Generating Accurate Electronic Health Assessment from Medical Graph

Zhichao Yang[1], **Hong Yu**[1,2]
[1] College of Information and Computer Sciences, University of Massachusetts Amherst
[2] Department of Computer Science, University of Massachusetts Lowell
zhichaoyang@cs.umass.edu hong_yu@uml.edu

Abstract

One of the fundamental goals of artificial intelligence is to build computer-based expert systems. Inferring clinical diagnoses to generate a clinical assessment during a patient encounter is a crucial step towards building a medical diagnostic system. Previous works were mainly based on either medical domain-specific knowledge, or patients' prior diagnoses and clinical encounters. In this paper, we propose a novel model for automated clinical assessment generation (MCAG). MCAG is built on an innovative graph neural network, where rich clinical knowledge is incorporated into an end-to-end corpus-learning system. Our evaluation results against physician generated gold standard show that MCAG significantly improves the BLEU and rouge score compared with competitive baseline models. Further, physicians' evaluation showed that MCAG could generate high-quality assessments.

1 Introduction

Electronic health record (EHR) is widely used by hospitals in the United States and other countries, resulting in an unprecedented amount of digital data or EHRs associated with patient encounters. In recent years, secondary use of EHRs has helped advance EHR-related computational approaches to foster precision medicine and a learning health system (Evans, 2017).

Rich clinical information is documented in the EHRs. Among many structures and formats in EHRs, a problem-oriented SOAP (Subjective, Objective, Assessment, and Plan) structure is commonly used by providers (Podder et al., 2020). Figure 1 illustrate an example of a SOAP note for an outpatient encounter. Typically, Chief Complaint includes a brief description of a patient's conditions and the reasons for the visit. The Subjective section is a detailed report of the patient's current conditions, such as source, onset, and duration of symptoms, mainly based on the patient's self-report. This section usually includes a history of present illness and symptoms, current medications, and allergies. The Objective section documents the results of physical exam findings, laboratory data, vital signs, and descriptions of imaging results. The Assessment section typically contains medical diagnoses and reasons that lead to medical diagnoses. The assessment is typically based on the content from the chief complaint, and the subjective and objective sections. The Plan section addresses treatment plans based on the assessment.

Inferring clinical diagnosis to generate an assessment is a crucial step during the patient encounter. Earlier expert systems were mainly knowledge-based, typically using decision rules. Later, machine learning approaches were developed, mainly used longitudinal electronic health records (EHR) to predict ICD codes (Subotin and Davis, 2014; Amoia et al., 2018), the diagnostic codes assigned to EHRs after each patient's visit or encounter. However, ICD codes are used mainly for billing purposes and have limitations (e.g., incomplete assignment) when used as the gold standard diagnoses labels (O'malley et al., 2005). In this study, we propose an alternative task. Instead of predicting ICD codes, we intend to build an expert system by directly generating medical assessments. We accomplish the task of automated assessment text generation using supervised machine-learning. Specifically, our system's input is the free-text of chief complaint, subjective sections, and objective sections. The output is the assessment. We train our supervised machine learning models based on the SOAP-structured EHR notes as a text to text generation NLP application. The challenges of this text to text generation include:

1. The length of assessment varies, from being

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3764–3773

short to being verbose. Since a) the assessment is mainly inferred (not a mere summary) from the corresponding subjective and objective sections, and b) assessment also includes reasons for diagnoses, thus the overlap between the input and output word tokens is small. Our EHR data shows that there is only 12.8% word overlap between subjective and objective sections and the corresponding assessments. This makes the text generation a challenging NLP task.

2. Both subjective and objective sections are verbose, containing abundant medical jargon, many of which are sparse (with low term frequency) and therefore could be considered as out-of-vocabulary words.

A strong baseline model for automated assessment generation is a Pointer-Generator model N2MAG (Hu et al., 2020). Although the model helps mitigate the out-of-vocabulary challenge, it however does not address the challenge of limited word overlap between the subjective and objective sections and the assessment.

Therefore, we propose a new model for automated clinical assessment generation (MCAG), which generates assessment using knowledge graph. Specifically, we treat our assessment generation as a concept-to-text generation problem. We first build a local or patient-specific concept graph by NLP-processing the free text of the subjective and objective sections. We then expand this patient-specific concept graph with background knowledge extracted from an external and comprehensive knowledge resource, the Unified Medical Language System (UMLS) (Bodenreider, 2004). Once we build the concept-graph, we train the MCAG model end-to-end. Our MCAG mitigates both challenges mentioned above. First, concept normalization (for example, "MI", "myocardial infarction" and "heart attack" can be mapped to the same concept) helps mitigate the out-of-vocabulary word (e.g., MI) challenge. The patient-specific concept graph helps generate the reasons for the diagnosis, and the expanded concept graph with the background knowledge helps infer novel text (diagnosis) not described in the input text (i.e., chief complaint, subjective and objective sections).

The contributions of our work are threefold:

(1) To our knowledge, this is the first study that explores using knowledge-graph to generate EHR texts.

(2) Our knowledge graph incorporates not only the local or patient specific concept relations extracted directly from EHR notes, but also rich background knowledge from an external knowledge graph.

(3) Through extensive experiments, our results show that both graph neural network architecture and expanded medical background information graph helps in generating accurate assessment.

2 Related work

2.1 Text generation in EHR

Motivated by sharing EHR note data without compromising patient privacy information, much work in EHR-related text generation focused on generating synthetic EHR notes. However, most of their work uses discrete features or text data as input, while we use graph, discrete features connected together with relations. Choi et al. (2017) proposed generating synthetic patient records using a combination of an autoencoder and generative adversarial networks (GAN). However, this method only generates high-dimensional discrete vaiables (e.g., diagnosis, medication, or procedure codes) that acts as patient records for secondary analysis instead of free text. Lee (2018) developed an encoder-decoder framework where the encoder's input consisted of numerous discrete variables (e.g., age and ICD codes), and the output of the decoder was chief complaint text. Guan et al. (2018) used the same GAN framework to generate the chief complaint using its EHR note text as the input but not the structured graph data formats that we propose. While most previous works generated short EHR text (usually less than 30 words) from either discrete variables or free text, our work targets a novel task: generating document-wise text from the medical graph.

The most relevant work is Hu et al. (2020), who proposed augmented attention-over-attention pointer-generator network to summarize the content from the "subjective" and "objective" sections. However, this summarization approach usually generates short and concise summaries. While the diagnosis information can be copied and pasted from the input text, the model is limited in generating novel content, which in our application, include differential diagnoses or other important related discussions that do not appear in the input text.

2.2 Structured data to text

Wiseman et al. (2017) studied the challenges of

CHIEF COMPLAINT: Medical Center Patient: <Patient name> <Acct .#> <MR#> <Date of Birth> <Date of Service> <Address> <Physician name> <dictation date> This is a summary of the follow up assessment for <Patient name> , who is pursuing weight loss surgery .

BACKGROUND
SUBJECTIVE: She attends today's nutrition consultation with the hopes of pursuing weight loss surgery . She tells me today that she has met with Dr . AAA in the Diabetes Center on her type 2 diabetes. She started on a new drug for her diabetes instead of lantus insulin. She tells me that as a result , she has been doing a lot better with having smaller portions and also she states that her blood sugars are between 134 and 137 mg/dL on average . She has yet to have another hemoglobin A1c reading , but that is scheduled for 2 weeks from now . As far as her eating , again , she states that her portions are smaller and she is making better choices ... EXERCISE: She reports that she is exercising 2-3 times a week for about a half an hour at Curves . ALLERGIES: Is allergic to lantus insulin .
OBJECTIVE: Height is 5 feet 9 inches , weight is 248 pounds , which indicates a 7-pound weight loss since her last nutrition appointment in 2016. BMI is 36, and excess body weight is 103 pounds .

ASSESSMENT: The patient attends today's nutrition consultation to address her struggle with obesity . She is doing distal better with her blood sugar control , and is doing better reducing her portion sizes . We talked about using Saxenda , a weight loss drug , to further help her . She indicates that she is being more mindful of her food choices , and has been steadily exercise . Based on her current blood sugar reading she believes her A1c will exhibit a downward trend .

Figure 1: An example of SOAP electronic health record note (deidentified). Colored words represent important medical keywords found by metamap tool.

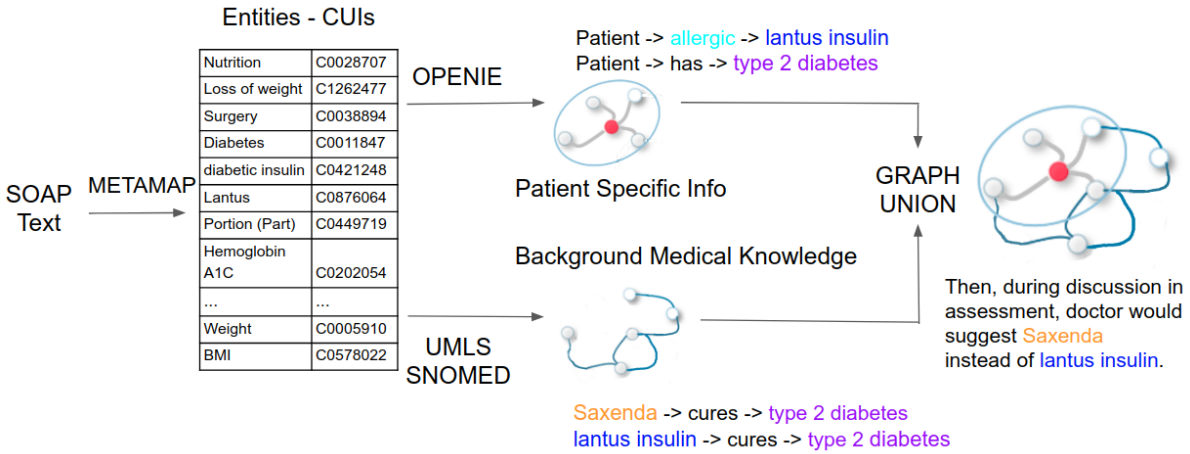

Figure 2: Our text to graph framework learns to build graph from electric health record text using automatic information extraction tools and health database with a real-world example why drug Saxenda is recommended. SOAP TEXT here are Subjective and Objective text in Figure 1.

applying neural networks to the data-to-text task. They introduced a large-scale dataset where a text review of a basketball game is paired with tables of team and player statistics (points, field goals, rebounds, etc.). However, these tasks focused on text generation from tables, where relation info is not included.

Due to the success of transformer model in applications such as machine translation and graph neural network, there is a recent trend to generate longer text (such as paragraph-level text) from structured data. Our work is most similar to (Koncel-Kedziorski et al., 2019), which further introduced a graph to text task by collecting 40k Semantic Scholar Corpus taken from the proceedings of AI conferences. Given a knowledge graph constructed by an automatic information extraction system and a scientific article's title, the goal is to generate a corresponding abstract. However, their graph only captures relevant information parallel to the text, but not extra info from the background. More specific dataset differences are shown in table 1.

3 Method

3.1 Text to Graph

To build a concept graph used later for assessment generation, we first need to build a Patient Specific Information Graph by extracting triples from text in the subjective and objective sections. We make use of OPENIE (Stanovsky et al., 2018) to extract triples, each of which consists of a subject (usually

the patient), object, and their open domain relation specified in the text. This graph should share most of patient's key clinical information stated in the subjective and objective sections of each EHR, including past diagnosis, symptoms, current medications, allergies and etc.

However, we also need to increase word overlap between the subjective and objective sections and the assessment section. Unified Medical Language System (UMLS) (Bodenreider, 2004) is applied to build a Background Medical Knowledge Graph. The UMLS includes a large biomedical thesaurus that is organized by concept (meaning) and concept relations from nearly 200 different professional medical vocabularies. This step allows nodes like symptoms, diagnosis, and treatment to be linked together, which constitute the patient's relevant background knowledge.

Before we build a medical concept graph for each EHR, we first need to extract all medical relevant entities as key clinical info. We use MetaMap (Aronson and Lang, 2010) to identify all key medical phrases and map them to certain medical concepts named as Concept Unique Identifiers (CUIs) in the Unified Medical Language System. The use of MetaMap allows us to associate extracted lexicons with their conceptual semantics, since words/phrases will be mapped to the same CUIs if they are semantically equivalent. For example, "MI," "myocardial infarction" and "heart attack" can now be mapped to the same concept. This mitigates the out-of-vocabulary word (e.g., MI) challenge.

To build a Patient Specific Info Graph G_s, we use OPENIE (Stanovsky et al., 2018) to extract all relevant relations mentioned in the text. We only include triples where CUIs exist because they represent key clinical info with respect to the specific patient. Since sentences from EHR text are not necessarily written in a grammatical manner, with clear subject-predicate-object structure, we rely on matching rules to identify spans of text corresponding to the symptomatic and other personal information of each patient (gender, age, etc.). We found that most graphs are centered around the patient entity as the red dot shown in Figure 2.

To build a Background Medical Knowledge Graph G_b for MCAG EXT model, we use UMLS SNOMED Clinical Terms Database (Bodenreider, 2004) to search for all potential connections between every pair of CUIs. If a 1-hop connection is found, we include both the new entity and relations to the graph.

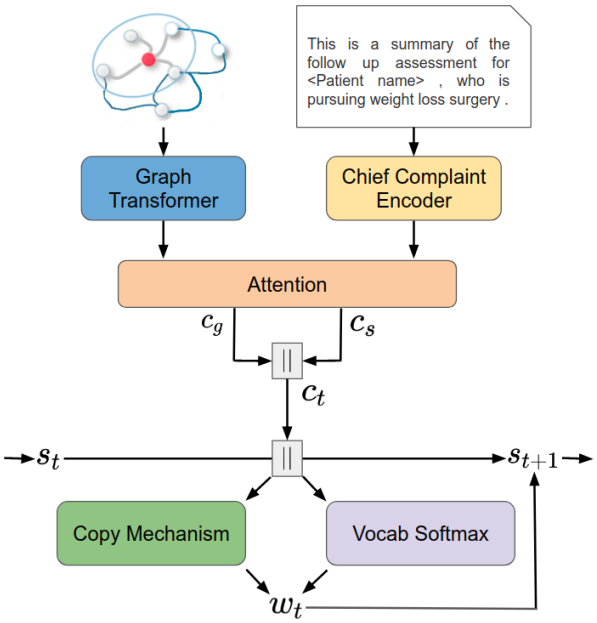

Figure 3: Our graph to text framework: learns to generate assessment from objective and subjective sections in graph and chief complaint in text.

We then combine nodes and relations from both the Background Medical Knowledge Graph G_b and Patient Specific Info Graph G_s, into a combined information graph G, by computing the graph union ($G = G_b \cup G_s$).

3.2 Graph to Text

We first apply graph neural network to knowledge graph with an encoder-decoder framework. As shown in figure 3, given a knowledge graph constructed by an automatic information extraction system in section 3.1 and the chief of complaint, the goal is to generate a corresponding assessment in text.

3.2.1 Encoder

To encode the graph, we use or graph attention neural network. First, to associate a node (mostly multiple words in a medical phrase) to the graph with a continuous representation, we use the last hidden state of a bidirectional RNN run over embeddings of each word in the entity phrase. The output of this embedding step is a matrix $H^0 = \{h_0^0, h_1^0, ..., h_N^0\}, h_i^0 \in \mathbb{R}^D$, (where N is is the number of nodes and D is the number of features in each node) which will serve as input (layer 0) to the graph transformer model. The layer then produces a new set of node features

$H^1 = \{h_0^1, h_1^1, ..., h_N^1\}, h_i^1 \in \mathbb{R}^{D'}$, as its first layer output. This step would be repeated for multiple layers to embed graph extensively.

In order to better encode the input features into next-level features, we use some extra parameters. First, a linear transformation is carried out by two weight matrix, $W_Q \in \mathbb{R}^{D' \times D}$ to obtain a Query matrix and $W_K \in \mathbb{R}^{D' \times D}$ to obtain a Key matrix, then we perform a self-attention to compute attention coefficients which indicate the importance of node j's features to node i.

$$e(h_i, h_j) = (W_Q h_i)^T W_K h_j \tag{1}$$

Then in order to match all attention weights of a probability from 0 to 1, a softmax operation is needed to re-scale the importance of all neighboring nodes N_i of node i.

$$\alpha_{ij} = \frac{exp(e(h_i, h_j))}{\sum_{k \in N_i} exp(e(h_i, h_k))} \tag{2}$$

Once attention weight α_{ij} is obtained, the contextualized representation h_i' of node i is obtained from attending over the connected nodes weighted by attention weight. To stabilize the learning process of self-attention, we employ multi-head attention.

$$h_i' = h_i + \|_{k=1}^{K} (\sum_{j \in N_i} \alpha_{ij}^k W_V^k h_j) \tag{3}$$

where $\|$ denotes the concatenation of the K attention heads, N_i denotes in neighborhood of node i, $W_V \in \mathbb{R}^{D' \times D}$ is used to obtain a Value matrix. Note that, by using concatenating from all heads, the returned output, h_i' , will consist of $K \times D'$ features (rather than D') for each node. Similar to their work (Vaswani et al., 2017), we use block networks, which consists of feedforward network with a non-linear transformation and layer normalization, to reduce the dimension back to D'.

This stacking method enables information to propagate through the majority of graph. Blocks are stacked L times to encode information among L hop nodes , with the layernorm output of layer $l-1$ taken as the input to layer l. The final output matrix $H^L = \{h_0^L, h_1^L, ..., h_N^L\}, h_i^L \in \mathbb{R}^D$ represents contextual information stored in all nodes and relations from the knowledge graph.

To encode the Chief Complaint section, we use a BiLSTM for Chief Complaint word embedding $P = \{p_0, p_1, ..., p_{|C|}\}, p_i \in \mathbb{R}^D$. where $|C|$ is the length of a Chief Complaint sentence. We use BiLSTM encoder instead of graph encoder because Chief Complaint is usually concise and each word could contain lots of information.

3.2.2 Decoder

In order to generate assessment based on the patient and background information input, we train an attention-based decoder with a copy mechanism to extract relevant content from both the knowledge graph and the chief complaint.

At each decoding timestep t we use decoder hidden state s_t to compute context vectors c_g for the graph and context vectors c_s for chief complaint sequence.

To compute context vectors c_g for the graph, we use similar approach shown in equation. 2 and 3. Instead using a specific node as query to be centered, here we replace it with decoder hidden state s_t of previous timestep t. Instead of using a neighborhood centered around a node, here we allow hidden representation from last layer h_j^L from every node V to attend on query.

$$c_g = s_t + \|_{k=1}^{K} (\sum_{j \in V} \alpha_j^k W_{DG}^k h_j^L) \tag{4}$$

$$\alpha_j = \frac{exp(e(s_t, h_j^L))}{\sum_{k \in V} exp(e(s_t, h_k^L))} \tag{5}$$

Similarly to the above equations, we calculate context vectors c_s for chief complaint sequence P following the functions below:

$$c_s = s_t + \|_{k=1}^{K} (\sum_{j \in |C|} \alpha_j^k W_{DT}^k p_j) \tag{6}$$

$$\alpha_j = \frac{exp(e(s_t, p_j))}{\sum_{k \in |C|} exp(e(s_t, p_k))} \tag{7}$$

Here, W_{DG} and W_{DT} are separate trainable decoder weights that differ from query, key, value in the encoder.

To predict the next hidden state, we construct the final context vector by concatenation $c_t = [c_g \| c_s]$. We then use an input-feeding decoder where both s_t and c_t are passed as input to the calculate the next timestep hidden state s_{t+1}. To predict the next word in abstract, the probability of each next token is calculated by scaling $[s_t \| c_t]$ to the vocabulary size with another weight matrix and taking a softmax.

	Abs	ESO	EAS
Vocab	77K	74K	39K
Tokens	5.8M	9.8M	2.2M
Avg Len	142	392	89
Entity Types	5	40	-
Avg Vert	12.42	7.91 (+5.29)	-
Avg Edge	4.43	4.02 (+2.62)	-

Table 1: Vocabulary size of document, number of total document tokens, average document length, number of unique entity types, average number of vertices, average number of edges for AGENDA Abstract(Abs), our EHR subjective and objective part (ESO) and our EHR assessment part (EAS). The average vertices and edges of ESO split into two parts. The first part represents data from patient specific Info graph, while the second one represents data from patient background Info graph.

4 Experiments

4.1 Datasets

We collected a corpus of 25.2K outpatient EHR notes from hospitals and medical centers , from which we randomly selected about 17.5K, 7.6K, and 100 notes for training, development, and test sets, respectively. Statistics of our dataset and a similar AGENDA dataset are available in table 1. However, our dataset is not parallel. Additional background information is added within our graph.

4.2 Baselines

We compare our MCAG against several baselines. In our graph model, we only keep Patient Specific Info Graph and left out Background Medical Knowledge Graph to test the need for it (MCAG Basic). Then, we compare it with augmented attention-over-attention pointer-generator network (N2MAG model) from Hu et al. (2020). We also compare the result of MCAG Basic with self-attention based architectures. We implemented a text to text vanilla transformer with 6 layers of encoder and decoder. To test the ability of Background Medical Knowledge Graph, we also compare the result of MCAG Ext to pretrained generation model on large corpus T5(Raffel et al., 2019), where T5-Small is the encoder-decoder model with 6 layers each, and T5-Base is the encoder-decoder model with 12 layers each. We further finetune these models on our dataset.

4.3 Implementation

Our models are trained end-to-end with EHR chief complaint text and relevant graph as input and corresponding assessment as target. We use SGD optimization with momentum (Qian, 1999) the best learning rate is 0.05 and momentum is 0.9 with gradient clipping. Models are trained for 25 epochs with early stopping (Prechelt, 1998) based on the validation loss, with most models stopping between 15 epochs. Each word is embeded into 500 vectors and the same dimension is used on hidden state size. As for graph encoder, we use a graph attention network (Veličković et al., 2018) with 6 layers with 4 heads. To encode chief complaint text, we use a 2 layer BiLSTM. To avoid penalizing repeatedly attending to the same locations, coverage loss weight is set to 0.5. During inference, we use beam search with a beam size of 4 and beam width of 6 to generate EHR assessments. To prevent overfitting, a dropout rate 0.1 (Srivastava et al., 2014) is used. For each method, experiments is run for 4 trials with random weight initialization, and the best model is selected to do evaluation for each method. We removed repeated sentences manually before evaluation. The whole experiment is carried out on 2 TITANX GPUs. Each model finished training within 12 hours. [1]

4.4 Evaluation Metrics

BLEU As a standard evaluation metric for text generation, BLEU (Papineni et al., 2002) measures the intersection of n-grams between the generated assessment and the gold assessment. A better generated assessment usually achieves higher BLEU score, as it shares more n-gram with the gold assessment.

ROUGE As a standard evaluation metric for summerization, ROUGE (Lin, 2004) also measures the intersection of n-grams between the generated assessment and the gold assessment. But unlike BLEU, it focuses on the n-grams appearing in the machine generated assessment as a measure of recall instead of precision. A better generated assessment usually achieves higher ROUGE score, as it shares more n-gram with the gold assessment.

Human evaluation While BLEU and other automatic metrics are objective metrics that could be applied to large-volume test set, we also ensure that our model works by human evaluation. We hired 4

[1] Our code and setting will be publicly available at https://github.com/whaleloops/mcag

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
N2MAG	9.726	5.449	2.12	1.412	22.334
Transformer	27.053	16.761	11.488	8.457	20.613
MCAG Basic	**27.926**	**17.117**	**12.158**	**9.046**	**23.289**
T5 Small	28.534	17.720	12.323	9.190	20.419
T5 Base	30.542	18.006	12.124	8.772	19.155
MCAG Ext	**38.731**	**26.667**	**20.299**	**15.942**	**30.662**

Table 2: Automatic scores of generated assessment from previous EHR sections. Transformer is the vanilla transformer with 6 layers encoder-decoder. T5 Small uses the same architecture but is pretrained on large corpus and T5 Base doubles the number of layers. MCAG Basic is the 6 layers encoder decoder model which generates assessment from patient specific info graph. MCAG Ext is the the same model which generates assessment from patient specific info graph and background info graph.

Model	Sentence Fluency	Keyword Coverage	Clinical Accuracy	Differential Discussion
N2MAG	2.92	2.14	2.07	1.97
MCAG Basic	3.31 (**+0.39**)	2.31 (+0.17)	2.10 (+0.03)	2.35 (+0.38)
MCAG Ext	3.48 (+0.17)	2.73 (+0.42)	3.13 (**+1.03**)	3.08 (**+0.73**)
Human	3.70 (+0.22)	3.23 (**+0.50**)	3.55 (+0.42)	3.38 (+0.30)

Table 3: Human evaluation results of generated assessment previous EHR sections. We report the mean scores for each evaluation metric of 30 EHR notes. Scores improved the most in each category are highlighted.

doctor experts to join our human evaluation.

We ask evaluators to compare each generated assessment and gold assessment from four perspectives: 1) Sentence Fluency: Is the generated assesssment semantic coherent and meaningful, (e.g. "get a flu shot" is good and "drink a flu shot" is bad). 2) Keywords Coverage: Does the keywords match between assessment and background? (Is the patient male or female? Age same? Times of visit same?). 3) Clinical Accuracy: Is the generated assessment semantically reasonable compared to the given background. 4) Differential Discussion: Coverage of elements in assessment (Does it contain Problem ? Differential Diagnoses? Discussion? Care/Politeness to patient). The grading scale for each perspective is from 1 to 5. For some specific generated and doctor written EHR used for evaluation, please refer to supplementary materials.

4.5 Results

As illustrated in Table 4. Given the chief complaint and background of a patient, we generate three assessments using our models mentioned in section 4.2. Intuitively, the more the generated assessments resembles the gold assessments, the better the model is. We report BLEU and ROUGE-L scores in Table 2 and mean human evaluation scores in Table 3.

According to experiment results, BLEU scores and scores in human evaluation are generally consistent with each other. We observe all BLEU scores are fairly low; we believe it is reasonable as there could be multiple ways to compose an assessment given background of a patient.

Graph based model leads to high precision. Compared to the graph transformer based models, the pointer generator are more susceptible to two sources of errors: 1) the pointer generator tends to generate shorter assessment centered upon a fewer number of medical keywords; 2) the pointer generator also lacks the ability to select multiple keywords and expand upon these keywords.

As shown in Table 4, the result produced from pointer-generator only contains 2 medical keywords in bold, while result produced from MCAG Basic contains 5. Within the test dataset, the average number of medical keywords extracted from N2MAG and MCAG Basic is 3.4 and 7.1 respectively.

Recall that Bleu measures precision: how often the tokens in the machine generated assessment appear in the doctor reference assessment. Rouge measures recall: how often the tokens in the doctor reference assessment appear in the machine generated assessment. Although MCAG without graph enhancement has much better BLEU score compared to pointer-generator, it does not improve a lot in ROUGE-L compared to pointer-generator. This shows that pointer-generator works as a summarization model, and its ability is restricted in keyword selection. As a result, it tends to generate shorter assessments, hence gaining a more favorable score on ROUGE-L (the gap between the pointer generator

Model	Text
N2MAG	ASSESSMENT: The patient attends today's **OBESITY CONSULTATION** . She seems to have a good amount of past nutrition **EDUCATION** .
MCAG Basic	ASSESSMENT: The patient attends today's nutrition **CONSULTATION** to **ADDRESS** her **OBESITY** issue . 1. She is doing better on all **BLOOD SUGAR MANAGEMENT** . 2. she is exercising many times a week . At this point , I do feel comfortable having her move **WEIGHT LOSS** next step in our program .
MCAG Ext	ASSESSMENT: The patient attends today's nutrition **CONSULTATION** to **ADDRESS** her struggle with **OBESITY** . She is doing better on **BLOOD SUGAR MANAGEMENT** and suggestions made by this provider . She has made a number of changes to her diet and lifestyle over the past few months . She is very engaged in our appointment today and asked appropriate **EXERCISE** questions to the education that was provided . We talked about using **Saxenda** as an alternative. At this point , I do believe that her **HEMOGLOBIN A1c** step **DOWNWARD** .

Table 4: An example of assessment generated by different models. The input and gold assessment could be found in Figure 1. MCAG Basic represents the model which generate assessment from patient specific info graph. MCAG Ext is the model which generate assessment from patient specific info graph and background info graph. Medical keywords selected from entities and relations in graph are marked as bold. N2MAG does not have graph, so MetaMap and some rules are used to find these medical keywords. More examples could be found in appendix.

and graph is closer according to ROUGE-L). This is also proven in human evaluation as well. MCAG without graph enhancement achieves a +0.03 point improvements in clinical accuracy, but +0.38 point improvements in differential discussion and +0.39 point improvements in sentence fluency. Comparing to pointer generator model, graph model shows more capability to include medical keywords and generate related discussions and differential diagnoses.

We further compare our MCAG Basic model with a non-pretrained text-to-text transformer model. While transformers can be seen as GNNs from an architecture perspective, our MCAG model use only keywords (graph) extracted from text as input, while this baseline transformer model uses more text as input. However, as shown in Table 4, their performance is similar to ours without using external knowledge. This shows that the medical assessment generation task relies mostly on keywords, and more irrelevant input would not do better in this task.

Incorporating background medical graphs gives better agreement with experts. Among two graph based models, enhancing the graph by expanding relevant background entities with UMLS would further improve the quality of the generated assessments. By comparing clinical keyword identified among the generated and gold assessment, this expanding technique can increase the clinical keyword overlap from 35% to 97%. Graph enhancements further significantly improves Clinical Accuracy by +1.03 and Differential Discussion by +0.73. But not so much in sentence fluency as the model architecture is not altered. This shows the importance of expanding relevant background entities from a graph level in this task as more information is given.

Explicit knowledge graph outperforms implicit pre-trained model. Even though pre-trained language models are able to answer queries structured as "fillin-the-blank" cloze statements, and Petroni et al. (2019) have shown that factual relational knowledge already presents within these pre-trained models, however, Poerner et al. (2019) have demonstrated that these pre-trained language models could only capture shallow information stored in the knowledge base, and incorporating BERT with entity embedding outperforms original BERT (Peters et al., 2019).

Here we present similar findings, but in text generation task. Within automatic evaluations shown in Table 2, our MCAG Ext model with graph enhancement outperforms pre-trained T5-Small, where the number of parameters is about the same. By doubling the number of layers, T5-Base only increases a little in BLEU but decreases slightly in ROUGE-L compared with T5-Small. Both pre-trained models outperform the non-pretrianed vanilla transformer. This may indicate that pre-trained language models from general web corpus contain only limited knowledge on a specific domain (i.e., medical). And explicitly integrate self-attention encoder with knowledge graph would im-

prove the quality of generation text compared to the pre-trained language model.

We also show that assessment generation is an arduous task. Even doctor written assessment gets a medium score of about 3.5 in Table 3 instead of the full 5 points.

5 Conclusion

In this paper, we propose a novel task of generating medical assessment from not only patient specific medical information but also relevant backgrounds. We adapt the graph transformer model to our task and meanwhile proposed an additional approach to address the lack of relevant background medical knowledge. Experiments show that graph transformer outperforms text pointer-generator model, even without the help of additional background medical knowledge. In addition, enhancing the graph with relevant medical knowledge could further improve the generated assessment quality. Experiments also show the current Text-to-Text Transformer pretrained on large corpus may learn limit medical domain-specific knowledge. Further generation quality improvements could be made by incorporating domain-specific knowledge graphs.

In the future, we plan to explore: (1) Probing tasks to randomly switch some entities to other irrelevant and improper tokens, and see if graph model is more resilient to these noises; (2) Many EHRs are follow-up EHRs that is based on the previous EHR. We wish to further expand EHRs in time step by applying temporal graph models to incorporate temporal information.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. Additionally, we thank Pengshan Cai, Shufan Wang, Dongxu Zhang and the UMass NLP group for suggestions that improved the paper's clarity, coverage of related work, and analysis experiments.

References

Marilisa Amoia, Frank Diehl, Jesus Gimenez, Joel Pinto, Raphael Schumann, Fabian Stemmer, Paul Vozila, and Yi Zhang. 2018. Scalable wide and deep learning for computer assisted coding. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 1–7, New Orleans - Louisiana. Association for Computational Linguistics.

Alan R. Aronson and François-Michel Lang. 2010. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association : JAMIA*, 17 3:229–36.

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl$_1$) : $D267 - -D270$.

Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. 2017. Generating multi-label discrete patient records using generative adversarial networks.

R. Scott Evans. 2017. Health records : Then , now , and in the future.

Jiaqi Guan, Runzhe Li, Sheng Yu, and Xuegong Zhang. 2018. Generation of synthetic electronic medical record text. *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*.

Baotian Hu, Adarsha Bajracharya, and Hong Yu. 2020. Generating medical assessments using a neural network model: Algorithm development and validation. *JMIR Med Inform*, 8(1):e14971.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2284–2293, Minneapolis, Minnesota. Association for Computational Linguistics.

Scott H. Lee. 2018. Natural language generation for electronic health records. *npj Digital Medicine*, 1(1).

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Kimberly J O'malley, Karon F. Cook, Matt D. Price, Kimberly Raiford Wildes, John F. Hurdle, and Carol M. Ashton. 2005. Measuring diagnoses: Icd code accuracy. *Health services research*, 40 5 Pt 2:1620–39.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference*

on Natural Language Processing (EMNLP-IJCNLP), pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Vivek Podder, Valerie Lew, and Sassan Ghassemzadeh. 2020. Soap notes. *StatPearls Publishing*.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2019. Bert is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised qa. *ArXiv*, abs/1911.03681.

Lutz Prechelt. 1998. Early stopping-but when? In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, page 55–69, Berlin, Heidelberg. Springer-Verlag.

Ning Qian. 1999. On the momentum term in gradient descent learning algorithms. *Neural Netw.*, 12(1):145–151.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, New Orleans, Louisiana. Association for Computational Linguistics.

Michael Subotin and Anthony Davis. 2014. A system for predicting ICD-10-PCS codes from electronic health records. In *Proceedings of BioNLP 2014*, pages 59–67, Baltimore, Maryland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.