Automatically Generating Case Notes From Digital Transcripts of Doctor-patient Conversations Using Text Mining

Nazmul Kazi[1] **and Indika Kahanda**[1]

[1]**Gianforte School of Computing, Montana State University, Bozeman, Montana, U.S.A**

Corresponding author:
Indika Kahanda[1]

Email address: indika.kahanda@montana.edu

ABSTRACT

Current health care systems require clinicians to spend a substantial amount of time to digitally document their interactions with their patients through the use of electronic health records (EHRs), limiting the time spent on face-to-face patient care. Moreover, the use of EHRs is known to be highly inefficient due to additional time it takes for completion, which also leads to clinician burnout. In this project, we explore the feasibility of developing an automated case notes system for psychiatrists using text mining techniques that will listen to doctor-patient conversations, generate digital transcripts using speech-to-text conversion, classify information from the transcripts in to relevant categories, and automatically generate structured case notes.

In our preliminary work, we develop a human powered doctor-patient conversation transcript annotator and obtain a gold standard dataset through National Alliance of Mental Illness (NAMI) Montana. We model the task of classifying parts of conversations in to six broad categories such as medical and family history as a supervised classification problem and apply several popular machine learning algorithms. According to our preliminary experimental results obtained through 5-fold cross validation, Support Vector Machines are able to classify an unseen transcript with an average AUROC (area under the receiver operating characteristic curve) score of 89%. Finally, we use part-of-speech (POS) tagging, grammatical rules of English language and verb conjugation, we generate written versions of the pieces of text belonging to different categories. These formal text are aggregated in to filling different sections of the EHR forms.

INTRODUCTION

An electronic health record (EHR) is a digital version of a patient's health record. EHRs were introduced as a means to improve 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.

In this work we explore the feasibility of automating the process of generating case notes for psychiatrists. Superficially, we envision a pipeline that records a doctor-patient conversation, generates the corresponding digital transcript of the conversation using speech-to-text and uses machine learning to predict the categories for pieces of text from that transcript. These relevant text is then converted to a more formal written version of text and are used for auto-populating the different sections of the

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27497v2 | CC BY 4.0 Open Access | rec: 28 Jan 2019, publ: 28 Jan 2019

No.	Sample	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: Sometime I hangout 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 1. Example of samples in original dataset.

EHR form. We conduct experiments with 17 artificial doctor-patient conversation transcripts obtained through National Alliance of Mental Illness (NAMI) Montana and present our preliminary results on the task of automatically generating a semi-structured case note using a digital transcript of a doctor-patient conversation.

BACKGROUND

EHRs were introduced to improve the health care system. 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 hand written notes (Payne et al., 2015). Nagy et al. 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 (Nagy et al., 2008). Kaufman et al. also developed a NLP-enabled dictation-based data entry where clinicians can write case notes over voice and able to reduce the time spent on writing case notes by more than 60% (Kaufman et al., 2016).

To utilize the full potential of EHRs, clinicians should generate case notes in both human and machine readable formats (Kaufman et al., 2016). Incomplete and unstructured case notes are difficult to search and access. Stetson et al. developed a tool called Physician Documentation Quality Instrument (PDQI) that clinicians can use to improve the quality of their case notes (Stetson et al., 2012).

Psychiatrists can reduce the time spent on writing case notes by using EHRs with dictation-based data entry. However, the quality of the case notes depends on the clinician's input and will require time and effort to write quality, complete and structured case notes. Psychiatrists mostly collect information from their patients through conversations and these conversations are the major source of their case notes. Many Speech-to-Text APIs are available in the market and can be used to generate digital transcripts by recording these doctor-patient conversations. Then a supervised machine learning model can be trained to classify the information from these digital transcripts. Further, POS tagging with regular expressions can be used to generate formal and structured text from these classified data. We develop a human powered transcript annotator and acquire gold-standard data with help of National Alliance on Mental Illness (NAMI) - Montana. Using these data and Support Vector Machines (SVMs), we develop a supervised machine learning model to classify transcripts into 6 categories, e.g. client details, with an average (area under the receiver operating characteristics curve) AUC score of 89% through 5-fold cross validations.

METHODS

We obtain gold-standard data that contains 17 annotated transcripts with 6 class labels to train our previous classification model. Each sample is a doctor-patient pair of statements where the statement(s) of the doctor are followed by the statement(s) of the patient as shown in Table 1. However, 10 more transcripts are also annotated with same 6 class labels and are received after training the previous model that remain unused. Six class labels and the number of samples per class label are listed in Table 3. The class label "Others" is used to annotate samples that don't fall under any other five class labels.

Each sample contains a doctor-patient pair of statements and it is observed that in the same sample one statement can belong to one class where the other statement(s) to a different class. For example, the first sentence of sample 2 in Table 1 should be annotated with Medical History whereas the rest of the

No.	Sample	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? Sometime I hangout 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 2. Example of samples in new dataset.

Class Labels	Number of samples		
	Old Dataset (17 Transcripts)	Unused 10 Transcripts	New Dataset
Chief Complaint	663	465	4001
Client Details	14	0	30
Family History	23	0	79
Medical History	47	20	141
Social History	63	107	388
Others	27	0	68

Table 3. Number of samples per category

sample with Social History. We model the classification problem as a multi-class problem and not as a multi-label problem. To overcome this limitation, we convert this dataset to a new dataset where each sample is either an assertive sentence or a question-answer (QA) pair by developing a python code that reads through the old dataset and outputs the new dataset. The code uses a function that identifies if a statement is a question. Upon identifying a question the code seeks for an assertive sentence and assigns the first assertive sentence as the answer to the question. Any question(s) found in between are put in the same sample. The code breaks each sample from the old dataset to QA pair(s) or assertive sentence(s) as separate samples in the new dataset and assigns the source class labels. All speaker labels, e.g. *Doctor:* and *Patient:* are removed. The new dataset is generated using all 27 transcripts and the number of samples per class label are listed in Table 3. Example of the new dataset is shown in Table 2 that is generated from the two samples shown in Table 1.

We develop a function to detect questions in text using grammatical rules of forming question in English language (**?**) and the function identifies below listed formats:

1. Any question that starts with a wh-word, e.g. what, how, who.
2. Any question that starts with an auxiliary verb or modal (including contractions e.g. isn't) followed by a noun or pronoun.
3. Any question that ends with an auxiliary verb or modal (including contractions) followed by a noun or pronoun.
4. Any assertive sentence that ends with comma followed by a verb, e.g. correct, right.

Python NLTK module is used for POS tagging for question detection but we observe that in some cases this module does very poor job tagging words with correct POS. For example, “can” is tagged as modal but “Can” as noun. Applying lowercase function leads to another problem, such as “i” is tagged as verb. So, RegEx is used beside this module to acquire correct POS tagging. Moreover, this module can not identify verbs in their contraction forms, e.g. ain't. So, a library containing all contraction forms of verbs is developed, used to identify the contraction forms in text and to generate their expanded forms before POS tagging, e.g. from “could've” to “could have”.

The new dataset is used to train a new supervised classification model using SVMs classifier with linear kernel. Stop word removal and Lemmatization are used as pre-processing and features are extracted

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27497v2 | CC BY 4.0 Open Access | rec: 28 Jan 2019, publ: 28 Jan 2019

using the Bag-of-Words model. AUC scores are collected through 5-fold cross validations and listed in Table 4.

To automatically generate a case note from an unseen digital transcript, the same python code that is used to generate the new dataset is used to break down the transcript into samples where each sample is either an assertive sentence or a QA pair. Each sample is annotated with one of the six class labels using the trained model. The samples with same class label reflects information of similar kind and should appear together in the same section in a case note. So, all samples with the same class label are grouped together; We will address this group as *class-specific group*. The class label “Others” reflects uninterested information and is ignored while generating case notes.

Converting all samples of all five class-specific groups to their formal form will result in generating a case note with 5 sections, one section per group. Formal form (or formal text) of a sample refers to the statements of that sample in assertive form, addressed in third person singular number with correct tense, verb form and sentence structure.

While generating formal text from samples, one benefit we get is that each sample is either an assertive sentence (one statement) or a QA pair (two or more statements). We develop the code to generate formal text in several steps. In the first step, we identify the number of statements in a sample. Samples with a single statement, requires minimum processing to generate the formal text. POS tagging from python module spaCy is used to identify the subject, main verb and the auxiliary verb(s) of the statement. If the subject is a first (I) or second person (you), the subject is replaced with a third person singular form (he/she). The gender of the patient can be acquired by querying the clinician. If the statement 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 unchanged. If the statement does not contain any auxiliary verb, the proper form of the main verb depends on the tense of the statement. If the statement is in present tense, the main verb is replaced with its third person singular form, e.g. *run* with *runs*. For statements in past tenses, the main verb is kept unchanged since the form of the verb is same for all persons, e.g. *took*. However, the system is not developed to check if a statement is in future tense since all future tenses contain at least one auxiliary verb, shall or will.

In a sample of multiple statements (QA pairs) the last statement is an assertive sentence and is the answer to the question. In this case, the source of 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 afterward removing the leading affirmation or negation word (e.g. Table 5, sample 4-5). If the answer does not start with any affirmation or negation word, the answer is further analyzed to see whether it a short answer. If not, the question is ignored and the answer is the formal text of the sample (e.g. Table 5, sample 6). In 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. We are presently working on to generate formal text from these kind of samples. While generating formal text all first and second person pronouns, regardless their position, are replaced with its 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 has no importance to be included in the formal texts.

To change a verb with its third person singular or any other forms, we initially searched for a library that is ready to install, import and use in python. The only library available in the market is NodeBox and is developed using python 2.7 whereas we use python 3.6 for this project. So, to make the library compatible with our code, we convert the code to the python 3.0 version.

RESULTS AND DISCUSSION

The SVMs classification model achieve an average AUC score of 90% and has no improvement over the previous model. However, these scores are not directly comparable since the structure of the samples are not same. The conversion process generates one or more samples from each old samples. Good number of samples in the new dataset have wrong annotation. The annotations of the new dataset need to be re-checked and should improve the accuracy of the model. Moreover, some of the samples in the new dataset contain a question with no answer paired with it. Sometimes, a patient asks a question and the answer from the doctor is in the next sample and the conversion process unable to pair the answers with such questions. Pairing the answer with the question is not challenging, rather to assign a class label to

Labels	Min	Max	Average
Chief Complaint	0.914	0.956	0.939
Client Details	0.586	0.933	0.760
Family History	0.878	0.980	0.914
Medical History	0.722	0.918	0.838
Social History	0.906	1.000	0.966
Others	0.909	0.984	0.957

Table 4. AUC Score through 5-fold cross validation, repeated 10 times.

No.	Sample	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 5. Formal Text Generation

that sample since often the samples are annotated with different class labels. Attempting to assigning a class label with probability will require to train another model targeting this task and may be more expensive than manual annotation. These new pairs with correct annotations should help to improve the accuracy of the model.

We were able to successfully generate formal text from samples with one limitation; generating formal text from samples where context need to be extracted from both the question and the answer. However, the system is able to generate formal text from samples in all other cases. Upon checking some generated case notes manually, we find that they are generated in assertive forms with correct persons, tenses, verb forms and sentence structure.

CONCLUSION AND FUTURE WORK

In this project, with some limitations, we were able to classify transcripts into QA-pair and sentence level and automatically generate case notes by generating formal texts. It proves that it is possible to automatically generate case notes for psychiatrists.

We are currently working on this tool to overcome its limitations. We also intend to build a prototype and send it to clinicians for testing. Some of the codes are developed using NLTK module but spaCy module has better performance comparing to NLTK, specially on POS tagging that is heavily used in this project. We will update our code that is using NLTK to spaCy for better performance. We will work on the new dataset and re-check all the annotations to acquire gold-standard data in the sentence level. We will classify few unseen transcripts in the sentence level using both models to get AUC scores that are comparable and the scores will help to determine if the new model has better performance over the old model. For simplicity, each question is paired with the next assertive sentence and doesn't guarantee that the assertive sentence is the answer to the question. We plan to implement another supervised learning model that will find and pair each question with the true answer. To generate better and complete case notes, we will update the system to generate formal text by extracting content from both the question and the answer. Further, we plan to use PDQI-9 (Stetson et al., 2012) to check the quality of our generated case notes.

REFERENCES

Hsiao and Hing (2014). Use and characteristics of electronic health record systems among office-based physician practices: United states, 2001–2013. *NCHS Data Brief, No 143. Hyattsville, MD: National Center for Health Statistics.*

Kaufman, D. R., Sheehan, B., Stetson, P., Bhatt, A. R., Field, A. I., Patel, C., and Maisel, J. M. (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).

Nagy, M., Hanzlicek, P., Zvarova, J., Dostalova, T., Seydlova, M., Hippman, R., Smidl, L., Trmal, J., and Psutka, J. (2008). Voice-controlled data entry in dental electronic health record. *Studies in health technology and informatics*, 136:529.

Payne, T. H., Corley, S., Cullen, T. A., Gandhi, T. K., Harrington, L., Kuperman, G. J., Mattison, J. E., McCallie, D. P., McDonald, C. J., Tang, P. C., 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.

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