An automated medical scribe for documenting clinical encounters*

Greg P. Finley, Erik Edwards, Amanda Robinson, Najmeh Sadoughi, James Fone, Mark Miller, David Suendermann-Oeft
EMR.AI Inc.
San Francisco, CA, USA
greg.finley@emr.ai

Michael Brenndoerfer
University of California
Berkeley, CA, USA

Nico Axtmann
DHBW, Karlsruhe, Germany

Abstract

A medical scribe is a clinical professional who charts patient–physician encounters in real time, relieving physicians of most of their administrative burden and substantially increasing productivity and job satisfaction. We present a complete implementation of an *automated* medical scribe. Our system can serve either as a scalable, standardized, and economical alternative to human scribes; or as an assistive tool for them, providing a first draft of a report along with a convenient means to modify it. This solution is, to our knowledge, the first automated scribe ever presented and relies upon multiple speech and language technologies, including speaker diarization, medical speech recognition, knowledge extraction, and natural language generation.

1 Introduction

A recent study from the University of Wisconsin concluded that primary care physicians spend almost two hours on tasks related to electronic medical record (EMR) systems for every one hour of direct patient care (Arndt et al., 2017). This result illustrates the omnipresent complaint that medical practitioners are overly burdened by the administrative overhead of their work.

One solution to this issue is to have someone other than the physician take care of most of the EMR-related work associated with patient care. In particular, a *medical scribe* assumes the role of a clinical paraprofessional entering information into and, in some cases, extracting required information from EMR systems during patient–physician encounters (Earls et al., 2017). Scribes produce data entries in real time, entering narrative and discrete data points into templates of the EMR system. They can be either physically present in the physician's office or connected through phone or internet, interacting with the EMR system offline or by way of remote desktop connections. The latter are referred to as "virtual scribes."

Several studies show that the use of human scribes saves physicians substantial time on documentation, improves their work-life balance, and enhances clinicians' productivity. The resulting revenue increase has the potential to be multiple times higher than the cost of the scribe (Koshy et al., 2010; Bastani et al., 2014; Earls et al., 2017).

Despite these considerable advantages, there are some drawbacks to using medical scribes:

- scribes require extended training time and cost before developing their full potential—e.g., Walker et al. (2016) found the average training cost to be $6,317;
- scribes are often medical or premedical students (Walker et al., 2016) who, after being sufficiently exposed to the training experience the position of a scribe offers, tend to move on to attend medical school full time[1]; this fast-paced turnover in conjunction with the aforementioned training time and cost greatly reduces their effectiveness;
- scribes are costly: Earls et al. (2017) states that their scribes are paid $39,750 p.a.; Walker et al. (2016) quotes an average salary of $15.91 per hour for their virtual scribes which equals approximately $29,000 p.a.; Brady and Shariff (2013) estimates the annual cost of an on-site scribe to be $49,000 and for a virtual scribe $23,000.

In order to mitigate these disadvantages while preserving the strengths of employing scribes in the

*Patent pending.

[1]E.g., Stanford University's scribe program is purposefully limited to 12 months and is designed to prepare future medical students (Lin et al., 2017).

first place, we have developed a fully automated scribe, a prototype of which is presented here. It makes use of a full stack of state-of-the-art speech and natural language processing (NLP) components which are concisely described in this paper. To the best of the authors' knowledge, this is the very first automated scribe implementation ever presented to the scientific community.

We see at least two main avenues for deploying this technology. The first is to serve as a direct stand-in for human scribes—useful in cases where hiring scribes is either economically or logistically infeasible. In this case, the output of our system would be subject to review and correction by the physician. The second is as an assistive tool to (human) virtual scribes: our system displays an inital draft of the report and a summary of the information present in the conversation. The virtual scribe will be able to make any necessary corrections either to this information, in which case the report can be re-generated, or directly to the text. Either way, the availability of our automated system promises to streamline the human scribe's work and increase their throughput dramatically. Note that a similar workflow is commonplace for transcribing dictated clinical reports: the dictation is passed through an automatic speech recognition (ASR) and formatting system, then manually corrected by professional transcriptionists off-site.

2 Design

The automated scribe features a linear processing pipeline of speech-processing modules followed by NLP modules. We briefly introduce and motivate all modules in this section, then describe each individually in the following sections.

The initial stages transform the recorded conversation into a text format: first, a speaker diarization module determines who is speaking when and uses this information to break the audio recording into segments coded for speaker, which are then passed through a medical ASR stage. These steps are described in Sections 3 and 4.

Following ASR, the scribe must convert a transcribed spontaneous conversation into a concise and fully formatted report. This goal is exemplified in Figure 1, which shows an excerpt of a conversation and its realization in the report. The system does not perform this translation directly—this would require enormous amounts of parallel data to solve, end to end, with any single technique. Instead, we employ a two-stage approach in which the scribe mines the conversation for information and saves it in a structured format, then exports this structured data to the final report. In this way, the bulk of the NLP work is divided into two well-studied problems: knowledge extraction (KE; Section 5) and natural language generation (NLG; Section 7). (Between these two stages, structured data is processed directly to prepare it for export [Section 6].) Generating structured data as an intermediate step has numerous other advantages: it can be kept in the patient's history for reference to improve results on future episodes; it can be used by other systems that process structured data (e.g. billing, decision support); and it can be corrected manually if needed, which can be less error-prone than correcting final text directly.

3 Speaker diarization

Speaker diarization is the "who spoke when" problem, also called speaker indexing (Wellekens, 2001; Miró et al., 2012; Moattar and Homayounpour, 2012). The input is audio features sampled at 100 Hz frame rate, and the output is frame-labels indicating speaker identify for each frame. Four labels are possible: speaker 1 (e.g. the doctor), speaker 2 (e.g. the patient), overlap (both speakers), and silence (within-speaker pauses and between-speaker gaps). Note that the great majority of doctor-patient encounters involve exactly two speakers. Although our method is easily generalizable to more speakers, we currently report on the two-speaker problem.

The diarization literature broadly distinguishes "bottom-up" vs. "top-down" approaches. The former (Gish et al., 1991) operate by merging neighboring frames by similarity (clustering); we found initial results unsatisfactory. The later operate with a prior model such as HMM–GMM (Hidden Markov, Gaussian mixture model) to represent the likely audio features and timing (transition) characteristics of dialogs. We have introduced our own top-down approach that utilizes a modified expectation maximization (EM) algorithm at decoding time to learn the current speaker and background silence characteristics in real time. It is coded in plain C for maximum efficiency and currently operates at $\sim$50 $\times$ real-time factor.

Diarization requires an expanded set of audio features compared to ASR. In ASR, only phoneme identity is of final interest, and so audio features

Conversation	Report
Dr: "okay great and in terms of your past medical history do you have any other medical conditions you have"	FAMILY MEDICAL HISTORY The patient's aunt had lung cancer.
Pt: "no i have not had any medical conditions but my auntie actually she had lung cancer so that's why i kind of. . . "	

Figure 1: An excerpt from a typical input and output for the NLP segment of the scribe. Note that the ASR output has no punctuation or case; the doctor ('Dr.') and patient ('Pt.') identifiers illustrate the contribution of the diarizer.

are generally insensitive to speaker characteristics. By contrast, in diarization, only speaker identity is of final interest. Also note that diarization performs a *de facto* speech activity detection (SAD), since states 1–3 vs. state 4 are speech vs. silence. Therefore features successful for SAD (Sadjadi and Hansen, 2013) are helpful to diarization as well. Accordingly, we use an expanded set of gammatone-based audio features for the total SAD + diarization + ASR problem (details to be reported elsewhere).

4 Speech recognition

ASR operates on the audio segments produced by the diarization stage, where each segment contains one conversational turn (1 speaker + possibly a few frames of overlap). Currently, the diarization and ASR stages are strictly separated and the ASR decoding operates by the same neural network (NN) methodology that we recently reported for general medical ASR (Edwards et al., 2017). In brief, the acoustic model (AM) consists of a NN trained to predict context-sensitive phones from the audio features; and the language model (LM) is a 3- or 4-gram statistical LM prepared with methods of interpolation and pruning that we developed to address the massive medical-vocabulary challenge. Decoding operates in real time by use of weighted finite-state transducer (WFST) methodology (Mohri et al., 2002; Allauzen et al., 2007; Povey et al., 2011) coded in C++. Our current challenge is to adapt the AM and LM to medical conversations, which have somewhat different statistics compared to dictations.

5 Knowledge extraction

Extracting information from spontaneous conversational speech is a notoriously difficult problem. There has been some recent work on extracting keywords (Habibi and Popescu-Belis, 2013) or facts such as biographical details (Jing et al., 2007), but it is unclear whether known methods are effective for clinical conversation specifically.

We apply a novel strategy to simplify the KE problem by tagging sentences and turns in the conversation based upon the information they are likely to contain. These classes overlap largely with sections in the final report—chief complaint, medical history, etc. Then, we apply a variety of strategies, depending on the type of information being extracted, on filtered sections of text.

We use hierarchical recurrent neural networks (RNNs) to tag turns and sentences with their predicted class; each sentence is represented by a single vector encoded by a word-level RNN with an attention mechanism. (Our approach is similar to the influential document classification strategy of Yang et al. (2016), although we classify the sentences individually rather than the entire document.) In most cases, we can generate a sentence vector from an entire speech turn; for longer turns, however, we have to detect sentence boundaries. This is essentially a punctuation restoration task, which we have successfully undertaken previously using RNNs with attention (Salloum et al., 2017).

To extract information from tagged sentences, we apply one or more of several strategies:

- Complete or partial string match to identify terms from ontologies. This is effective for concepts which do not vary much in representation, such as certain medications.
- Extractive rules using regular expressions, which are well suited to predictable elements such as medication dosages, or certain temporal expressions (e.g. dates and durations).
- Other unsupervised or knowledge-based strategies, such as Lesk-style approaches (Lesk, 1986) in which semantic overlap with dictionary definitions of terms is used to normalize semantically equivalent phrases, as has been done successfully for medical concepts (Melton et al., 2010). This might be suitable for concepts that usually vary in expression, such as descriptions of symptoms.

- Fully supervised machine learning approaches, which we employ for difficult or highly specialized tasks—for example, identifying when a patient complains of symptoms generally worsening.

The KE stage also relies on extractive summary techniques where necessary, in which entire sentences may be copied directly if they refer to information that is relevant but difficult to represent in our structured type system—for example, a description of how a patient sustained a workplace injury. (To handle such cases using natural language understanding is a highly complex problem requiring a domain-general solution, which is beyond the scope of the medical scribe.) At a later stage, extracted text is processed to fit seamlessly into the final report (e.g. changing pronouns).

6 Processing structured data

Following the information extraction stage is a module which performs several functions to validate the structured knowledge and prepare it for NLG. This often entails correcting for any gaps or inconsistencies in the extracted knowledge, as may occur when there is critical information that is not explicitly mentioned during the encounter (as is frequently the case), or if there are errors in diarization, ASR, or KE. Typically, problems can be resolved through a series of logical checks or by relying on other structured data in the patient's history (when available). If not, conflicts or grave omissions can be flagged for the user.

Wherever appropriate, data is also encoded in structures compatible with the HL7 FHIR v3 standard (Bender and Sartipi, 2013) to facilitate interoperability with other systems. As a concrete example, if the physician states an intent to prescribe a medication, a FHIR MedicationRequest resource is generated. The output of this stage can be made available to the user if he or she wishes to amend the structured information, and any changes can be propagated instantly to NLG.

7 Natural language generation

The NLG module produces and formats the final report. Medical reports follow a loosely standardized format, with sections appearing in a generally predictable order and with well-defined content within each section. Our strategy is a data-driven templatic approach supported by a finite-state "grammar" of report structure.

The template bank consists of sentence templates annotated for the structured data types necessary to complete them. We fill this bank by clustering sentences from a large corpus of medical reports according to semantic and syntactic similarity. The results of this stage are manually curated to ensure that strange or imprecise sentences cannot be generated by the system, and to ensure parsimony in the resulting type system. Kondadadi et al. (2013) employ a similar method of clustering and manual review to quickly and effectively generate a full template bank from data.

Using the same reports, we induce the grammar using a probabilistic finite-state graph, where each arc outputs a sentence and a single path through the graph represents one actual or possible report. Decoding optimizes the maximal use of structured data and the likelihood of the path chosen. The grammar helps to improve upon one common criticism of templatic NLG approaches, which is the lack of variation in sentences (van Deemter et al., 2005), in a way that does not require any "inflation" of the template bank with synonyms or paraphrases: during decoding, different semantically equivalent templates may be selected based on context and the set of available facts, thus replicating the flow of natural language in existing notes.

Note that, as format can vary considerably by note type, specialty, and healthcare provider, we build separate NLG models to handle each type of output.

Finally, all notes pass through a processor that handles reference and anaphora (e.g. replacing some references to the patient with gender pronouns), truecasing, formatting, etc.

8 Conclusion

The presented automated scribe can take over or supplement the role of human scribes documenting encounters between patients and physicians. At the current stage, the system is still limited in its functionality and scope, and major enhancements are being made to improve performance and content coverage of several of the involved components. In particular, we plan to expand the use of machine learning techniques as soon as enough data has been accumulated in various pilot studies currently underway. Additionally, we are working to compile a large set of parallel inputs and outputs to allow for a true end-to-end evaluation of the system.

References

C Allauzen, M Riley, J Schalkwyk, and M Mohri. 2007. OpenFst: a general and efficient weighted finite-state transducer library. In *Proc CIAA*, volume LNCS 4783, pages 11–23. Springer.

BG Arndt, JW Beasley, MD Watkinson, JL Temte, W-J Tuan, CA Sinsky, and VJ Gilchrist. 2017. Tethered to the EHR: primary care physician workload assessment using EHR event log data and time-motion observations. *Ann Fam Med*, 15(5):419–426.

A Bastani, B Shaqiri, K Palomba, D Bananno, and W Anderson. 2014. An ED scribe program is able to improve throughput time and patient satisfaction. *Am J Emerg Med*, 32(5):399–402.

D Bender and K Sartipi. 2013. HL7 FHIR: an Agile and RESTful approach to healthcare information exchange. In *Proc Int Symp CBMS*, pages 326–331. IEEE.

K Brady and A Shariff. 2013. Virtual medical scribes: making electronic medical records work for you. *J Med Pract Manage*, 29(2):133–136.

K van Deemter, M Theune, and E Krahmer. 2005. Real versus template-based natural language generation: a false opposition? *Comput Linguist*, 31(1):15–24.

ST Earls, JA Savageau, S Begley, BG Saver, K Sullivan, and A Chuman. 2017. Can scribes boost FPs' efficiency and job satisfaction? *J Fam Pract*, 66(4):206–214.

E Edwards, W Salloum, GP Finley, J Fone, G Cardiff, M Miller, and D Suendermann-Oeft. 2017. Medical speech recognition: reaching parity with humans. In *Proc SPECOM*, volume LNCS 10458, pages 512–524. Springer.

H Gish, M-H Siu, and JR Rohlicek. 1991. Segregation of speakers for speech recognition and speaker identification. In *Proc ICASSP*, volume 2, pages 873–876. IEEE.

M Habibi and A Popescu-Belis. 2013. Diverse keyword extraction from conversations. In *Proc ACL*, volume 2, pages 651–657. ACL.

H Jing, N Kambhatla, and S Roukos. 2007. Extracting social networks and biographical facts from conversational speech transcripts. In *Proc ACL*, pages 1040–1047. ACL.

R Kondadadi, B Howald, and F Schilder. 2013. A statistical NLG framework for aggregated planning and realization. In *Proc ACL*, pages 1406–1415. ACL.

S Koshy, PJ Feustel, M Hong, and BA Kogan. 2010. Scribes in an ambulatory urology practice: patient and physician satisfaction. *J Urol*, 184(1):258–262.

ME Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proc SIGDOC*, pages 24–26. ACM.

S Lin, K Osborn, A Sattler, I Nelligan, D Svec, A Aaronson, and E Schillinger. 2017. Creating the medical school of the future through incremental curricular transformation: the Stanford Healthcare Innovations and Experiential Learning Directive (SHIELD). *Educ Prim Care*, 28(3):180–184.

GB Melton, S Moon, M Bridget, and S Pakhomov. 2010. Automated identification of synonyms in biomedical acronym sense inventories. In *Proc Louhi Workshop*, pages 46–52. ACL.

XA Miró, S Bozonnet, N Evans, C Fredouille, G Friedland, and O Vinyals. 2012. Speaker diarization: a review of recent research. *IEEE Trans Audio Speech Lang Process*, 20(2):356–370.

MH Moattar and MM Homayounpour. 2012. A review on speaker diarization systems and approaches. *Speech Commun*, 54(10):1065–1103.

M Mohri, FCN Pereira, and M Riley. 2002. Weighted finite-state transducers in speech recognition. *Comput Speech Lang*, 16(1):69–88.

D Povey, G Boulianne, L Burget, O Glembek, NK Goel, M Hannemann, P Motlícek, Y Qian, P Schwarz, and J Silovsky. 2011. The Kaldi speech recognition toolkit. In *Proc ASRU*, pages 1–4. IEEE.

SO Sadjadi and JHL Hansen. 2013. Unsupervised speech activity detection using voicing measures and perceptual spectral flux. *IEEE Signal Process Lett*, 20(3):197–200.

W Salloum, GP Finley, E Edwards, M Miller, and D Suendermann-Oeft. 2017. Deep learning for punctuation restoration in medical reports. In *Proc Workshop BioNLP*, pages 159–164. ACL.

KJ Walker, W Dunlop, D Liew, MP Staples, M Johnson, M Ben-Meir, HG Rodda, I Turner, and D Phillips. 2016. An economic evaluation of the costs of training a medical scribe to work in emergency medicine. *Emerg Med J*, 33(12):865–869.

CJ Wellekens. 2001. Seamless navigation in audio files. In *Proc Odyssey*, pages 9–12. ISCA.

Z Yang, D Yang, C Dyer, X He, A Smola, and E Hovy. 2016. Hierarchical attention networks for document classification. In *Proc NAACL-HLT*, pages 1480–1490. ACL.