resources (Cunningham, 2002) to obtain a set of syntactic annotations associated with the input query and to classify the query. Once this is done, it becomes straight-forward for the Linguistic Component to automatically create the Query-Triples. Then, these query triples are

[1] AquaLog is available as Open Source https://sourceforge.net/projects/aqualog

[2] A plug-in mechanism and a generic API ensure that different Knowledge Representation languages can be used.

further processed and interpreted by the Relation Similarity Service Component (RSS), which uses lexical resources and the ontology to map them to ontology-compliant semantic markup or triples.

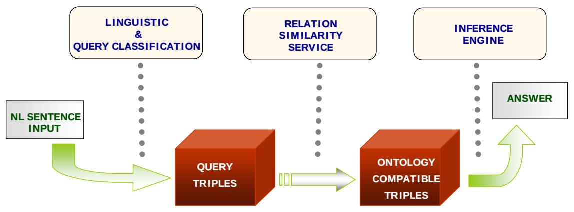

LINGUISTIC & QUERY CLASSIFICATION
RELATION SIMILARITY SERVICE
INFERENCE ENGINE
NL SENTENCE INPUT
QUERY TRIPLES
ONTOLOGY COMPATIBLE TRIPLES
ANSWER

**Fig. 1**. AquaLog data model

## 2 Linguistic Component

The Linguistic Component's task is to map the NL input query to the Query-Triple. GATE (Cunningham, 2002) infrastructure and resources (e.g. processing resources like ANNIE) are part of the Linguistic Component.

After the execution of the GATE controller, a set of syntactic annotations associated with the input query are returned. These annotations include information about sentences, tokens, nouns and verbs. For example, we get voice and tense for the verbs and categories for the nouns, such as determinant, singular/plural, conjunction, possessive, determiner, preposition, existential, wh-determiner, etc. When developing AquaLog we extended the set of annotations returned by GATE, by identifying terms, relations, question indicators (which/who/when. etc.) and patterns or types of questions. This is achieved through the use of *Jape grammars*, which consist of a set of *phases*, that run sequentially, and each phase is defined as a set of pattern rules, which allow us to recognize regular expressions using previous annotations in documents.

Thanks to this architecture that takes advantage of the *Jape grammars*, although we can still only deal with a subset of NL, it is possible to extend this subset in a relatively easy way by updating the regular expressions in the *Jape grammars*. This ensures the easy portability of the system with respect to both ontologies and natural languages. Currently, the linguistic component, through the *Jape grammars*, dynamically identifies around 14 different linguistic categories or intermediate representations, including: basic queries requiring an affirmation/negation or a description as an answer; or the big set of queries constituted by a wh-question (such as the ones starting with: what, who, when, where, are there any, does anybody/anyone or how many, and imperative commands like list, give, tell, name, etc.), like "are there any PhD students in dotkom?" where the relation is implicit or unknown or "which is the job title of John?" where no information about the type of the expected answer is provided; etc.

Categories tell us not only the kind of solution that needs to be achieved, but also they give an indication of the most likely common problems that the system will need to deal with to understand this particular NL query and in consequence it guides the process of creating the equivalent intermediate representation. Categories are the driving force to generate an answer by combining the triples in an appropriate way. For example, in "who are the academics involved in the semantic web?" the triple will be of the form <generic term, relation, second term>, i.e. <academics, involved, semantic web>. A query with a equivalent triple representation is "which technologies has KMi produced?", where the triple will be <technologies, has produced, KMi>. However, a query like "are there any PhD students in akt?" has another equivalent representation, where the relation is implicit or unknown <phd students, ?, akt> . Other queries may provide little information about the type of the expected answer, i.e. "what is the job title of John?", or they can be just a generic enquiry about someone or something, i.e. "who is Vanessa?", "what is an ontology?"

At this stage we do not have to worry about getting the representation completely right as the interpretation is completely domain independent. The role of the triple-based intermediate representation is simply to provide an easy way to represent the NL query and to manipulate the input for the RSS. Consider the request "List all the projects in the knowledge media institute about the semantic web", where both "in knowledge media institute" and "about semantic web" are modifiers (i.e. they modify the meaning of other syntactic constituents). The problem here is to identify the constituent to which each modifier has to be attached. The RSS is responsible for resolving this ambiguity through the use of the ontology, or by interacting with the user. The linguistic component's task is therefore to pass the ambiguity problem to the RSS through the intermediate representation.

Nevertheless, a query can be a composition of two basic queries. In this case, the intermediate representation usually consists of two triples, one triple per relationship. There are different ways in which queries can be combined. Firstly, queries can be combined by using a “and” or “or” conjunction operator, as in “which projects are funded by epsrc and are about semantic web?”. This query will generate two Query-Triples: <projects, *funded*, epsrc> and <projects, *?*, semantic web> and the subsequent answer will be a combination of both lists obtained after resolving each triple. Secondly, a query may be conditioned to a second query, as in “which researchers wrote publications related to social aspects?” which generates the Query-Triples <researchers, *wrote*, publications> and <*which are*, *related*, social aspects>,where the second clause modifies one of the terms in the first triple. In this example, ambiguity cannot be solved by linguistic procedures; therefore the term to be modified by the second clause remains uncertain.

## 3 Relation Similarity Service

This is the backbone of the QA system. The RSS component is invoked after the NL query has been transformed into a term-relation form and classified into the appropriate category. Essentially the RSS tries to make sense of the input query by looking at the structure of the ontology, string metrics[3], WordNet, and a domain-dependent lexicon obtained by the Learning Mechanism.

In any non-trivial NL system, it is important to deal with the various sources of ambiguity. Some sentences are structurally ambiguous and although general world knowledge does not resolve this ambiguity, within a specific domain it may happen that only one of the interpretations is possible. The key issue here is to determine some constraints derived from the domain knowledge and to apply them in order to resolve ambiguity. Whether the ambiguity cannot be resolved by domain knowledge the only reasonable course of action is to get the user to choose between the alternative readings. Moreover, since every item on the onto-triple is an entry point in the KB or ontology the user has the possibility to navigate through them. In fact, to ensure user acceptance of the system justifications are provided for every step of the user interaction.

[3] http://secondstring.sourceforge.net/

## 4 Related Work

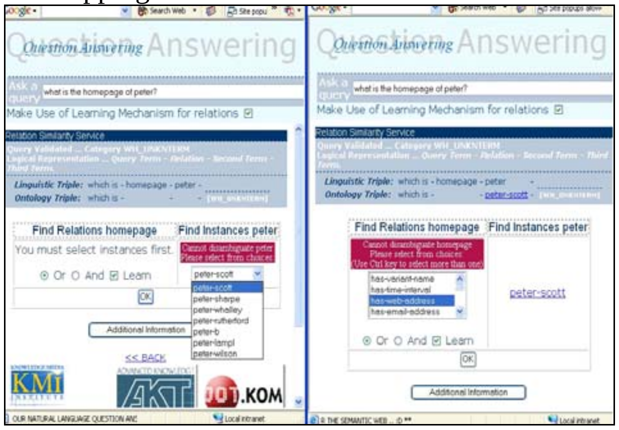
This scenario is similar to research in NL queries to databases (NLIDB). However, the SW provides a new and potentially important context in which results from this research area can be applied. There are linguistic problems common in most kinds of NL understanding systems, see (Androutsopoulos, 1995) for an overview of the state of the art. In contrast with the latest generation of NLIDB systems (see (Popescu, 2003) for recent work) AquaLog uses an intermediate representation from the representation of the user’s query (NL front end) to the representation of an ontology compliant triple, from which an answer can be directly inferred. It takes advantage of the use of ontologies in a way that the entire process highly portable and it is easy to handle unknown vocabulary. For instance, in PRECISE (Popescu, 2003) the problem of finding a mapping from the tokenization to the database requires that all tokens must be distinct, questions with unknown words are not semantically tractable and cannot be handled. In contrast with PRECISE, AquaLog interpret the user query by means of the ontology vocabulary and structure in order to make sense of unknown vocabulary which appears not to have any match.

Current work on QA is somewhat different in nature from AquaLog as they are open-domain systems. QA applications to text typically involve (Hirschman, 2001) identifying the semantic type of the entity sought by the question (a date, a person…); and determining key words or relations to be use in matching candidate answers. Moreover, as pointed by Srihari et al. (Srihari, 2004) Named Entity (NE) tagging is often necessary. The main differences between AquaLog and open-domains systems are: (1) it is not necessary to build hierarchies or heuristic to recognize NE, as all the semantic information needed is in the ontology. (2) AquaLog has mechanisms to exploit the relationships to understand a query. Nevertheless, the RSS goal is to map the relationships in the Query-Triple into an ontology-compliant-triple. Both AquaLog and open-domain systems attempt to find synonyms plus their morphological variants. AquaLog also automatically classifies the question before hand, based on the kind of triple needed, while most of the open-domain QA systems classify questions according to their answer target. The triple contains information not only about the answer expected, but also about the relationships of the other terms in the query. To conclude, other QA systems also follow a relational data model (triple-based), e.g. the START "object-property-value" approach (Katz, 2002).

## 5 AquaLog in action: illustrative example.

For demonstration purposes AquaLog application is used with the AKT ontology in the context of the academic domain in our department (Lei, 2006), e.g., AquaLog translates the query "what is the homepage of Peter who has an interest on the semantic web?" into a conjunction of ontology-compliant non-ground triples: <what is?, has-web-address, peter-scott> & <person?, has-research-interest, Semantic Web area>.

Consider the query "what is the homepage of Peter?" on Fig. 2. Given that the system is unable to disambiguate between Peter-Scott, Peter-Sharpe, etc, user feedback is required. Also the user is call to disambiguate that "homepage" is the same that "has-web-address" as it is the first time the system came across this term, no synonyms have been identified, and the ontology does not provide further ways to disambiguate. The system will learn the mapping and context for future occasions.



what is the homepage of peter?

Query Validated ... Category WH_UNKNTERM
Logical Representation ... Query Term - Relation - Second Term - Third Term.

Linguistic Triple: which is - homepage - peter -
Ontology Triple: which is - - - [WH_UNKNTERM]

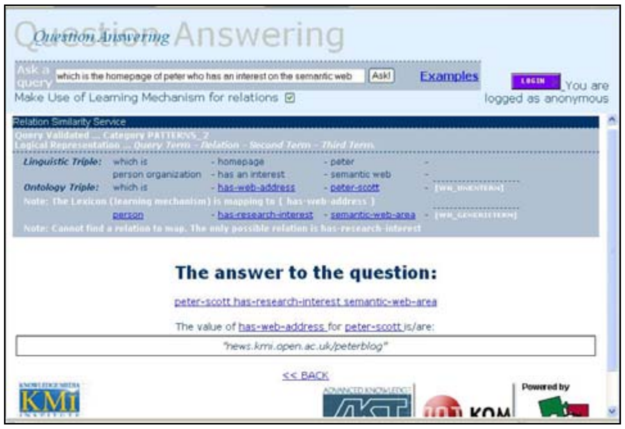
You must select instances first.
Cannot disambiguate peter Please select from choices
peter-scott, peter-sharpe, peter-whalley, peter-rutherford, peter-b, peter-lampl, peter-wilson

Linguistic Triple: which is - homepage - peter -
Ontology Triple: which is - - peter-scott - [WH_UNKNTERM]

Cannot disambiguate homepage Please select from choices (Use Ctrl key to select more than one)
has-variant-name, has-time-interval, has-web-address, has-email-address
peter-scott

**Fig. 2.** Example of user disambiguation

On Fig. 3 we are asking for the web address of Peter, who has an interest in SW. In this case AquaLog does not need any assistance from the user, given that only one of the Peters has an interest in SW. Also the similarity relation between "homepage" and "has-web-address" has been learned by the Learning Mechanism. When the RSS comes across a query like that it has to access to the ontology information to recreate the context and complete the ontology triples. In that way, it realizes that "who has an interest on the Semantic Web" is a modifier of the term "Peter".


Question Answering
Ask a query
Ask!
Examples
LOGIN
You are logged as anonymous
Make Use of Learning Mechanism for relations
Relation Similarity Service
<< BACK
Powered by

which is the homepage of peter who has an interest on the semantic web

Query Validated ... Category PATTERNS_2
Logical Representation ... Query Term - Relation - Second Term - Third Term.

| | | | | |
|---|---|---|---|---|
| Linguistic Triple: | which is | - homepage | - peter | - |
| | person organization | - has an interest | - semantic web | - |
| Ontology Triple: | which is | - has-web-address | - peter-scott | - [WH_UNKNTERM] |
| Note: The Lexicon (learning mechanism) is mapping to ( has-web-address ) | | | | |
| | person | - has-research-interest | - semantic-web-area | - [WH_GENERICTERM] |
| Note: Cannot find a relation to map. The only possible relation is has-research-interest | | | | |

The answer to the question:

peter-scott has-research-interest semantic-web-area

The value of has-web-address for peter-scott is/are:

"news.kmi.open.ac.uk/peterblog"

**Fig. 3.** Example of AquaLog disambiguation

## References

Androutsopoulos, I., Ritchie, G.D., Thanisch P.: Natural Language Interfaces to Databases - An Introduction. *Natural Language Engineering*, 1(1) (1995) 29-81.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (2002).*

Hirschman, L., Gaizauskas, R.: Natural Language question answering: the view from here. *Natural Language Engineering, Special Issue on Question Answering*, 7(4) (2001) 275-300.

Katz, B., et al.: Omnibase: Uniform Access to Heterogeneous Data for QA. In Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB) (2002)

Lopez, V., Pasin, M., and Motta, E. AquaLog: An Ontology-portable Question Answering System for the Semantic Web. In proceeding of the ESWC (2005).

Mc Guinness, D.: Question Answering on the Semantic Web. IEEE Intelligent Systems, 19(1), 2004.

Popescu, A., M., Etzioni, O., Kautz, H., A.: Towards a theory of natural language interfaces to databases. *In Proceedings of the 2003 International Conference on Intelligent User Interfaces*, (2003) 149-157

Srihari, K., Li, W., Li, X.: Information Extraction Supported QA, In T. Strzalkowski & S. Harabagiu (Eds.), in *Advances in Open- Domain Question Answering*. Kluwer Academic Publishers (2004)

Lei, Y., Sabou, M., Lopez, V., Zhu, J., Uren, V. and Motta. E. An Infrastructure for Acquiring high Quality Semantic Metadata. *In proceedings of the 3rd European Semantic Web Conference*. Montenegro, (2006).

# Knowtator: A Protégé plug-in for annotated corpus construction

**Philip V. Ogren**
Division of Biomedical Informatics
Mayo Clinic
Rochester, MN, USA
Ogren.Philip@mayo.edu

## Abstract

A general-purpose text annotation tool called Knowtator is introduced. Knowtator facilitates the manual creation of annotated corpora that can be used for evaluating or training a variety of natural language processing systems. Building on the strengths of the widely used Protégé knowledge representation system, Knowtator has been developed as a Protégé plug-in that leverages Protégé's knowledge representation capabilities to specify annotation schemas. Knowtator's unique advantage over other annotation tools is the ease with which complex annotation schemas (e.g. schemas which have constrained relationships between annotation types) can be defined and incorporated into use. Knowtator is available under the Mozilla Public License 1.1 at http://bionlp.sourceforge.net/Knowtator.

## 1 Introduction

Knowtator is a general-purpose text annotation tool for creating annotated corpora suitable for evaluating Natural Language Processing (NLP) systems. Such corpora consist of texts (e.g. documents, abstracts, or sentences) and *annotations* that associate structured information (e.g. POS tags, named entities, shallow parses) with extents of the texts. An *annotation schema* is a specification of the kinds of annotations that can be created. Knowtator provides a very flexible mechanism for defining annotation schemas. This allows it to be employed for a large variety of corpus annotation tasks.

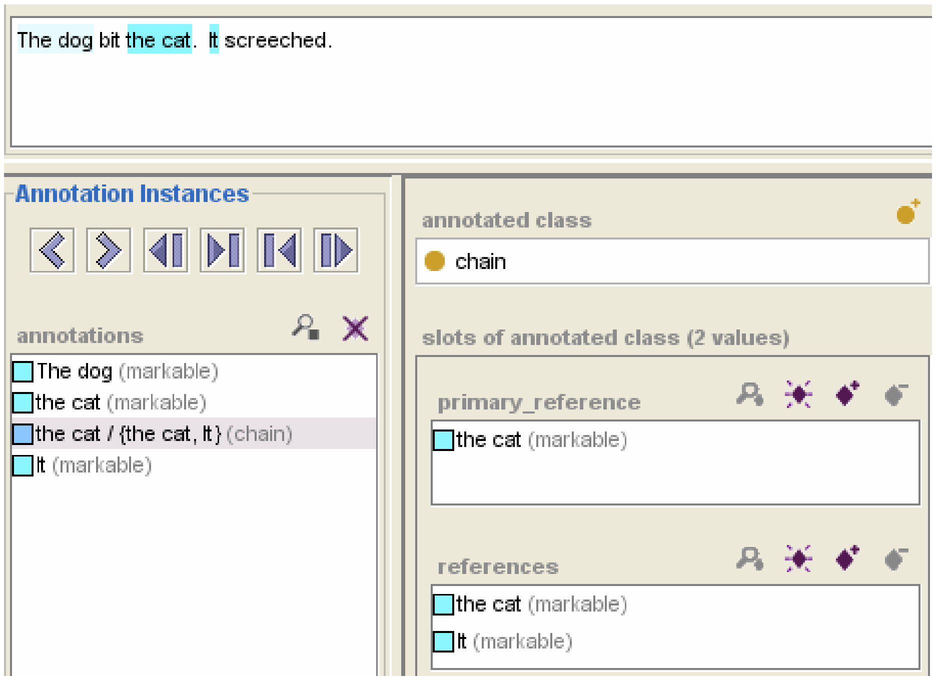
Protégé is a widely used knowledge representation system that facilitates construction and visualization of knowledge-bases (Noy, 2003)[1]. A Protégé knowledge-base typically consists of class, instance, slot, and facet frames. Class definitions represent the concepts of a domain and are organized in a subsumption hierarchy. Instances correspond to individuals of a class. Slots define properties of a class or instance and relationships between classes or instances. Facets constrain the values that slots can have.

Protégé has garnered widespread usage by providing an architecture that facilitates the creation of third-party plug-ins such as visualization tools and inference engines. Knowtator has been implemented as a Protégé plug-in and runs in the Protégé environment. In Knowtator, an annotation schema is defined with Protégé class, instance, slot, and facet definitions using the Protégé knowledge-base editing functionality. The defined annotation schema can then be applied to a text annotation task without having to write any task specific software or edit specialized configuration files. Annotation schemas in Knowtator can model both syntactic (e.g. shallow parses) and semantic phenomena (e.g. protein-protein interactions).

## 2 Related work

There exists a plethora of manual text annotation tools for creating annotated corpora. While it has been common for individual research groups to build customized annotation tools for their specific

[1] http://protege.stanford.edu


The dog bit the cat. It screeched.

| annotations |
|---|
| The dog (markable) |
| the cat (markable) |
| the cat / {the cat, It} (chain) |
| It (markable) |

annotated class: chain

| primary_reference |
|---|
| the cat (markable) |

| references |
|---|
| the cat (markable) |
| It (markable) |

**Figure 1** Simple co-reference annotations in Knowtator

annotation tasks, several text annotation tools have emerged in the last few years that can be employed to accomplish a wide variety of annotation tasks. Some of the better general-purpose annotation tools include Callisto[2], WordFreak[3] (Morton and LaCivita, 2003), GATE[4], and MMAX2[5]. Each of these tools is distributed with a limited number of annotation tasks that can be used 'out of the box.' Many of the tasks that are provided can be customized to a limited extent to suit the requirements of a user's annotation task via configuration files. In Callisto, for example, a simple annotation schema can be defined with an XML DTD that allows the creation of an annotation schema that is essentially a tag set augmented with simple (e.g. string) attributes for each tag. In addition to configuration files, WordFreak provides a plug-in architecture for creating task specific code modules that can be integrated into the user interface.

A complex annotation schema might include hierarchical relationships between annotation types and constrained relationships between the types. Creating such an annotation schema can be a formidable challenge for the available tools either because configuration options are too limiting or because implementing a new plug-in is too expensive or time consuming.

## 3 Implementation

### 3.1 Annotation schema

Knowtator approaches the definition of an annotation schema as a knowledge engineering task by leveraging Protégé's strengths as a knowledge-base editor. Protégé has user interface components for defining class, instance, slot, and facet frames. A Knowtator annotation schema is created by defining frames using these user interface components as a knowledge engineer would when creating a conceptual model of some domain. For Knowtator the frame definitions model the phenomena that the annotation task seeks to capture.

As a simple example, the co-reference annotation task that comes with Callisto can be modeled in Protégé with two class definitions called *markable* and *chain*. The *chain* class has two slots *references* and *primary_reference* which are constrained by facets to have values of type *markable*. This simple annotation schema can now be used to annotate co-reference phenomena occur-

[2] http://callisto.mitre.org
[3] http://wordfreak.sourceforge.net
[4] http://gate.ac.uk/. GATE is a software architecture for NLP that has, as one of its many components, text annotation functionality.
[5]http://mmax.eml-research.de/.

ring in text using Knowtator. Annotations in Knowtator created using this simple annotation schema are shown in Figure 1.

A key strength of Knowtator is its ability to relate annotations to each other via the slot definitions of the corresponding annotated classes. In the co-reference example, the slot *references* of the class *chain* relates the *markable* annotations for the text extents 'the cat' and 'It' to the *chain* annotation. The constraints on the slots ensure that the relationships between annotations are consistent.

Protégé is capable of representing much more sophisticated and complex conceptual models which can be used, in turn, by Knowtator for text annotation. Also, because Protégé is often used to create conceptual models of domains relating to biomedical disciplines, Knowtator is especially well suited for capturing named entities and relations between named entities for those domains.

### 3.2 Features

In addition to its flexible annotation schema definition capabilities, Knowtator has many other features that are useful for executing text annotation projects. A consensus set creation mode allows one to create a gold standard using annotations from multiple annotators. First, annotations from multiple annotators are aggregated into a single Knowtator annotation project. Annotations that represent agreement between the annotators are consolidated such that the focus of further human review is on disagreements between annotators.

Inter-annotator agreement (IAA) metrics provide descriptive reports of consistency between two or more annotators. Several different match criteria (i.e. what counts as agreement between multiple annotations) have been implemented. Each gives a different perspective on how well annotators agree with each other and can be useful for uncovering systematic differences. IAA can also be calculated for selected annotation types giving very fine grained analysis data.

Knowtator provides a pluggable infrastructure for handling different kinds of text source types. By implementing a simple interface, one can annotate any kind of text (e.g. from xml or a relational database) with a modest amount of coding.

Knowtator provides stand-off annotation such that the original text that is being annotated is not modified. Annotation data can be exported to a simple XML format.

Annotation filters can be used to view a subset of available annotations. This may be important if, for example, viewing only named entity annotations is desired in an annotation project that also contains many part-of-speech annotations. Filters are also used to focus IAA analysis and the export of annotations to XML.

Knowtator can be run as a stand-alone system (e.g. on a laptop) without a network connection. For increased scalability, Knowtator can be used with a relational database backend (via JDBC).

Knowtator and Protégé are provided under the Mozilla Public License 1.1 and are freely available with source code at http://bionlp.sourceforge.net/Knowtator and http://protege.stanford.edu, respectively. Both applications are implemented in the Java programming language and have been successfully deployed and used in the Windows, MacOS, and Linux environments.

## 4 Conclusion

Knowtator has been developed to leverage the knowledge representation and editing capabilities of the Protégé system. By modeling syntactic and/or semantic phenomena using Protégé frames, a wide variety of annotation schemas can be defined and used for annotating text. New annotation tasks can be created without writing new software or creating specialized configuration files. Knowtator also provides additional features that make it useful for real-world multi-person annotation tasks.

## References

Thomas Morton and Jeremy LaCivita. 2003. *WordFreak: An Open Tool for Linguistic Annotation*, Proceedings of NLT-NAACL, pp. 17-18.

Noy, N. F., M. Crubezy, et al. 2003. *Protege-2000: an open-source ontology-development and knowledge-acquisition environment.* AMIA Annual Symposium Proceedings: 953.

# Automatic Cluster Stopping with Criterion Functions and the Gap Statistic

**Ted Pedersen** and **Anagha Kulkarni**
Department of Computer Science
University of Minnesota, Duluth
Duluth, MN 55812 USA
{tpederse,kulka020}@d.umn.edu
http://senseclusters.sourceforge.net

**Abstract**

SenseClusters is a freely available system that clusters similar contexts. It can be applied to a wide range of problems, although here we focus on word sense and name discrimination. It supports several different measures for automatically determining the number of clusters in which a collection of contexts should be grouped. These can be used to discover the number of senses in which a word is used in a large corpus of text, or the number of entities that share the same name. There are three measures based on clustering criterion functions, and another on the Gap Statistic.

## 1 Introduction

Word sense and name discrimination are problems in unsupervised learning that seek to cluster the occurrences of a word (or name) found in multiple contexts based on their underlying meaning (or identity). The assumption is made that each discovered cluster will represent a different sense of a word, or the underlying identity of a person or organization that has an ambiguous name.

Existing approaches to this problem usually require that the number of clusters to be discovered (*k*) be specified ahead of time. However, in most realistic settings, the value of *k* is unknown to the user. Here we describe various *cluster stopping measures* that are now implemented in SenseClusters (Purandare and Pedersen, 2004) that will group $N$ contexts into *k* clusters, where the value of *k* will be automatically determined.

Cluster stopping can be viewed as a problem in model selection, since a number of different models (i.e., clustering solutions) are created using different values of *k*, and the one that best fits the observed data is selected based on a criterion function. This is reminiscent of earlier work on sequential model selection for creating models of word sense disambiguation (e.g., (O'Hara et al., 2000)), where it was found that forward sequential search strategies were most effective. These methods start with simpler models and then add to them in a stepwise fashion until no further improvement in model fit is observed. This is in fact very similar to what we have done here, where we start with solutions based on one cluster, and steadily increase the number of clusters until we find the best fitting solution.

SenseClusters supports four cluster stopping measures, each of which is based on interpreting a clustering criterion function in some way. The first three measures (PK1, PK2, PK3) look at the successive values of the criterion functions as *k* increases, and try to identify the point at which the criterion function stops improving significantly. We have also created an adaptation of the Gap Statistic (Tibshirani et al., 2001), which compares the criterion function from the clustering of the observed data with the clustering of a null reference distribution and selects the value of $k$ for which the difference between them is greatest.

In order to evaluate our results, we sometimes conduct experiments with words that have been manually sense tagged. We also create *name con-*

*flations* where some number of names of persons, places, or organizations are replaced with a single name to create pseudo or false ambiguities. For example, in this paper we refer to an example where we have replaced all mentions of *Sonia Gandhi* and *Leonid Kuchma* with a single ambiguous name.

Clustering methods are typically either partitional or agglomerative. The main difference is that agglomerative methods start with 1 or $N$ clusters and then iteratively arrive at a pre–specified number (*k*) of clusters, while partitional methods start by randomly dividing the contexts into *k* clusters and then iteratively rearranging the members of the *k* clusters until the selected criterion function is maximized. In this work we have used K-means clustering, which is a partitional method, and the $H2$ criterion function, which is the ratio of within–cluster similarity ($I2$) to between–cluster similarity ($E1$).

## 2 Methodology

In word sense or name discrimination, the number of contexts $(N)$ to cluster is usually very large, and considering all possible values of *k* from $1...N$ would be inefficient. As the value of *k* increases, the criterion function will reach a plateau, indicating that dividing the contexts into more and more clusters does not improve the quality of the solution. Thus, we identify an upper bound to *k* that we refer to as *deltaK* by finding the point at which the criterion function only changes to a small degree as *k* increases.

According to the $H2$ criterion function, the higher its ratio of within–cluster similarity to between–cluster similarity, the better the clustering. A large value indicates that the clusters have high internal similarity, and are clearly separated from each other. Intuitively then, one solution to selecting *k* might be to examine the trend of $H2$ scores, and look for the smallest *k* that results in a nearly maximum $H2$ value.

However, a graph of $H2$ values for a clustering of the 2 sense name conflation *Sonia Gandhi* and *Leonid Kuchma* as shown in Figure 1 (top) reveals the difficulties of such an approach. There is a gradual curve in this graph and there is no obvious *knee point* (i.e., sharp increase) that indicates the appropriate value of *k*.

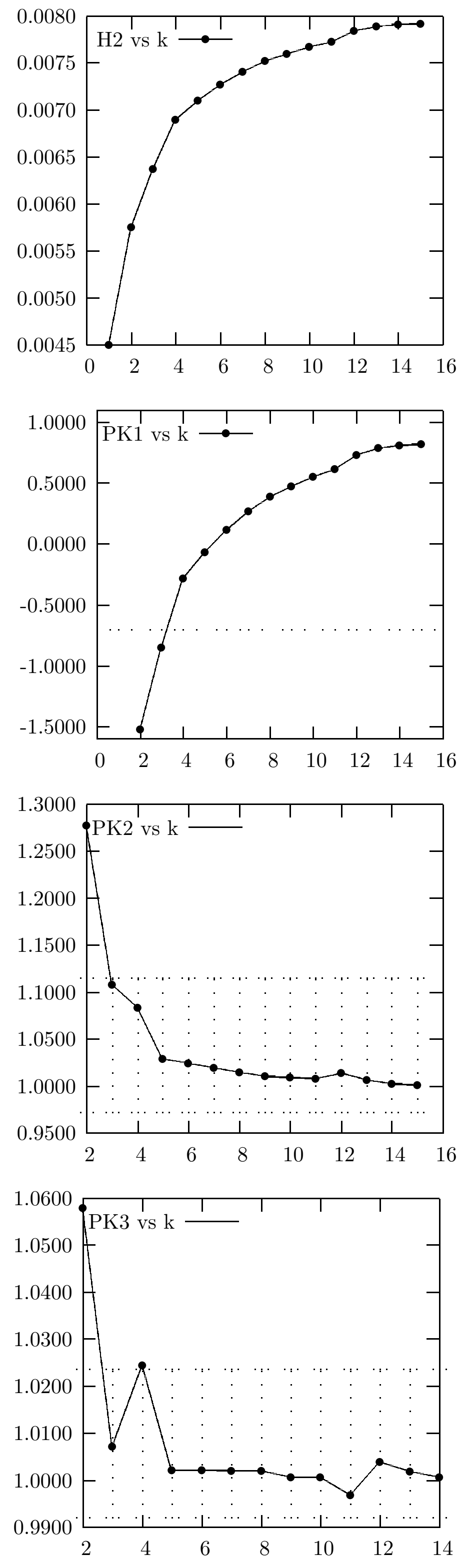


Figure 1: H2 (top) and PK1, PK2, and PK3 for the name conflate pair *Sonia Gandhi* and *Leonid Kuchma*. The predicted number of senses is 2 for all the measures.

### 2.1 PK1

The $PK1$ measure is based on (Mojena, 1977), which finds clustering solutions for all values of k from $1..N$, and then determines the mean and standard deviation of the criterion function. Then, a score is computed for each value of *k* by subtracting the mean from the criterion function, and dividing by the standard deviation. We adapt this technique by using the $H2$ criterion function, and limit *k* from $1...deltaK$:

$$PK1(k) = \frac{H2(k) - mean(H2[1...deltaK])}{std(H2[1...deltaK])} \tag{1}$$

To select a value of *k*, a threshold must be set. Then, as soon as $PK1(k)$ exceeds this threshold, *k-1* is selected as the appropriate number of clusters. Mojena suggests values of 2.75 to 3.50, but also states they would need to be adjusted for different data sets. We have arrived at an empirically determined value of -0.70, which coincides with the point in the standard normal distribution where 75% of the probability mass is associated with values greater than this.

We observe that the distribution of $PK1$ scores tends to change with different data sets, making it hard to apply a single threshold. The graph of the $PK1$ scores shown in Figure 1 illustrates the difficulty : the slope of these scores is nearly linear, and as such any threshold is a somewhat arbitrary cutoff.

### 2.2 PK2

$PK2$ is similar to (Hartigan, 1975), in that both take the ratio of a criterion function at *k* and *k-1*, in order to assess the relative improvement when increasing the number of clusters.

$$PK2(k) = \frac{H2(k)}{H2(k-1)} \tag{2}$$

When this ratio approaches 1, the clustering has reached a plateau, and increasing *k* will have no benefit. If $PK2$ is greater than 1, then we should increase *k*. We compute the standard deviation of $PK2$ and use that to establish a boundary as to what it means to be "close enough" to 1 to consider that we have reached a plateau. Thus, $PK2$ will select *k* where $PK2(k)$ is the closest to (but not less than) 1 + standard deviation(PK2[1...*deltaK*]).

The graph of $PK2$ in Figure 1 shows an *elbow* that is near the actual number of senses. The critical region defined by the standard deviation is shaded, and note that $PK2$ selected the value of *k* that was outside of (but closest to) that region. This is interpreted as being the last value of *k* that resulted in a significant improvement in clustering quality. Note that here $PK2$ predicts 2 senses, which corresponds to the number of underlying entities.

### 2.3 PK3

$PK3$ utilizes three *k* values, in an attempt to find a point at which the criterion function increases and then suddenly decreases. Thus, for a given value of *k* we compare its criterion function to the preceding and following value of *k*:

$$PK3(k) = \frac{2 \times H2(k)}{H2(k-1) + H2(k+1)} \tag{3}$$

The form of this measure is identical to that of the Dice Coefficient, although in set theoretic or probabilistic applications Dice tends to be used to compare two variables or sets with each other.

$PK3$ is close to 1 if the $H2$ values form a line, meaning that they are either ascending, or they are on the plateau. However, our use of *deltaK* eliminates the plateau, so in our case values of 1 show that *k* is resulting in consistent improvements to clustering quality, and that we should continue. When $PK3$ rises significantly above 1, we know that *k+1* is not climbing as quickly, and we have reached a point where additional clustering may not be helpful. To select $k$ we select the largest value of $PK3(k)$ that is closest to (but still greater than) the critical region defined by the standard deviation of $PK3$.

$PK3$ is similar in spirit to (Salvador and Chan, 2004), which introduces the L measure. This tries to find the point of maximum curvature in the criterion function graph, by fitting a pair of lines to the curve (where the intersection of these lines represents the selected *k*).

## 2.4 The Gap Statistic

SenseClusters includes an adaptation of the Gap Statistic (Tibshirani et al., 2001). It is distinct from the measures PK1, PK2, and PK3 since it does not attempt to directly find a knee point in the graph of a criterion function. Rather, it creates a sample of reference data that represents the observed data as if it had no meaningful clusters in it and was simply made up of noise. The criterion function of the reference data is then compared to that of the observed data, in order to identify the value of *k* in the observed data that is least like noise, and therefore represents the best clustering of the data.

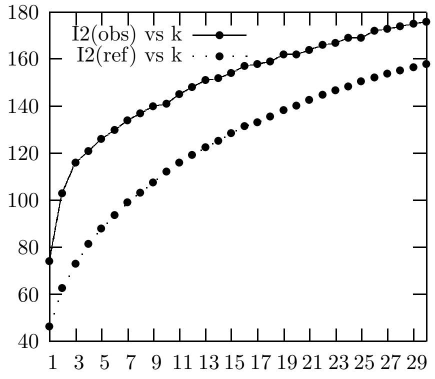
To do this, it generates a null reference distribution by sampling from a distribution where the marginal totals are fixed to the observed marginal values. Then some number of replicates of the reference distribution are created by sampling from it with replacement, and each of these replicates is clustered just like the observed data (for successive values of *k* using a given criterion function).

The criterion function scores for the observed and reference data are compared, and the point at which the distance between them is greatest is taken to provide the appropriate value of *k*. An example of this is seen in Figure 2. The reference distribution represents the noise in the observed data, so the value of *k* where the distance between the reference and observed data is greatest represents the most effective clustering of the data.

Our adaption of the Gap Statistic allows us to use any clustering criterion function to make the comparison of the observed and reference data, whereas the original formulation is based on using the within–cluster dispersion.

## 3 Acknowledgments

This research is supported by a National Science Foundation Faculty Early CAREER Development Award (#0092784).



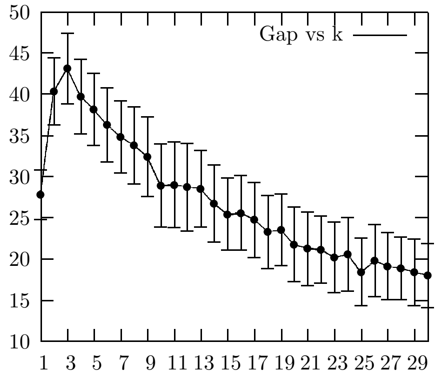


Figure 2: $I2$ for observed and reference data (top) and the Gap between them (bottom) for the name conflate pair *Sonia Gandhi* and *Leonid Kuchma*. The predicted number of senses is 3.

## References

J. Hartigan. 1975. *Clustering Algorithms*. Wiley, New York.

R. Mojena. 1977. Hierarchical grouping methods and stopping rules: An evaluation. *The Computer Journal*, 20(4):359–363.

T. O'Hara, J. Wiebe, and R. Bruce. 2000. Selecting decomposable models for word-sense disambiguation: The grling-sdm system. *Computers and the Humanities*, 34(1–2):159–164.

A. Purandare and T. Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 41–48, Boston, MA.

S. Salvador and P. Chan. 2004. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Proceedings of the 16th IEEE International Conference on Tools with AI*, pages 576–584.

R. Tibshirani, G. Walther, and T. Hastie. 2001. Estimating the number of clusters in a dataset via the Gap statistic. *Journal of the Royal Statistics Society (Series B)*, pages 411–423.

# Automating the Creation of Interactive Glyph-supplemented Scatterplots for Visualizing Algorithm Results

**Dinoj Surendran**
Department of Computer Science
University of Chicago
`dinoj@cs.uchicago.edu`

**Abstract**

Ndaona is a Matlab toolkit to create interactive three-dimensional models of data often found in NLP research, such as exploring the results of classification and dimensionality reduction algorithms. Such models are useful for teaching, presentations and exploratory research (such as showing where a classification algorithm makes mistakes).

Ndaona includes embedding and graphics parameter estimation algorithms, and generates files in the format of Partiview (Levy, 2001), an existing free open-source fast multidimensional data displayer that has traditionally been used in the planetarium community. Partiview[1] supports a number of enhancements to regular scatterplots that allow it to display more than three dimensions' worth of information.

## 1 Supplemented Scatterplots

Scatterplots are not the most efficient way of representing information (Grinstein et al., 2001). However, they are intuitive and stable (Wong and Bergeron, 1997), and can be supplemented in several ways. We describe some of these augmentations in Section 1, basic Ndaona usage in Section 2, and finally a couple of some embedding methods in Section 3.

[1]`http://niri.ncsa.uiuc.edu/partiview`

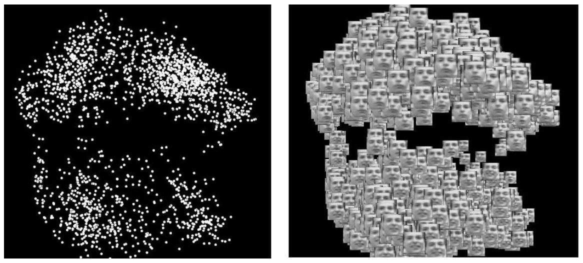


Figure 1: Regular and glyph-supplemented scatterplots showing how a linear kernel can separate happy and sad faces from the Frey Face dataset.

### 1.1 Glyphs

Glyphs are icons that provide a visual representation of a datum. A simple example of a glyph is a filled-in circle whose color and size convey two additional dimensions of information. More complex examples have been designed to present more information (Tukey and Tukey, 1981) (Ward, 2002). Partiview can use any image as a glyph, as long as all images used can fit in graphics memory.

For example, Figure 1 shows faces from the Frey Face Dataset[2] in linear kernel space; two faces are close together then the vectors $u$ and $v$ representing have a high value of $u^T v$. In this case, each point has a natural visual representation — the face itself. And we therefore use the faces as lossless glyphs, with each glyph representing a 560-dimensional vector (20 x 28 pixel image).

A second example is in Figure 2. It shows Mandarin syllables in a tone recognition experiment (Surendran et al., 2005), with two syllables close

[2]Thanks to Sam Roweis for placing this data on his site.

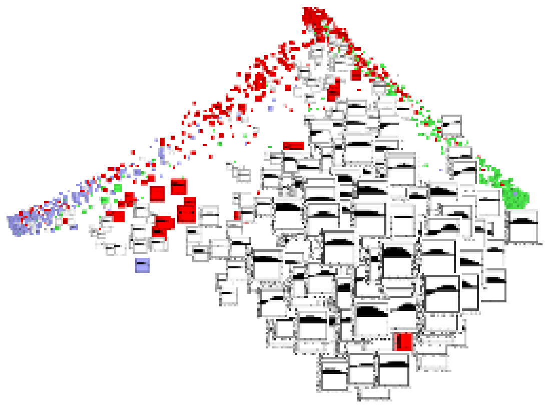


Figure 2: A close-up screenshot of a 3D glyph-supplemented scatterplot showing the performance of a linear Support Vector Machine (SVM) on a 4-way Mandarin syllable tone classification task. Ndaona embedded syllables so that those classified similarly by the SVM are close together. The glyph for each syllable represents the 20-dimensional vector input to the SVM. Syllables with the same tone are represented by glyphs of the same color; the white syllables in the foreground have falling tone.

together if the classification algorithm made similar predictions of their tone. The algorithm received for each syllable a 20-dimensional vector that described its normalized pitch contour of the syllable. In this case, a histogram of the pitch contour, with the area between the pitch contour and the horizontal axis shaded for enhanced visibility, results in a highly informative glyph.

A close-up look at the low tone syllables reveals that the algorithm 'thinks' that any syllable whose pitch contour decreases towards the end has a falling tone which is what linguists expect. We can also tell that many of the mistakes made by the algorithm are due to the features and not the algorithm itself. For instance, the several high tone syllables that are close to the cluster of low-tone-syllables (and would thus be classified as having low tone by the algorithm) do in fact have a falling pitch contour.

### 1.2 Three Dimensions

Partiview users can smoothly spin, zoom, and move the 3d scatterplot model even when it contains hundreds of thousands of points. Interaction and motion deal with difficulties (described in the information visualization and statistical graphics literature) of visually estimating distances when a third dimension is used (Jacoby, 1998).

### 1.3 Graphs

While Partiview is not the best software for displaying graphs, it can display lines of varying width and color. This permits two bits of information to be displayed about binary relations between points.

### 1.4 Information about points

While text labels can be placed at points, this often results in visual clutter. It is better to give the user the option of only having labels displayed when actively requested. This option is called 'linking'.

Partiview has some linking capability. When a user clicks on a point (and presses 'p'), the command window displays information about it, such as its position, several features, and any 'comment' provided by the user. For example, Figures 3 and 4 show the results of a 13-class dialog act classification task — the user supplied as comments the words said during each dialog act. Some of these can be seen in the command window of each screenshot.

### 1.5 Brushing

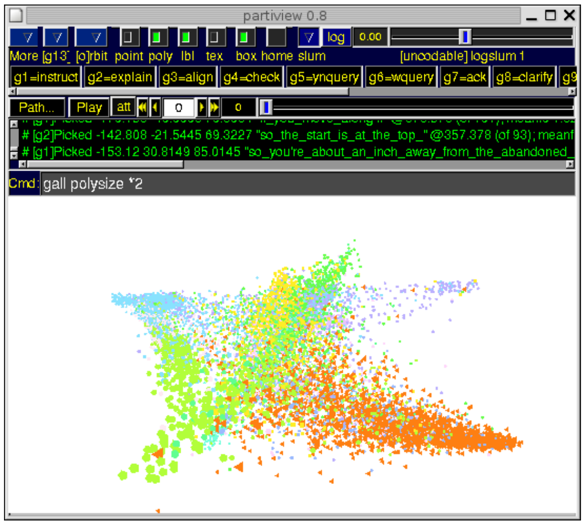
Brushing is the ability for users to select subsets of data points and apply certain operations to them, such as toggling their visibility (masking), changing their color or size (Becker and Cleveland, 1987).

Partiview supports this very well, and it is possibly the most important feature available for data exploration. For example, we can change the colors of points to be that of any attribute of the data, including its original features. This helps investigate what original features the algorithm is actually using.

For example, in Figure 3 color represents class, while in Figure 4 color represents duration. The color could just as easily be changed to represent other attributes of the data; Ndaona estimates Partiview parameters required for consistent behavior across attributes by normalizing the color map for each attribute.

### 1.6 Animation

Partiview supports animated data. Ndaona has been written so that one can deal with various combinations of static and dynamic (time-varying) graphical elements, such as fixed points and varying edges, or dynamic points and fixed edges (i.e. the edges always join the same points), or both dynamic points and edges, fixed points and dynamic attributes, and so on. The only difference to the user is that s/he provides a cell array (list) of matrices for the dynamic element instead of a single matrix.



```
# [g2]Picked -142.808 -21.5445 69.3227 "so_the_start_is_at_the_top_" @357.378 (of 93); meanf
# [g1]Picked -153.12 30.8149 85.0145 "so_you're_about_an_inch_away_from_the_abandoned_
```

```
gall polysize *2
```

Figure 3: Partiview screenshot of a Ndaona-made model showing the result of a dialog act classification algorithm. Each point represents a dialog act, and all acts of the same type are colored identically.

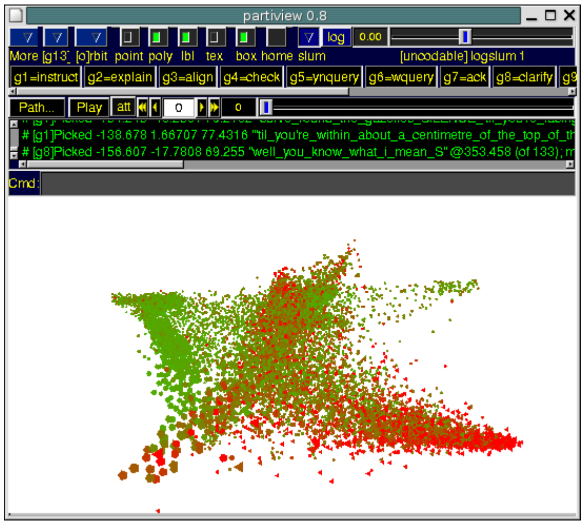
## 2 Usage

For flexibility of input, arguments to Ndaona are supplied in parameter-value pairs. For example, say P is a $N \times 3$ matrix representing the 3D coordinates of $N$ points and Q is a list of $N$ images representing the glyphs of each point. Ndaona includes tools to create such images, or the user can provide their own JPEGs. Either of the equivalent commands

```
ndaona('POSITIONS',P,'PICTURES',Q)
ndaona('POS',P, 'PICS',Q)
```

creates a 3D model with the pictures in Q represented at the positions for each point, such as that in Figures 1 and 3. Graphics parameters controlling picture sizes are automatically estimated by Ndaona.

Now suppose that the $N$ points have time-varying positions. Making P a list of $N \times 3$ matrices and



```
# [g1]Picked -138.678 1.66707 77.4316 "'til_you're_within_about_a_centimetre_of_the_top_of_th
# [g8]Picked -156.607 -17.7808 69.255 "well_you_know_what_i_mean_S" @353.458 (of 133); m
```

Figure 4: As for Figure 3 but now color represents duration. Shorter acts (top left) are green.

using the same command as above creates a time-varying scatterplot with the images moving about.

If this was classification and the true class of each point is stored in a $N$-dimensional vector L, then

```
ndaona('POS',P,'PICS',Q,'CLASS',L)
```

creates the 3D model with the pictures colored by class, with the same color for points of the same class, such as that in Figure 2. Also, Partiview provides a button for each class that toggles the visibility of data points in that class. If each point has $A$ attributes stored in a $N \times A$ matrix F, then

```
ndaona('POS',P,...,'ATTRIBUTES',F)
```

creates a model as before, but with brushing available. The colors of each point can be changed according to the $r$-th attribute by typing "color a$r$", where a$r$ is the automatically assigned name for the $r$-th attribute. (Users can provide attribute names with another parameter-value pair.)

If the $N$ points also form the nodes of a (connected or not) graph with $N_e$ edges, then if the edges are represented by a $N_e \times 3$ matrix or a sparse $N_e \times N_e$ matrix G, the command

```
ndaona('POS',P,...,'EDGES',G)
```

creates the same scatterplot, overlaid with edges.

Additional parameter-value pairs in Ndaona can be used to fine-tune graphics parameters, create files in directory structures that are easy to compress and distribute, change background color, etc.

## 3 Embedding into Three Dimensions

When visualizing the results of algorithms, users may not have a three-dimensional embedding already available. However, algorithms have been proposed to produce such embeddings, and we now describe some of those available in Ndaona. Ndaona also implements basic dimensionality reduction algorithms such as Principal Components Analysis, Laplacian Eigenmaps, and Isomap.

### 3.1 Classification Probabilities

If users have a $N \times K$ matrix $\mathsf{S}$ of prediction probabilities from a $K$-class classification algorithm, with $\mathsf{S}(n, k)$ having the probability (estimated by the algorithm) that the $n$-th point is in class $k$, then this can be supplied instead.

Ndaona uses the Parametric Embedding algorithm (Iwata et al., 2004) to find a low-dimensional embedding of the $N$ points so that pairs of points that were given similar predictions by the classification algorithm (i.e. have low Kullback-Leibler distance between their prediction probability distributions) are closer together.

### 3.2 Kernel Matrices

Support vector machines (SVMs) and related methods depend on pairwise similarities of points, in the form of a kernel matrix whos $(i, j)$-th entry represents the similarity of the $i$-th and $j$-th points. Shawe-Taylor and Christianini (2004) suggest using the eigenvectors corresponding to the three smallest positive eigenvalues of the Laplacian of the $N \times N$ kernel matrix to define a $N \times 3$ positions matrix. Ndaona implements an alternative that, in our experience, works better — using the normalized Laplacian of the kernel matrix (with negative entries replaced by zero).

## 4 Conclusion

Ndaona is an interface package that helps reearchers produce compelling visual representations of their data. Its output is a (time-varying) 3d model that can be displayed by Partiview, an external data viewer. Future plans include adding more scalable embedding algorithms, and allowing other output formats. Ndaona, documentation, and examples of models created with it, can be found at http://people.cs.uchicago.edu/∼dinoj/ndaona

## References

R A Becker and W S Cleveland. 1987. Brushing scatterplots. *Technometrics*, 29(2):127–142.

G Grinstein, M Trutschl, and U Cvek. 2001. High-dimensional visualizations. In *Proceedings of the 7th Data Mining Conference-KDD*.

T Iwata, K Saito, N Ueda, S Stromsten, T L Griffiths, and Joshua B Tenenbaum. 2004. Parametric embedding for class visualization. In *Advances in Neural Information Processing Systems 17*.

William G. Jacoby. 1998. *Statistical Graphics for Visualizing Multivariate Data*. Sage University Papers Series on Quantitative Applications in the Social Sciences 07-120, Thousand Oaks, CA.

Stuart Levy. 2001. Interactive 3-d visualization of particle systems with partiview. In *Astrophysical Supercomputing Using Particles (I.A.U. Symposium Proceedings)*, volume 208, pages 85–91. International Astronomical Union.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Dinoj Surendran, Gina-Anne Levow, and Yi Xu. 2005. Tone recognition in mandarin using focus. In *Proceedings of the 9th European Conference of Speech Communication and Technology*.

P A Tukey and J W Tukey. 1981. Summarization; smoothing; supplementing views. In Vic Barnett, editor, *Interpreting Multivariate Data*, pages 245–275. John Wiley and Sons.

Matthew O. Ward. 2002. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3/4):194–210.

Pak Chung Wong and R Daniel Bergeron. 1997. Thirty years of multidimensional multivariate visualization. In Gregory M Nielson, Hans Hagan, and Heinrich Muller, editors, *Scientific Visualization - Overviews, Methodologies and Techniques*, pages 3–33, Los Alamitos, CA. IEEE Computer Society Press.

# SconeEdit: A Text-guided Domain Knowledge Editor

**Alicia Tribble**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
atribble@cs.cmu.edu

**Benjamin Lambert**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
benlambert@cmu.edu

**Scott E. Fahlman**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sef@cs.cmu.edu

## Abstract

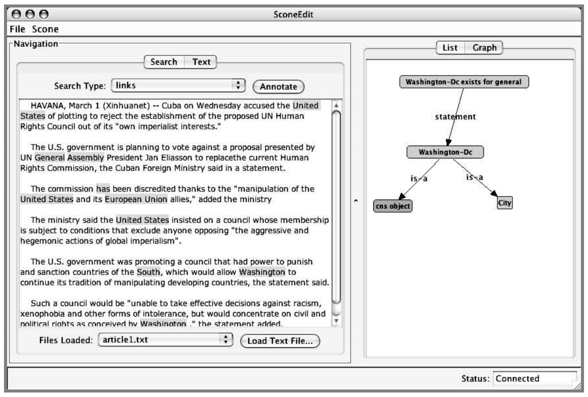
We will demonstrate SconeEdit, a new tool for exploring and editing knowledge bases (KBs) that leverages interaction with domain texts. The tool provides an annotated view of user-selected text, allowing a user to see which concepts from the text are in the KB and to edit the KB directly from this Text View. Alongside the Text View, SconeEdit provides a navigable KB View of the knowledge base, centered on concepts that appear in the text. This unified tool gives the user a text-driven way to explore a KB and add new knowledge.

## 1 Introduction

We will demonstrate SconeEdit, a new tool for exploring and editing knowledge bases that integrates domain text. SconeEdit expands on the function of traditional ontology editors by showing the user an interactive text window (Text View) where the user can view and edit concepts from the knowledge base as highlighted terms in their original context. The Text View augments a traditional KB View, allowing the user to leverage existing knowledge as well as domain-focused text examples to perform a variety of knowledge-based tasks.

Consider the task of assessing the quality of a knowledge base as a resource for a new AI or natural language system. In SconeEdit, a user can view the knowledge base alongside a text document from the target domain. SconeEdit searches for instances of KB concepts in the text and highlights them in the Text View. Already the user can see a concise visual sample of the coverage of the KB for this domain.

Now the user can work with the KB View and Text View together to navigate the ontology. Double-clicking on a highlighted concept like "keyboard" opens a detailed view of that concept in the KB View. Inside the KB View, the user can click on the superclass of the *keyboard* concept to see the concept *computer input device* and all of its children. Next, SconeEdit selectively highlights all instances of *computer input device* in the text. The system uses type inference from the KB to highlight "mouse", "touchpad", and "wireless keyboard." If "scanner" appears in the text but isn't included in the knowledge base, the user can spot the omission quickly.



HAVANA, March 1 (Xinhuanet) -- Cuba on Wednesday accused the United States of plotting to reject the establishment of the proposed UN Human Rights Council out of its "own imperialist interests."

The U.S. government is planning to vote against a proposal presented by UN General Assembly President Jan Eliasson to replacethe current Human Rights Commission, the Cuban Foreign Ministry said in a statement.

The commission has been discredited thanks to the "manipulation of the United States and its European Union allies," added the ministry

The ministry said the United States insisted on a council whose membership is subject to conditions that exclude anyone opposing "the aggressive and hegemonic actions of global imperialism".

The U.S. government was promoting a council that had power to punish and sanction countries of the South, which would allow Washington to continue its tradition of manipulating developing countries, the statement said.

Such a council would be "unable to take effective decisions against racism, xenophobia and other forms of intolerance, but would concentrate on civil and political rights as conceived by Washington ," the statement added.

Figure 1. The SconeEdit Interface

*Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume*, pages 284–287,
New York City, June 2006. ©2006 Association for Computational Linguistics

In this way, domain text is used as a measuring tool for coverage of domain knowledge. Our demonstration allows the user to try SconeEdit and to explore the interaction of text and knowledge.

## 2 The Knowledge Base

SconeEdit is a software client to the Scone Knowledge Base System, or simply "Scone" (Fahlman, 2005). Scone is an efficient, open-source knowledge base (KB) system being developed in the Language Technologies Institute of Carnegie Mellon University. Scone is intended to be a practical KB system that can be used as a component in a wide range of AI and natural language software applications. One of the goals in developing Scone is to make it easy to use, especially when adding new knowledge.

The SconeEdit interface makes Scone more usable in several ways: the Text View display gives the user a convenient and intuitive starting point for exploring the knowledge base. SconeEdit also provides an easy way of adding knowledge to the KB without learning the formal input language for Scone. This demonstration focuses on the effectiveness of SconeEdit and Scone together, but the design principles of SconeEdit are applicable to knowledge bases written in other formalisms.

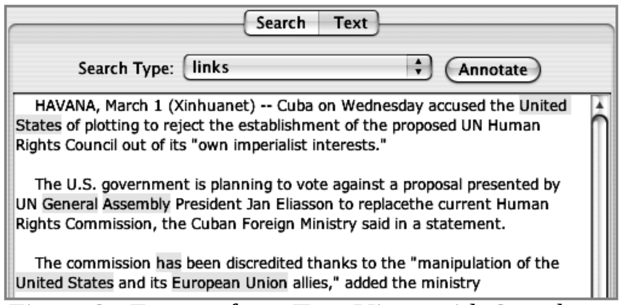
Figure 1 shows the SconeEdit window with a document and KB loaded. The left side of the interface contains the Text View, and the KB View is on the right. Each of these views is described in detail below.

## 3 Architecture

### 3.1 Text View

In a traditional ontology browser, the user starts looking for concepts of interest by typing words and phrases into a search field. This is the model for several existing tools, including the VisDic viewer for WordNet (Horák and Smrž, 2004), the INOH ontology viewer (INOH, 2004), and the Gene Ontology viewer presented by Koike and Takagi (2004), among others.

SconeEdit improves on this browsing paradigm by giving a user who is unfamiliar with the knowledge base an easy way to start exploring. Rather than generating a series of guesses at what may be



HAVANA, March 1 (Xinhuanet) -- Cuba on Wednesday accused the United States of plotting to reject the establishment of the proposed UN Human Rights Council out of its "own imperialist interests."

The U.S. government is planning to vote against a proposal presented by UN General Assembly President Jan Eliasson to replacethe current Human Rights Commission, the Cuban Foreign Ministry said in a statement.

The commission has been discredited thanks to the "manipulation of the United States and its European Union allies," added the ministry

Figure 2. Excerpt from Text View, with Search and Text Tabs

covered by the KB, the user can load natural language text into SconeEdit from a file or the system clipboard. We take an article from Xinhuanet News Service (Xinhuanet, 2006) as an example. Figure 2 shows an excerpt of this text after it has been loaded.

When the text file is loaded, it appears in the Text Tab of the Text View pane. SconeEdit highlights all strings that it can identify as concepts from the knowledge base. In this example, "Washington" is correctly identified as the city, not the state. In many cases the concept may be ambiguous from the string alone. SconeEdit currently uses dynamic programming to highlight the longest-matching concept names it can find (see Section 5). More sophisticated disambiguation is a priority for our future work.

The result of highlighting is a concise visual representation of what is "known" about that text. The Text View helps a user find relevant knowledge quickly, even in a large general-domain KB. Clicking on any highlighted term in the Text View brings up a hierarchical representation of that concept in the KB View.

### 3.2 KB View

The KB View contains two tabs: a Graph Tab and a List Tab. The Graph Tab displays an excerpt from the knowledge base as a network of linked concepts with one *focus concept* in the center. When the user clicks on a highlighted concept in the Text View, a graph focused on that concept appears in the Graph Tab. Continuing with our Xinhuanet example, Figure 3 shows the Graph Tab after a user has clicked on "Washington" in the text. The Graph View now displays concepts that are closely related to *Washington-Dc* in the knowledge base.

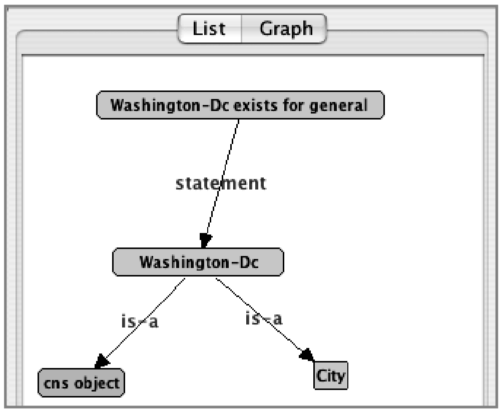
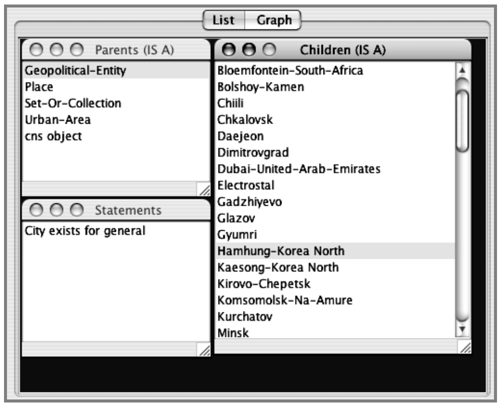


Figure 3. KB View, Graph Tab of *Washington-Dc*



Geopolitical-Entity
Place
Set-Or-Collection
Urban-Area
cns object

City exists for general

Bloemfontein-South-Africa
Bolshoy-Kamen
Chilli
Chkalovsk
Daejeon
Dimitrovgrad
Dubai-United-Arab-Emirates
Electrostal
Gadzhiyevo
Glazov
Gyumri
Hamhung-Korea North
Kaesong-Korea North
Kirovo-Chepetsk
Komsomolsk-Na-Amure
Kurchatov
Minsk

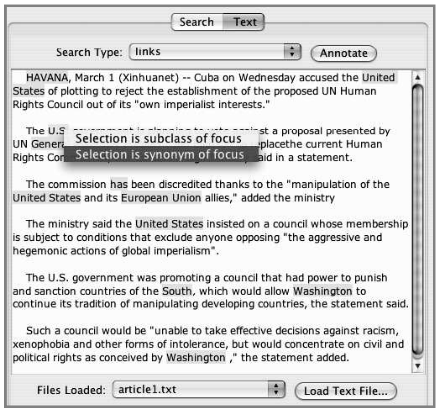
Figure 4. KB View, List Tab of *City*

Clicking on any of these related concepts in the Graph Tab moves the focus of the graph to that concept.

The List Tab shows an alternative view of the same *focus concept*. It displays KB information as a set of property lists. As in the Graph Tab, the user can double-click on any concept in the List Tab to bring that concept into focus. When the focus concept is densely connected to other concepts in the KB, the List Tab can be easier to interpret than the Graph Tab. In general, research has shown that preference for the list style or graph style is personal and varies from user to user (Tribble and Rosé, 2006). Figure 4 shows the List Tab, focused on the concept *City*.

## 4 Adding Knowledge

Browsing the knowledge base in this way gives the user a detailed, domain-targeted view of its contents. A natural extension of this paradigm is to allow the user to edit the KB while browsing. For example, a user may encounter a concept in the text that is not present in the knowledge base. SconeEdit allows the user to simply click on a word in the text to create a new concept in the KB (see Figure 5). To specify where the new concept belongs, the user navigates to the appropriate location in the KB View (List Tab or Graph Tab).



HAVANA, March 1 (Xinhuanet) -- Cuba on Wednesday accused the United States of plotting to reject the establishment of the proposed UN Human Rights Council out of its "own imperialist interests."

The U.S st a proposal presented by UN Genera placethe current Human Rights Con aid in a statement.

The commission has been discredited thanks to the "manipulation of the United States and its European Union allies," added the ministry

The ministry said the United States insisted on a council whose membership is subject to conditions that exclude anyone opposing "the aggressive and hegemonic actions of global imperialism".

The U.S. government was promoting a council that had power to punish and sanction countries of the South, which would allow Washington to continue its tradition of manipulating developing countries, the statement said.

Such a council would be "unable to take effective decisions against racism, xenophobia and other forms of intolerance, but would concentrate on civil and political rights as conceived by Washington ," the statement added.

Figure 5. Adding a concept synonym

The user can also modify an existing KB concept by adding English synonyms. For example, the word "United States" may be highlighted in a text example, while "U.S." is not. To add a synonym for the "United States" concept, the user navigates to this concept in the KB View, and then clicks on the text "U.S.". A menu offers the choice of adding a synonym to the existing focus concept. Figure 5 illustrates this process.

## 5 Identifying KB Concepts in Text

Elements in a Scone knowledge base represent specific concepts, rather than words or word senses. Each concept is linked with a list of English names (words or phrases). This association between Scone elements and English names is many-to-many.

To map a sentence to the set of concepts that appear there, a dynamic-programming alignment is performed using the English names in the KB as a dictionary. SconeEdit searches for an alignment that covers as much of the input text as possible. The result of aligning an input string with concepts is a set of triples, each consisting of a concept, an offset, and a length. These triples are used directly by the Text Tab to highlight substrings and associate them with KB concepts.

Consider the sentence "Washington, D.C. is a city." Table 1 shows some example Scone concepts and their English names. Given a knowledge

| Concept Name | English Names |
|---|---|
| *Washington-State* | “Washington”, “Washington State”, |
| *Washington-Dc* | “Washington”, “Washington, D.C.” |
| *City* | “city” |

Table 1. Example concepts and their English Name lists

base with these concepts, SconeEdit returns the alignment: (concept: *Washington-DC, offset:* 1, length: 16) (concept: *City*, offset: 23, length: 4).

## 6 Planned Features

A single node in the KB could have hundreds or thousands of outgoing links. For readability, the browser must select a subset of these links to display to the user. We plan to leverage Scone's reasoning ability, along with SconeEdit's document-driven design, to select which nodes are likely to be relevant to the user in the context of the loaded document(s). For example, a user who views subclasses of *disease* in a medical ontology may be presented with thousands of disease types. If the current document loaded into SconeEdit is a document about *food*, Scone may be able to prune the subclasses it lists to only food-borne illnesses.

Another feature we hope to add is better integration with an entire corpus. The current system allows the user to work with individual documents. This could be extended to allow a user to navigate to a particular concept in the knowledge base and retrieve all documents in a corpus containing that concept (in its various forms). These documents could then be used to generate more KB concepts of interest.

## 7 Related Work

To the best of our knowledge, existing ontology and KB editors and viewers do not specifically focus on editing and viewing an ontology or KB in the context of natural language text. Other ontology editors such as Protégé (Gennari, 2002) and OntoEdit (Sure, 2002) offer many features for generating complex ontologies, but do not provide the rich interaction with domain text that is the focus of SconeEdit. The CNet Big Picture (CNet News Online, 2000) is one example of a system that does link ontology knowledge to text, but the concepts in the ontology are limited to a small fixed set.

## Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. The authors would like to thank Vasco Pedro, Eric Nyberg, and Tim Isganitis for their contributions to SconeEdit.

## References

CNet News Online. 2000. *The Big Picture*, http://news.com.com/The+Big+Picture/2030-12_3-5843390.html.

Scott E. Fahlman. 2006. *Scone User's Manual*, http://www.cs.cmu.edu/~sef/scone/.

J. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubezy, H. Eriksson, N. F. Noy, S. W. Tu. 2002. The Evolution of Protégé: An Environment for Knowledge-Based Systems Development. *International Journal of Human-Computer Interaction*, 58(1), pp. 89—123.

Aleš Horák and Pavel Smrž. 2004. VisDic -- WordNet Browsing and Editing Tool. *Proceedings of GWC 2004*, pp. 136—141.

INOH, 2004. INOH Ontology Viewer Website. http://www.inoh.org:8083/ontology-viewer/.

Asako Koike and Toshishisa Takagi, 2004. Gene/protein/family name recognition in biomedical literature. In *Proceedings of BioLINK 2004: Linking Biological Literature, Ontologies, and Databases*, pp. 9-16.

Alicia Tribble and Carolyn Rosé. 2006. Usable Browsers for Ontological Knowledge Acquisition. To appear in *Proceedings of CHI-2006*. Montréal, Canada. April 22-27, 2006.

Xinhuanet. 2006. US accused of blocking approval of new UN human rights body. http://news.xinhuanet.com/english/2006-03/02/content_4247159.htm.

Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer and D. Wenke. OntoEdit: Collaborative Ontology Engineering for the Semantic Web. In *Proceedings of the first International Semantic Web Conference 2002 (ISWC 2002).*

# Factoid Question Answering with Web, Mobile and Speech Interfaces

**E.W.D. Whittaker** **J. Mrozinski** **S. Furui**
Dept. of Computer Science
Tokyo Institute of Technology
2-12-1, Ookayama, Meguro-ku
Tokyo 152-8552 Japan
{edw,mrozinsk,furui}@furui.cs.titech.ac.jp

## Abstract

In this paper we describe the web and mobile-phone interfaces to our multi-language factoid question answering (QA) system together with a prototype speech interface to our English-language QA system. Using a statistical, data-driven approach to factoid question answering has allowed us to develop QA systems in five languages in a matter of months. In the web-based system, which is accessible at `http://asked.jp`, we have combined the QA system output with standard search-engine-like results by integrating it with an open-source web search engine. The prototype speech interface is based around a VoiceXML application running on the Voxeo developer platform. Recognition of the user's question is performed on a separate speech recognition server dedicated to recognizing questions. An adapted version of the Sphinx-4 recognizer is used for this purpose. Once the question has been recognized correctly it is passed to the QA system and the resulting answers read back to the user by speech synthesis. Our approach is modular and makes extensive use of open-source software. Consequently, each component can be easily and independently improved and easily extended to other languages.

## 1 Introduction

The approach to factoid question answering (QA) that we adopt was first described in (Whittaker et al., 2005b) where the details of the mathematical model and how it was trained for English were given. The approach has been successfully evaluated in the 2005 text retrieval conference (TREC) question answering track evaluations (Voorhees and Trang Dang, 2005) where our group placed eleventh out of thirty participants (Whittaker et al., 2005a). Although the TREC QA task is substantially different to web-based QA this evaluation showed that our approach works and provides an objective assessment of its quality. Similarly, for our Japanese language system we have evaluated the performance of our approach on the NTCIR-3 QAC-1 task (Whittaker et al., 2005c). Although our Japanese experiments were applied retrospectively, the results would have placed us in the mid-range of participating systems. In (Whittaker et al., 2006b) we described how our approach could be used for the rapid development of web-based QA systems in five very different languages. It was shown that a developer proficient with the tools, and with access to suitable training data, could build a system in a new language in around 10 hours. In (Whittaker et al., 2006a) we evaluated the performance of the systems for four of our five languages. We give a brief summary of our approach to QA in Section 2.

In this paper we introduce our web-based QA system which is publicly accessible at `http://asked.jp`, permitting questions in English, Japanese, Chinese, Russian and Swedish and is discussed in Section 3. Since answers in factoid QA are inherently well-suited to display on small screens we have also made a mobile-phone interface which is accessible at the same address when using an HTML browser from a mobile phone. This is discussed in Section 4. There are several other QA systems on the web including Brainboost (Brainboost, 2005) and Lexxe (Lexxe, 2005) but they only try to answer questions in English and do not have convenient mobile interfaces.

Entering whole questions rather than just keywords is tedious especially on a mobile-phone so we have also begun to look at speech interfaces. In this paper we describe a prototype speech interface to our English-language QA system. This prototype is currently intended primarily as a platform for further research into speech recognition and answering of questions from an acoustic modelling point-of-view (e.g. low-bandwidth, low-quality VoIP channel), from a language modelling perspective (e.g. irregular word order in questions vs. text, and very large out-of-vocabulary problem) and also in terms of dialog modelling. There have been several attempts at speech interfaces to QA systems in the literature e.g. (Schofield and Zheng, 2003) but as far as we know ours is the only system that is publicly accessible. We discuss this interface in Section 5.

## 2 Statistical pattern classification approach to QA

The answer to a question depends primarily on the question itself but also on many other factors such as the person asking the question, the location of the person, what questions the person has asked before, and so on. For simplicity, we choose to consider only the dependence of an answer $A$ on the question $Q$. In particular, we hypothesize that the answer $A$ depends on two sets of features extracted from $Q$: $W = \mathcal{W}(Q)$ and $X = \mathcal{X}(Q)$ as follows:

$$P(A \mid Q) = P(A \mid W, X), \qquad (1)$$

where $W$ can be thought of as a set of $l_W$ features describing the "question-type" part of $Q$ such as *who, when, where, which,* etc. and $X$ is a set of features comprising the "information-bearing" part of $Q$ i.e. what the question is actually about and what it refers to. For example, in the questions, *Where is Mount Everest?* and *How high is Mount Everest?* the information-bearing component is identical in both cases whereas the question-type component is different.

Finding the best answer $\hat{A}$ involves a search over all $A$ for the one which maximizes the probability of the above model:

$$\hat{A} = \arg\max_A P(A \mid W, X). \qquad (2)$$

This is guaranteed to give us the optimal answer in a maximum likelihood sense if the probability distribution is the correct one. Making various conditional independence assumptions to simplify modelling we obtain the final optimisation criterion:

$$\arg\max_A \underbrace{P(A \mid X)}_{\substack{retrieval\\ model}} \cdot \underbrace{P(W \mid A)}_{\substack{filter\\ model}}. \qquad (3)$$

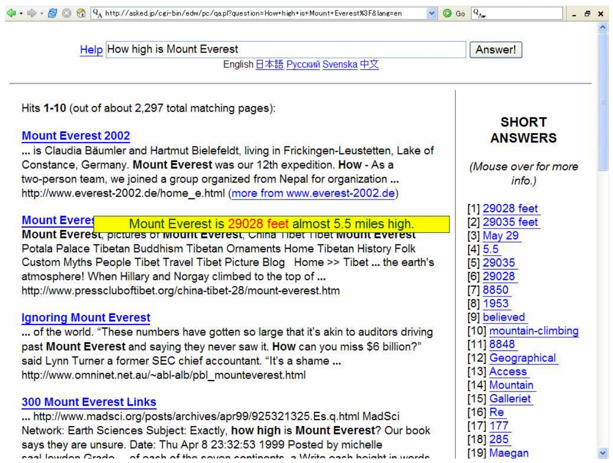
The $P(A \mid X)$ model is essentially a language model which models the probability of an answer sequence $A$ given a set of information-bearing features $X$. It models the proximity of $A$ to features in $X$. This model is referred to as the *retrieval model.*

The $P(W \mid A)$ model matches an answer $A$ with features in the question-type set $W$. Roughly speaking this model relates ways of asking a question with classes of valid answers. For example, it associates names of people or companies with *who*-type questions. In general, there are many valid and equiprobable $A$ for a given $W$ so this component can only re-rank candidate answers retrieved by the retrieval model. Consequently, we call it the *filter model.*

## 3 Web interface

The web-based interface to our QA systems has been accessible at `http://asked.jp` since December 2005 and although still primarily a research system and not widely advertised it attracts around five unique users a day. Currently we do not perform language detection for an input question so the user must first select a language-specific system before inputting a question in a language other than English.

In Figure 1 we show the results page for the question *"How high is Mount Everest?"*. As can be seen the left-hand side of the page contains the familiar title, link and summaries of pages relevant to the query that is common to most of today's web search engines. These results are produced by an open-source web search engine which is run locally and currently contains about 100 million web-pages in its database. Down the right-hand side of the results page we present the answers that were found by our QA system. These answers are presented in order of probability as determined by Equation (3). When the mouse is rolled over an answer a Java-script pop-up box is displayed that shows more context for a given answer. This allows the user to determine more rapidly the validity of an answer and also partially compensates for inaccurate answer identification by the system. Each answer can also be clicked on whereupon the user is redirected to the page from which the answer was taken. This re-direction is effected through a redirect via our own web-server so that for a given question we can see which answers were clicked on. Eventually, it is hoped this could be used for unsupervised system adaptation.


http://asked.jp/cgi-bin/edw/pc/qa.pl?question=How+high+is+Mount+Everest%3F&lang=en

Help How high is Mount Everest Answer!

English 日本語 Русский Svenska 中文

Hits **1-10** (out of about 2,297 total matching pages):

Mount Everest 2002
... is Claudia Bäumler and Hartmut Bielefeldt, living in Frickingen-Leustetten, Lake of Constance, Germany. **Mount Everest** was our 12th expedition. **How** - As a two-person team, we joined a group organized from Nepal for organization ...
http://www.everest-2002.de/home_e.html (more from www.everest-2002.de)

Mount Everes
Mount Everest is 29028 feet almost 5.5 miles high.
**Mount Everest**, pictures of **Mount Everest**, China Tibet Tibet **Mount Everest** Potala Palace Tibetan Buddhism Tibetan Ornaments Home Tibetan History Folk Custom Myths People Tibet Travel Tibet Picture Blog Home >> Tibet ... the earth's atmosphere! When Hillary and Norgay climbed to the top of ...
http://www.presscluboftibet.org/china-tibet-28/mount-everest.htm

Ignoring Mount Everest
... of the world. "These numbers have gotten so large that it's akin to auditors driving past **Mount Everest** and saying they never saw it. **How** can you miss $6 billion?" said Lynn Turner a former SEC chief accountant. "It's a shame ...
http://www.omninet.net.au/~abl-alb/pbl_mounteverest.html

300 Mount Everest Links
... http://www.madsci.org/posts/archives/apr99/925321325.Es.q.html MadSci Network: Earth Sciences Subject: Exactly, **how high** is **Mount Everest**? Our book says they are unsure. Date: Thu Apr 8 23:32:53 1999 Posted by michelle

**SHORT ANSWERS**

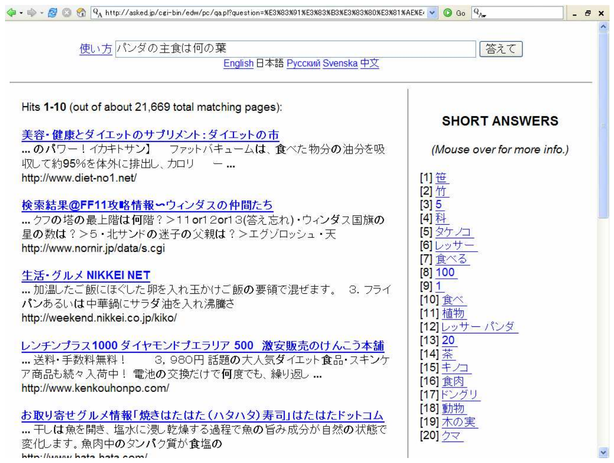
*(Mouse over for more info.)*

[1] 29028 feet
[2] 29035 feet
[3] May 29
[4] 5.5
[5] 29035
[6] 29028
[7] 8850
[8] 1953
[9] believed
[10] mountain-climbing
[11] 8848
[12] Geographical
[13] Access
[14] Mountain
[15] Galleriet
[16] Re
[17] 177
[18] 285
[19] Maegan

Figure 1: Results page for *"How high is Mount Everest?"*.

The same basic layout and design is repeated for each of the five language-specific systems. In Figure 2 we show the results page for the Japanese question of *"What plant do Pandas eat?"*.

The average response time to present the full results page for a question in each language is currently around 10 seconds. The web-search and QA systems are run in parallel and the outputs combined when both are complete.


http://asked.jp/cgi-bin/edw/pc/qa.pl?question=%E3%83%91%E3%83%B3%E3%83%80%E3%81%AE%E

使い方 パンダの主食は何の葉 答えて

English 日本語 Русский Svenska 中文

Hits **1-10** (out of about 21,669 total matching pages):

美容・健康とダイエットのサプリメント：ダイエットの市
... のパワー！イカキトサン】　ファットバキュームは、食べた物分の油分を吸収して約95%を体外に排出し、カロリ　ー ...
http://www.diet-no1.net/

検索結果@FF11攻略情報～ウィンダスの仲間たち
... クフの塔の最上階は何階？＞11or12or13(答え忘れ)・ウィンダス国旗の星の数は？＞5・北サンドの迷子の父親は？＞エグゾロッシュ・天
http://www.nornir.jp/data/s.cgi

生活・グルメ NIKKEI NET
... 加温したご飯にほぐした卵を入れ玉かけご飯の要領で混ぜます。　3. フライパンあるいは中華鍋にサラダ油を入れ沸騰さ
http://weekend.nikkei.co.jp/kiko/

レンチンプラス1000 ダイヤモンドプエラリア 500　激安販売のけんこう本舗
... 送料・手数料無料！　　3，980円 話題の大人気ダイエット食品・スキンケア商品も続々入荷中！ 電池の交換だけで何度でも、繰り返し ...
http://www.kenkouhonpo.com/

お取り寄せグルメ情報「焼きはたはた（ハタハタ）寿司」はたはたドットコム
... 干しは魚を開き、塩水に浸し乾燥する過程で魚の旨み成分が自然の状態で変化します。魚肉中のタンパク質が食塩の

**SHORT ANSWERS**

*(Mouse over for more info.)*

[1] 笹
[2] 竹
[3] 5
[4] 科
[5] タケノコ
[6] レッサー
[7] 食べる
[8] 100
[9] 1
[10] 食べ
[11] 植物
[12] レッサー パンダ
[13] 20
[14] 茶
[15] キノコ
[16] 食肉
[17] ドングリ
[18] 動物
[19] 木の実
[20] クマ

Figure 2: Results page for *"What plant do Pandas eat?"* in Japanese.

## 4 Mobile-phone interface

Since the priorities with a mobile-phone interface revolve around speed, display size and cost to the user, the interface is basically a whittled down version of the web-based interface described in the previous section. The only requirement for being able to use the mobile phone interface is that the phone must contain an HTML browser. In countries like Japan this has been fairly standard for many years but it is expected that this will become more common worldwide in the near future with the continued roll-out of 3G mobile phone services.

For the mobile-phone interface the standard web-search results section has been removed entirely from the results section and instead only the top 20 short answers are displayed without pop-up boxes or corresponding context. Such a strategy minimizes the number of bytes transmitted and ensures that most answers are adequately displayed on most mobile-phone interfaces with a minimum amount of scrolling. Although not yet implemented we aim to allow users to click on an answer and be taken to the part of the page that contains the answer rather than loading a whole page which could sometimes be several megabytes in size.

## 5 Speech interface

Our implementation of the speech interface to the QA system was greatly simplified by the availability of the Voxeo developer platform[1] which provides free access, for development purposes, to a VoiceXML browser running our application. The application can be accessed through: (i) a U.S. telephone number at `(800) 289-5570` then `PIN:9991423955`; (ii) SIP VoIP clients at (`SIP:9991423955sip.voxeo.net`); (iii) Free World Dialup at (`**86919991423955`); and (iv) SkypeOut at (`+99000936 9991423955`).

Since most VoiceXML applications are designed for use with small vocabulary, rule-based grammars we only use VoiceXML and Voxeo's browser to handle negotiation of the questions and answers with the user through simple dialogs. The recognition of the question itself is performed using a dedicated large-vocabulary speech recognizer with a language model (LM) trained on English-language questions. The speech recognizer we use is the open-source Sphinx-4 recognizer (Walker et al., 2004) which runs in a server mode and has been adapted to use more complex LMs than those permitted by the default ARPA format word $N$-gram LMs. Currently we use a linear interpolation of a word and class-based trigram LM each of which were trained on a large corpus of English-language questions (Hallmarks, 2002)—the same data used to train the English-language QA system (Whittaker et al., 2005b).

## 6 Conclusion and Further work

Having recapped a basic overview of our statistical approach to question answering (QA), in this paper we have described the web and mobile-phone interfaces to our multi-language QA system and how they can be accessed. In addition, we have described our first attempt at a prototype speech interface which will be used as a platform for future research. Eventually our aim is to make the QA performance of the speech interface the same as that obtained through the web and mobile-phone interfaces. This will be achieved through a combination of acoustic, language and dialog model adaptation on the speech side, and making the QA system more robust to underspecified and errorful questions on the QA side. We think these demonstration systems show significant progress has already been made and give a hint of how information access to QA systems might be achieved in the near future.

[1]`http://www.voxeo.com/developers`

## 7 Acknowledgments

This research was supported by JSPS and the Japanese government 21st century COE programme. The authors also wish to thank Dietrich Klakow for all his contributions.

## References

Brainboost. 2005. `http://www.brainboost.com`.

Academic Hallmarks. 2002. Knowledge Master Educational Software. PO Box 998, Durango, CO 81302 http://www.greatauk.com/.

Lexxe. 2005. `http://www.lexxe.com`.

E. Schofi eld and Z. Zheng. 2003. A Speech Interface for Open-domain Question-answering. In *Proceedings of the 41st Annual Meeting of the ACL*, Sapporo, Japan, July.

E.M. Voorhees and H. Trang Dang. 2005. Overview of the TREC 2005 Question Answering Track. In *Proceedings of the 14th Text Retrieval Conference*.

W. Walker et al. 2004. Sphinx-4: A Flexible Open Source Framework for Speech Recognition. Technical report, Sun Microsystems Inc.

E.W.D. Whittaker, P. Chatain, S. Furui, and D. Klakow. 2005a. TREC2005 Question Answering Experiments at Tokyo Institute of Technology. In *Proceedings of the 14th Text Retrieval Conference*.

E.W.D. Whittaker, S. Furui, and D. Klakow. 2005b. A Statistical Pattern Recognition Approach to Question Answering using Web Data. In *Proceedings of Cyberworlds*.

E.W.D. Whittaker, J. Hamonic, and S. Furui. 2005c. A Unifi ed Approach to Japanese and English Question Answering. In *Proceedings of NTCIR-5*.

E.W.D. Whittaker, J. Hamonic, T. Klingberg, Y. Dong, and S. Furui. 2006a. Monolingual Web-based Factoid Question Answering in Chinese, Swedish, English and Japanese. In *Proceedings of the Workshop on Multi-language Question Answering, EACL*.

E.W.D. Whittaker, J. Hamonic, T. Klingberg, Y. Dong, and S. Furui. 2006b. Rapid Development of Web-based Monolingual Question Answering Systems. In *Proceedings of ECIR2006*.

# AUTOMATED QUALITY MONITORING FOR CALL CENTERS USING SPEECH AND NLP TECHNOLOGIES

*G. Zweig, O. Siohan, G. Saon, B. Ramabhadran, D. Povey, L. Mangu and B. Kingsbury*

IBM T.J. Watson Research Center, Yorktown Heights, NY 10598

## ABSTRACT

This paper describes an automated system for assigning quality scores to recorded call center conversations. The system combines speech recognition, pattern matching, and maximum entropy classification to rank calls according to their measured quality. Calls at both ends of the spectrum are flagged as "interesting" and made available for further human monitoring. In this process, the ASR transcript is used to answer a set of standard quality control questions such as "did the agent use courteous words and phrases," and to generate a question-based score. This is interpolated with the probability of a call being "bad," as determined by maximum entropy operating on a set of ASR-derived features such as "maximum silence length" and the occurrence of selected n-gram word sequences. The system is trained on a set of calls with associated manual evaluation forms. We present precision and recall results from IBM's North American Help Desk indicating that for a given amount of listening effort, this system triples the number of bad calls that are identified, over the current policy of randomly sampling calls. The application that will be demonstrated is a research prototype that was built in conjunction with IBM's North American call centers.

## 1. INTRODUCTION

Every day, tens of millions of help-desk calls are recorded at call centers around the world. As part of a typical call center operation a random sample of these calls is normally re-played to human monitors who score the calls with respect to a variety of quality related questions, e.g.

- Was the account successfully identified by the agent?
- Did the agent request error codes/messages to help determine the problem?
- Was the problem resolved?
- Did the agent maintain appropriate tone, pitch, volume and pace?

This process suffers from a number of important problems: first, the monitoring at least doubles the cost of each call (first an operator is paid to take it, then a monitor to evaluate it). This causes the second problem, which is that therefore only a very small sample of calls, e.g. a fraction of a percent, is typically evaluated. The third problem arises from the fact that most calls are ordinary and uninteresting; with random sampling, the human monitors spend most of their time listening to uninteresting calls.

This work describes an automated quality-monitoring system that addresses these problems. Automatic speech recognition is used to transcribe 100% of the calls coming in to a call center, and default quality scores are assigned based on features such as key-words, key-phrases, the number and type of hesitations, and the average silence durations. The default score is used to rank the calls from worst-to-best, and this sorted list is made available to the human evaluators, who can thus spend their time listening only to calls for which there is some a-priori reason to expect that there is something interesting.

The automatic quality-monitoring problem is interesting in part because of the variability in how hard it is to answer the questions. Some questions, for example, "Did the agent use courteous words and phrases?" are relatively straightforward to answer by looking for key words and phrases. Others, however, require essentially human-level knowledge to answer; for example one company's monitors are asked to answer the question "Did the agent take ownership of the problem?" Our work focuses on calls from IBM's North American call centers, where there is a set of 31 questions that are used to evaluate call-quality. Because of the high degree of variability found in these calls, we have investigated two approaches:

1. Use a partial score based only on the subset of questions that can be reliably answered.
2. Use a maximum entropy classifier to map directly from ASR-generated features to the probability that a call is bad (defined as belonging to the bottom 20% of calls).

We have found that both approaches are workable, and we present final results based on an interpolation between the two scores. These results indicate that for a fixed amount of listening effort, the number of bad calls that are identified approximately triples with our call-ranking approach. Surprisingly, while there has been significant previous scholarly research in automated call-routing and classification in the call center , e.g. [1, 2, 3, 4, 5], there has been much less in automated quality monitoring per se.

## 2. ASR FOR CALL CENTER TRANSCRIPTION

### 2.1. Data

The speech recognition systems were trained on approximately 300 hours of 6kHz, mono audio data collected at one of the IBM call centers located in Raleigh, NC. The audio was manually transcribed and speaker turns were explicitly marked in the word transcriptions but not the corresponding times. In order to detect speaker changes in the training data, we did a forced-alignment of the data and chopped it at speaker boundaries. The test set consists of 50 calls with 113 speakers totaling about 3 hours of speech.

### 2.2. Speaker Independent System

The raw acoustic features used for segmentation and recognition are perceptual linear prediction (PLP) features. The features are

| Segmentation/clustering | Adaptation | WER |
|---|---|---|
| Manual | Off-line | 30.2% |
| Manual | Incremental | 31.3% |
| Manual | No Adaptation | 35.9% |
| Automatic | Off-line | 33.0% |
| Automatic | Incremental | 35.1% |

**Table 1**. ASR results depending on segmentation/clustering and adaptation type.

| Accuracy | Top 20% | Bottom 20% |
|---|---|---|
| Random | 20% | 20% |
| QA | 41% | 30% |

**Table 2**. Accuracy for the Question Answering system.

mean-normalized 40-dimensional LDA+MLLT features. The SI acoustic model consists of 50K Gaussians trained with MPE and uses a quinphone cross-word acoustic context. The techniques are the same as those described in [6].

### 2.3. Incremental Speaker Adaptation

In the context of speaker-adaptive training, we use two forms of feature-space normalization: vocal tract length normalization (VTLN) and feature-space MLLR (fMLLR, also known as constrained MLLR) to produce canonical acoustic models in which some of the non-linguistic sources of speech variability have been reduced. To this canonical feature space, we then apply a discriminatively trained transform called fMPE [7]. The speaker adapted recognition model is trained in this resulting feature space using MPE.

We distinguish between two forms of adaptation: off-line and incremental adaptation. For the former, the transformations are computed per conversation-side using the full output of a speaker independent system. For the latter, the transformations are updated incrementally using the decoded output of the speaker adapted system up to the current time. The speaker adaptive transforms are then applied to the future sentences. The advantage of incremental adaptation is that it only requires a single decoding pass (as opposed to two passes for off-line adaptation) resulting in a decoding process which is twice as fast. In Table 1, we compare the performance of the two approaches. Most of the gain of full offline adaptation is retained in the incremental version.

#### *2.3.1. Segmentation and Speaker Clustering*

We use an HMM-based segmentation procedure for segmenting the audio into speech and non-speech prior to decoding. The reason is that we want to eliminate the non-speech segments in order to reduce the computational load during recognition. The speech segments are clustered together in order to identify segments coming from the same speaker which is crucial for speaker adaptation. The clustering is done via k-means, each segment being modeled by a single diagonal covariance Gaussian. The metric is given by the symmetric K-L divergence between two Gaussians. The impact of the automatic segmentation and clustering on the error rate is indicated in Table 1.

| Accuracy | Top 20% | Bottom 20% |
|---|---|---|
| Random | 20% | 20% |
| ME | 49% | 36% |

**Table 3**. Accuracy for the Maximum Entropy system.

| Accuracy | Top 20% | Bottom 20% |
|---|---|---|
| Random | 20% | 20% |
| ME + QA | 53% | 44% |

**Table 4**. Accuracy for the combined system.

## 3. CALL RANKING

### 3.1. Question Answering

This section presents automated techniques for evaluating call quality. These techniques were developed using a training/development set of 676 calls with associated manually generated quality evaluations. The test set consists of 195 calls.

The quality of the service provided by the help-desk representatives is commonly assessed by having human monitors listen to a random sample of the calls and then fill in evaluation forms. The form for IBM's North American Help Desk contains 31 questions. A subset of the questions can be answered easily using automatic methods, among those the ones that check that the agent followed the guidelines e.g.

- Did the agent follow the appropriate closing script?
- Did the agent identify herself to the customer?

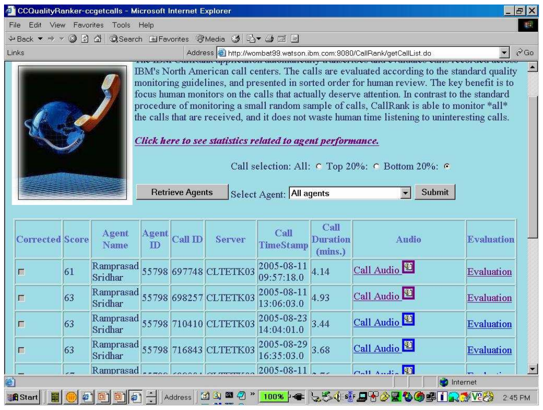
But some of the questions require human-level knowledge of the world to answer, e.g.

- Did the agent ask pertinent questions to gain clarity of the problem?
- Were all available resources used to solve the problem?

We were able to answer 21 out of the 31 questions using pattern matching techniques. For example, if the question is "Did the agent follow the appropriate closing script?", we search for "THANK YOU FOR CALLING", "ANYTHING ELSE" and "SERVICE REQUEST". Any of these is a good partial match for the full script, "Thank you for calling, is there anything else I can help you with before closing this service request?" Based on the answer to each of the 21 questions, we compute a score for each call and use it to rank them. We label a call in the test set as being *bad*/*good* if it has been placed in the bottom/top 20% by human evaluators. We report the accuracy of our scoring system on the test set by computing the number of *bad* calls that occur in the bottom 20% of our sorted list and the number of *good* calls found in the top 20% of our list. The accuracy numbers can be found in Table 2.

### 3.2. Maximum Entropy Ranking

Another alternative for scoring calls is to find arbitrary features in the speech recognition output that correlate with the outcome of a call being in the bottom 20% or not. The goal is to estimate the probability of a call being *bad* based on features extracted from the automatic transcription. To achieve this we build a maximum



CCQualityRanker-ccgetcalls - Microsoft Internet Explorer

Address http://wombat99.watson.ibm.com:9080/CallRank/getCallList.do

IBM's North American call centers. The calls are evaluated according to the standard quality monitoring guidelines, and presented in sorted order for human review. The key benefit is to focus human monitors on the calls that actually deserve attention. In contrast to the standard procedure of monitoring a small random sample of calls, CallRank is able to monitor *all* the calls that are received, and it does not waste human time listening to uninteresting calls.

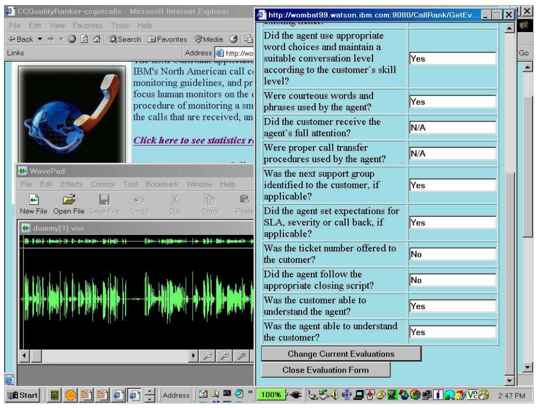
*Click here to see statistics related to agent performance.*

Call selection: All: ○ Top 20%: ○ Bottom 20%: [✓ ●]

Retrieve Agents | Select Agent: All agents | Submit

| Corrected | Score | Agent Name | Agent ID | Call ID | Server | Call TimeStamp | Call Duration (mins.) | Audio | Evaluation |
|---|---|---|---|---|---|---|---|---|---|
| ☐ | 61 | Ramprasad Sridhar | 55798 | 697748 | CLTETK03 | 2005-08-11 09:57:18.0 | 4.14 | Call Audio | Evaluation |
| ☐ | 63 | Ramprasad Sridhar | 55798 | 698257 | CLTETK03 | 2005-08-11 13:06:03.0 | 4.93 | Call Audio | Evaluation |
| ☐ | 63 | Ramprasad Sridhar | 55798 | 710410 | CLTETK03 | 2005-08-23 14:04:01.0 | 3.44 | Call Audio | Evaluation |
| ☐ | 63 | Ramprasad Sridhar | 55798 | 716843 | CLTETK03 | 2005-08-29 16:35:03.0 | 3.68 | Call Audio | Evaluation |
|  |  | Ramprasad |  |  |  | 2005-08-11 |  |  |  |

**Fig. 1**. Display of selected calls.



http://wombat99.watson.ibm.com:9080/CallRank/GetEv...

|  |  |
|---|---|
| Did the agent use appropriate word choices and maintain a suitable conversation level according to the customer`s skill level? | Yes |
| Were courteous words and phrases used by the agent? | Yes |
| Did the customer receive the agent`s full attention? | N/A |
| Were proper call transfer procedures used by the agent? | N/A |
| Was the next support group identified to the customer, if applicable? | Yes |
| Did the agent set expectations for SLA, severity or call back, if applicable? | Yes |
| Was the ticket number offered to the cutomer? | No |
| Did the agent follow the appropriate closing script? | No |
| Was the customer able to understand the agent? | Yes |
| Was the agent able to understand the customer? | Yes |

**Fig. 2**. Interface to listen to audio and update the evaluation form.

entropy based system which is trained on a set of calls with associated transcriptions and manual evaluations. The following equation is used to determine the score of a call $C$ using a set of $N$ predefined features:

$$P(class/C) = \frac{1}{Z} exp(\sum_{i=1}^{N} \lambda_i f_i(class, C)) \qquad (1)$$

where $class \in \{bad, not-bad\}$, $Z$ is a normalizing factor, $f_i()$ are indicator functions and $\{\lambda_i\}_{\{i=1,N\}}$ are the parameters of the model estimated via iterative scaling [8].

Due to the fact that our training set contained under 700 calls, we used a hand-guided method for defining features. Specifically, we generated a list of VIP phrases as candidate features, e.g. "THANK YOU FOR CALLING", and "HELP YOU". We also created a pool of generic ASR features, e.g. "number of hesitations", "total silence duration", and "longest silence duration". A decision tree was then used to select the most relevant features and the threshold associated with each feature. The final set of features contained 5 generic features and 25 VIP phrases. If we take a look at the weights learned for different features, we can see that if a call has many hesitations and long silences then most likely the call is *bad*.

We use $P(bad|C)$ as shown in Equation 1 to rank all the calls. Table 3 shows the accuracy of this system for the bottom and top 20% of the test calls.

At this point we have two scoring mechanisms for each call: one that relies on answering a fixed number of evaluation questions and a more global one that looks across the entire call for hints. These two scores are both between 0 and 1, and therefore can be interpolated to generate one unique score. After optimizing the interpolation weights on a held-out set we obtained a slightly higher weight (0.6) for the maximum entropy model. It can be seen in Table 4 that the accuracy of the combined system is greater that the accuracy of each individual system, suggesting the complementarity of the two initial systems.

## 4. END-TO-END SYSTEM PERFORMANCE

### 4.1. Application

This section describes the user interface of the automated quality monitoring application. As explained in Section 1, the evaluator scores calls with respect to a set of quality-related questions after listening to the calls. To aid this process, the user interface provides an efficient mechanism for the human evaluator to select calls, e.g.

- All calls from a specific agent sorted by score
- The top 20% or the bottom 20% of the calls from a specific agent ranked by score
- The top 20% or the bottom 20% of all calls from all agents

The automated quality monitoring user interface is a J2EE web application that is supported by back-end databases and content management systems [1] The displayed list of calls provides a link to the audio, the automatically filled evaluation form, the overall score for this call, the agent's name, server location, call id, date and duration of the call (see Figure 1). This interface now gives the agent the ability to listen to interesting calls and update the answers in the evaluation form if necessary (audio and evaluation form illustrated in 2). In addition, this interface provides the evaluator with the ability to view summary statistics (average score) and additional information about the quality of the calls. The overall system is designed to automatically download calls from multiple locations on a daily-basis, transcribe and index them, thereby making them available to the supervisors for monitoring. Calls spanning a month are available at any given time for monitoring purposes.

### 4.2. Precision and Recall

This section presents precision and recall numbers for the identification of "bad" calls. The test set consists of 195 calls that were manually evaluated by call center personnel. Based on these manual scores, the calls were ordered by quality, and the bottom 20% were deemed to be "bad." To retrieve calls for monitoring, we sort the calls based on the automatically assigned quality score and return the worst. In our summary figures, precision and recall are plotted as a function of the number of calls that are selected for monitoring. This is important because in reality only a small number of calls can receive human attention. Precision is the ratio

[1]In our case, the backend consists of DB2 and IBM's Websphere Information Integrator for Content and the application is hosted on Websphere 5.1.)

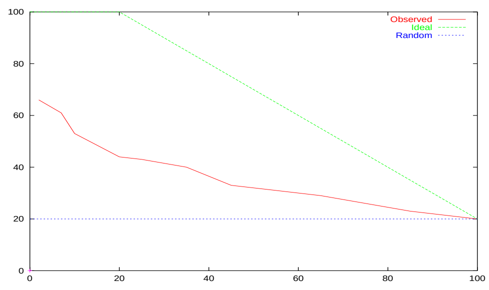


**Fig. 3**. Precision for the bottom 20% of the calls as a function of the number of calls retrieved.

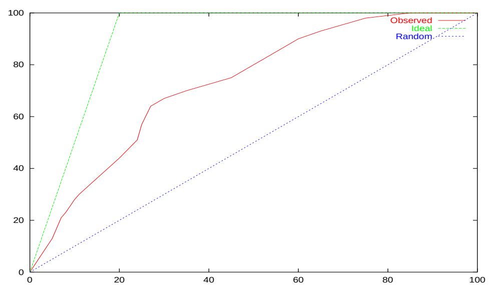


**Fig. 4**. Recall for the bottom 20% of the calls.

of bad calls retrieved to the total number of calls monitored, and recall is the ratio of the number of bad calls retrieved to the total number of bad calls in the test set. Three curves are shown in each plot: the actually observed performance, performance of random selection, and oracle or ideal performance. Oracle performance shows what would happen if a perfect automatic ordering of the calls was achieved.

Figure 3 shows precision performance. We see that in the monitoring regime where only a small fraction of the calls are monitored, we achieve over 60% precision. (Further, if 20% of the calls are monitored, we still attain over 40% precision.)

Figure 4 shows the recall performance. In the regime of low-volume monitoring, the recall is midway between what could be achieved with an oracle, and the performance of random-selection.

Figure 5 shows the ratio of the number of bad calls found with our automated ranking to the number found with random selection. This indicates that in the low-monitoring regime, our automated technique triples efficiency.

### 4.3. Human vs. Computer Rankings

As a final measure of performance, in Figure 6 we present a scatterplot comparing human to computer rankings. We do not have calls that are scored by two humans, so we cannot present a human-human scatterplot for comparison.

## 5. CONCLUSION

This paper has presented an automated system for quality monitoring in the call center. We propose a combination of maximum-entropy classification based on ASR-derived features, and question answering based on simple pattern-matching. The system can either be used to replace human monitors, or to make them more

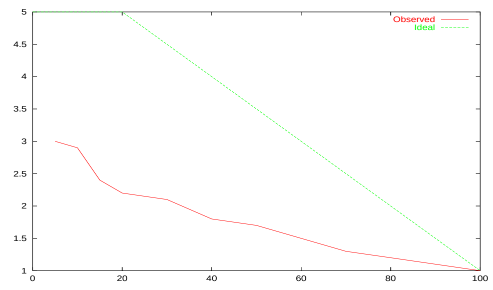


**Fig. 5**. Ratio of bad calls found with QTM to Random selection as a function of the number of bad calls retrieved.

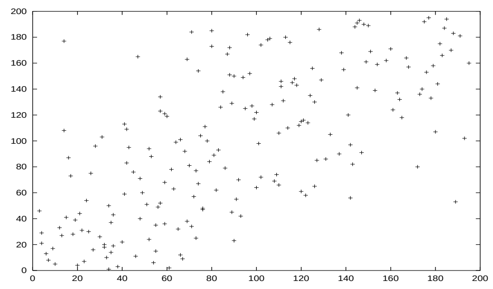


**Fig. 6**. Scatter plot of Human vs. Computer Rank.

efficient. Our results show that we can triple the efficiency of human monitors in the sense of identifying three times as many bad calls for the same amount of listening effort.

## 6. REFERENCES

[1] J. Chu-Carroll and B. Carpenter, "Vector-based natural language call routing," *Computational Linguistics*, 1999.

[2] P. Haffner, G. Tur, and J. Wright, "Optimizing svms for complex call classification," 2003.

[3] M. Tang, B. Pellom, and K. Hacioglu, "Call-type classification and unsupervised training for the call center domain," in *ARSU-2003*, 2003.

[4] D. Hakkani-Tur, G. Tur, M. Rahim, and G. Riccardi, "Unsupervised and active learning in automatic speech recognition for call classification," in *ICASSP-04*, 2004.

[5] C. Wu, J. Kuo, E.E. Jan, V. Goel, and D. Lubensky, "Improving end-to-end performance of call classification through data confusion reduction and model tolerance enhancement," in *Interspeech-05*, 2005.

[6] H. Soltau, B. Kingsbury, L. Mangu, D. Povey, G. Saon, and G. Zweig, "The ibm 2004 conversational telephony system for rich transcription," in *Eurospeech-2005*, 2005.

[7] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *ICASSP-2005*, 2004.

[8] A. Berger, S. Della Pietra, and V. Della Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, 1996.

# Author Index