An Intelligent Listening Framework for Capturing Encounter Notes from a Doctor-Patient Dialog

Jeffrey Klann
MIT, Cambridge MA
Regenstrief Institute, Indianapolis IN
Indiana University, Indianapolis IN
jklann@regenstrief.org

Peter Szolovits
MIT, Cambridge MA
psz@mit.edu

Abstract

Capturing accurate and machine-interpretable primary data from clinical encounters is a challenging task, yet critical to the integrity of the practice of medicine. We explore the intriguing possibility that technology can help accurately capture structured data from the clinical encounter using a combination of automated speech recognition (ASR) systems and tools for extraction of clinical meaning from narrative medical text. Our goal is to produce a displayed evolving encounter note, visible and editable (using speech) during the encounter. This is very ambitious, and so far we have taken only the most preliminary steps. Here we report a simple proof-of-concept system and the design of the more comprehensive one we are building, discussing both the engineering design and challenges encountered. Without a formal evaluation, we were encouraged by our initial results, so we conclude with proposed next steps.

1. Introduction

Capturing accurate structured primary data from clinical encounters is critical to the integrity of medical practice. Furthermore, research in translational medicine also depends on our ability to document patients' clinical conditions so that we can relate these to the enormous new data sets that we can gather about patients' genes. Unfortunately, many studies document deficiencies in the record-keeping process as currently practiced by clinicians. Early studies show that actual medical records often fail to include critical information. A 1971 Army study reported critical missing data from the medical record in 10–70% of cases [17]. A 1975 study found significant discrepancies between 51 tape-recorded doctor-patient conversations and their record [18]. More recent studies of a similar sort demonstrate that the problem persists to today's generation of physicians [12, 7]. Although current practice recommends the adoption of computerized records and computerized physician order entry [9], these trends are met with resistance in part because they take additional time from the practice of already busy doctors. [14]

We explore the intriguing possibility that we can bring technology to bear on the problem of accurately capturing machine-interpretable data from the clinical encounter using a combination of automated speech recognition (ASR) systems and tools for extraction of clinical meaning from narrative medical text. Our goal is to instrument the locale of a clinical encounter (such as a doctor's office or examination room) with one or more microphones that listen to two-party conversations, transcribe them using ASR technology, annotate them using medical natural language processing (MNLP) tools, and then integrate the data they have extracted into a displayed evolving structured encounter note that is visible to both physician and patient and that can be edited by them using a natural speech and pointing interface to correct errors and complete the record.

Other researchers have mounted efforts to capture accurate patient records more automatically through technology (e.g. [13], [10]), but our research is timely and novel for a number of reasons. For one, most serious efforts in this area occurred long enough ago that the technologies available were inadequate to the task. The last decade has brought many serious improvements to ASR and MNLP. Second, we believe we are suggesting a new approach – one that will utilize conversational interaction in an office visit and will enable the patient and provider to interact with the system during the encounter. Third, most research in ASR has focused on transcribing speech (e.g. [2]); however, we propose to use ASR in an entirely different way. Rather than capturing a simple transcript, we are using MNLP techniques to extract a structured and coded encounter summary.

This is a very ambitious goal, and so far we have taken

978-1-4244-2890-8/08/$25.00 ©2008 IEEE

only the most preliminary steps toward its fulfillment. Here we report a simple proof-of-concept system and the more comprehensive one we are building. The proof-of-concept permits a lash-up of Dragon's well-known Naturally Speaking ASR system [11] with an MNLP system called CaRE (Category and Relationship Extractor) built in our laboratory by Sibanda [15]. These two components allow us to experiment with recording at least one side of a conversation, finding clinically significant terms in the recognized speech, and summarizing them in a draft of the encounter note. The more comprehensive system uses GATE, the General Architecture for Text Engineering [5] to more thoroughly integrate the different components of this task.

2. Engineering and Integration

We chose to use one of the most successful commercial ASR systems available, Dragon's Naturally Speaking (DNS), for interpretation of speech inputs. Colleagues at Nuance, which produces and markets DNS, have made available to us a well-documented System Development Kit (SDK) for DNS that allowed us to integrate its capabilities with other programs. They have also given us use of several copies of their Medical Edition, which is widely used as a transcription tool for doctors and has demonstrated good accuracy on medical speech [6]. CaRE, the text-based language processing tool we adopted, was implemented in a combination of Java and Perl programs that also invoke a number of large pre-packaged utilities such as a Support Vector Machine (SVM) based learning system [4], the Brill tagger [3] that uses statistical models to identify the likely parts of speech of words, and the Link Grammar Parser [16] that determines the syntactic structure of sentences and sentence fragments. It also includes custom programs that make use of a local copy of the UMLS metathesaurus [8] to identify the semantic types of words and phrases found in the text. Applied to text from hospital discharge summaries, CaRE achieved an F-measure above 90% for retrieval of relevant medical concepts. Sibanda also describes a component that recognizes relationships among words and phrases, but we have not yet exploited this capability.

Our proof-of-concept system consists of a Java program that presents to DNS a text window (hidden from the user) into which it can type, much as it normally does when used for simple dictation.[1] This program observes this input window and, when enough input has been gathered, invokes CaRE to try to interpret those data. It then presents the interpretation in a second window, highlighting words and terms that have been identified as clinically important ones, and showing by color highlighting the semantic types of the recognized terms. The left side of Figure 2 shows an example of spoken text from a one-sided conversation interpreted by this system. Though the simple approach taken here was sufficient to persuade us that the larger task was feasible, its architecture is clearly not sufficient to handle the many additional interactive components that will be needed for the overall project. Without a formal evaluation of this system, we noted that it was able to make a reasonable interpretation of uncorrected ASR output. Although there are a few false positives, many concepts are correctly recognized with the proper category, including some multi-word phrases that are not built into the UMLS. Because CaRE was trained on text from discharge summaries rather than doctor-patient conversations, we also believe that its performance can be improved significantly once we train it on appropriate corpora (which we do not yet have).

Although the eventual system must include multi-speaker ASR and utilize further MNLP techniques to not only recognize concepts but also fully interpret a two-party conversation, we were encouraged by our initial results. Therefore, we have developed an intelligent listening framework (ILF) that is a step toward our long-term goal of a system that will capture all the relevant data from a doctor-patient encounter into a well-structured encounter note.

```
<TextWithNodes>
  <Node id="0"/> Dr.  <Node id="3"/>
  <Node id="4"/>I<Node id="5"/>
  <Node id="6"/>believe<Node id="13"/>
  <Node id="14"/>I<Node id="15"/>
  <Node id="16"/>have<Node id="20"/>
  <Node id="21"/>a<Node id="22"/>
  <Node id="23"/>brain<Node id="28"/>
  <Node id="29"/>tumor<Node id="34"/>
  .  <Node id="36"/>
</TextWithNodes>
```

Figure 1: A dictated utterance with embedded GATE annotation nodes, as XML.

ILF is implemented as a Java program, running in Microsoft Windows, that uses the DNS SDK to control the background operation of DNS and that also controls GATE to create documents from the outputs of the speech interpretation process. ILF is built as a flexible tool with adjustable granularity parameters to control how often recognized text is sent to an MNLP package for processing and how often the .WAV recording of actual raw speech is sent to disk.[2] GATE is itself a large, Java-based integrated toolkit with useful facilities for managing corpora, multiple annotations,

[1] At the time of its construction, we did not yet have access to the DNS SDK, hence we adopted this more straightforward, if awkward, approach.

[2] It is an unfortunate problem with the DNS SDK that this speech dump must not be done continuously, because dumping to disk always inserts a sentence break. This is because DNS normally uses its interpretation of the beginning of the next utterance to decide whether the pause that caused it to recognize two utterances does or does not correspond to an actual sentence break. Our technique of specifying a granularity attempts to minimize this flaw.

978-1-4244-2890-8/08/$25.00 ©2008 IEEE

None
Disease
MED
Test
Result
Symptom
Drug

Paul I hate to tell you this but you ve got cancer ***By*** *cancer I* ***mean*** *lymphoma* We re going to have to put you on CHEMOTHERAPY Your WBC that s your **white blood cell count** is going to drop below ***2000*** I m very sorry about this but I think we can give you a prescription of LORAZEPAM to relieve your **anxiety** It s possible *that* during the course of TREATMENT you ll have *multiple organ failure* For example your ***lungs are*** going to fill with phlegm It s also possible that you will have **angina** or an *MI* ***that*** is a *myocardial infarction* Do you have any questions for me Well of COURSE we will continue to monitor your condition with **regular blood tests** I just don t know what the results will be

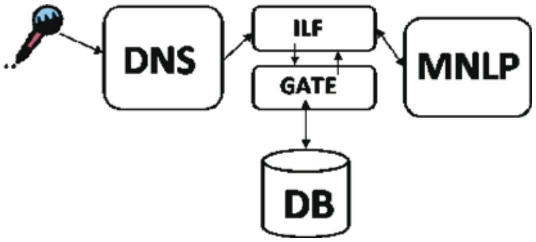

Figure 2: Proof-of-concept output (left), as "heard" by DNS and interpreted by CaRE, and ILF architechture (right). The left pane shows color-coded CaRE-recognized concepts from single party dictation. When combined with other MNLP technologies, this information will be used to automatically produce a coded encounter note.

and interactive text mark-up [5]. We plan to re-create the MNLP processing capabilities of CaRE within the GATE framework, to allow us to experiment with variations that combine different methods for accomplishing its tasks. ILF allows more complex interfacing with MNLP than our original prototype. For example, the MNLP framework can report to ILF that it does not have enough unprocessed text to accurately interpret the speech and can request that more be captured. Currently we use a dummy processing module that simply accepts its inputs without altering or further annotating them. The ILF architecture is diagrammed on the right side of Figure 2. An example of an ILF XML utterance output appears in Figure 1. The two types of annotations ILF automatically produces after DNS interpretation are shown in Figure 3.

Two of the principal advantages of using the DNS SDK rather than the simple dictation-to-text interface of our initial effort are that (1) the SDK can capture the actual input sounds into a .WAV file, and (2) we can query it for alternative interpretations of any portion of the speech signal. When the actual sounds being interpreted are dumped into a file by DNS, it can be instructed also to record the start and stop times of each *utterance* (these are the segments of speech between natural breaks such as pauses) and of each *word* (the units identified by the ASR algorithm). In addition, the SDK can provide a list of the top alternative interpretations of the last utterance and its confidence score for each word in the top choice. Therefore it will be possible, in a future, more integrated system, to go back and reinterpret segments of the speech input if what was transcribed does not appear to make sense. It should also be possible to build recognizers for non-speech noise sources that may occur often in our target clinical setting, such as a cough or a baby crying. With the recorded timing information for each element of the interpreted text, such a recognizer could identify segments of input that should be omitted from interpretation. Another, yet more powerful possible design that is not supported by the current SDK would permit ILF to provide feedback to the DNS recognition algorithm based on the semantic plausibility of what is being recognized.

```
<ANNOTATION ID="13" TYPE="12" STARTNODE="0"
ENDNODE="36">
  <Feature>
    <Name className="java.lang.String">rank</Name>
    <Value className="java.lang.Integer">12</Value>
  </Feature>
  <Feature>
    <Name className="java.lang.String">phrase</Name>
    <Value className="java.lang.String">Dr.  Doctor
       Eppel Levi had a brain tumor<Value>
  </Feature>
</ANNOTATION>
```

```
<ANNOTATION ID="27" TYPE="UTT" STARTNODE="0"
ENDNODE="36">
  <Feature>
    <Name className="java.lang.String">start</Name>
    <Value className="java.lang.Double">0.000</Value>
  </Feature>
  <Feature>
    <Name className="java.lang.String">end</Name>
    <Value className="java.lang.Double">3.033</Value>
  </Feature>
  <Feature>
    <Name className="java.lang.String">dbid</Name>
    <Value className="java.lang.Integer">28</Value>
  </Feature>
</ANNOTATION>
```

Figure 3: Two annotations for the utterance represented by Figure 1. The first is the 12th alternative phrase choice. The second is the sound dump timing and associated WAV file id.

3. Difficulties Encountered

As with much of contemporary software engineering, the greatest challenges in combining these tools has been to deal with the many incompatible components. Some of the difficulties we have encountered will illustrate this theme:

First, CaRE consists of a complex set of interrelated tools developed in various programming languages and originally deployed in a Linux environment. DNS runs only in Microsoft Windows and is strongly coupled to such Windows-only technologies as ActiveX controls and COM objects. GATE has been developed on a Java platform. Consequently, tying all of these pieces together required considerable effort including utilizing a Java-COM bridge[3] and developing workarounds for the missing Unix facilities that tie CaRE's pieces together. For future extensions, we will almost certainly need to recode much of CaRE in Java, to allow its proper integration into the GATE framework.

Second, although we find the graphical user interface presented by that system to be robust and relatively easy to use, we have had the opposite experience with the Application Programmer's Interface (API). We found places where its behavior is not predictable from the documentation, and others where documented calls simply do not work. One particular area where we encountered serious problems has been in utilizing GATE's persistence tools, which should work best with an actual database backend. GATE describes support for the free PostgreSQL database, but due to GATE's poor support of the latest release and Windows' poor support of the prior release, we struggled to connect GATE and PostgreSQL. We were able to overcome many of these problems, but we are still unable to use GATE's database-backed persistent document implementation reliably. It appears that some undiscovered error in its implementation causes changes sometimes not to be communicated to the database record.

4. Challenges and Next Steps

The current version of ILF is quite functional for the two tasks described here: (1) capturing transcribed speech and metadata from a single speaker and (2) inputting these into a database-backed text engineering framework. We have mentioned the need to incorporate into GATE the CaRE-like abilities to interpret the transcribed text into a concise and valid structured record of the encounter. This should be a "mere matter of programming," because we have previously built similar systems. We believe, however, that there are several other major challenges facing us in our work on this project.

First, current ASR systems seem built for use by a single user, not the pair (at least) that participate in a clinical encounter. Thus DNS expects that every utterance heard by it comes from the same speaker, hence it applies the same language and speaker model to all inputs. This is clearly wrong in our setting, and will lead to degraded performance if, for example, one party to the conversation is female and the other male, or if one has a very different accent than the other, making any language model a poor fit for both. There are, in the research laboratory, ASR systems designed to be far more speaker independent than DNS, and perhaps they could be adapted to our task. We have also considered running two instances of DNS, one listening to the doctor, the other the patient. It is not currently possible to run more than one instance of the software on a single machine, which is a shame in the era where two, four and even eight-core personal computers are becoming commonplace. Thus our current plan is to use two computers to interpret the two participants' inputs, and then to use a network-based coordination protocol to assure synchrony between what is said by the two parties.

Second, although doctors may be willing to train a system to their voice patterns and speaking styles, patients certainly will not have the opportunity or time to do so. DNS does come with a generic language model that claims to be able to handle the ASR task without any training, but such use clearly degrades accuracy. Again, we may need to use more research-stage systems that have been designed explicitly for such audiences if the DNS models are inadequate.

Third, the quality and placement of microphones seems to be critical to good ASR performance. Indeed, DNS recommends use of a headset microphone, which may be suitable for dictation but is probably not acceptable in a clinical encounter. We have used such microphones in our experiments so far. Alternatives include high-quality lapel microphones, whose placement farther from the speaker's mouth puts them at a disadvantage, but which may be sufficiently unobtrusive to be acceptable. A better option would be an array microphone, which uses dynamic signal processing techniques with an array of microphone inputs, typically arranged in a line, to isolate sounds that come from a specific direction and distance in the space before them. These can be used several feet from the speaker, and thus do not interfere with the speaker's freedom of movement. Such systems had been quite expensive, but continued price reductions have now made them available for under $100. Unfortunately, our very limited experience with one such system suggests that they do not perform as well for ASR as the headset-mounted microphones.

Fourth, we must accumulate a significant set of doctor-patient conversations to use as training data in developing the statistical models that go into CaRE and similar sys-

[3] We settled on JACOB [1], which in our evaluation was the most stable and functional of the open source tools.

978-1-4244-2890-8/08/$25.00 ©2008 IEEE

tems. In addition, if we find that our initial serial approach to the interpretation task does not yield sufficient accuracy, we may need to develop a more sophisticated integration between various components of ILF so that quality measurements in different parts of the system can control the effort expended by other parts to reach some globally optimal interpretation.

Finally, the challenge of creating a primarily speech-based interface that will allow a doctor and patient to correct a visually-presented encounter record seems daunting. Clearly, dictation-oriented commands such as "delete last paragraph" are completely inappropriate to this setting. Instead, such corrections need to be made through natural speech, based on the semantics of what the system believes and shows. Thus, we should expect statements more like "no, he suffered his heart attack in 1985, not 1995." We are unaware of existing techniques for doing this, which raises both the risks and rewards of our approach.

This first pass at our ambitious goal of automating documentation of clinical encounters yielded two positive results. First, it encouraged us that, given enough time and effort, the goal is reachable. Second, it gave us a realistic understanding of the strengths and weaknesses of the state of the art and helped us to anticipate and plan for the challenges that lie ahead.

References

[1] D. Adler. A java-com bridge. http://sourceforge.net/projects/jacob-project, Feb 2008.

[2] J. Bellegarda. Statistical techniques for robust asr: review and perspectives. In *EUROSPEECH-1997*, pages KN33–KN36, 1997.

[3] E. Brill. A simple rule-based part of speech tagger. In *HLT '91*, pages 112–116, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[4] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines*. Cambridge University Press, Cambridge, UK, 2000.

[5] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. Gate: A framework and graphical development environment for robust nlp tools and applications. In *Association for Computational Linguistics Proceedings*, pages 168–175, 2002.

[6] E. G. Devine, S. A. Gaehde, and A. C. Curtis. Comparative evaluation of three continuous speech recognition software packages in the generation of medical reports. *JAMIA*, 7(5):462–468, September-October 2000.

[7] N. Haapanen, S. Miilunpalo, M. Pasanen, P. Oja, and I. Vuori. Agreement between questionnaire data and medical records of chronic diseases in middle-aged and elderly finnish men and women. *American Journal of Epidemiology*, 145(8):762–769, Apr 1997.

[8] B. L. Humphreys and D. A. Lindberg. The umls project: making the conceptual connection between users and the information they need. *Bulletin of the Medical Library Association*, 81(2):170–177, 1993.

[9] IOM. *The computer-based patient record: an essential technology for health care*. National Academy Press, 1997.

[10] R. Lacson, R. Barzilay, and W. Long. Automatic analysis of medical dialogue in the home hemodialysis domain: structure induction and summarization. *J Biomed Inform*, 39:541–555, Oct 2006.

[11] Nuance Communications. Dragon naturally speaking. http://www.nuance.com/naturallyspeaking, Feb 2008.

[12] C. Patricoski, K. Shannon, and G. Doyle. The accuracy of patient encounter logbooks used by family medicine clerkship students. *Family Medicine*, 30(7):487–9, 1998.

[13] S. Shiffman, M. Detmer, and et al. A continuous-speech interface to a decision support system: I. techniques to accomodate for misrecognized input. In *J Am Med Informatics Assoc*, volume 2, pages 36–45, 1995.

[14] K. Shu, D. Boyle, C. Spurr, J. Horsky, H. Heiman, P. O'Connor, J. Lepore, and D. Bates. Comparison of time spent writing orders on paper with computerized physician order entry. *Medinfo*, 10:1207–1211, 2001.

[15] T. Sibanda. *Was the Patient Cured? : Understanding Semantic Categories and their Relationship in Patient Records*. Thesis, MIT, 2006.

[16] D. Temperley. An introduction to the link grammar parser. http://www.link.cs.cmu.edu/link/dict/introduction.html, 1999.

[17] H. M. Tufo and J. J. Speidel. Problems with medical records. *Medical Care*, 9(6):509–517, Nov-Dec 1971.

[18] A. E. Zuckerman, B. Starfield, C. Hochreiter, and B. Kovasznay. Validating the content of pediatric outpatient medical records by means of tape-recording doctor-patient encounters. *Pediatrics*, 56(3):407–411, September 1975.

978-1-4244-2890-8/08/$25.00 ©2008 IEEE