

Review

Word embedding:

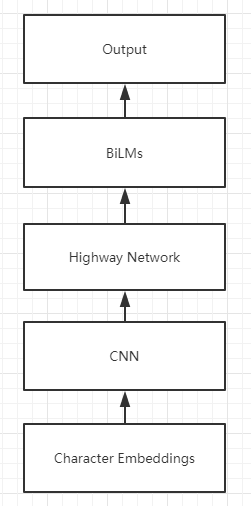
One-hot: This representation is sparse and may suffer the curse of dimensionality with the increase of vocabulary size, meanwhile, it can’t represent relation among words.

Distributed: It encodes words into continuous low-dimensional vectors so that closely-related words are not far away from each other in vector space, which reveals correlation of words.

CoVE: It brings beneficial of transfer learning to NLP tasks, but it requires a great deal of parallel corpus.

ELMo: It can obtain richer word representations by collapsing outputs of all biLM layers into a single vector with a task specific weighting rather than just utilizing outputs of the top layer. But it is limited in a way by the insufficient feature extraction capability of LSTM.

ELMO

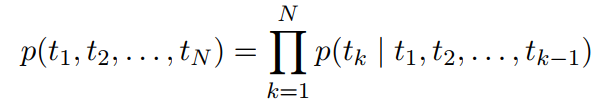


1. Bidirectional language models

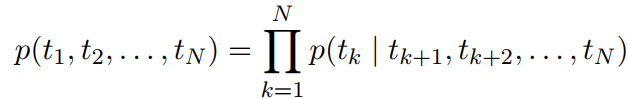
LM:

We usually construct a language model by constructing a string probability distribution.

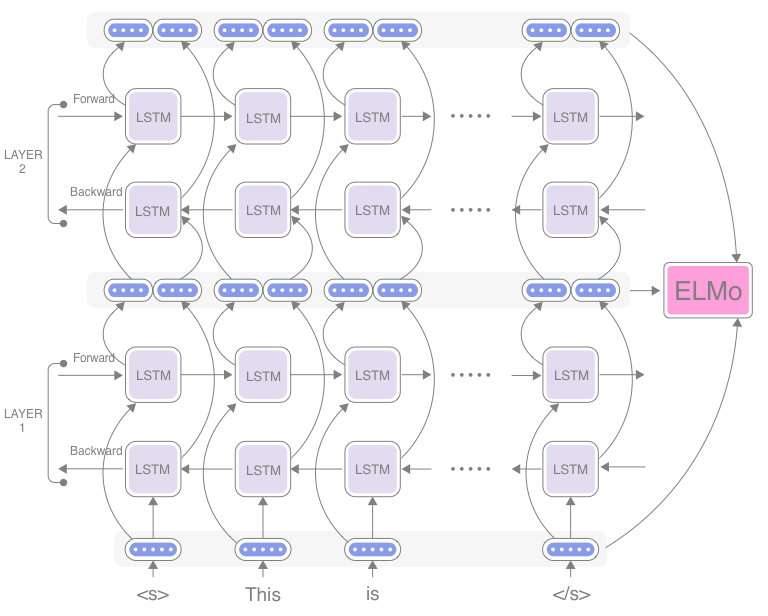
Forward language model



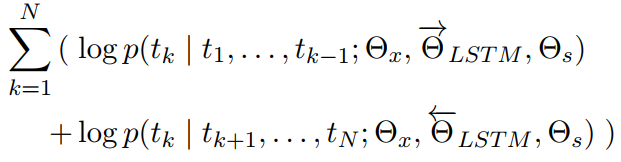
Backward LM



biLM



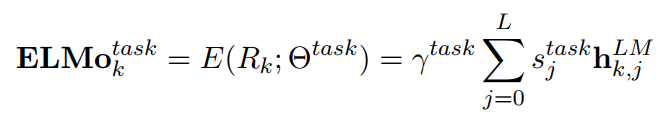
The formulation jointly maximizes the log likelihood of the forward and backward directions:

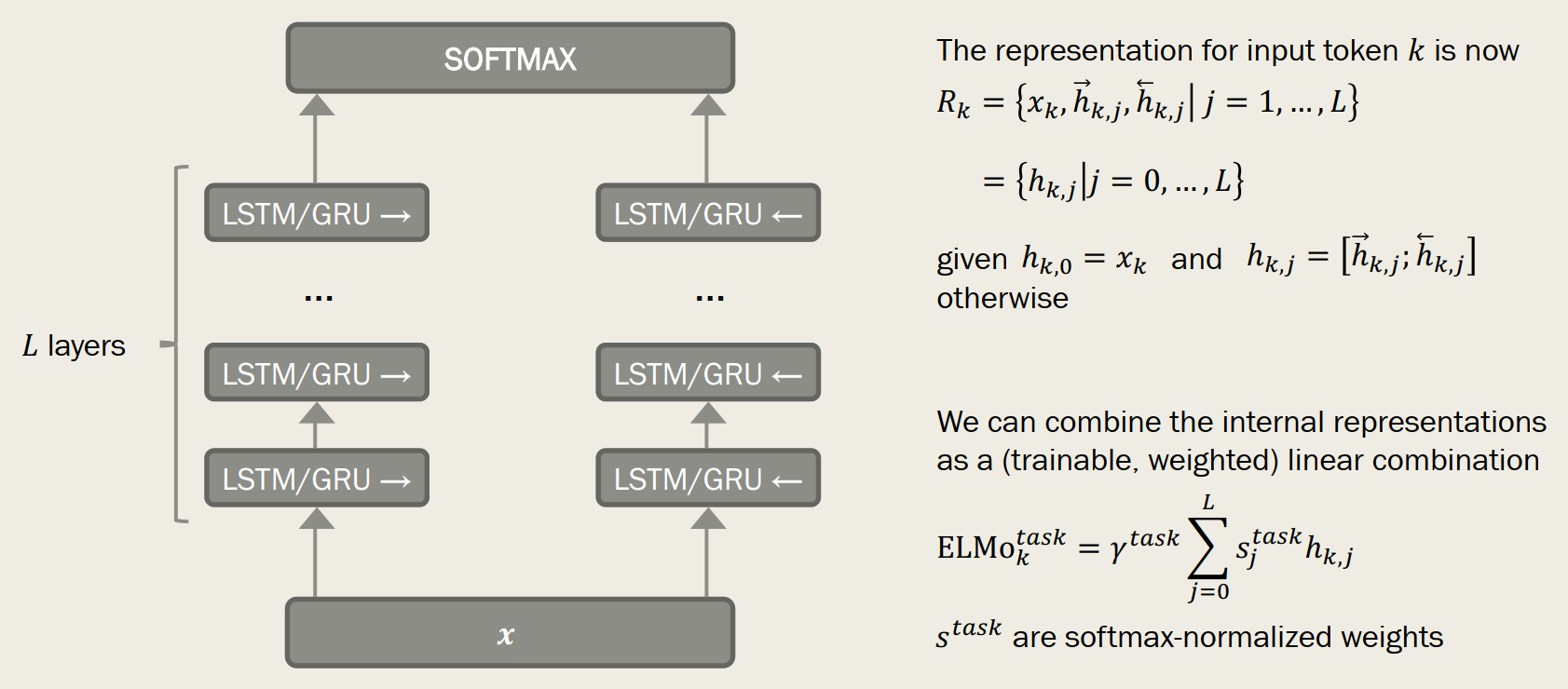


At each position k, each LSTM layer outputs a context-dependent representation

.

The top layer LSTM output is used to predict the next token tk+1 with a softmax layer.



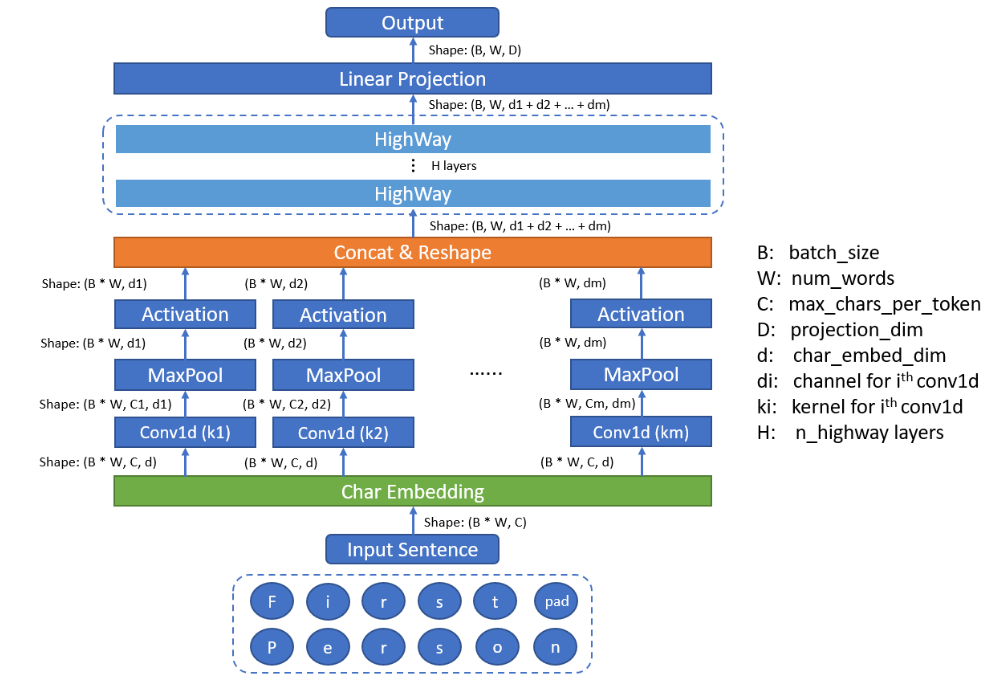


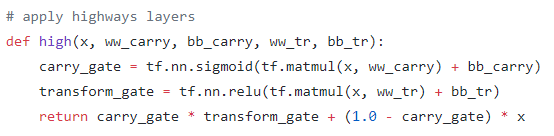
different layers of deep biRNNs encode different types of information

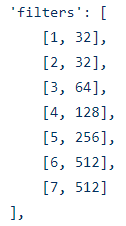
Instead of words, use sub-word units (e.g. characters), and pass them through a CNN

before presenting the now-dense representations to the architecture. Settling OOV problem.

ELMo’s inputs are characters rather than words. They can thus take advantage of sub-word units to compute meaningful representations even for out-of-vocabulary words (like FastText).







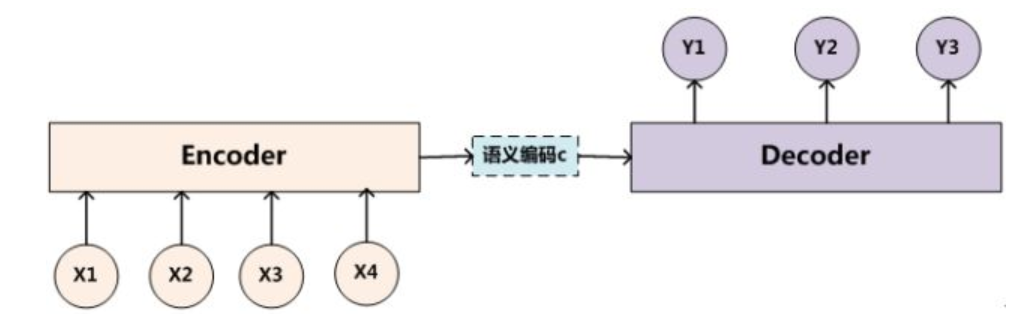
Matthew E. Peters, Mark Neumann, Mohit Iyyer, MattGardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

<https://blog.csdn.net/Magical_Bubble/article/details/89160032>

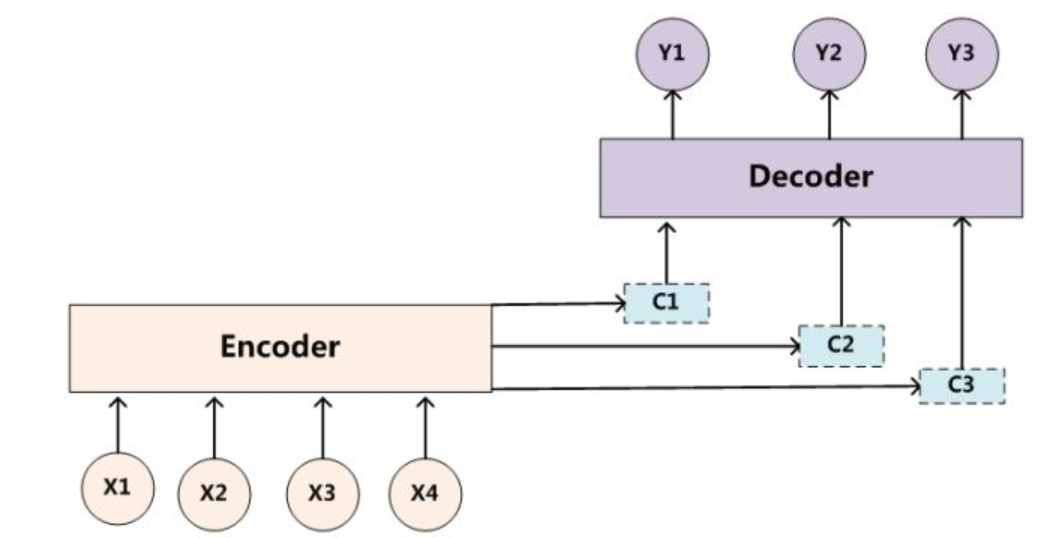
<https://github.com/allenai/bilm-tf/blob/master/bilm/model.py>

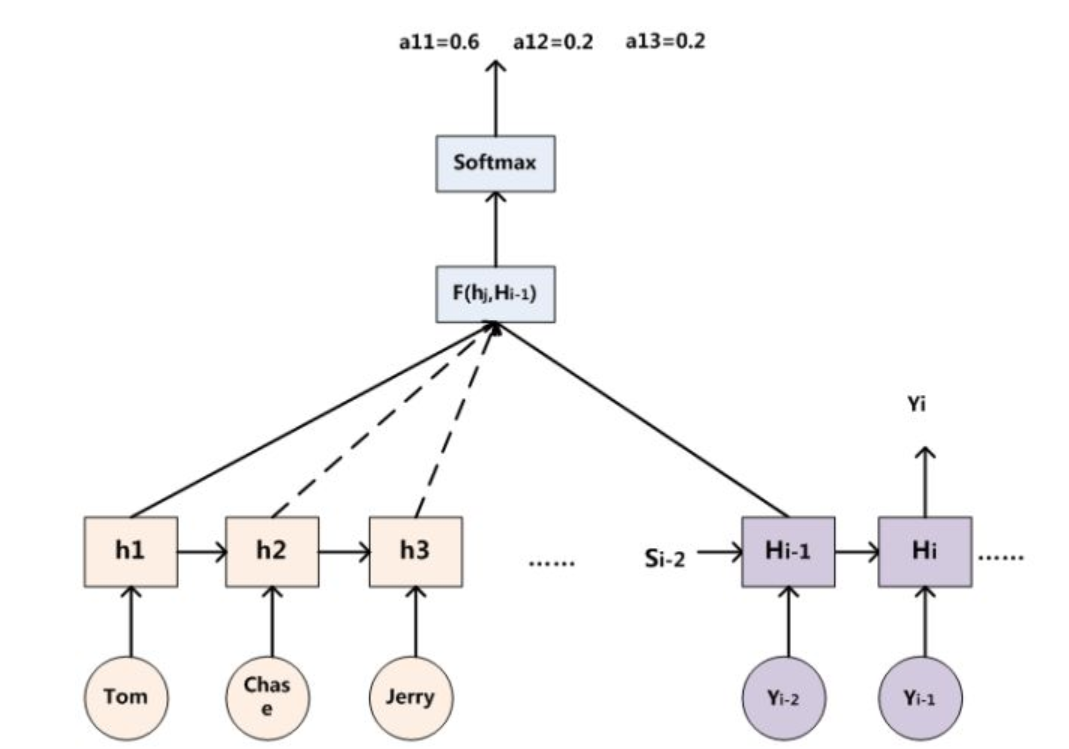
<https://zhuanlan.zhihu.com/p/49271699>

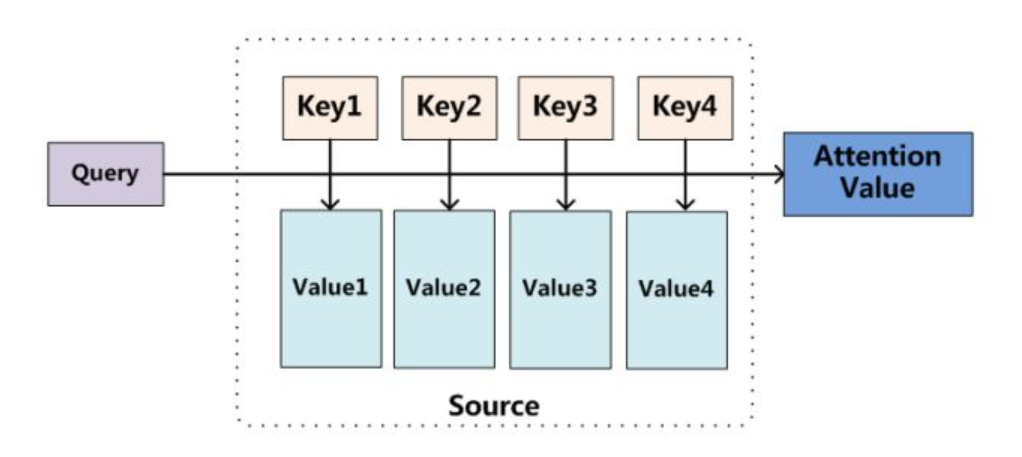
Attention

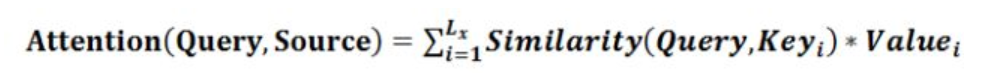


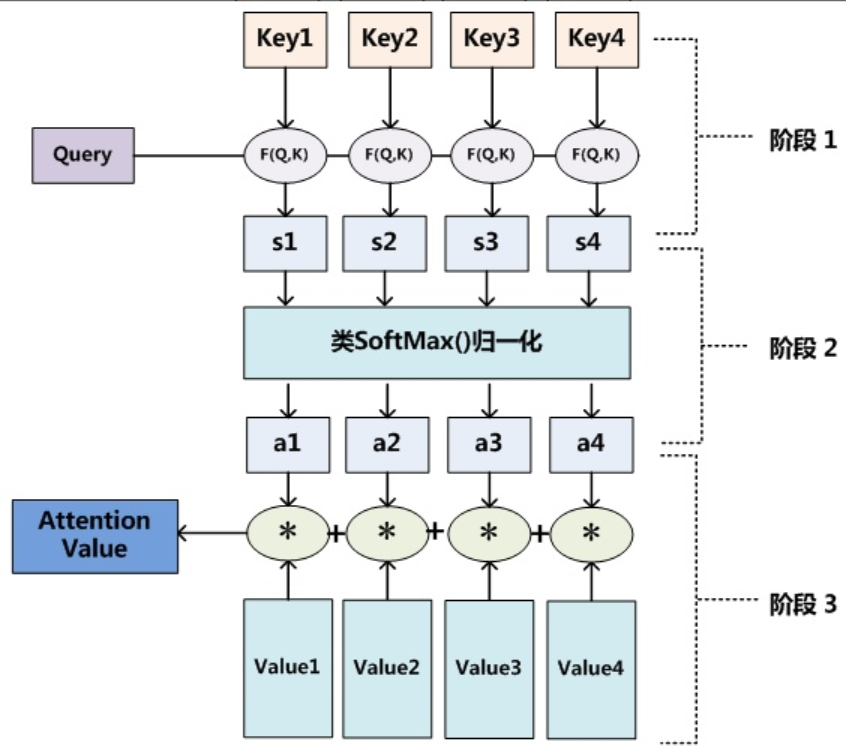
Soft attention:



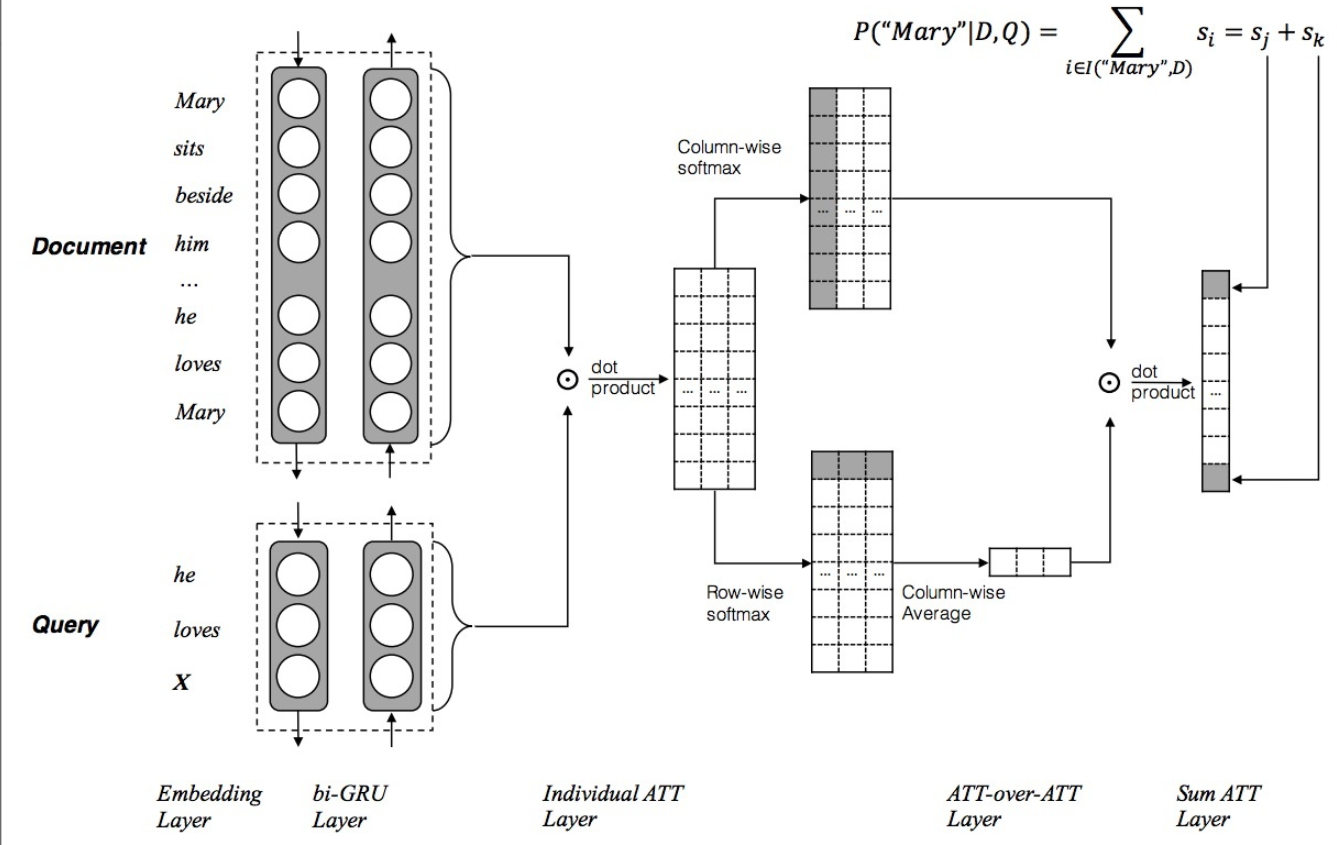






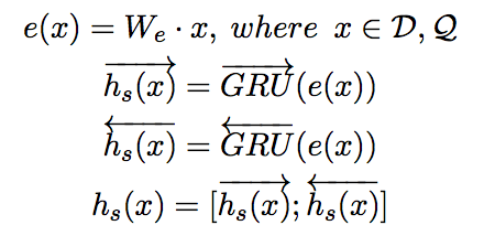


Attention over attention:



Word embedding:

1. Use one-hot method to encode all words in D(document) and Q(query), and then put them into a word embedding layer so that we get e(x).
2. After e(x) going through biLM layer, we could get h\_doc and h\_query.



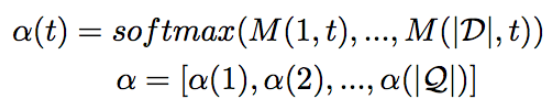
 

Pairwise matching score matrix:



Individual attention:

Column-wise softmax get the correlation of document to query.



Attention over attention:



