Sentence-Level End-to-End Data-to-Text Generation

Tony Wong

Department of Computer Science, Yale University, New Haven, CT

LILY Lab

Introduction

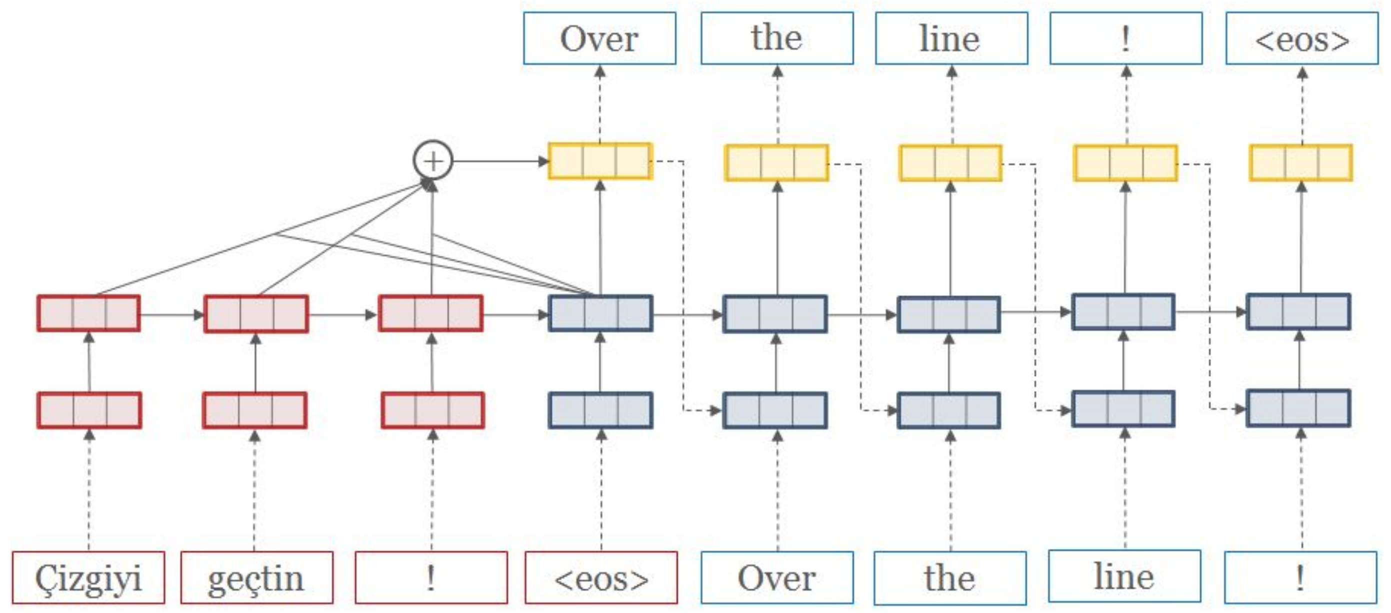
Data-to-text generation (D2T NLG), a subtask of natural language generation (NLG) is gaining popularity in a lot of applications such as sports commentary generation and weather report generation. D2T NLG aims to transform structured inputs (such as tables, json files) into unstructured natural language similar to those written by human. Classical NLG approach here involves breaking down the process into generation into four phases: content selection, document structuring, microplanning and surface realization. Such an architecture is usually involved and requires careful planning and design in each stage. Yet in another sense, D2T NLG can also be regarded as a neural machine translation (NMT) task, which is trained end-to-end and has a relatively simple architecture. While more data with more complex schema might require a hierarchical structure, restaurant ratings data are relatively simple and an NMT model might suffice.

Materials and Methods

The E2E dataset is a dataset with information about restaurants. It contains 50k pairs of dialogue-act-based meaning representations with around 8.1 references on average. Each meaning representation is first converted into structured linear natural language sentences using a fixed rule: [] is converted to “is” and “,” is converted to “and”. Thus, matching with the output sentence, the task of D2T NLG is converted into an English-to-English translation task.

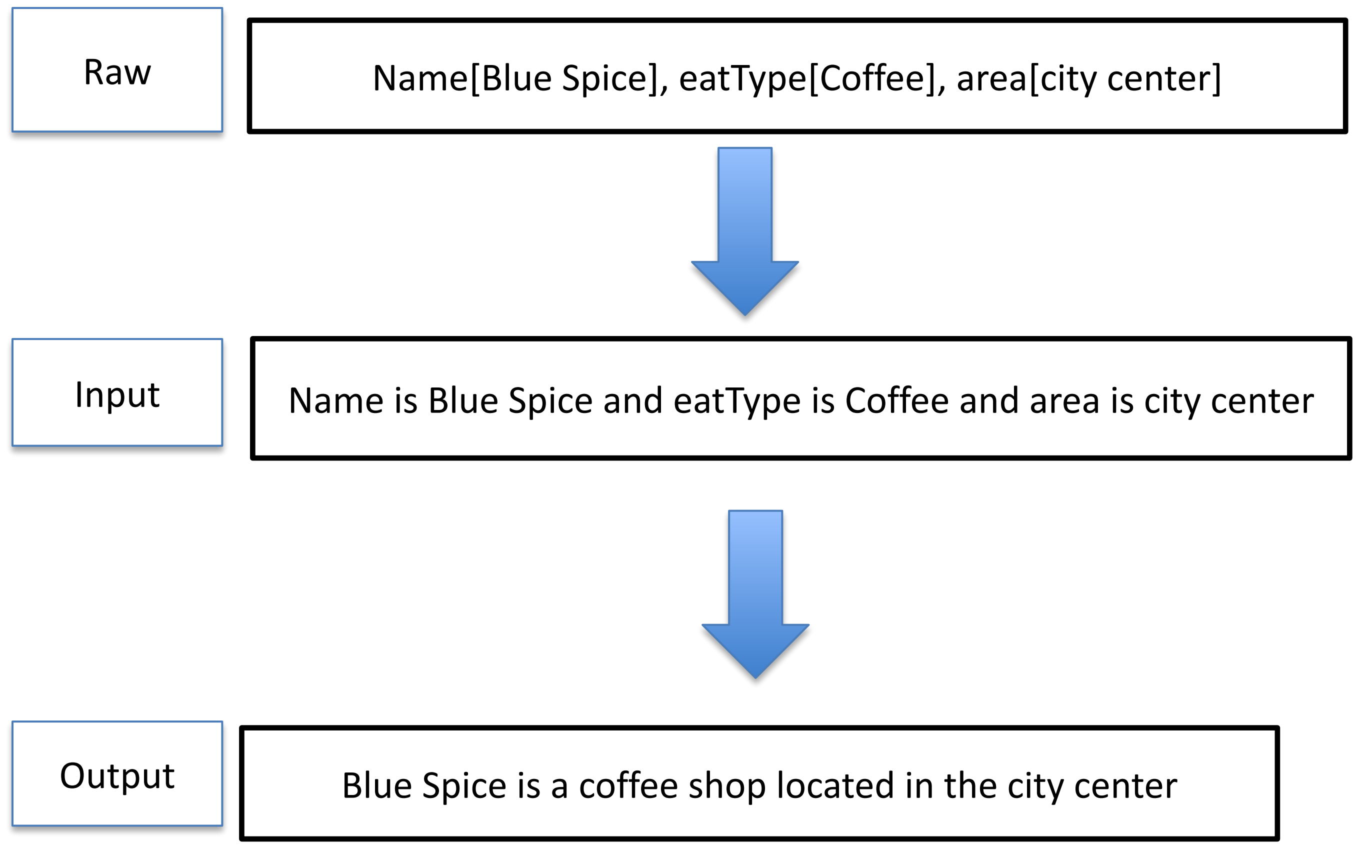
After such data preprocessing, an end-to-end system is trained in the OpenNMT framework to translate the preprocessed sentences into more natural natural language sentences generated by human. OpenNMT is an open source framework based on the neural translation encoder-decoder model. It sees the task of translation as a conditional language modeling by modeling the probability of target sentence given a source sentence. The encoder model consists of an recurrent neural network (RNN) mapping each word to a hidden vector, which the decoder then incorporates with previous generated words’ hidden representation to generate the next word in the sequence. Such a representation is passed through a softmax layer to obtain the next word distribution, which is also affected by hidden vectors as well.

Figure 1. The OpenNMT model. Taken from https://opennmt.net



Attribute	Data Type	Example
name	String	Zizzi
eatType	Dictionary	Coffee shop
familyFriendly	Boolean	Yes
Customer rating	String	low
priceRange	Dictionary	low
Food	Dictionary	French
Near	String	Rainbow Vegetarian Cafe
Area	Enumerable	Riverside

Table 1. Example of E2E attributes and values



Raw: Name[Blue Spice], eatType[Coffee], area[city center]

Input: Name is Blue Spice and eatType is Coffee and area is city center

Output: Blue Spice is a coffee shop located in the city center

Figure 2. Transformation of e2e input

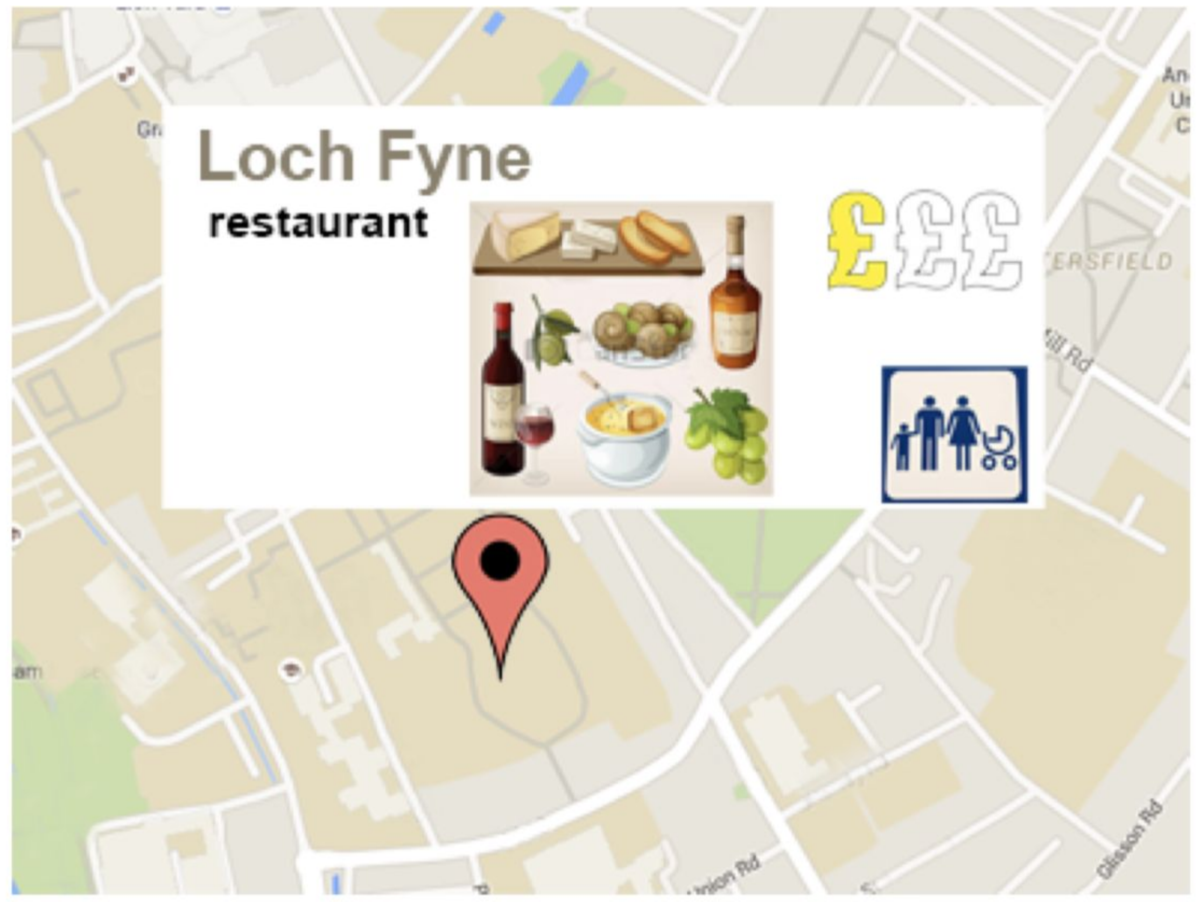
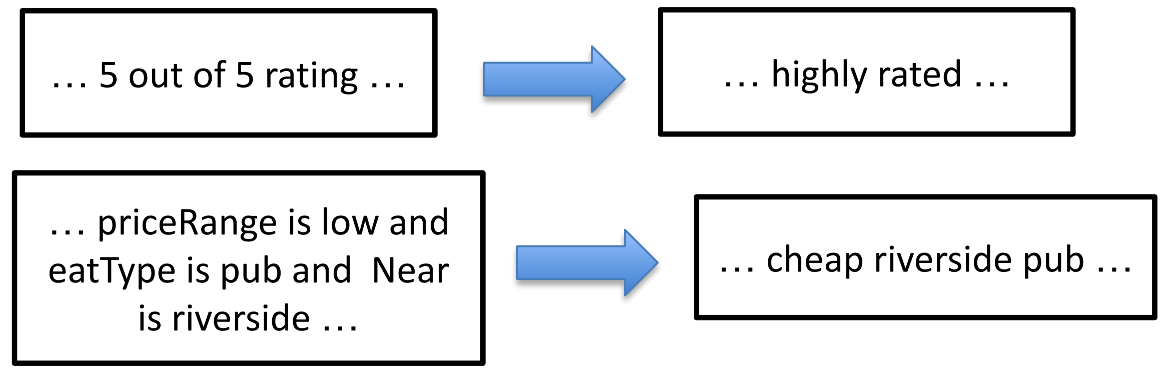


Figure 3. Collection of E2E data. Taken from https://arxiv.org/pdf/1706.09254.pdf



… 5 out of 5 rating … → … highly rated …

… priceRange is low and eatType is pub and Near is riverside … → … cheap riverside pub …

Figure 4. Successful example of language variation.

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L
F-1	0.346	0.134	0.055	0.026	0.312
Recall	0.344	0.130	0.052	0.024	0.307
Precision	0.388	0.155	0.065	0.031	0.343

Table 2. Automatic Evaluation Results: ROUGE

	BLEU	NIST	Meteor
Mean	0.137	1.635	0.294

Table 3. Automatic Evaluation Results: BLEU, NIST, Meteor

	Fluency (1-5)	Variation (1-5)	Precision (0-1)	Recall (0-1)
Mean	4.9	4.5	0.55	0.62
Median	5	5	0.5	0.67

Table 4. Manual Evaluation Results.

Results

After the model is trained with 100,000 steps validating every 500 steps, the generated outputs are evaluated using automatic metrics such as BLEU, NIST, METEOR and ROUGE. Human evaluation is also performed based on (1) fluency (2) variation (3) recall (4) precision of the input information.

The metrics across all four metrics are not very satisfactory. The ROUGE scores decrease from ROUGE-1 to ROUGE-4. The output also do not score well on NIST, METEOR and BLEU. A manual evaluation uncovers more patterns underlying the output sentences. In particular, the linguistic quality of the output generated is excellent and shows almost no grammatical mistakes. Also, the model is able to transform language produced by splitting one sentence into multiple, adding adjectives in front and also paraphrasing. Yet the recall and accuracy are not satisfactory. In particular there are two main problems: 1) The generated output hallucinates, often adding keys that are not present in the input. This causes the unsatisfactory precision of the output 2) In addition, the generated output often puts in the wrong values to the keys, and this is the main cause to the unsatisfactory recall of the output sentence.

Conclusion

The results show that such a simple NMT model is inadequate to generate text even for such simple sentences under a simple schema for a D2T NLG task. Further suggestions would involve introducing a hierarchical planning model as in the classic NLG task, exploiting the inner structure within the meaning representation into better capture the relations between entities, or training a GAN to avoid hallucination. Also, the model can also be pretrained on related corpus to further improves language variety.

Acknowledgement

I would like to thank Professor Dragomir Radev for the guidance throughout the project.