

Generation of clinical letters from keywords or structural data



Description:

Based on text generation technologies to generate formatted clinical letters from keywords or structural data.

Week 1 (previous)

Milestone(Week 1):

- ☒ Understand clinical letters format and the background of text generation

LETTER A	LETTER B
<p>Oxford Radcliffe Hospitals </p> <p>NHS Trust</p> <p>2nd March 2010</p> <p>DERMATOLOGY DEPARTMENT Churchill Hospital Old Road Headington Oxford OX3 7LJ</p> <p>Dr J Smith The Oxford Surgery London Road Oxford OX1 1AB</p> <p>Dear Dr Smith</p> <p>Re: Andrew JONES - DOB 24 January 1978 ***FICTIONAL PATIENT FOR RESEARCH ONLY*** ***DO NOT FILE LETTER IN PATIENT NOTES***</p> <p>Diagnoses:</p> <ol style="list-style-type: none">1. Erythrodermic psoriasis2. Hypertension3. Polycythemia4. Non-alcoholic steatohepatitis5. Obesity <p>I reviewed Andrew again in clinic today. His psoriasis has now flared and he became erythrodermic within a few weeks of stopping psoralen plus ultraviolet A (PUVA). His Psoriasis Area and Severity Index (PASI) score is now 45/72 and his Dermatology Life Quality Index (DLQI) is 16/30. Systemic treatments are now very limited on the grounds of his comorbidities. A biologic agent is now our preferred option.</p> <p>I have given him written information regarding Adalimumab (TNF blocker). We have most of his baseline investigations excluding a chest X-ray and tuberculosis ELISPOT which I have requested. I have asked Mr Jones to attend your surgery to check that his immunisation schedule is complete.</p> <p>I will review him again in four weeks with the results.</p> <p>Yours sincerely</p> <p>Dr James McLeod Consultant Dermatologist</p>	<p>Oxford Radcliffe Hospitals </p> <p>NHS Trust</p> <p>2nd March 2010</p> <p>DERMATOLOGY DEPARTMENT Churchill Hospital Old Road Headington Oxford OX3 7LJ</p> <p>Dr J Smith The Oxford Surgery London Road Oxford OX1 1AB</p> <p>Dear Dr Smith</p> <p>Re: Andrew JONES - DOB 24 January 1978 ***FICTIONAL PATIENT FOR RESEARCH ONLY*** ***DO NOT FILE LETTER IN PATIENT NOTES***</p> <p>Diagnoses:</p> <ol style="list-style-type: none">1. Erythrodermic psoriasis2. Hypertension3. Polycythemia4. Non-alcoholic steatohepatitis5. Obesity <p>I reviewed Andrew again in clinic today. His psoriasis has now flared and he became erythrodermic within a few weeks of stopping psoralen plus ultraviolet A (PUVA). His Psoriasis Area and Severity Index (PASI) score is now 45/72 and his Dermatology Life Quality Index (DLQI) is 16/30. Systemic treatments are now very limited on the grounds of his comorbidities. A biologic agent is now our preferred option.</p> <p>Investigations:</p> <ol style="list-style-type: none">1. Chest X-ray2. Tuberculosis ELISPOT3. Other baseline investigations in place <p>Treatments:</p> <ol style="list-style-type: none">1. Written information regarding Adalimumab (TNF blocker)2. Patient to attend GP surgery to check immunisation schedule complete <p>Follow up:</p> <p>Review in 4 weeks with the results</p> <p>Yours sincerely,</p> <p>Dr James McLeod Consultant Dermatologist</p>

Basic ideas for Week1:

About the main structure of the model:

- First, collect and prepare a dataset of clinical letters. This dataset should be included different types of letters and be **cleaned and preprocessed** to remove irrelevant information.
- Utilize a text generation model such as RNNs or transformers, training it on the clinical letter dataset with the desired keywords as input.
- Fine-tune the model by **adjusting its parameters** to improve its performance in generating clinical letters.
- Input the desired keywords to generate a clinical letter that incorporates those keywords and **follows the style of the training data**.
- Carefully review the generated letter to ensure it meets the necessary requirements.**
- If the model has not been fine-tuned or the dataset is not extensive, the generated letter may require editing for accuracy.

RNN is widely used in the text generation area, for CNN, which contributed mainly to computer vision, can also be used in this project. For the further goal, GAN would be tried to improve the performance of the generation. (because the training dataset cannot cover all the letter conditions)

Furthermore, if the model can achieve a satisfying result, I would try to use EHR to generate clinical letters, which means more processes to preprocess and clean the input of the generation model.

Thus, for the next coming week, I would try to build an RNN structured model to try basic text generation.

P.S.:

For the keywords/structural data, would the words be in the correct order? If not, how to manage it?

Is comparing with manually generated necessary?

$$Recall = \frac{Similar - Tokens}{total - Tokens - in - Training - Summary} \quad (7)$$

Evaluation: Rouge Precision: It actually gives us the system summary which is relevant or needed and is calculated as:

$$Precision = \frac{Similar - Tokens}{total - Tokens - in - Generated - Summary} \quad (8)$$

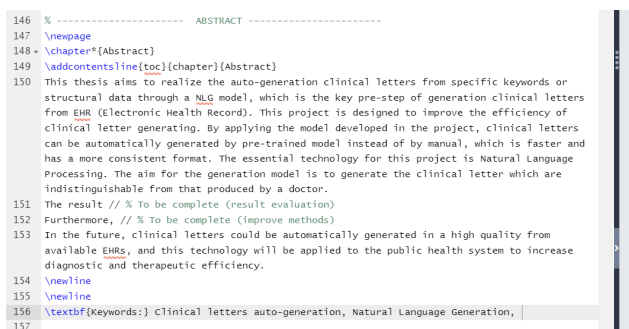
BLEU, METOR, ROUGH..... Unsupervised self-evaluation

Dataset ?

Week 2 (current)

Milestone(Week 2):

- ☒ Prepare for the **proposal**:
 - Developed a latex template for dissertation
 - Writing the basic **Abstract**



- ☒ Generate simple **NLG model** (debugging for encoding and tokenizer)

- **RNN**

```
# Create sequences of input data
num_keywords = 10
sequences = []
for i in range(num_keywords, len(text)):
    sequence = text[i-num_keywords:i+1]
    sequences.append(sequence)

# One-hot encode the input and output data
```

```

x, y = [], []
for sequence in sequences:
    encoded = tokenizer.texts_to_sequences([sequence])[0]
    for i in range(1, len(encoded)):
        x.append(encoded[:i])
        y.append(encoded[i])
y = to_categorical(y, num_classes=total_words)

# Define the model
model = Sequential()
model.add(Embedding(len(tokenizer.word_index)+1, 10, input_length=10))
model.add(Reshape((10, 1)))
model.add(LSTM(50))
model.add(Dense(len(tokenizer.word_index)+1, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam')

# Train the model
model.fit(encoded, epochs=100, batch_size=1, verbose=2)

```

- **Google T5** Reading

For transformer, which always has a better performance in results than baseline model.

Google's T5 is a Text-To-Text Transfer Transformer which is a shared NLP framework where all NLP tasks are reframed into a unified text-to-text-format where the input and output are always text strings.

Finished reproduction of code.

Basic ideas for Week2:

- Compared with traditional RNN, CNN, transformer based on attention mechanism has a better performance on TEXT-generation.
- Should pre-train a based model to adapt the generate text to a new writing style(PubMed abstract)

For coming week3:

- Learn more about **T5 transformer**
- Try to adjust it to fit my objective
- Test for some **clinical text**
- Get more source for writing **Introduction-background**