Automatically Generating Psychiatric Case Notes From Digital Transcripts of Doctor-Patient Conversations

Nazmul Kazi
Gianforte School of Computing
Montana State University
Bozeman, MT, USA
kazinazmul.hasan@montana.edu

Indika Kahanda
Gianforte School of Computing
Montana State University
Bozeman, MT, USA
indika.kahanda@montana.edu

Abstract

Electronic health records (EHRs) are notorious for reducing the face-to-face time with patients while increasing the screen-time for clinicians leading to burnout. This is especially problematic for psychiatry care in which maintaining consistent eye-contact and non-verbal cues are just as important as the spoken words. In this ongoing work, we explore the feasibility of automatically generating psychiatric EHR case notes from digital transcripts of doctor-patient conversation using a two-step approach: (1) predicting semantic topics for segments of transcripts using supervised machine learning, and (2) generating formal text of those segments using natural language processing. Through a series of preliminary experimental results obtained through a collection of synthetic and real-life transcripts, we demonstrate the viability of this approach.

1 Introduction

An electronic health record (EHR) is a digital version of a patient's health record. EHRs were introduced as a means to improve the health care system. EHRs are real-time and store patient's records in one place and can be shared with other clinicians, researchers and authorized personals instantly and securely. The use and implementation of EHRs were spurred by the 2009 US Health Information Technology for Economic and Clinical Health (HITECH) Act and 78% office-based clinicians reported using some form of EHR by 2013 (Hsiao and Hing, 2014).

Presently, all clinicians are required to digitally document their interactions with their patients using EHRs. These digital documents are called case notes. Manually typing case notes is time consuming (Payne et al., 2015) and limits the face-to-face time with their patients, which leads to both patient dis-satisfaction and clinician burnout. Limited face-to-face time is especially disadvantageous for working with mental health patients where the psychiatrist could easily miss a non-verbal cue highly important for the correct diagnosis. Moreover, EHR's usability related problems lead to unstructured and incomplete case notes (Kaufman et al., 2016) which are difficult to search and access.

Due to the above-mentioned downsides of EHRs, there have been recent attempts for developing novel methods for incorporating various techniques and technologies such as natural language processing (NLP) for improving the EHR documentation process. In 2015, American Medical Informatics Association reported time-consuming data entry is one of the major problems in EHRs and recommended to improve EHRs by allowing multiple modes of data entry such as audio recording and handwritten notes (Payne et al., 2015). Nagy et al. (2008) developed a voice-controlled EHR system for dentists, called *DentVoice*, that enables dentists to control the EHR and take notes over voice and without taking off their gloves while working with their patients. Kaufman et al. (2016) also developed an NLP-enabled dictation-based data entry where clinicians can write case notes over voice and able to reduce the time by more than 60%.

Psychiatrists mostly collect information from their patients through conversations and these conversations are the primary source of their case notes. In a long-term project in collaboration with National Alliance of Mental illness (NAMI) Montana and the Center for Mental Health Research and Recovery (CMHRR) at Montana State University, we envision a pipeline that automatically records a doctor-patient conversation, generates the corresponding digital transcript of the conversation using speech-to-text API and uses natural language processing and machine learning tech-

Proceedings of the 2nd Clinical Natural Language Processing Workshop, pages 140–148
Minneapolis, Minnesota, June 7, 2019.

niques to predict and/ or extract important pieces of information from the text. This relevant text is then converted to a more formal written version of the text and are used for auto-populating the different sections of the EHR form.

In this work, we focus on the back-end of the above mentioned pipeline, i.e. we explore the feasibility of populating sections of EHR form using the information extracted from a digital transcript of a doctor-patient conversation. In order to gather gold-standard data, we develop a human powered digital transcript annotator and acquire annotated versions of digital transcripts of doctor-patient conversations with the help domain experts. As the first step in our two-step approach, we develop a machine learning model that can predict the semantic topics of segments of conversations. Then we develop natural language processing techniques to generate a formal written text using the corresponding segments. In this paper, we present our preliminary findings from these two tasks; Figure 1 depicts the high-level overview of our two-step approach.

Previous studies most related to our work are (1) Lacson et al. (2006) predicting semantic topics for medical dialogue turns in the home hemodialysis, and (2) Wallace et al. (2014) automatically annotating topics in transcripts of patient-provider interactions regarding antiretroviral adherence. While both studies successfully use machine learning for predicting semantic topics (albeit different topics to ours) they do not focus on the development of NLP models for text summarization (i.e. formal text generation).

The rest of the paper is structured as follows. We describe our two-step approach, data collection and processing, machine learning models and natural language processing methods in chapter 2. In chapter 3, we report and discuss the performance of our methods. We summarize our findings, discuss limitations and potential future work in chapter 4.

2 Methods

2.1 Approach

As depicted in Figure 1, we divide the task of generating case notes from digital transcripts of doctor-patient conversations into two sub tasks: (1) using supervised learning models to predict semantic topics for segments of the transcripts and then (2) using natural language processing models to generate a more formal (i.e. written) version of the text which goes in to the corresponding section of the EHR form.

These semantic topics are suggested by the domain experts from NAMI Montana and correspond to the main sections of a typical EHR form. They are (1) Client details: personal information of a patient, such as name, age, birth date etc., (2) Chief complaint: refers to the information regarding a patient's primary problem for which the patient is seeking medical attention., (3) Medical history: any past medical condition(s), treatment(s) and record(s), (4) Family history: indicates medical history of a family member of the patient, and (5) Social history: refers to information about patient's social interactions, e.g. friends, work, family dinner etc. We call these semantic categories "EHR categories" interchangeably. The *formal text* is essentially the summary text that the clinician would write or type into the EHR form based on the interaction with the patient.

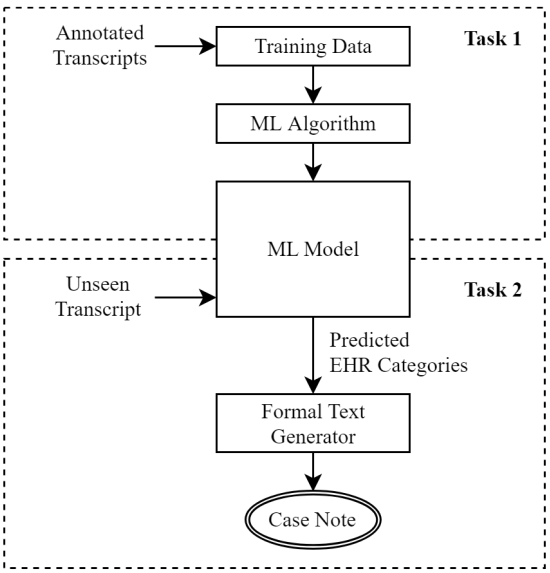

Figure 1: High-level overview of our approach. Task1: Predicting EHR categories. Task 2: Formal text generation. ML: Machine Learning. EHR: Electronic Heallth Record.

2.2 Transcripts of doctor-patient dialogue

Our raw dataset is composed of 18 digital transcripts of doctor-patient conversations and covers 11 presenting conditions. The presenting conditions are Attention-deficit/ hyperactivity disorder (ADHD), Alzheimer's disease, Anger, Anorexia, Anxiety, Bipolar, Borderline Personality Disor-

der (BPD), Depression, Obsessive Compulsive Disorder (OCD), Post Traumatic Stress Disorder (PTSD) and Schizophrenia. All transcripts are labeled with speaker tags "Doctor:" and "Patient:" to indicate the words uttered by each individual.

Thirteen of these transcripts are *synthetic* in that they are handwritten (i.e. typed) by a domain expert from NAMI Montana who has years of experience working with mental illness patients. Hence, each synthetic transcript represents a real case scenario of conversation between a patient (suffering from one of the presenting conditions mentioned above) and a psychiatric doctor/ clinician who verbally interviews the patient in a 2-person dialogue set up. Table 1 reports summary statistics.

Rest of the five transcripts are part of *Counseling & Therapy* database[1] from the Alexander Street website. Hence, we refer to them as AS transcripts for the rest of the paper. Each of these AS transcripts is generated from a real-life conversation between a patient and a clinician. Majority of these transcripts cover multiple mental conditions.

In order to annotate transcripts using semantic topics mentioned above, we develop a human-powered transcript annotator as shown in Figure 2, a responsive web application, that takes digital transcripts as input, breaks down each transcript into segments where each segment starts with a speaker tag (Doctor: or Patient:) and generates samples by pairing each doctor segment with the followed by patient segment. The application displays the generated samples, from one transcript at a time, in the same order as they appear in the transcript and allows the user to annotate them with one of the six semantic topics.

A group of three annotators including two domain-experts from NAMI Montana use the above annotator tool to single-annotate (through collaboration) all 18 transcripts. As highlighted in Figure 2, annotations are added at the *conversation pair* level. We define the conversation pair as the entire text associated with a consecutive pair of "Doctor:" and "Patient:" speaker tags. Each conversation pair is annotated with one of the five topics (i.e. EHR categories). These labels are based on the main focus/ subject/ topic of the corresponding conversation pair as judged by

[1]https://search.alexanderstreet.com/health-sciences/counseling-therapy

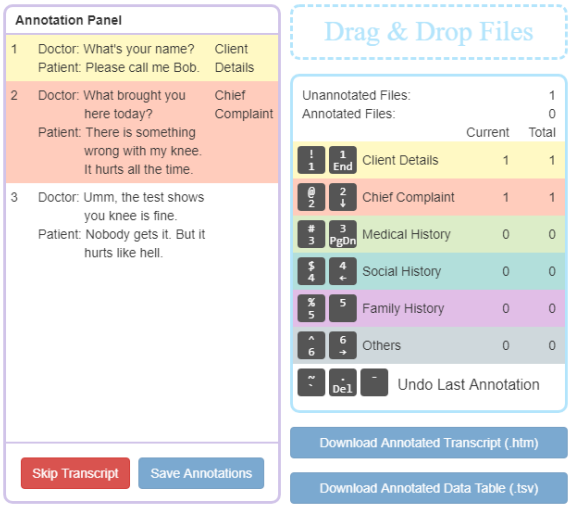

Figure 2: Screen shot of the human-powered transcript annotator. Left panel displays an example transcript while the semantic concepts are shown on the right.

the expert annotators. Any conversation pair that was found to be irrelevant to the five categories is annotated with a new category called "Others". Conversation pair level annotations eliminated the challenges in annotating a question or an answer on their own without the proper context provided by the preceding/ following sentences.

2.3 Task 1: Predicting EHR categories

In this task, we use the annotated digital transcripts to generate the training data to train supervised classification models using two different approaches. These two approaches mainly differ in how the transcripts were segmented into examples (i.e. training instances) for generating the training datasets as described in the sections 2.3.1 and 2.3.2. Regardless of the approach, we label the examples with one of the six class labels analogs to the semantic topics (EHR categories): Client Details, Chief Complaint, Family History, Social History, Medical History and Others.

2.3.1 Training data - Model 1

In this approach, we build a training dataset by taking a conversation pair as a single example (i.e. instance). Each example contains at least two sentences where the first sentence is spoken by the doctor and the second sentence is spoken by the patient. The class label for each example is the corresponding annotation from the original transcript; this results in six class labels. A short examples of the training dataset and distribution of class labels are reported in Tables 2 and 3.

Property	Synthetic Transcripts			AS Transcripts		
	Total	Mean	STDEV	Total	Mean	STDEV
No. Sentences	1930	148.4	55.6	1390	278.0	74.9
No. Questions	513	39.4	19.8	188	37.6	7.1
No. Dialogue turns	861	66.2	40.0	684	136.8	55.0
No. Sentences spoken by the Doctor	751	57.7	30.0	581	116.2	44.3
No. Sentences spoken by the Patient	1179	90.6	33.2	809	161.8	60.5

Table 1: Summary statistics on 13 synthetic transcripts vs. 5 (AS) Alexander Street transcripts.

No.	Example	Class Label
1	Doctor: How many voices do you hear? Patient: Two. They talk all the time.	Chief Complaint
2	Doctor: Your record shows that you take antidepressants pills regularly. Do you hang out with your parents, co-workers or friends? Do you talk to them? Patient: Sometimes I hang out with my mom. Yes, I talk to my co-workers but only for work. I used to have a friend who moved couple months ago and we don't talk anymore.	Social History

Table 2: Examples in Model 1 training data.

Class Label	Synthetic		All
	Model 1	Model 2	
Chief Complaint	309	870	1746
Client Details	32	88	198
Family History	28	101	149
Medical History	34	74	85
Others	12	174	264
Social History	19	51	110
Total	434	1358	2552

Table 3: Distribution of class labels in training data. All: represents Model 2 training data with all 18 transcripts.

Segmenting the transcripts into training examples in this fashion is convenient because there is a one-to-one mapping between the semantic topics in the original annotated transcripts and the class labels of the examples; additional reconciliation is not needed. However, sometimes, the doctor or the patient talks about more than one topic (inside the same conversation pair). For example, although example 2 in Table 2 is labeled with Social History, the conversation pair is composed of information relevant to both the medical history and social history. Therefore, segmenting the transcript to smaller pieces could be more beneficial for improved overall performance. This is the motivation for the second approach mentioned in the next section.

2.3.2 Training data - Model 2

In this approach, we use a finer-level granularity (than conversation pairs) for segmenting the transcripts for generating training examples. We start with the Model 1 training data and tokenize the text of each example at the sentence level by identifying the sentence boundaries using sentence tokenizer in NLTK[2]. We first assign labels to each sentence based on the class label of the original source (i.e. conversation pair). Then, one of the human annotators manually reviewed the class labels and makes corrections if needed.

However, labeling at the sentence-level is also challenging because the information that defines the topic (class label) lies in the question and is sometimes followed by a short answer, e.g. Table 4 example 1. We also observe the opposite scenario where the answer holds the context, e.g. Table 4 example 2, and scenarios where the information lies in both the question and the answer, e.g. Table 4 example 3. So, it is understood that without pairing the questions with their corresponding answers (or being aware of the context provided by the question or the answer), it is challenging even for human annotators to label these sentences individually. However, We also observe that a

[2]https://www.nltk.org/

question is commonly followed by its corresponding answer in the form of a non-interrogative sentence. Therefore, we use the following approach to overcome the above challenge.

We first combine the grammatical rules of the English language in forming a question (British Council, 2019) and spaCy[3], an industrial-strength natural language processing API, to identify the questions in the transcript. Then, to preserve the context, we pair each question with the following non-interrogative sentence and combine them into a single example. In other words, Model 2 training instances can be single sentences or a conversation pair or anything in between. Several examples of Model 2 dataset is shown in Table 5. These examples correspond to the Model 1 examples depicted in Table 2.

#	Question-answer pair	Class Label
1	Doctor: How old are you? Patient: 23.	Client Details
2	Doctor: Who do you take? Patient: I take Ibuprofen.	Medical History
3	Doctor: What is your name? Patient: Name is a game.	Chief Compliant

Table 4: Question-answer pair dependency.

2.3.3 Machine learning models

To explore the feasibility of classifying information from digital transcripts, we train separate supervised learning classifiers using both training datasets (i.e. Model 1 and Model 2). Specifically, since each instance is annotated with exactly one class label (out of six), we model this as a multi-class problem and use the one-vs-rest (Bishop, 2006) classification strategy.

We apply Support Vector Machines (SVMs) as our machine learning algorithm (which was found to be the best performer in an initial study in comparison with a few other popular machine learning algorithms: k-Nearest Neighbors, Naïve Bayes, Decision Tree, Neural Networks – data not shown). We use stop word removal and lemmatization for pre-processing and Bag-of-Words model for feature extraction. We use scikit learn (Pedregosa et al., 2011) python machine learning library for implementing these models. For our preliminary experiments reported in this paper, we do not use any model checking or parameter tuning and use default settings.

[3]https://spacy.io/

2.3.4 Task 2: Formal text generation

Due to the error-prone nature of Model 1 training data described above, we exclusively use Model 2 training data for the formal text generation. The high-level idea is that in order to generate a case note for an unseen transcript, we first segment the transcript at the Model 2 granularity and predict the EHR categories using the Model 2 classifier. Then instances are grouped based on their predicted EHR categories. Generating case notes with sentences as they appear in the transcripts (i.e. verbatim) will result in redundant case notes that will be difficult to search for important information. An assertive sentence generated by gathering information from a question-answer pair will be easier to read and concise. Therefore, for each category, a formal written version of the text is generated using the method described below. We ignore 'Others' category in our current setup because they represent irrelevant information and any information under this class is likely not important for case note.

In order to generate formal text from an instance, the entire text needs to be rewritten using an assertive sentence, subject in third person singular form, correct tense, verb form and sentence structure. We concatenate each piece of formal text within the category to form a paragraph. Thus, our method results in generating a case note composed of five paragraphs corresponding to the first five EHR categories.

As illustrated in Figure 3, our method generates formal text in several steps. As mentioned above, a sample can be either a sentence or a question-answer pair (as depicted in Table 5). First, we identify the number of sentences in the example text. Examples composed of a single sentence (e.g. Table 7, examples 1-3) requires minimal processing to generate formal text. We use part-of-speech tagging from python module spaCy to identify the subject, main verb and the auxiliary verb(s) of the sentences. If the subject is a first (I) or second person (you), the subject is replaced with the third person singular form (he/she). Clinicians typically collect personal information, such as name, gender and contact information, prior to their conversation or appointment and so they can be fed into our model as input to generate accurate case notes.

No.	Example	Class Label
1	How many voices do you hear? Two.	Chief Complaint
2	They talk all the time.	Chief Complaint
3	Your record shows that you take antidepressants pills regularly.	Medical History
4	Do you hang out with your parents, co-workers or friends? Do you talk to them? Sometimes I hang out with my mom.	Social History
5	Yes, I talk to my co-workers but only for work.	Social History
6	I used to have a friend who moved couple months ago and we don't talk anymore.	Social History

Table 5: Examples in Model 2 training data.

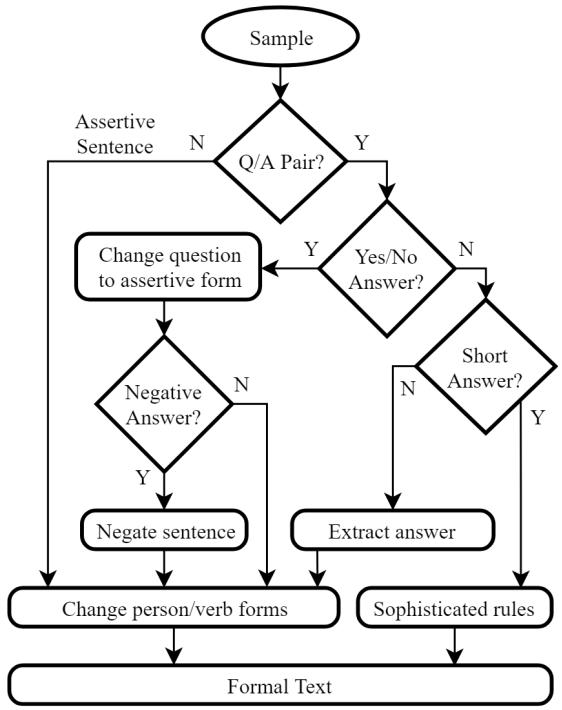

Figure 3: An overview of formal text generation steps.

If the sentence contains auxiliary verb(s), the first auxiliary verb is replaced with its third person singular form, e.g. *am* with *is*, and the second auxiliary verb, if any, and the main verb are kept as they are. If the sentence does not contain any auxiliary verbs, the proper form of the main verb depends on the tense of the sentence. If the sentence is in the present tense, the main verb is replaced with its third person singular form, e.g. *run* with *runs*. For sentences in the past tense, the main verb is kept unchanged since the form of the verb is the same for all persons, e.g. *took*. A sentence in future tense contains at least one auxiliary verb, *shall* or *will*, and therefore our method processes the sentence as a sentence in the present tense; there is no need to add any additional functionality to cover this tense.

If an instance is composed of multiple sentences, the last sentence is always a non-interrogative sentence and is the answer to the question posed in the very first sentence. In this case, the formal text depends on both the question and the answer. If the answer starts with an affirmation or negation word (e.g. yes, no, yeah, never), the question is changed to an affirmative or negative sentence, respectively, and the assertive sentence is added as a separate sentence after removing the leading affirmation or negation word (e.g. Table 7, examples 4-5). If the answer does not start with any affirmation or negation word, the answer is further analyzed to see whether it is a short answer. If not, the question text is ignored and the answer text is returned as the formal text (e.g. Table 7, example 6).

In the case of short answers, an answer alone does not provide the full context to construct the formal text and we need to rely on both the question and the answer. For e.g. the wh- questions (e.g. when, who) are usually followed by a relatively short answer that requires context from the question text as well. This required more sophisticated rules and we are presently working on generating formal text for this scenario. Examples and the intended "ideal" formal text for them are given in Table 8.

While generating formal text, all first and second person pronouns, regardless their position, are replaced with their third person singular form and the verbs are also replaced with its third person singular form, where applicable. Regular expressions are used to remove leading words (e.g. ok, right, yes, and, but, hmm) from the assertive sentences that have no importance to be included in the formal texts. This functionality was imple-

mented using NodeBox[4] Python library.

2.4 Experimental setup and metrics

In terms of Task 1, we evaluate our supervised machine learning models using 5 fold stratified cross-validation and the performance is reported using the AUROC (Area Under the ROC Curve) scale (Bewick et al., 2004). A score of 1 corresponds to the performance of an ideal classifier whereas a score of 0.5 relates to the performance of a random classifier. Because Task 2 (formal text generation aspect) of the project is a work-in-progress, we highlight the scenarios that our model is able to handle and mention the more challenging scenarios in future work.

3 Results and Discussion

In an initial experiment, we assessed the performance of Model 1 and Model 2 training data using the 13 synthetic transcripts. According to our preliminary results, SVMs with linear kernel performs the best with a macro-average AUROC score of 79% for Model 1. For Model 2, the SVMs classifier achieves a macro-average AUROC score of 81%. However, note that these numbers are not directly comparable because Model 1 training instances are different from that of Model 2. Still, this suggested that Model 2 is superior in performance. This is intuitive because Model 2 training data is a more refined dataset as described previously. This observation, coupled with the fact that Model 2 data are more conducive to formal text generation, we used Model 2 training data for the rest of the experiments.

Next, we assessed the performance of Model 2 using all the transcripts (i.e. 13 synthetic and 5 AS transcripts). There is a clear performance dip (0.81 vs. 0.76) when the AS transcripts are added to the training data. This is intuitive because we believe the AS transcripts may have lead to data that is harder to generalize for the classifiers. The reason is that the majority of them is associated with multiple presenting conditions and hence the content of the questions and answers may be broader than synthetic transcripts. Also, the language characteristics between the synthetic and AS transcripts have a noticeable difference according to Table 1. However, this provides valuable insight into the importance of the robustness of the classifier. In other words, caution must be exercised when synthetic data are used for training machine learning models. Note that we did not conduct a separate experiment with only the AS transcripts because the number of examples for some of the ill-represented classes were deemed inadequate.

Class Label	AUROC	STDEV
Chief Complaint	0.74	0.02
Client Details	0.73	0.04
Family History	0.77	0.04
Medical History	0.78	0.07
Others	0.84	0.03
Social History	0.67	0.06
Macro-average	0.76	

Table 6: Performance of Model 2 training data using all transcripts (13 artificial and 5 AS). Performance collected through 5-fold cross validation, repeated 10 times.

We observe that the performance for the individual semantic topics (EHR categories) fall in the range of 0.67 (Social History) and 0.84 (Others) as depicted in Table 6. But there is no correlation between the class distribution and the performance as evident from Table 3. Overall, these numbers suggest that the words of the transcript are reasonably informative for differentiating EHR categories but there is definitely room for improvement. One such improvement may come from focusing on the *type* of the words in addition to their lexical value. This view is supported by the top 5 tokens identified by the classifier as the most important tokens for each category (Table 9). For example, many of the top words for Family History are names of family members. We also emphasize that the performance reported is from models that work with BoW features and default parameter values, suggesting that the use of a comprehensive feature/ model selection procedure would likely yield better results.

As mentioned above, our formal text generation module is able to handle the scenarios listed in Table 7. However, instances in which the context lies in both the question and the answer (e.g. Table 4 example 3) are clearly more challenging and hence would require sophisticated rules. In such cases, the challenge is to extract information from both the question as well as the answer and to form an assertive sentence using the combined information. We are currently working on this scenario.

[4]https://www.nodebox.net/code/index.php/Linguistics

No.	Example	Generated Formal Text
1	I do not seem to be coping with things.	He does not seem to be coping with things.
2	I woke up about 4 am last night.	He woke up about 4 am last night.
3	My sister said I should come.	His sister said he should come.
4	Do you have any sort of hallucination and delusion? No.	He does not have any sort of hallucination and delusion.
5	Has this been going on for some time? Yeah, a few months really.	This has been going on for some time. A few months really.
6	Ok, so what is brought you here today? My sister's noticed, I am just a bit fed up really with some mood swings.	His sister's noticed, he is just a bit fed up really with some mood swings.

Table 7: Formal Text Generation: example inputs and the generated text.

No.	Example	Ideal Formal Text
1	Where do you work? A shop near the mall.	He works in a shop near the mall.
2	When did you wake up last night? It was before 4.	He woke up before 4 last night.
3	When did that happen? Then I was 10.	That happened when he was 10.
4	How often do you exercise? Not that much, I play basketball on Mondays and go for a run on Wednesdays and Saturdays.	He does not exercises much. He plays basketball on Mondays and goes for a run on Wednesdays and Saturdays.
5	Which color shall we use? Red, use red.	We shall use red.
6	In what way does he push her? Not like with hands, just ignores her to make her mad.	He does not push her with hands, just ignores her to make her mad.

Table 8: Formal Text Generation: challenging examples (requiring sophisticated rules) and their *ideal* formal text.

Class Label	Top five features
Chief Complaint	percent, stuff, feeling, number, feel
Client Details	meet, learned, write, pack, style
Family History	cousin, supportive, dad, married, family
Medical History	teen, asthma, dr., prozac, advair
Others	lab, ok, let, right, thank
Social History	comment, wellbutrin, racist, share, friend

Table 9: List of top five features per category used by the machine learning classifier.

Table 8 depicts examples from this scenario and the ideal formal text that must be generated.

4 Conclusion and Future Work

In this work, we focus on the problem of automatically generating case notes from digital transcripts of doctor-patient conversations, using a two-step approach: (1) predicting EHR categories and (2) generating formal text. On the task of predicting semantic topics for segments of the transcripts, we develop a supervised learning model while for the subsequent task of generating a formal version of the text from those segments, we develop a natural language processing model. According to preliminary experimental results obtained using a set of annotated synthetic and real-life transcripts, we demonstrate that our two-step approach is a viable option for automatically generating case notes from digital transcripts of doctor-patient conversations.

However, as noted previously, this is an ongoing project. The immediate attention is paid to handling the case of generating case notes for examples related to short answers given in Table 8. Due to the complexity of this scenario, sophisticated rules that make use for entities identified in the text must be utilized. We plan to transcribe authentic doctor-patient interactions and train a new classification model using these transcripts. We also intend to build a prototype and send it to clinicians

for testing using PDQI-9 (Stetson et al., 2012) to check the quality of our generated case notes.

5 Acknowledgements

We like to thank Matt Kuntz from NAMI Montana and CMHRR at Montana State University for his valuable contributions in bringing forth the vision, providing insight as well as assisting with gold standard data. We would also like to thank Cheryl Bristow from NAMI Montana for generating the synthetic transcripts and assistance in annotating transcripts.

References

Viv Bewick, Liz Cheek, and Jonathan Ball. 2004. Statistics review 13: receiver operating characteristic curves. *Critical care*, 8(6):508.

Christopher M Bishop. 2006. *Pattern recognition and machine learning*. springer.

British Council. 2019. Questions and negatives. *Learn English British Council, retrived from: https://learnenglish.britishcouncil.org/en/english-grammar/questions-and-negatives*.

Hsiao and Hing. 2014. Use and characteristics of electronic health record systems among office-based physician practices: United states, 20012013. *NCHS Data Brief, No 143. Hyattsville, MD: National Center for Health Statistics.*

David R Kaufman, Barbara Sheehan, Peter Stetson, Ashish R Bhatt, Adele I Field, Chirag Patel, and James Mark Maisel. 2016. Natural language processing-enabled and conventional data capture methods for input to electronic health records: a comparative usability study. *JMIR medical informatics*, 4(4).

Ronilda C Lacson, Regina Barzilay, and William J Long. 2006. Automatic analysis of medical dialogue in the home hemodialysis domain: structure induction and summarization. *Journal of biomedical informatics*, 39(5):541–555.

Miroslav Nagy, Petr Hanzlicek, Jana Zvarova, Tatjana Dostalova, Michaela Seydlova, Radim Hippman, Lubos Smidl, Jan Trmal, and Josef Psutka. 2008. Voice-controlled data entry in dental electronic health record. *Studies in health technology and informatics*, 136:529.

Thomas H Payne, Sarah Corley, Theresa A Cullen, Tejal K Gandhi, Linda Harrington, Gilad J Kuperman, John E Mattison, David P McCallie, Clement J McDonald, Paul C Tang, et al. 2015. Report of the amia ehr-2020 task force on the status and future direction of ehrs. *Journal of the American Medical Informatics Association*, 22(5):1102–1110.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Peter D Stetson, Suzanne Bakken, Jesse O Wrenn, and Eugenia L Siegler. 2012. Assessing electronic note quality using the physician documentation quality instrument (pdqi-9). *Applied clinical informatics*, 3(2):164.

Byron C Wallace, M Barton Laws, Kevin Small, Ira B Wilson, and Thomas A Trikalinos. 2014. Automatically annotating topics in transcripts of patient-provider interactions via machine learning. *Medical Decision Making*, 34(4):503–512.