ChiNet

Nil Adell **Joseph Cornelius** **Alexander Nedergaard** **Lama Saouma**

Feel free to add more sections but those listed here are strongly recommended.

1 Introduction

You can keep this short. Ideally you introduce the task already in a way that highlights the difficulties your method will tackle.

2 Methodology

Your idea. You can rename this section if you like. Early on in this section – but not necessarily first – make clear what category your method falls into: Is it generative? Discriminative? Is there a particular additional data source you want to use?

3 Model

The math/architecture of your model. This should formally describe your idea from above. If you really want to, you can merge the two sections. **Xander is on this**

We define our sentence RNN as

$$h_i^s = GRU(s_i; h_{i-1}^s)$$

where s denotes the embedded sentence. We denote the final hidden state of the sentence RNN as r^s.
We then define our document RNN as

$$h_i^d = GRU(r_i^s; h_{i-1}^d)$$

and similarly denote the final hidden state of the document RNN as r^d.
Now, we define our generator RNN as

$$h_i^g = GRU(y_i; h_{i-1}^g)$$

where y_i is the embedded word generated at the previous time step. We set y_0 to the embedded stop-word. Unlike the sentence and document RNN, where the initial hidden states h_0^s and h_0^d are set to 0, we initialize our generator hidden state as

$$h_0^g = r^d + z$$
$$z \sim \mathcal{N}(0, 1)$$

We determine the generated word from the generator hidden state using Gumbel-Softmax:

$$\pi_i = softmax(h_i^g W_{d \to e} W_e^T)$$
$$g_i = -log(-log(u))$$
$$u \sim Uniform(0, 1)$$
$$t_i = relu(h_i^g W_t) + \epsilon$$
$$p_i = softmax(\frac{log(\pi_i) + g_i}{t_i})$$

$$y_i = p_i W_e$$

where $W_{d \to e}$ is a transformation matrix from document to embedding space, W_e is the embedding matrix, W_t is a matrix used to determine the temperature t_i and ϵ is a small number to ensure that t_i is positive. We continue to generate words until the stop-word is generated (we use π_i to check) or the maximum sentence length is reached. We then stack the words to obtain the generated embedded sentence $\bar{s}$.

The discriminator score is defined as

$$D = \sigma(r^d W_{d \to s} (r_t^s)^T)$$

where $W_{d \to s}$ denotes a transformation matrix from document space to sentence space and σ is the sigmoid function. The discriminator assigns the score 1 to target sentences r_t^s that it considers the most likely ending given the document context r_d and 0 to the least likely.
We use attention to assign weights to the inputs of the document RNN, based on their similarity to the target sentence. The weighted sentences are determined as

$$\tilde{r}_i^s = r_i^s \cdot a_i$$

$$a_i = r_t^s W_A (r_i^s)^T$$

where r_t^s denotes the target sentence, W_A is an attention matrix and $\cdot$ denotes the scalar product. We do not use attention when determining the document context for sentence generation, as this would provide information about the ground truth target sentence to the generator.

Now, given a story without the ending (distilled to a single document representation using the sentence and document RNNs) and two endings (both distilled to sentence representations using the sentence RNN), we can determine the most likely ending as having the highest discriminator score D.

4 Training

What is your objective? How do you optimize it? **Xander is on this**

5 Experiments

This **must** at least include the accuracy of your method on the validation set.

6 Conclusion

You can keep this short, too.