Problem Statement

Given a user query and candidate passages corresponding to each, the task is to mark the most relevant passage which contains the answer to the user query.

In the data, there are 10 passages for each query out of which only one passage is correct. Therefore, only one passage is marked as label 1 and all other passages for that query are marked as label 0. Our goal is to rank the passages by scoring them such that the actual correct passage gets as high score as possible.

Mathematical Formulation

Each passage will be scored correspondingly for its relevance with the query with the more relevant ones awarded a higher score than others. We will evaluate using Mean Reciprocal Rank (MRR) which is defined as follows:

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\mathrm{rank}_i}.$$

where rank_i is the rank of the correct passage in the prediction file for the i^{th} query across all Q queries.

Approach

The mapping of a question $q = \{w_1, w_2, \ldots, w_T\}$ to its vector representation, r_q is done using a single-layered unidirectional GRU based encoder network. We call this part of the model the question encoder ENC_Q.

For each time step i with input x_i and previous hidden state h_{i-1}, we compute the updated hidden state $h_i = GRU(x_i, h_{i-1})$ by

$$u_i = \sigma\,(W(u)x_i + U(u)h_{i-1} + b(u)) \quad \ldots \quad (1)$$
$$r_i = \sigma\,(W(r)x_i + U(r)h_{i-1} + b(r)) \quad \ldots \quad (2)$$
$$\tilde{h}_i = \tanh\,(W x_i + r_i \circ U h_{i-1} + b(h) \quad \ldots \quad (3)$$
$$h_i = u_i \circ \tilde{h}_i + (1 - u_i) \circ h_{i-1} \quad \ldots \quad (4)$$

where σ is the sigmoid activation function, ◦ is an elementwise product, $W(z), W(r), W \in R^{n_H \times n_L}$, $U(z), U(r), U \in R^{n_H \times n_H}$, n_H is the hidden size, and n_L is the input size.

The question encoder ENC_Q first uses the word representation function $REP_W(w_T)$ to generate vector representations for all words w_t, $t = 1, \ldots, T$ (as described in the next paragraph), which are subsequently fed to the RNN until all words have been seen. Starting with the initial hidden state h_0, the GRU of the question encoder RNN iteratively updates its hidden state h_t after processing each word according to Equations (3) to (6), where the word representation vector $REP_W(w_T)$ is fed as input to the GRU (i.e. $x_T = REP_W(w_T)$). The final hidden state h_T (produced after processing the last word represented by $REP_W(w_T)$) is returned by ENC_Q as the representation of question q.

As word embeddings, we use GloVe vectors provided on the GloVe website . Such pre-trained word embeddings implicitly incorporate word semantics inferred from a large text corpus based on the distributional semantics hypothesis. This hypothesis can be phrased as “words with similar meanings occur in similar contexts”, which in our case translates to similar vectors for words with similar meanings. Using those pre-trained word embeddings allows us to better handle synonyms and find better matches between words in the question and passages. In addition, during testing, it allows to handle words that have not been seen during training.

A similar procedure is followed to encode the passages. Each passage $p = \{w_1, w_2, \ldots, w_T\}$ is mapped to its vector representation, r_p using a single-layered unidirectional GRU based encoder network called ENC_P. The passage is first split into words $w_p^1, w_p^2, \ldots$, each word is embedded (as described previously), and then the word embeddings are fed into a single-layer word-level GRU-based encoder ENCR that takes the final state of its RNN as the representation of the passage, that is $r_p = ENC_P(\{w_p^1, w_p^2, \ldots, \})$

Now, the scoring of each passage can be done by simply taking a cosine between the passage vectors and corresponding query vector.

To maximise the final MRR score, we have to rank the correct passage as high as possible so we choose loss function as $1\text{-}S_a$ for the correct passage and as S_a for the incorrect ones. This takes care of the fact for rank to be higher, not only does the model need to find the correct passage, but it also needs to figure out the wrong ones.

Code

In the files attached, I have included two main python files namely, text2ctf.py and model.py. Running text2ctf.py on the data converts it into ctf files from which I can take input and compute using model.py.

Evaluation Metric

I obtained the data for evaluation and validation from Microsoft AI challenge dataset comprising of 500K queries and 5 million passages. The model was trained on 90% of the data and validated on 10% of the same. The Evaluation was done by finding the Mean Reciprocal Rank MRR of the complete dataset.

After 150 epochs on the training set, an MRR of 0.53 was achieved.

Other Approaches

The approached used above is from a paper “Neural Network-based Question Answering over Knowledge Graphs on Word and Character Level” where the model was trained using a Knowledge graphs and so the above method works best when supporting facts are not marked during training. But for our purposes we might not always have supporting facts. So, I have come across another paper namely “Dynamic Memory Networks for Visual and Textual Question Answering” . In it they propose a new input module which uses a two level encoder

with a sentence reader and input fusion layer to allow for information flow between sentences. For the memory, they propose a modification to gated recurrent units (GRU)). The new GRU formulation incorporates attention gates that are computed using global knowledge over the facts. Unlike before, the new model does not require that supporting facts (i.e. the facts that are relevant for answering a particular question) are labeled during training. The model learns to select the important facts from a larger set.