Automatic Generation of Electromyogram Diagnosis Report: Task and Dataset

Anonymous ACL submission

Abstract

Report-writing of electromyogram can be problematic for under-experienced physicians and time-consuming for experienced physicians. In this paper, we explore to generate textual report from tabular diagnostic records of electromyogram. We construct the first dataset for this task and demonstrate results of some baseline approaches.

1 Introduction

Electromyography (EMG) refers to the muscle bioelectrical pattern recorded with an electromyograph (Ni et al., 2020). It is one of the major diagnostic tools for identifying and characterizing disorders of the motor unit (Daube, 2002). After the EMG examination, the physicians will get the records of the electrical signals and perform a two-step analysis. Firstly, they analyze the wave form and convert signals to tabular data with pre-defined format. Secondly, they interpret the tabular data to a diagnosis report (Boon et al., 2008).

Figure 1 shows an anonymized EMG diagnostic report. The reports consist of two sections, Findings and Impression. The Findings section lists the key diagnostic results revealed in the tabular data. As for the Impression section, it contains an anatomic or physiologic diagnosis but not a final clinical diagnosis. The Impression should be brief, yet clear and disclose as much information as possible. Intuitively, Findings can be seen as a summary of tabular information, while Impression needs to be inferred in conjunction with the physician’s clinical experience (Katirji, 2002). In this paper, we focus on the task of automatic report generation from EMG tabular data.

There is already a considerable of work for medical report generation (Jing et al., 2019; Liu et al., 2019b; Zhang et al., 2020b)). However, they mainly focus on x-ray images. Here, we introduce a new dataset which contains anonymized tabular result of electrophysiological examination and corresponding diagnostic reports written by physicians and demonstrate a pipeline to generate diagnostic reports from tabular data of EMG examination. This is a the first attempt in this field.

Considering the heterogeneity of the Findings and Impression, we treat the generation of EMG diagnostic report as two tasks, we generate Findings and Impression separately from tabular information of the electrophysiological examination. Both tasks are formalized as table-to-text generation tasks. We trained neural-based models on these two tasks and tried to learn physicians’ clinical experience in EMG diagnosis from a large number of real diagnostic reports.

2 Dataset and Task Description

2.1 Dataset

In this section, we introduce our new annotated dataset MIME (Medical Information Mart for Elect-
in the table, and the last item corresponds to the description of the unit state. We emphasize that such an evaluation scheme is most appropriate when evaluating generations that are primarily intended to summarize information. While Impression needs to be inferred in conjunction with the physician’s clinical experience and there is very little overlap between Impression and tabular information. Designing evaluation metrics for Impression will be more difficult, and we will leave it for future work.

Table 1 gives some basic statistics for our MIME dataset. The vocabulary size is 549, which indicates that the lexicon is very limited in our EMG diagnostic report setting. The average number of records in the table is 266, and the average length of Findings and Impression are 82 and 29, respectively.

## 2.2 Task Description

In this paper, we treat the generation of EMG diagnostic report as a table-to-text task. We generate both Findings and Impression from tabular information of the electrophysiological examination using a pre-trained language model (GPT2) (Radford et al., 2019; Zhao et al., 2019), and two other non-pretrained models as baselines. Details of these models are described in the next section.

### 3 Methods

In this section, we will first introduce some notations, and then we will describe how to generate EMG diagnostic reports using our models by specifying how are our models organized and how is input arranged.

#### 3.1 Notations

Consider the following notations:

- We use \( r_1, r_2, ..., r_n \) to denote a table \( T \), and for a regular table, each \( r \) represents a cell in the table and have a 2-tuples form that contains column name \( r.k \) (key), and cell value \( r.v \) (value).
- We use \( y_1, y_2, ..., y_{|T|} \) denotes a piece of text \( Y \), and each \( y \) is a token or a word.
- Our dataset consists of \((T, Y)\) pairs and it’s worth noting that although we have multiple tables or text segments, we can encode them in exactly the same way, therefore, for convenience, we use the tuple \((T, Y)\) to represent input and output respectively.

#### 3.2 Compared Models

##### 3.2.1 Long Short Term Memory (LSTM)

Our model follows the standard encoder-decoder architecture (Bahdanau et al., 2014), where the encoder encodes the table into hidden representations and the decoder generates text conditioned on these representations.

The first layer of the network consists in learning two embedding matrices to embed the record keys and values. Each record embedding is computed by a linear projection on their concatenation. We use a bidirectional LSTM (Hochreiter and Schmidhuber, 1997).
1977) on top of the cell embedding to obtain the table representation. After the table is represented as a sequence of vectors, a decoder based on LSTM (Hochreiter and Schmidhuber, 1997) is applied to generate text token by token.

### 3.2.2 Transformer

We linearize the table and feed the records into standard Transformer (Vaswani et al., 2017). The linearization of the table consists of a concatenation of row cells. And since each cell (i.e. record) is represented by the key and value, We concatenate them together and get the representation of the cell using a layer of MLP, which is same as the record embedding layer described above.

### 3.2.3 GPT2

We follow previous work on linearizing knowledge base as natural language (Liu et al., 2019a; Zhang et al., 2020a) to propose “table linearization”, which uses template to flatten the table $T$ as a document $P_T = w_1, \ldots, w_T|Y|$ fed into pre-trained language models to generate statement $Y$, where we use $w_i$ to denote the $i$-th word in the generated paragraph $P_T$ and $|T|$ to denote the length of the paragraph (the word $w_i$ is either a table entry or a functional word in the template). The original table $T$ is transformed into a paragraph by horizontally scanning each cell in the table.

After table linearization, we directly feed the paragraph $P_T$ as the input to the pre-trained GPT-2 model and generate the output sentence $Y$. We finetune the model on MIME by maximizing the likelihood of $p(Y|P_T; \beta)$, with $\beta$ denoting the parameters of GPT-2 model (Radford et al., 2019; Zhao et al., 2019).
We use ROUGE and BLEU is perhaps a reasonably effective way of evaluating text generation, we note that it primarily rewards fluent text generation, rather than generations that capture the most important information in the database which is extremely important for medical diagnosis. Our proposed IE system can be used as an approximation to solve this evaluation challenge. The result for Findings generation is in the Table 2 and the result for Impression generation is in the Table 3.

<table>
<thead>
<tr>
<th>Model</th>
<th>B-1</th>
<th>B-2</th>
<th>B-4</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
<th>TC</th>
<th>TM</th>
<th>CS-acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>58.9</td>
<td>54.9</td>
<td>48.6</td>
<td>80.0</td>
<td>66.2</td>
<td>76.0</td>
<td>35.0</td>
<td>29.0</td>
<td>60.4</td>
</tr>
<tr>
<td>Transformer</td>
<td>72.0</td>
<td>68.4</td>
<td>62.2</td>
<td>85.4</td>
<td>74.2</td>
<td>81.9</td>
<td>42.4</td>
<td>34.5</td>
<td>72.3</td>
</tr>
<tr>
<td>GPT2</td>
<td>76.4</td>
<td>73.8</td>
<td>69.5</td>
<td>88.3</td>
<td>80.0</td>
<td>86.1</td>
<td>52.1</td>
<td>42.5</td>
<td>88.4</td>
</tr>
</tbody>
</table>

Table 3: Overall performance of different models for Impression generation.

4.1 Result and Analysis

We use ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002) scores to evaluate our model. And we report BLEU-1, BLEU-2, BLEU-4 scores and the $F_1$ scores for unigram (ROUGE-1) and bigram (ROUGE-2) and longest common subsequence overlap (ROUGE-L).

We also propose two information retrieval (IR) based metrics. These metrics compare the gold and generated descriptions and measure to what extent the extracted facts are aligned or differ. First, we apply an information extraction (IE) system to extract quintuple in Findings. The value ranges of the first four items in the quintuple can be obtained directly from the tables of the training set. The last item is obtained from our manually labeled test set (only 12). For example, in the sentence Tibial nerve H reflex latency upper limit of normal, an IE tool will extract the pair (Tibial nerve, -, H reflex, latency, normal). Second, we compute two metrics on the extracted information:

- **Tuple Coverage (TC)** estimates how well the generated description containing the gold description in terms of mentioned quintuple. Obviously, based on this simple entity extraction IE system, each item in the 5-tuple may contain multiple elements at the same time. When only the extracted quintuple contains the truly labeled quintuple, we call it tuple coverage. For example, quintuple (ulnar nerve/tibial nerve, -, H reflex, latency, normal) covers quintuple (tibial nerve, -, H reflex, latency, normal).
- **Tuple Matching (TM)** measures how well the system is able to generate text containing factual (i.e., correct) facts. If and only if the two tuples are exactly the same, we call it a match.

While ROUGE and BLEU is perhaps a reasonably effective way of evaluating text generation, we note that it primarily rewards fluent text generation, rather than generations that capture the most important information in the database which is extremely important for medical diagnosis. Our proposed IE system can be used as an approximation to solve this evaluation challenge. The result for Findings generation is in the Table 2 and the result for Impression generation is in the Table 3.

As is shown in the table, all models get relatively good textual overlap with reference text. And the pre-trained model achieves the best results on all metrics benefit from the rich language information contained in it. The extractive metrics provide further insight into the behavior of the models. We first note that on the gold documents $y_{1:T}$, the extractive model reaches 70.5 coverage and 50.9 match rate. Using the LSTM model, generation only has a tuple coverage (TC) of 35.0 indicating that 4-tuples are often generated incorrectly. The best pre-trained model improves this value to 52.1, a significant improvement and potentially the cause of the improved ROUGE and BLEU score, but still far below gold. It is worth noting that all the models seem to get a relatively high prediction accuracy for the fifth item on the accurately matched quadruples. This shows that in the Findings generation task, it is more difficult to locate a specific position in the table than to describe its state after finding the precise location.

5 Conclusions

This paper explores the automatic generation of electromyogram diagnostic report. We formalize the generation as two tasks, namely, table-to-findings and findings-to-impressions. To evaluate the generation results, we introduce both token-level and fact-level evaluations. Results of some baselines on our self-constructed dataset are demonstrated.
References


A Related Work

Data-to-text generation Wiseman, Shieber, and Rush (Wiseman et al., 2017) introduced a document-scale data-to-text dataset with relatively large table records and long reference texts and proposed extraction based evaluation metrics for automatically evaluating generation quality. More specifically, they introduced an information extraction module to evaluate content generation, and ordering of the data-to-document model. Puduppully, Dong and Lapata (Puduppully et al., 2019a) model a content-selection and-planning module separate from text generation, with the idea that introducing a direct signal, i.e. a loss on orderly selection of table records would improve generation performance. Gong, Feng, Qin, Bing and Liu. (Gong et al., 2019) presented a hierarchical encoder that learn records’ representation along row and column and obtain row-level representation for subsequent decoding. Jain et al. (Jain et al., 2018) proposed a mixed hierarchical attention based encoder-decoder model to leverage the structural information in tables. Puduppully, Dong and Latapa (Puduppully et al., 2019b) propose an entity-centric architecture such that instead of treating entities as ordinary tokens, they create dynamically updated entity-specific representations and generates text using hierarchical attention on table and entity memory cell.

Automatic Medical Report generation Jing, Xie and Xing (Jing et al., 2018) proposed a co-attention mechanism to localize regions containing abnormalities and generate descriptive texts for them. Jing, Wang and Xing (Jing et al., 2019) proposed a multi-agent framework to exploit the structural features within report sections for generating Chest X-ray Reports where they have two agents for generating text about abnormal and normal results separately with the observation that the distribution between abnormality and normality is imbalanced and the wordings are quite different in text describing abnormal and normal results. Liu et al. (Liu et al., 2019b) proposed a generation model which hierarchically first chooses topics and then generates words from topics and they optimized the model for clinical correctness which a proposed clinically coherent reward via reinforcement learning. Zhang, Merck, Tsai, Manning and Langlotz (Zhang et al., 2020b) leveraged an existing information extraction module to extract a zero-one vector of 14 dimension indicating the presence or absence of 14 clinical observations in chest radiology reports and apply reinforcement learning with a factual correctness reward to improve the factuality of generated reports.