Deep Learning - Attention

Jian Li

IIIS, Tsinghua

Attention Mechanism

- Neural processes involving attention: studied in Neuroscience and Computational Neuroscience.
- Visual attention: many animals focus on specific parts of their visual inputs to compute the adequate responses.
- This principle has a large impact on neural computation as we need to select the most pertinent piece of information, rather than using all available information, a large part of it being irrelevant to compute the neural response.

Material from https://blog.heuritech.com/2016/01/20/attention-mechanism/

Image Captioning –

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Previous Model

Previous image captioning RNN models.

The problem with this method is that, when the model is trying to generate the next word of the caption, this word is usually describing only a part of the image.

Using the whole representation of the image h to condition the generation of each word cannot efficiently produce different words for different parts of the image.

This is exactly where an attention mechanism is helpful.

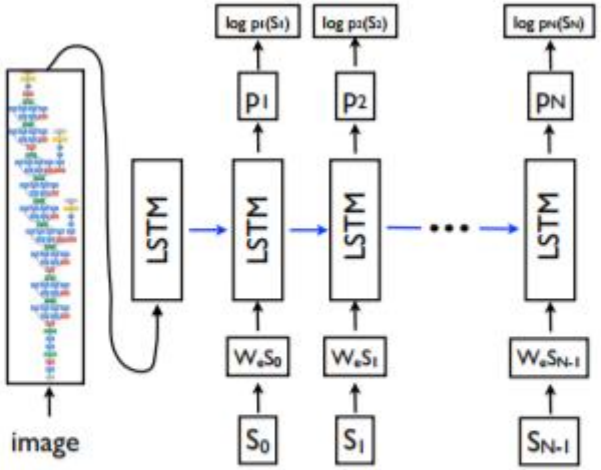


Figure 3. LSTM model combined with a CNN image embedder (as defined in [12]) and word embeddings. The unrolled connections between the LSTM memories are in blue and they correspond to the recurrent connections in Figure 2. All LSTMs share the same parameters.

Examples

Figure 3. Visualization of the attention for each generated word. The rough visualizations obtained by upsampling the attention weights and smoothing. (top)"soft" and (bottom) "hard" attention (note that both models generated the same captions in this example).

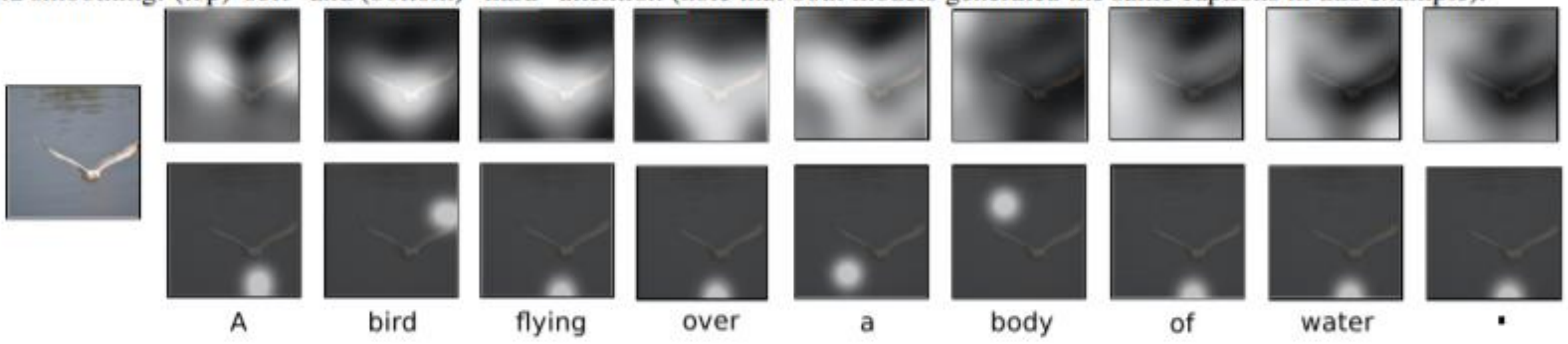
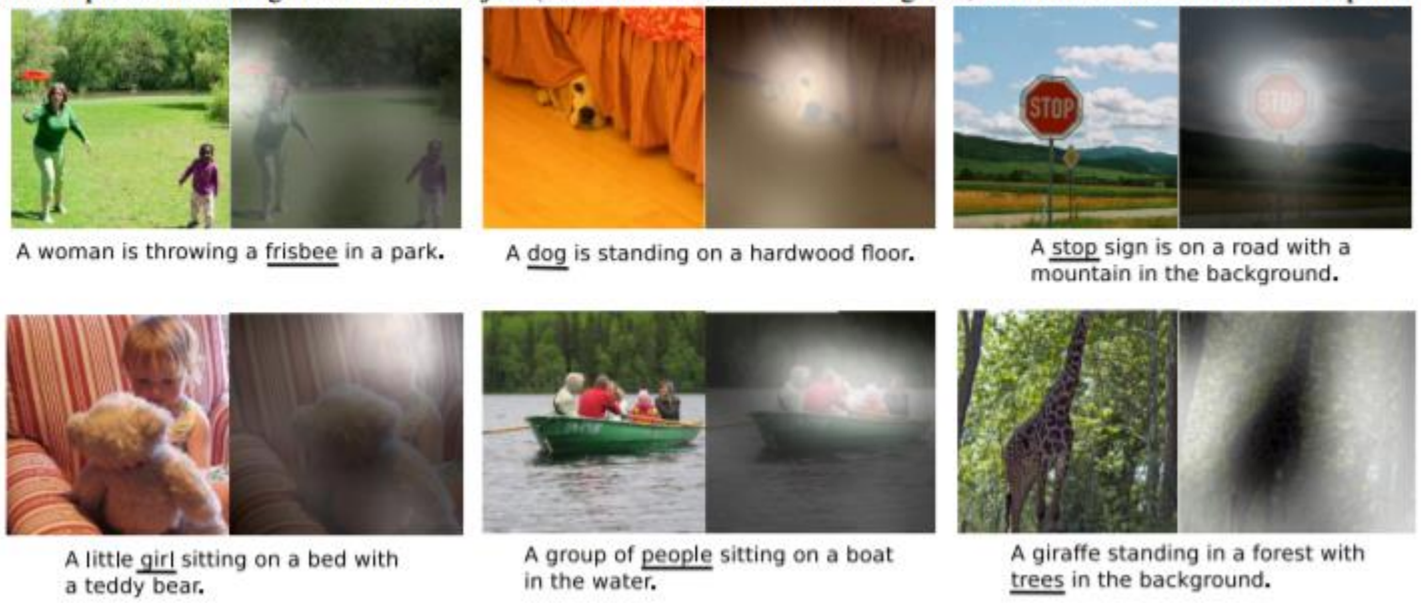


Figure 4. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)



A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

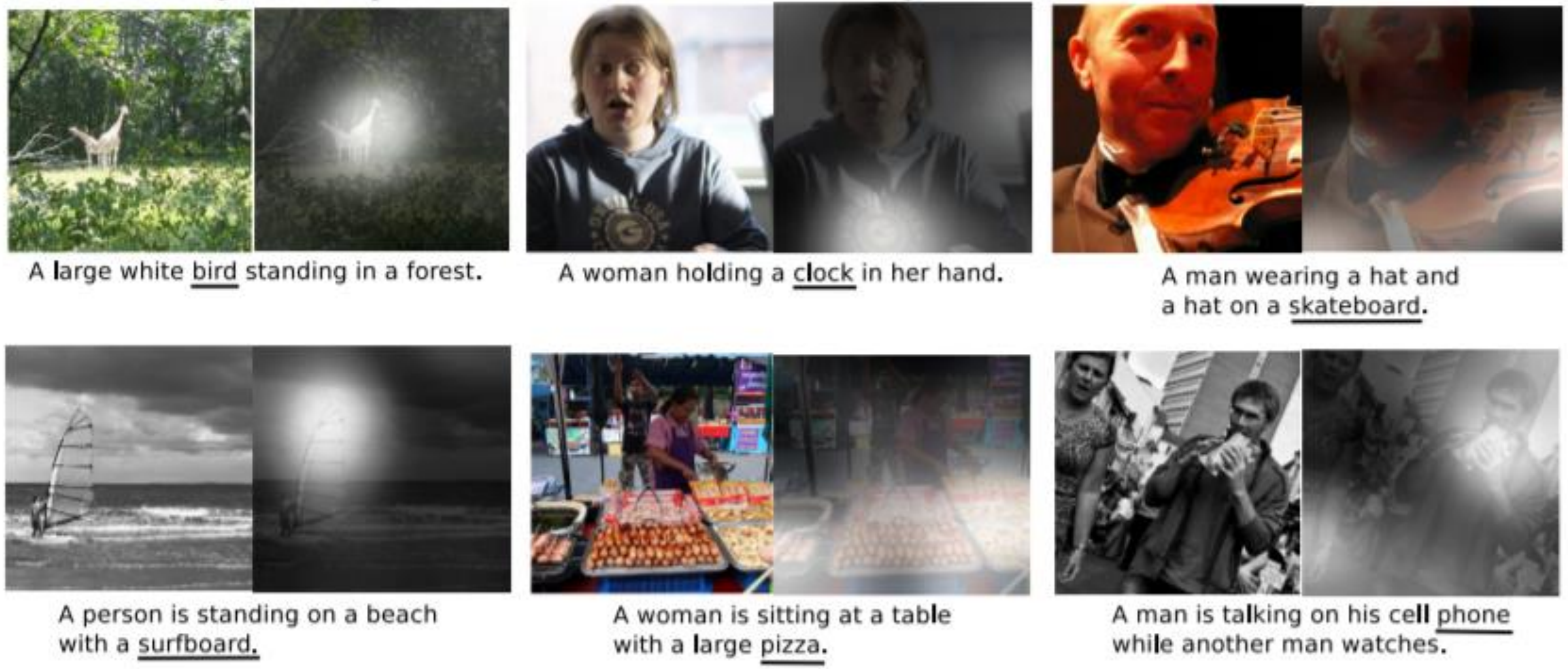
A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Examples

mistakes

Figure 5. Examples of mistakes where we can use attention to gain intuition into what the model saw.



A large white bird standing in a forest.

A woman holding a clock in her hand.

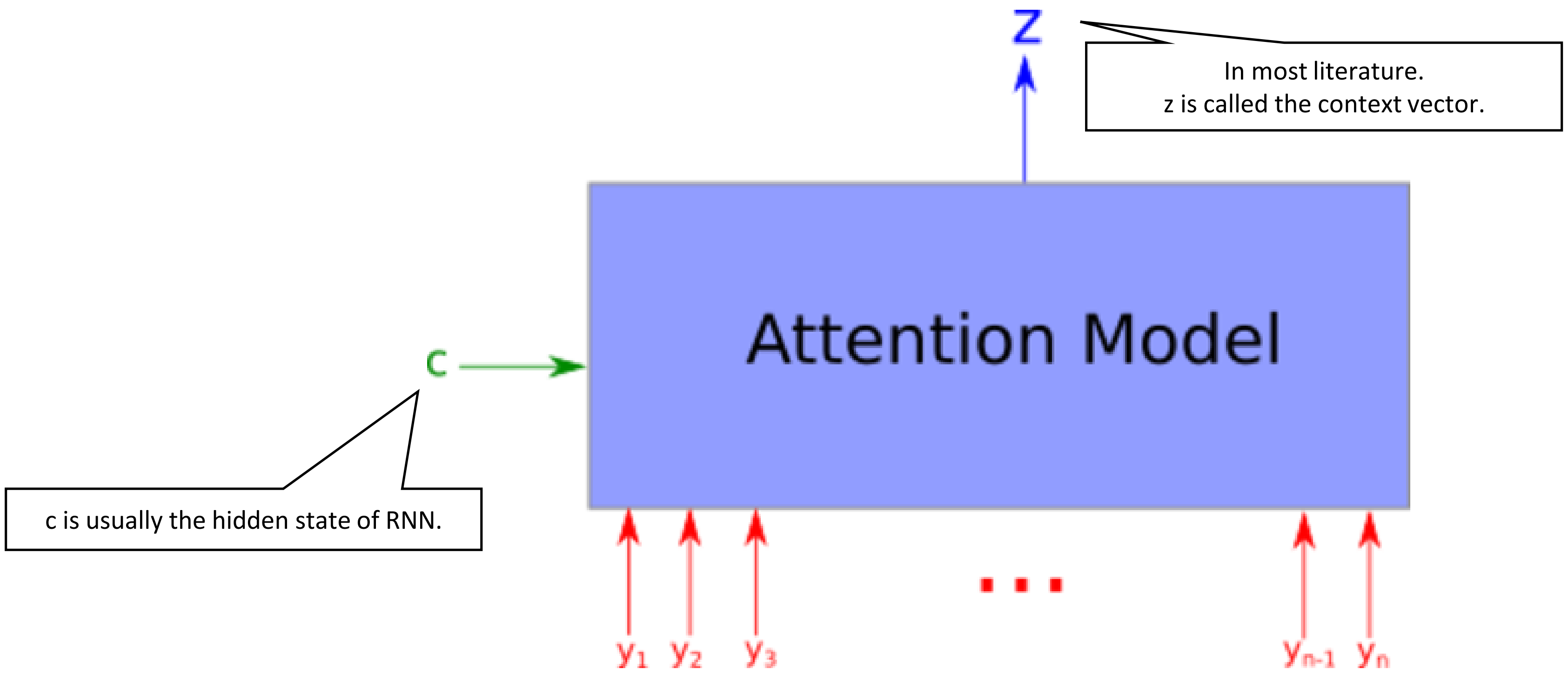
A man wearing a hat and a hat on a skateboard.

A person is standing on a beach with a surfboard.

A woman is sitting at a table with a large pizza.

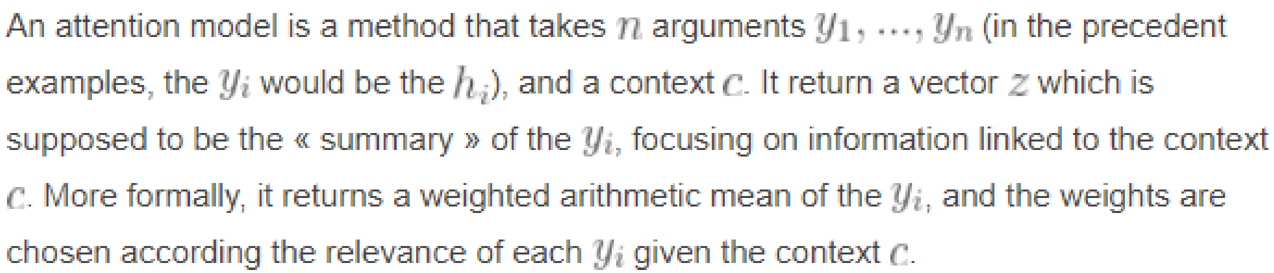
A man is talking on his cell phone while another man watches.

Basic attention mechanism



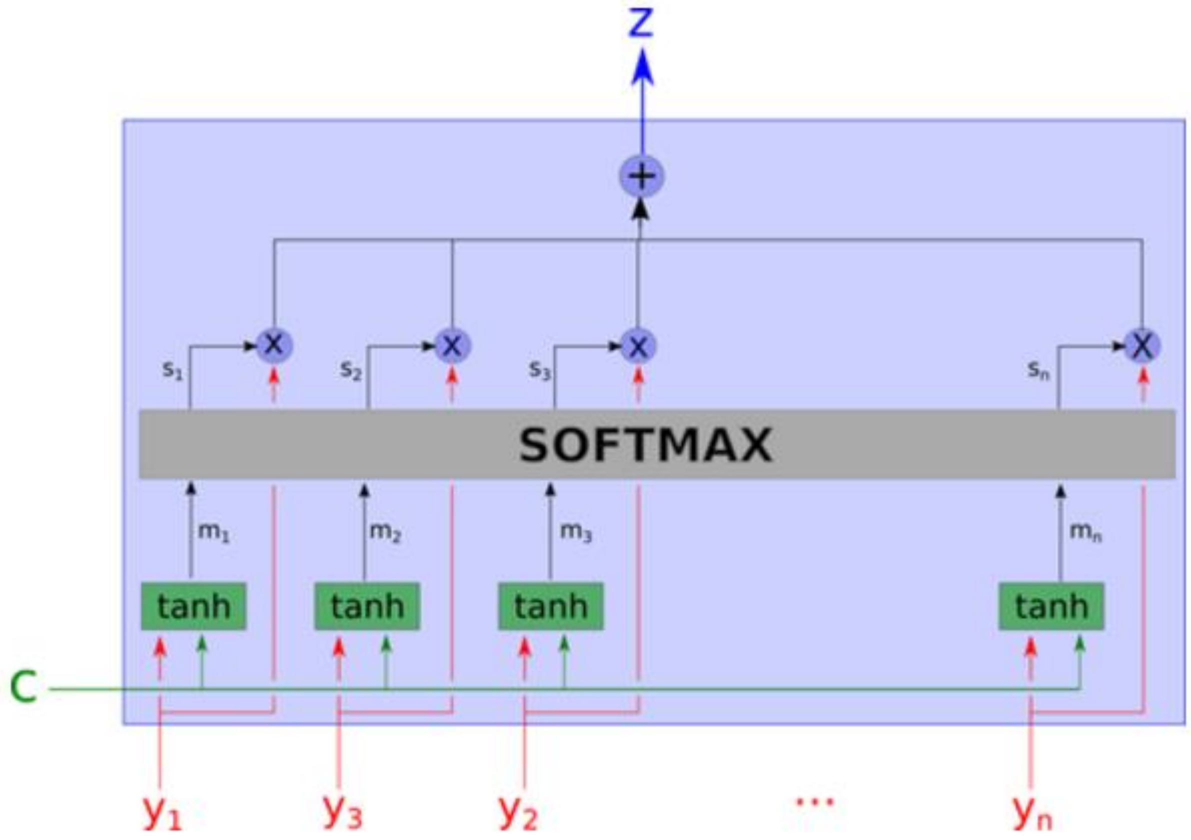
In most literature.
z is called the context vector.

c is usually the hidden state of RNN.

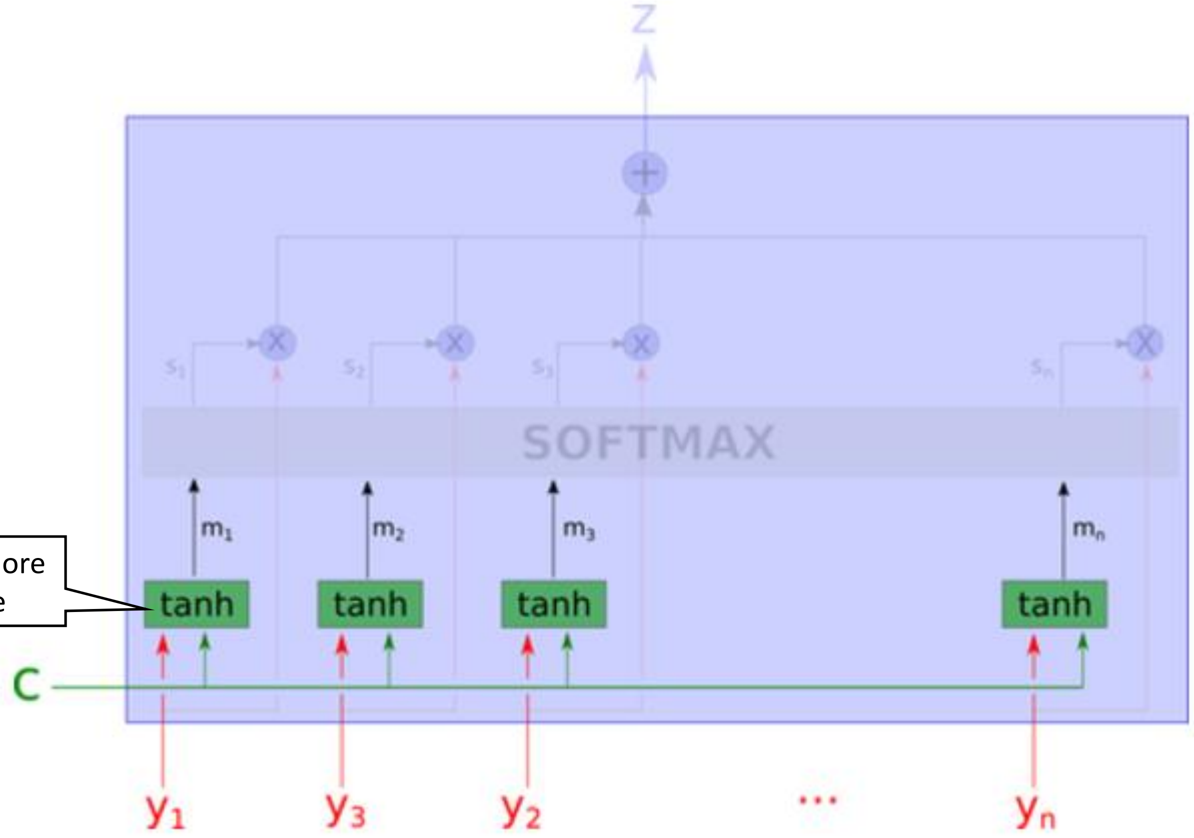


An attention model is a method that takes n arguments $y_1, \ldots, y_n$ (in the precedent examples, the y_i would be the h_i), and a context c. It return a vector z which is supposed to be the « summary » of the y_i, focusing on information linked to the context c. More formally, it returns a weighted arithmetic mean of the y_i, and the weights are chosen according the relevance of each y_i given the context c.

Details


z
+
x
x
x
X
s_1
s_2
s_3
s_n
SOFTMAX
m_1
m_2
m_3
m_n
tanh
tanh
tanh
tanh
C
y_1
y_3
y_2
...
y_n

Details



We can also use a more general MLP here

$$m_i = \tanh\left(W_{cm} c + W_{ym} y_i\right)$$

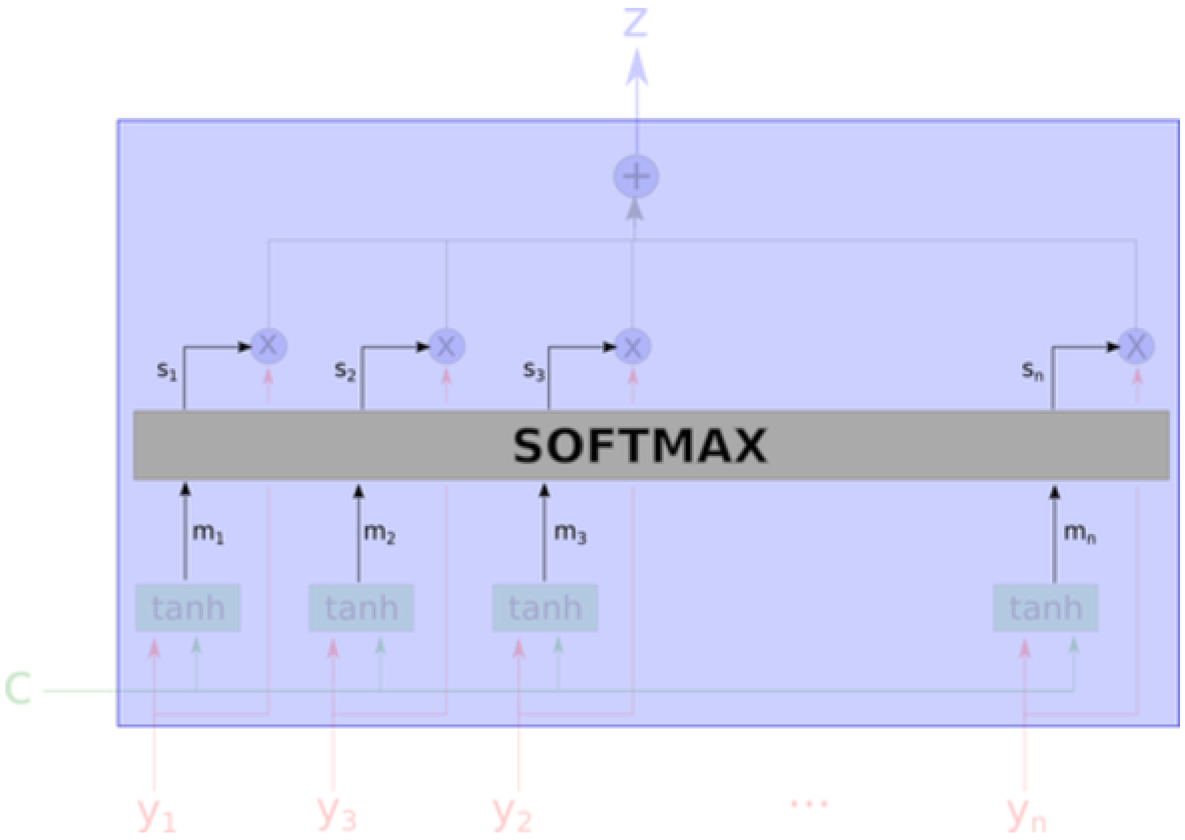
m_i measures the relevance of c and y_i

learnt parameter

learnt parameter

Details

Compute the weight using softmax



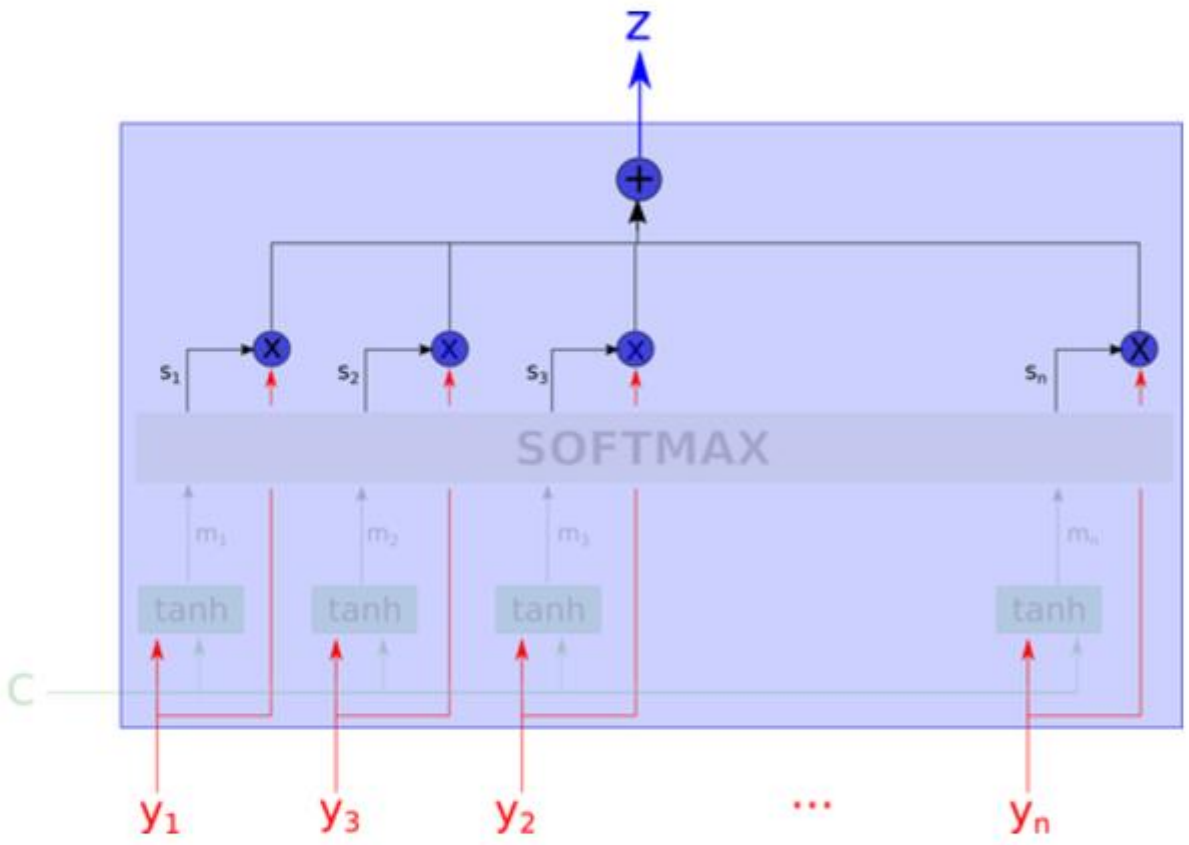
$$s_i \propto \exp\left(\langle w_m, m_i \rangle\right)$$

learnt parameter

$$\sum_i s_i = 1$$

Details

The final output is a convex combination of the inputs yi

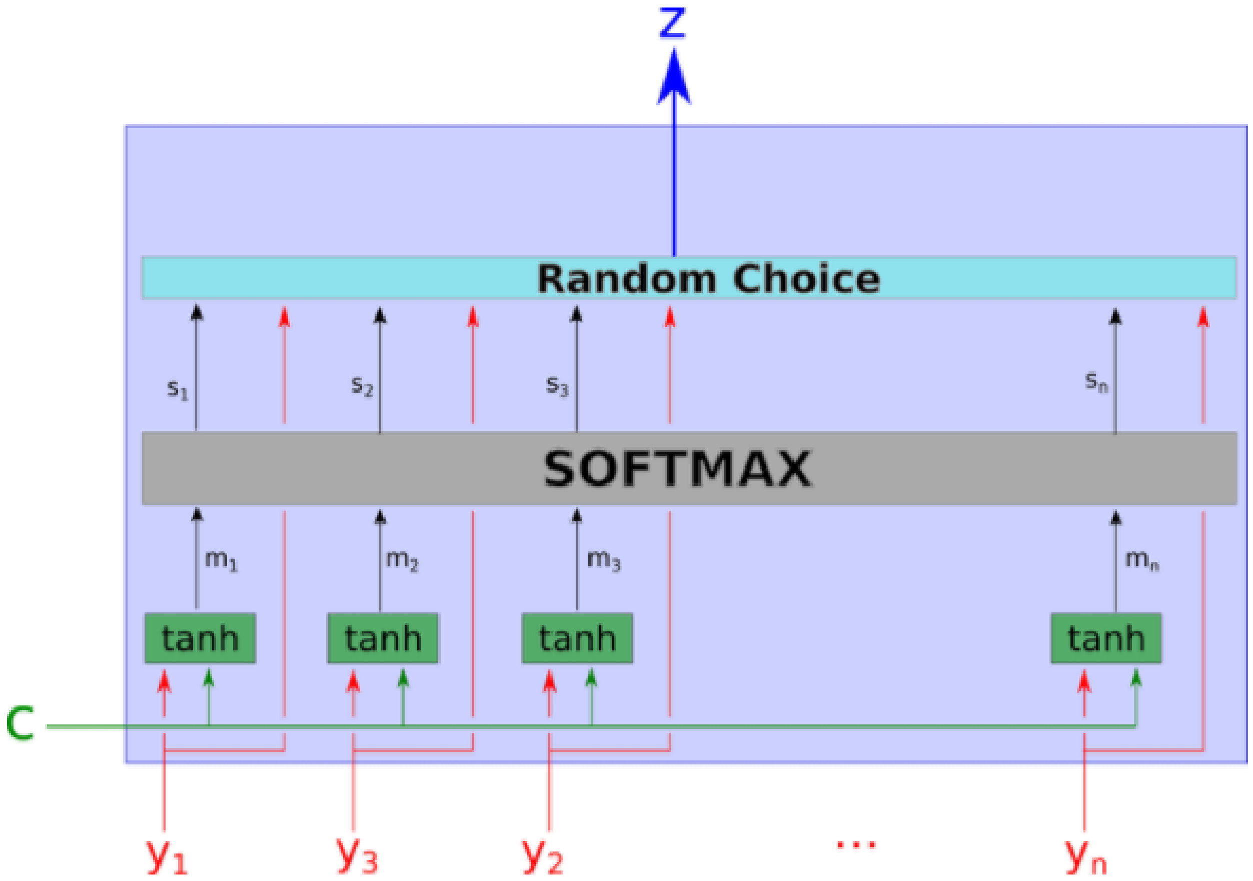


$$z = \sum_i s_i y_i$$

Soft attention. Differentiable. Can be trained end-to-end using BP

Details

- Hard attention (not very popular)

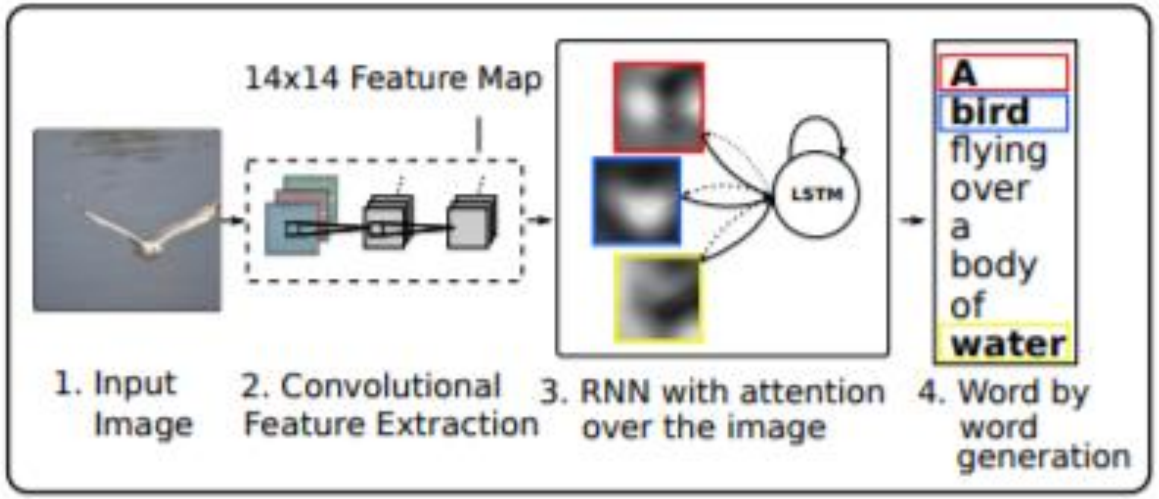


A Hard Attention model. The output is a random choice of one of the y_i, with probability s_i.

The forward computation requires sampling.
So the objective needs to optimize certain log-likelihood.
Omitted here.

Framework

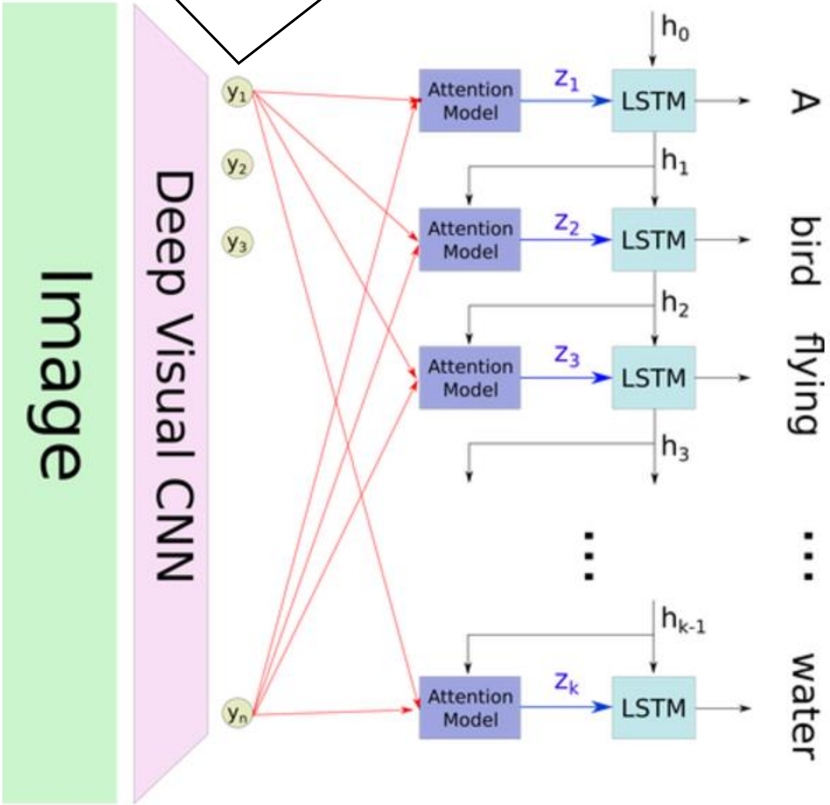
Figure 1. Our model learns a words/image alignment. The visualized attentional maps (3) are explained in Sections 3.1 & 5.4



14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

- LSTM at different time steps attend to different part of the image

Lower convolutional layer. Each y_i (a feature) corresponds to a certain portion of the image



Attention model for image captioning

Metrics

BLEU score: BLEU (bilingual evaluation understudy) is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. (next page)

METEOR: another metric (google by yourself)

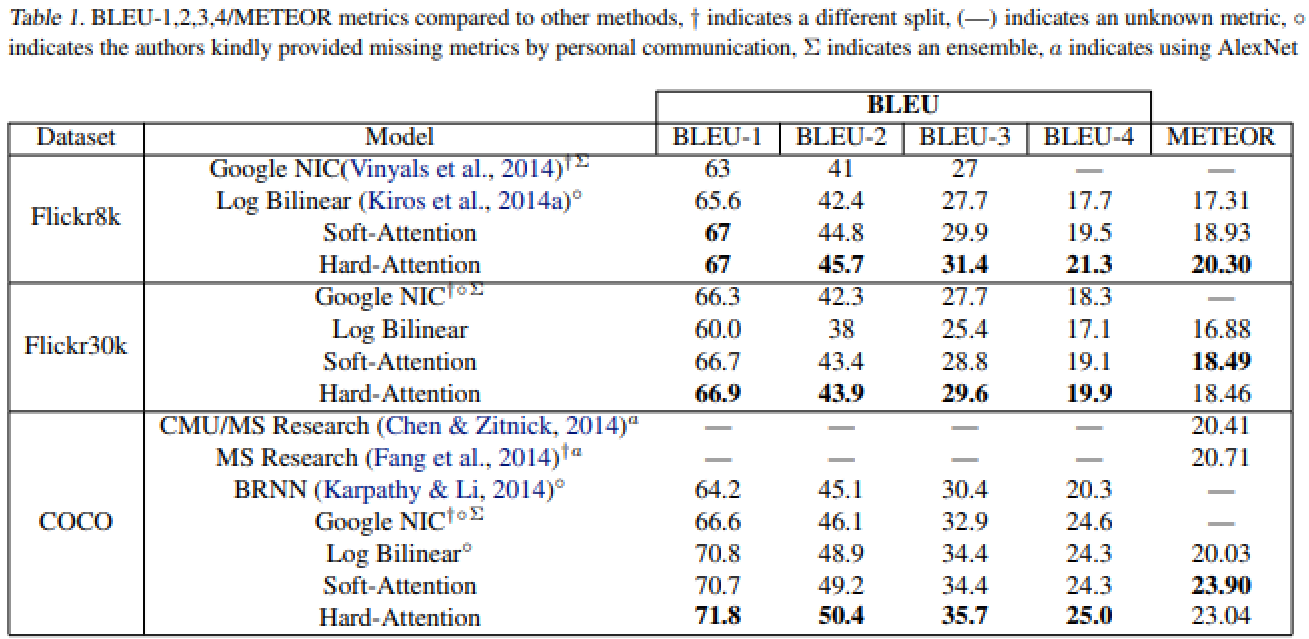


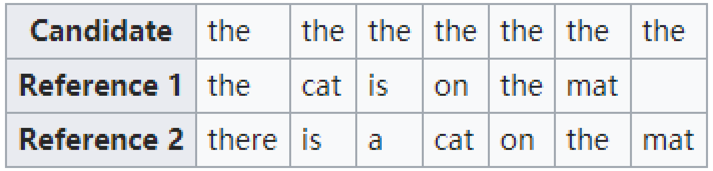
Table 1. BLEU-1,2,3,4/METEOR metrics compared to other methods, † indicates a different split, (—) indicates an unknown metric, ◦ indicates the authors kindly provided missing metrics by personal communication, Σ indicates an ensemble, *a* indicates using AlexNet

		BLEU				
Dataset	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr8k	Google NIC(Vinyals et al., 2014)$^{\dagger\Sigma}$	63	41	27	—	—
	Log Bilinear (Kiros et al., 2014a)$^{\circ}$	65.6	42.4	27.7	17.7	17.31
	Soft-Attention	**67**	44.8	29.9	19.5	18.93
	Hard-Attention	**67**	**45.7**	**31.4**	**21.3**	**20.30**
Flickr30k	Google NIC$^{\dagger\circ\Sigma}$	66.3	42.3	27.7	18.3	—
	Log Bilinear	60.0	38	25.4	17.1	16.88
	Soft-Attention	66.7	43.4	28.8	19.1	**18.49**
	Hard-Attention	**66.9**	**43.9**	**29.6**	**19.9**	18.46
COCO	CMU/MS Research (Chen & Zitnick, 2014)^{a}	—	—	—	—	20.41
	MS Research (Fang et al., 2014)$^{\dagger a}$	—	—	—	—	20.71
	BRNN (Karpathy & Li, 2014)$^{\circ}$	64.2	45.1	30.4	20.3	—
	Google NIC$^{\dagger\circ\Sigma}$	66.6	46.1	32.9	24.6	—
	Log Bilinear$^{\circ}$	70.8	48.9	34.4	24.3	20.03
	Soft-Attention	70.7	49.2	34.4	24.3	**23.90**
	Hard-Attention	**71.8**	**50.4**	**35.7**	**25.0**	23.04

This is an early paper. There are many follow-up papers with better results.

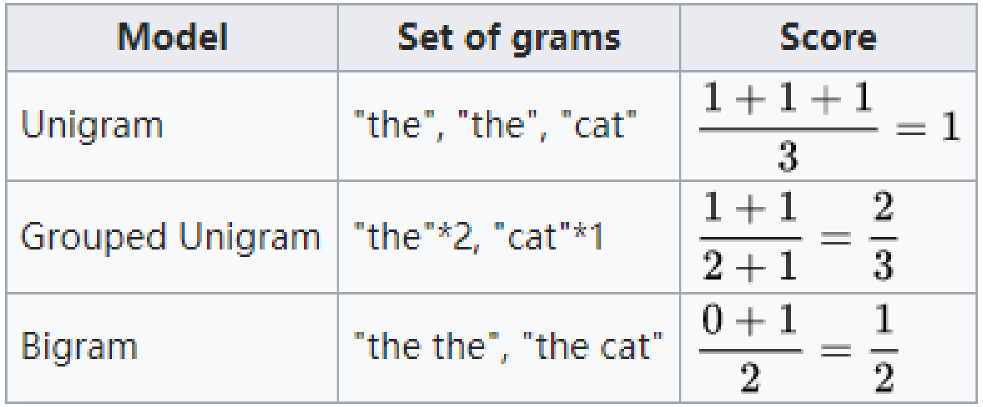
BLEU

Example of poor machine translation output with high precision



Candidate	the	the	the	the	the	the	the
Reference 1	the	cat	is	on	the	mat	
Reference 2	there	is	a	cat	on	the	mat

Comparing metrics for candidate "the the cat"



Model	Set of grams	Score
Unigram	"the", "the", "cat"	$\frac{1+1+1}{3} = 1$
Grouped Unigram	"the"*2, "cat"*1	$\frac{1+1}{2+1} = \frac{2}{3}$
Bigram	"the the", "the cat"	$\frac{0+1}{2} = \frac{1}{2}$

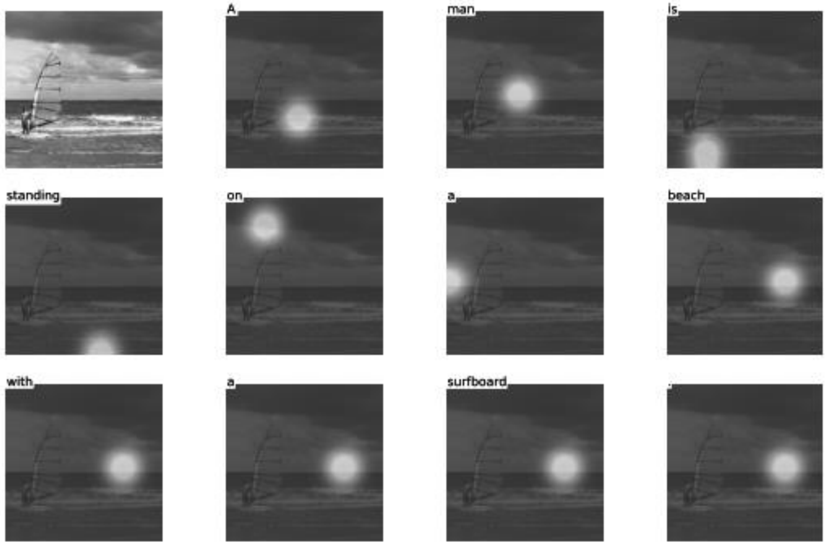
For each word in the candidate translation, the algorithm takes its maximum total count, m_{max} in any of the reference translations. In the example above, the word "the" appears twice in reference 1, and once in reference 2. m_{max} = 2.

M=#words in the candidate that also appear in the reference (7 in the example)
W=total number of words in the candidate translation (7 in the example)

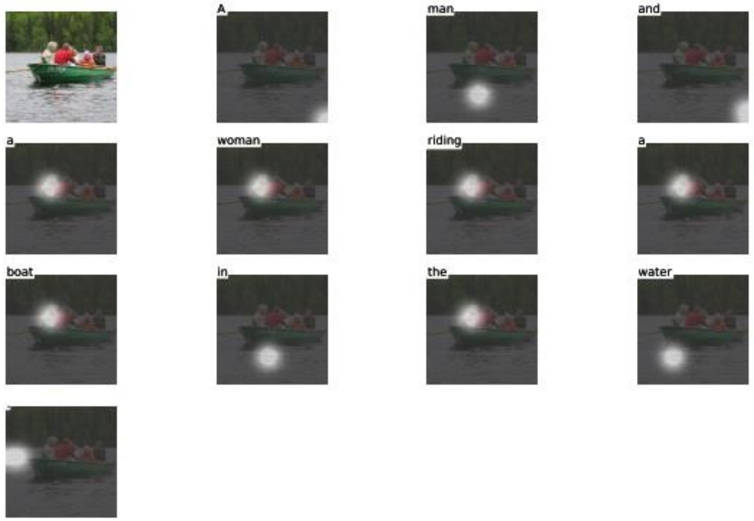
BLEU= min(m_max, M)/W   (2/7 in the example)

We also look at 2-grams (BLEU-2), 3-grams (BLEU-2), and so on.

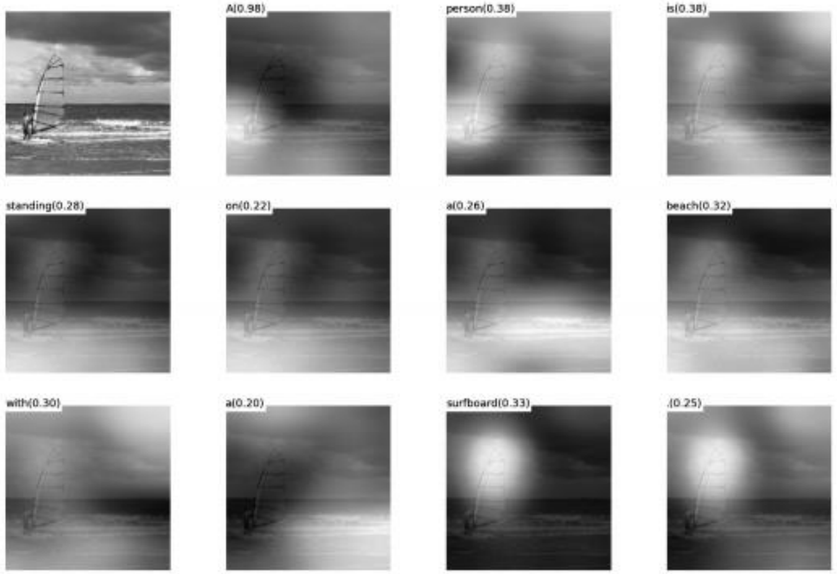
BLEU score favors short translation and thus has been criticized (but still the most popular metric) “hard” (a) and “soft” (b) attention model. White indicates the regions where the model roughly attends to



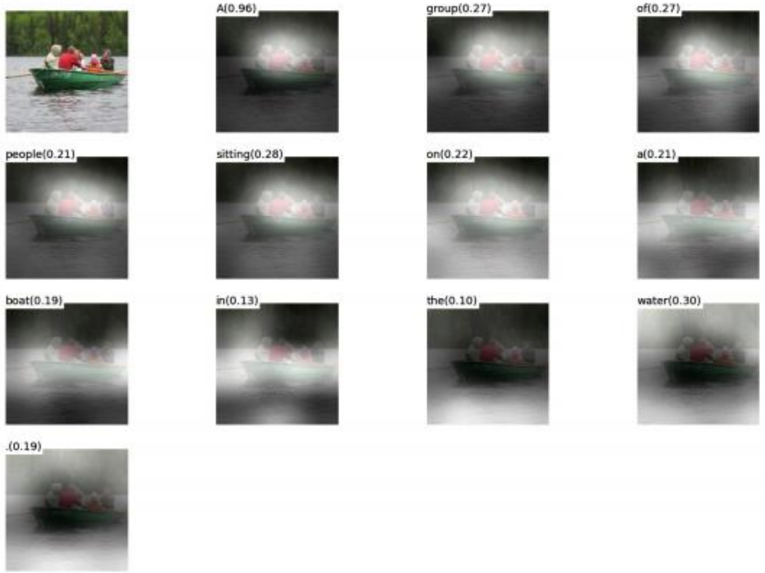
(a) A man is standing on a beach with a surfboard.



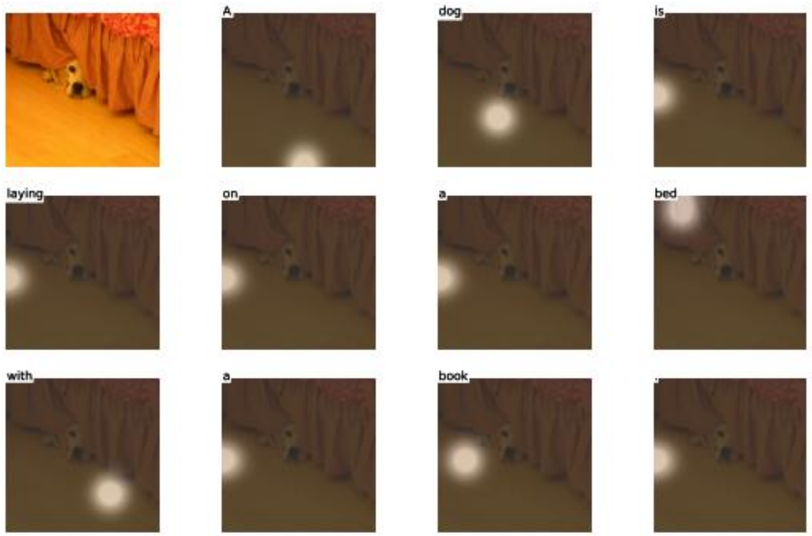
(a) A man and a woman riding a boat in the water.



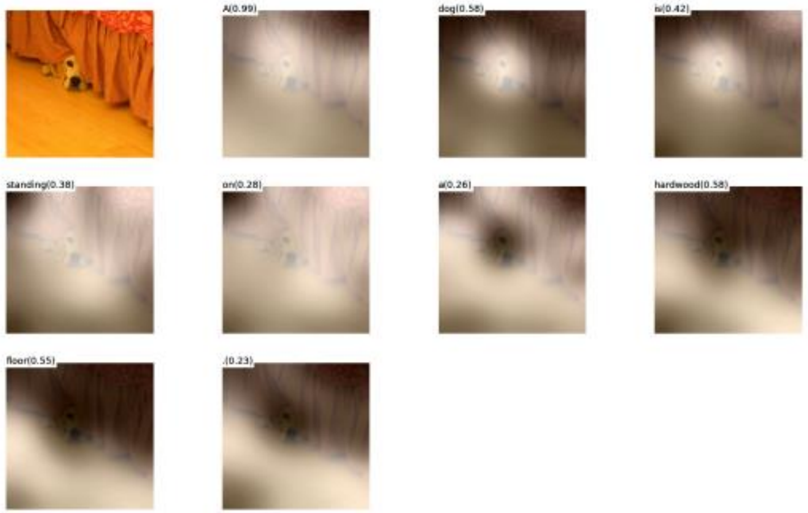
(b) A person is standing on a beach with a surfboard.



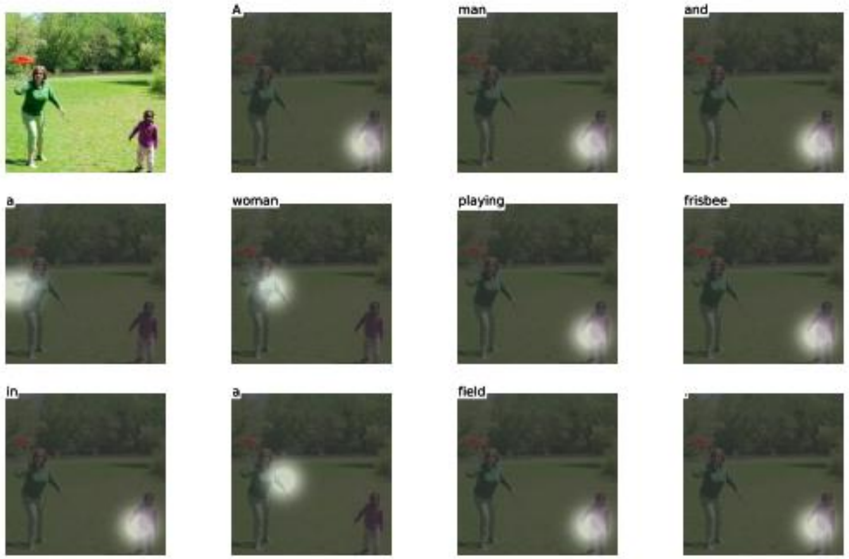
(b) A group of people sitting on a boat in the water.



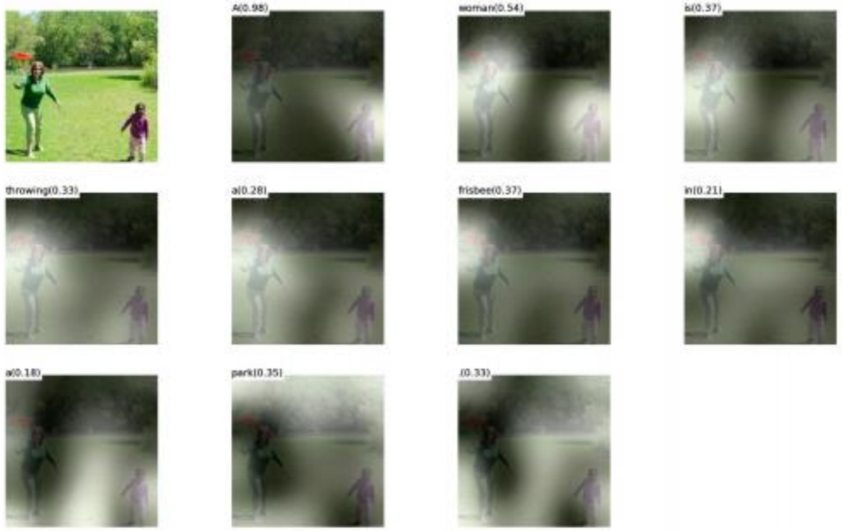
(a) A dog is laying on a bed with a book.



(b) A dog is standing on a hardwood floor.



(a) A man and a woman playing frisbee in a field.

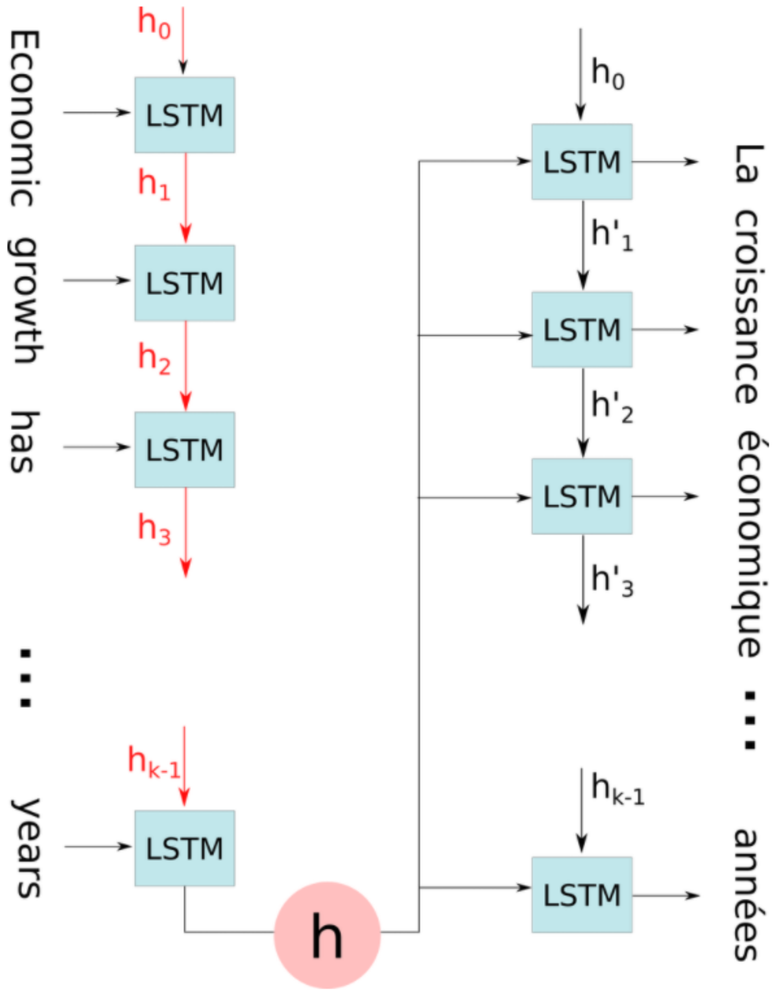


(b) A woman is throwing a frisbee in a park.

Attention in sequence-to-sequence e.g., machine translation

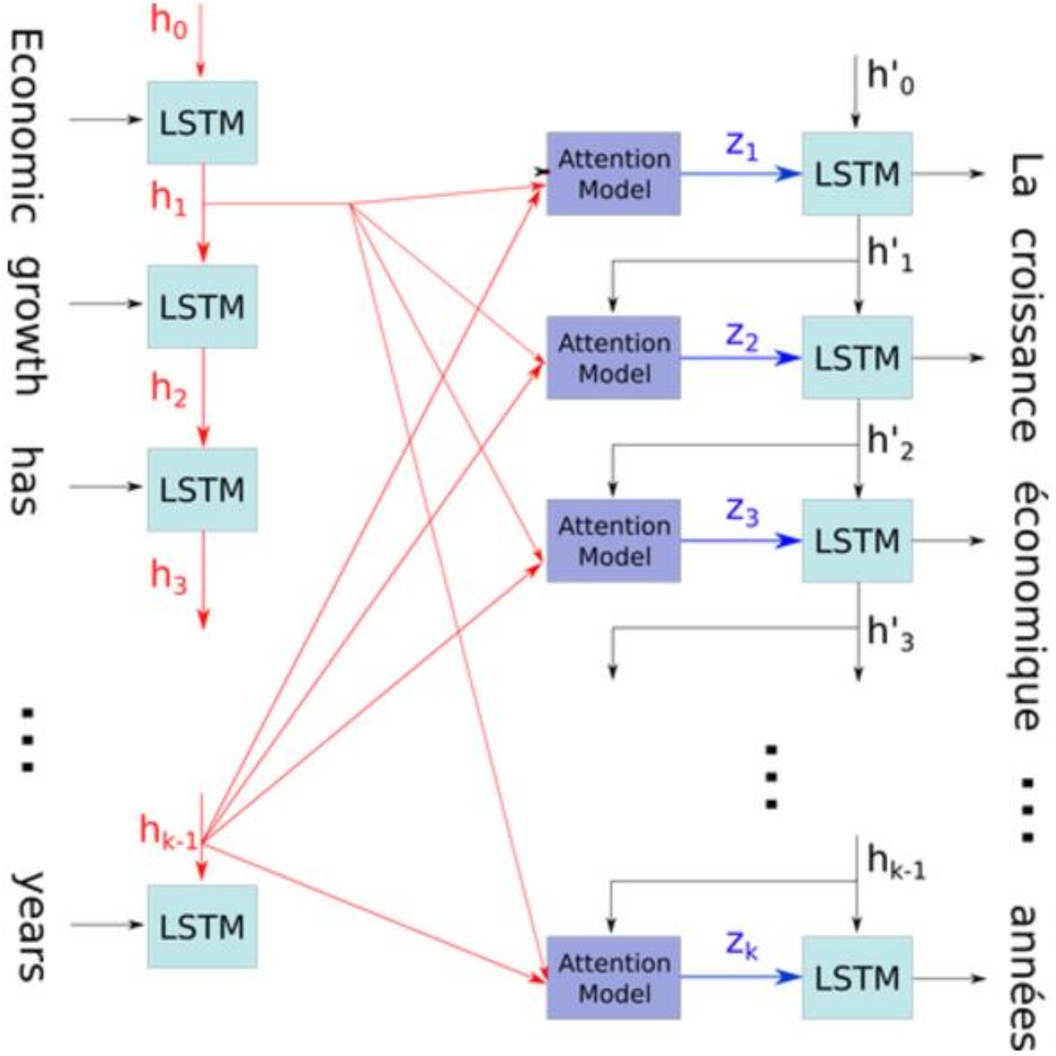
Neural machine translation by jointly learning to align and translate

An MT model without attention

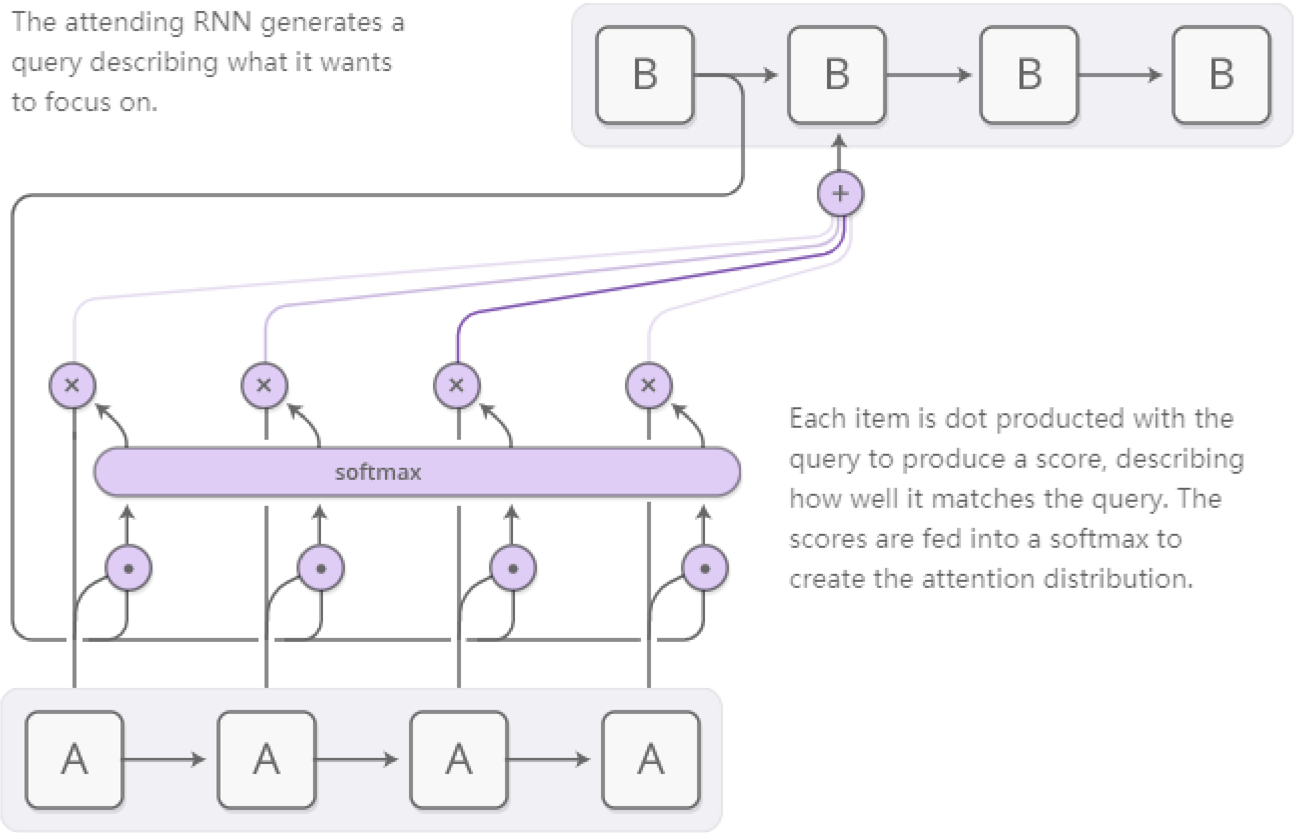

Economic growth has ... years
h0
h1
h2
h3
hk-1
LSTM
h
h0
h'1
h'2
h'3
hk-1
La croissance économique ... années

A model for translation without attention.

With attention



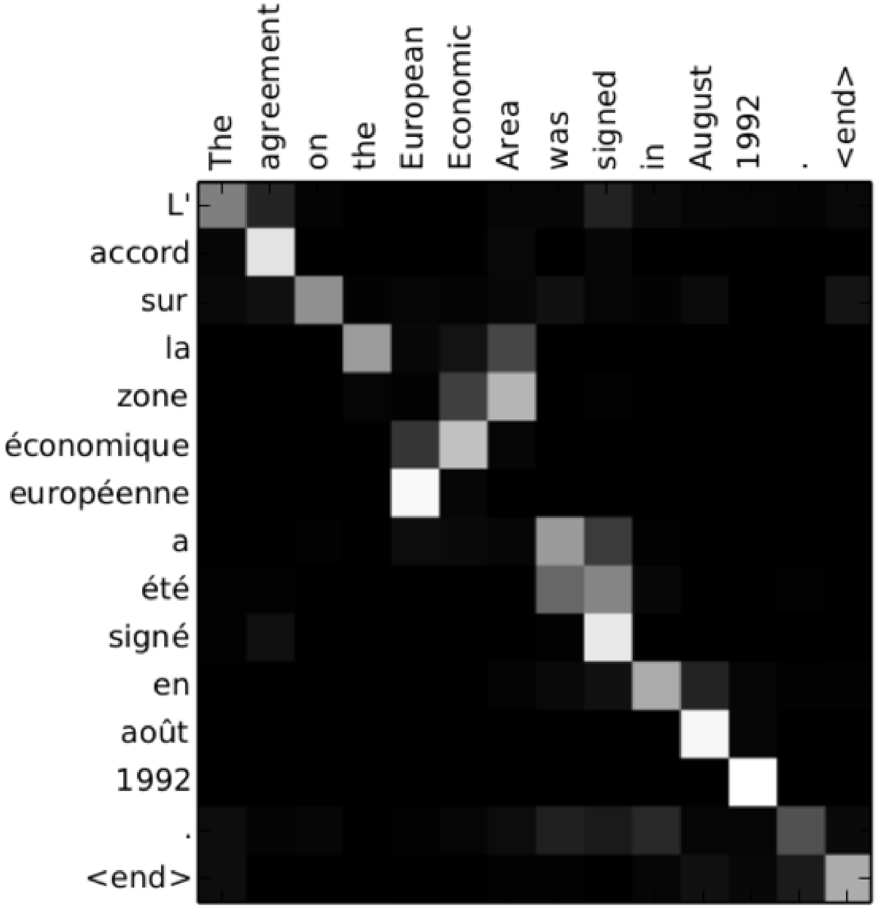
Attention model for Translation.



The attending RNN generates a query describing what it wants to focus on.

Each item is dot producted with the query to produce a score, describing how well it matches the query. The scores are fed into a softmax to create the attention distribution.

Other animations:
https://distill.pub/2016/augmented-rnns/#attentional-interfaces

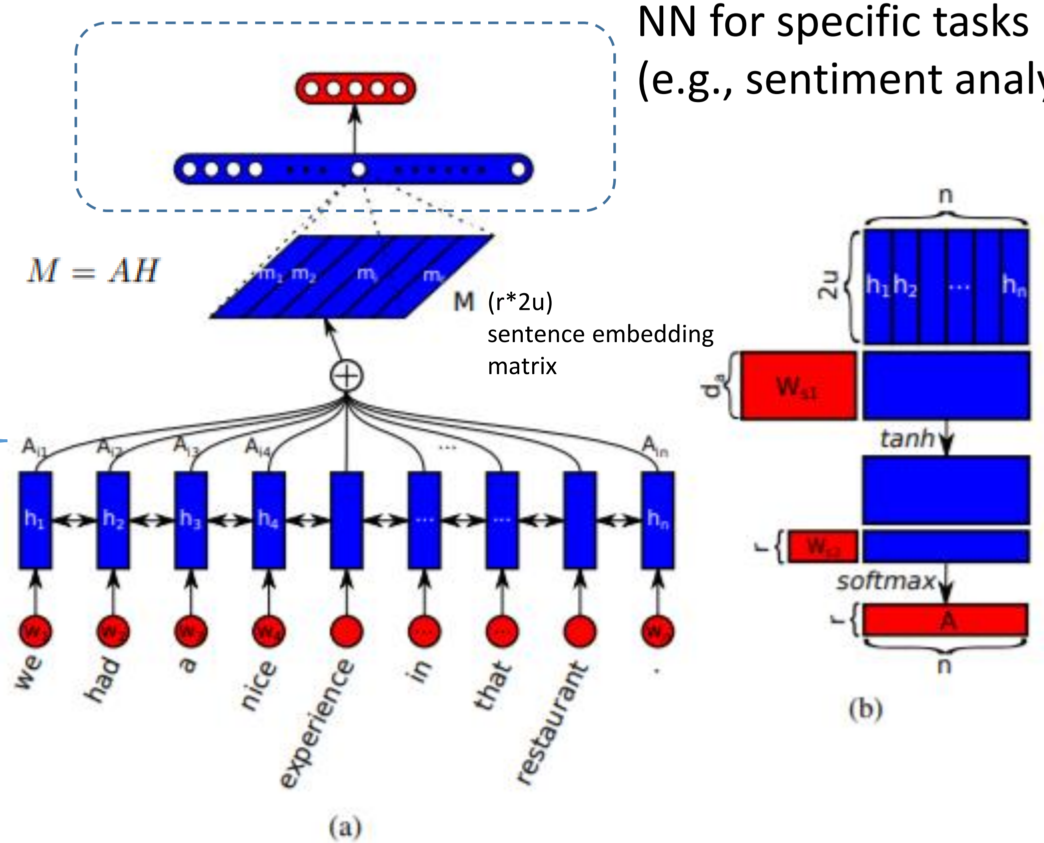

The
agreement
on
the
European
Economic
Area
was
signed
in
August
1992
.
<end>
L'
accord
sur
la
zone
économique
européenne
a
été
signé
en
août
1992
.
<end>

Self-Attention

A structured self-attentive sentence embedding

A STRUCTURED SELF-ATTENTIVE SENTENCE EMBEDDING

NN for specific tasks
(e.g., sentiment analysis)



M (r*2u)
sentence embedding
matrix

Summation weight

Bi-LSTM

$S = (\mathbf{w_1}, \mathbf{w_2}, \cdots \mathbf{w_n})$

w_i word embedding
(trainable)

Computing the summation weights based on h_i

$$\mathbf{a} = softmax\left(\mathbf{w_{s2}} tanh\left(W_{s1} H^T\right)\right)$$

Multi-hop attention:
(attent to several parts of a sentence)

$$A = softmax\left(W_{s2} tanh\left(W_{s1} H^T\right)\right)$$

Figure 1: A sample model structure showing the sentence embedding model combined with a fully connected and softmax layer for sentiment analysis (a). The sentence embedding M is computed as multiple weighted sums of hidden states from a bidirectional LSTM ($\mathbf{h_1}$, ..., $\mathbf{h_n}$), where the summation weights (A_{i1}, ..., A_{in}) are computed in a way illustrated in (b). Blue colored shapes stand for hidden representations, and red colored shapes stand for weights, annotations, or input/output.

A STRUCTURED SELF-ATTENTIVE SENTENCE EMBEDDING

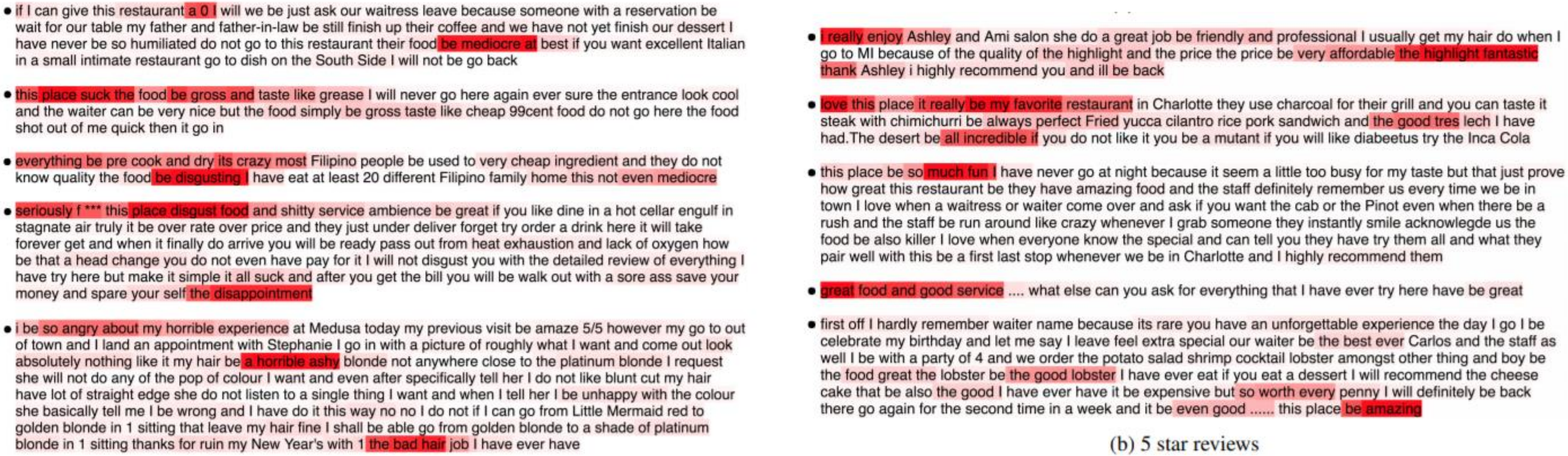
- RM The embedding matrix M can suffer from redundancy problems if the attention mechanism always provides similar summation weights for all the r hops

- A penalization term: $P = \left\|(AA^T - I)\right\|_F{}^2$

  Want A to have orthonormal rows

- This term will be added to the final objective (with a multiplicative factor)
  - E.g., suppose we want to do sentiment analysis (yelp dataset. Yelp reviews, labels: stars 1-5; so it is a classification task. The main objective is the cross entropy.)

Visualization



- if I can give this restaurant a 0 I will we be just ask our waitress leave because someone with a reservation be wait for our table my father and father-in-law be still finish up their coffee and we have not yet finish our dessert I have never be so humiliated do not go to this restaurant their food be mediocre at best if you want excellent Italian in a small intimate restaurant go to dish on the South Side I will not be go back
- this place suck the food be gross and taste like grease I will never go here again ever sure the entrance look cool and the waiter can be very nice but the food simply be gross taste like cheap 99cent food do not go here the food shot out of me quick then it go in
- everything be pre cook and dry its crazy most Filipino people be used to very cheap ingredient and they do not know quality the food be disgusting I have eat at least 20 different Filipino family home this not even mediocre
- seriously f *** this place disgust food and shitty service ambience be great if you like dine in a hot cellar engulf in stagnate air truly it be over rate over price and they just under deliver forget try order a drink here it will take forever get and when it finally do arrive you will be ready pass out from heat exhaustion and lack of oxygen how be that a head change you do not even have pay for it I will not disgust you with the detailed review of everything I have try here but make it simple it all suck and after you get the bill you will be walk out with a sore ass save your money and spare your self the disappointment
- i be so angry about my horrible experience at Medusa today my previous visit be amaze 5/5 however my go to out of town and I land an appointment with Stephanie I go in with a picture of roughly what I want and come out look absolutely nothing like it my hair be a horrible ashy blonde not anywhere close to the platinum blonde I request she will not do any of the pop of colour I want and even after specifically tell her I do not like blunt cut my hair have lot of straight edge she do not listen to a single thing I want and when I tell her I be unhappy with the colour she basically tell me I be wrong and I have do it this way no no I do not if I can go from Little Mermaid red to golden blonde in 1 sitting that leave my hair fine I shall be able go from golden blonde to a shade of platinum blonde in 1 sitting thanks for ruin my New Year's with 1 the bad hair job I have ever have

(a) 1 star reviews

- I really enjoy Ashley and Ami salon she do a great job be friendly and professional I usually get my hair do when I go to MI because of the quality of the highlight and the price the price be very affordable the highlight fantastic thank Ashley i highly recommend you and ill be back
- love this place it really be my favorite restaurant in Charlotte they use charcoal for their grill and you can taste it steak with chimichurri be always perfect Fried yucca cilantro rice pork sandwich and the good tres lech I have had.The desert be all incredible if you do not like it you be a mutant if you will like diabeetus try the Inca Cola
- this place be so much fun I have never go at night because it seem a little too busy for my taste but that just prove how great this restaurant be they have amazing food and the staff definitely remember us every time we be in town I love when a waitress or waiter come over and ask if you want the cab or the Pinot even when there be a rush and the staff be run around like crazy whenever I grab someone they instantly smile acknowlegde us the food be also killer I love when everyone know the special and can tell you they have try them all and what they pair well with this be a first last stop whenever we be in Charlotte and I highly recommend them
- great food and good service .... what else can you ask for everything that I have ever try here have be great
- first off I hardly remember waiter name because its rare you have an unforgettable experience the day I go I be celebrate my birthday and let me say I leave feel extra special our waiter be the best ever Carlos and the staff as well I be with a party of 4 and we order the potato salad shrimp cocktail lobster amongst other thing and boy be the food great the lobster be the good lobster I have ever eat if you eat a dessert I will recommend the cheese cake that be also the good I have ever have it be expensive but so worth every penny I will definitely be back there go again for the second time in a week and it be even good ...... this place be amazing

(b) 5 star reviews

Text Entailment

- SNLI corpus
- Classification task:

Given many Sentence pairs

Classified as

1. entailment
2. contradiction
3. neutral

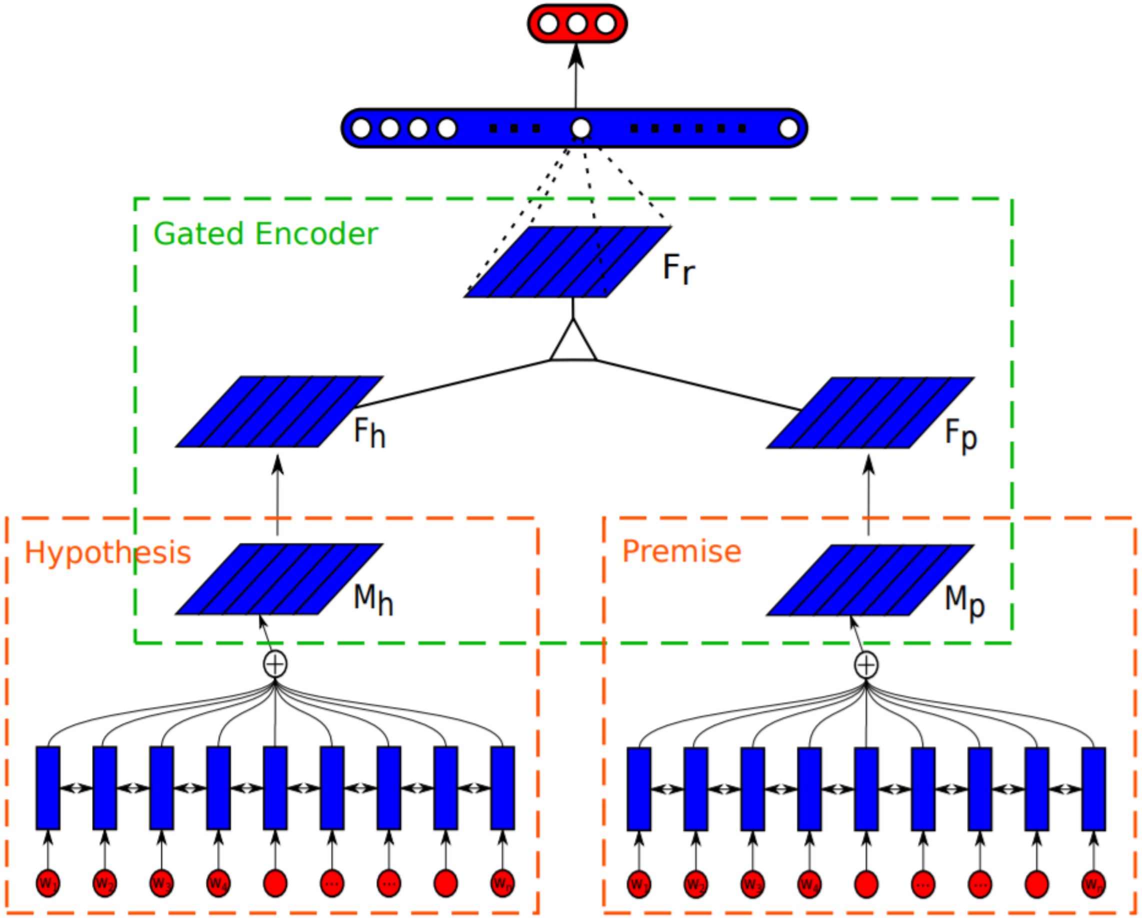


Figure 7: Model structure used for textual entailment task.

Text Entailment

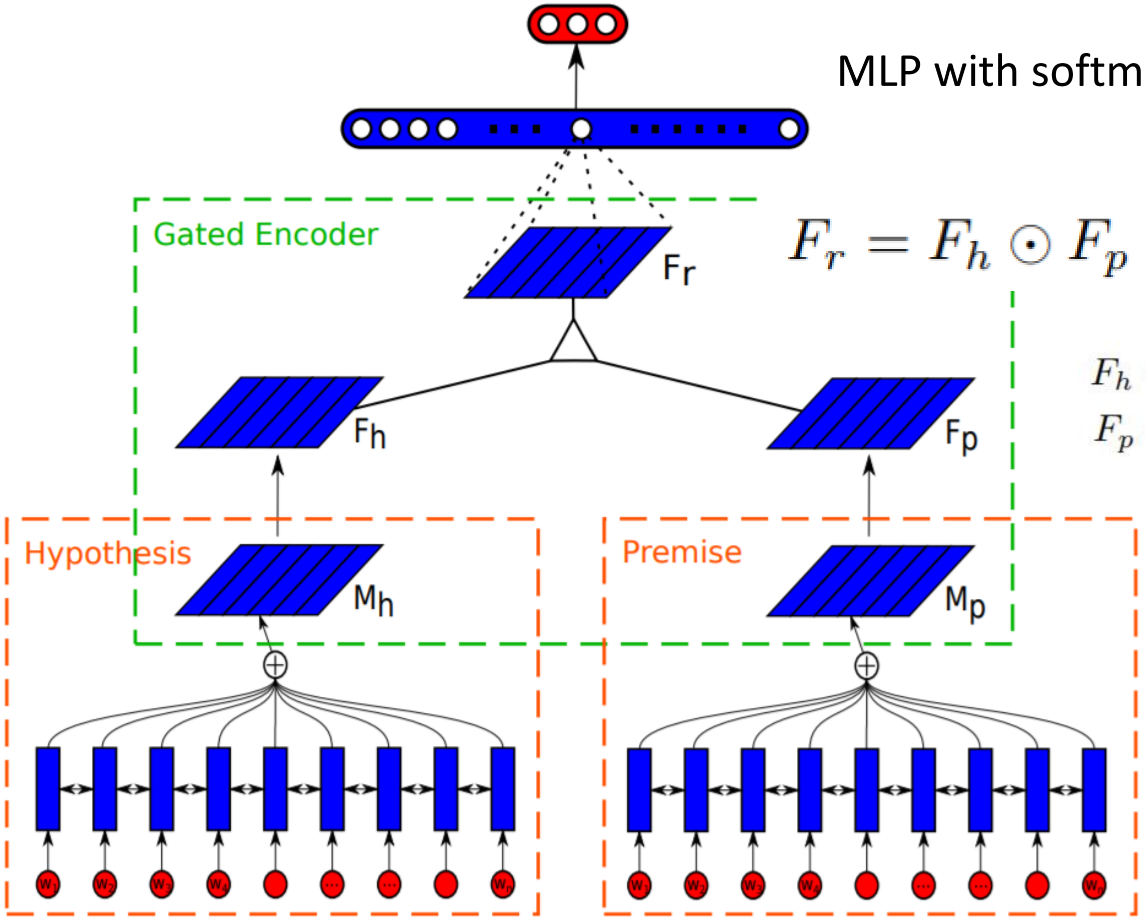


Figure 7: Model structure used for textual entailment task.

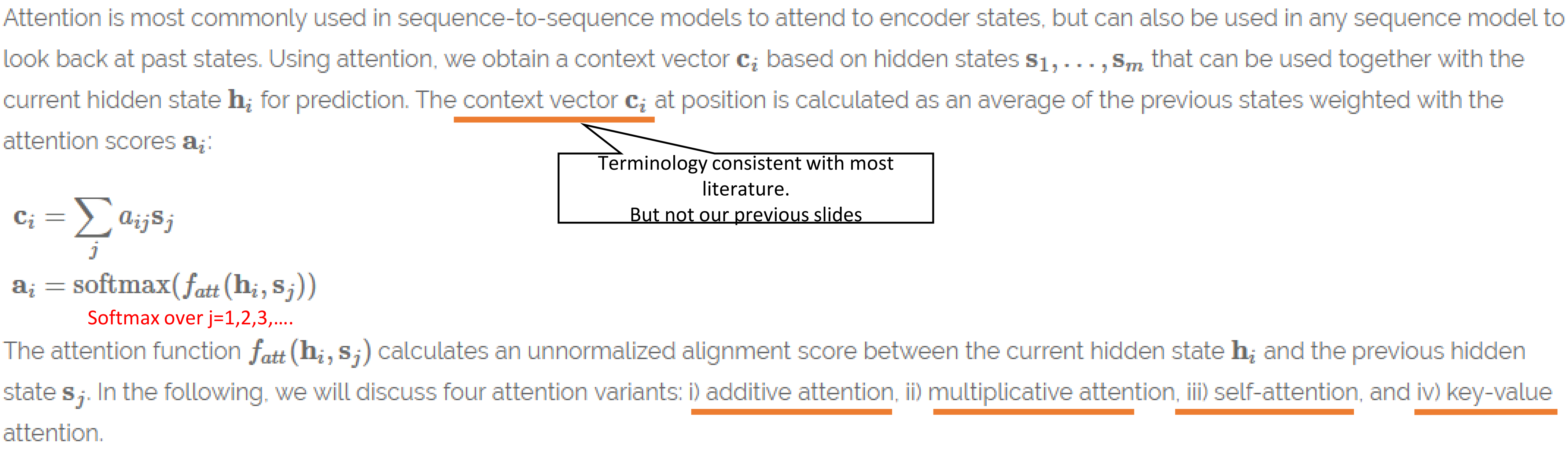
MLP with softmax (3 classes)

$F_r = F_h \odot F_p$ element-wise product

$$F_h = batcheddot(M_h, W_{fh})$$
$$F_p = batcheddot(M_p, W_{fp})$$

First we multiply each row in the matrix embedding by a different weight matrix. Repeating it over all rows, corresponds to a batched dot product between a 2-D matrix and a 3-D weight tensor

Summary



Attention is most commonly used in sequence-to-sequence models to attend to encoder states, but can also be used in any sequence model to look back at past states. Using attention, we obtain a context vector $\mathbf{c}_i$ based on hidden states $\mathbf{s}_1, \dots, \mathbf{s}_m$ that can be used together with the current hidden state $\mathbf{h}_i$ for prediction. The context vector $\mathbf{c}_i$ at position is calculated as an average of the previous states weighted with the attention scores $\mathbf{a}_i$:

Terminology consistent with most literature.
But not our previous slides

$$\mathbf{c}_i = \sum_j a_{ij} \mathbf{s}_j$$

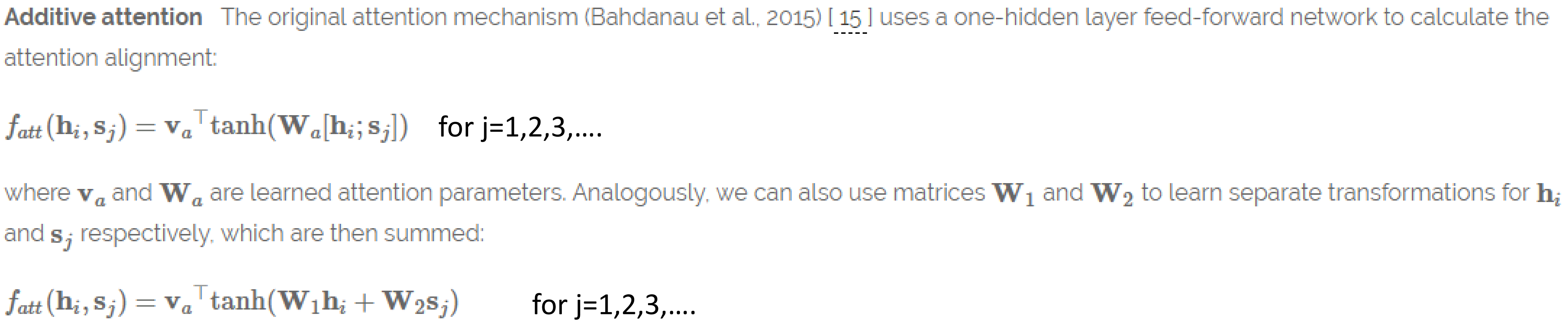
$$\mathbf{a}_i = \text{softmax}(f_{att}(\mathbf{h}_i, \mathbf{s}_j))$$

Softmax over j=1,2,3,….

The attention function $f_{att}(\mathbf{h}_i, \mathbf{s}_j)$ calculates an unnormalized alignment score between the current hidden state $\mathbf{h}_i$ and the previous hidden state $\mathbf{s}_j$. In the following, we will discuss four attention variants: i) additive attention, ii) multiplicative attention, iii) self-attention, and iv) key-value attention.

Material From http://ruder.io/deep-learning-nlp-best-practices/

Summary

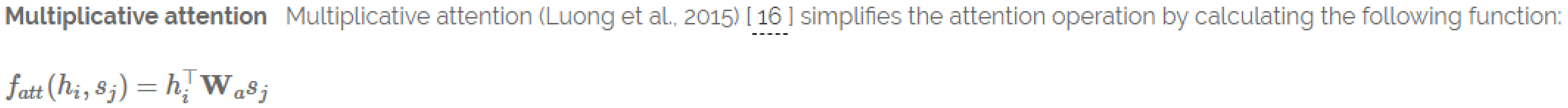


Additive attention The original attention mechanism (Bahdanau et al., 2015) [15] uses a one-hidden layer feed-forward network to calculate the attention alignment:

$f_{att}(\mathbf{h}_i, \mathbf{s}_j) = \mathbf{v_a}^\top \tanh(\mathbf{W}_a[\mathbf{h}_i; \mathbf{s}_j])$ for j=1,2,3,....

where $\mathbf{v}_a$ and $\mathbf{W}_a$ are learned attention parameters. Analogously, we can also use matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ to learn separate transformations for $\mathbf{h}_i$ and $\mathbf{s}_j$ respectively, which are then summed:

$f_{att}(\mathbf{h}_i, \mathbf{s}_j) = \mathbf{v_a}^\top \tanh(\mathbf{W}_1\mathbf{h}_i + \mathbf{W}_2\mathbf{s}_j)$ for j=1,2,3,....



Multiplicative attention Multiplicative attention (Luong et al., 2015) [16] simplifies the attention operation by calculating the following function:

$f_{att}(h_i, s_j) = h_i^\top \mathbf{W}_a s_j$

Self-attention Without any additional information, however, we can still extract relevant aspects from the sentence by allowing it to attend to itself using self-attention (Lin et al., 2017) [18]. Self-attention, also called intra-attention has been used successfully in a variety of tasks including reading comprehension (Cheng et al., 2016) [38], textual entailment (Parikh et al., 2016) [39], and abstractive summarization (Paulus et al., 2017) [40].

We can simplify additive attention to compute the unnormalized alignment score for each hidden state $\mathbf{h}_i$:

$$f_{att}(\mathbf{h}_i) = \mathbf{v}_a{}^\top \tanh(\mathbf{W}_a \mathbf{h}_i)$$

In matrix form, for hidden states $\mathbf{H} = \mathbf{h}_1, \ldots, \mathbf{h}_n$ we can calculate the attention vector $\mathbf{a}$ and the final sentence representation $\mathbf{c}$ as follows:

$$\begin{aligned}
&\mathbf{a} = \text{softmax}(\mathbf{v}_a \tanh(\mathbf{W}_a \mathbf{H}^\top)) \\
&\mathbf{c} = \mathbf{H}\mathbf{a}^\top
\end{aligned}$$

Rather than only extracting one vector, we can perform several hops of attention by using a matrix $\mathbf{V}_a$ instead of $\mathbf{v}_a$, which allows us to extract an attention matrix $\mathbf{A}$:

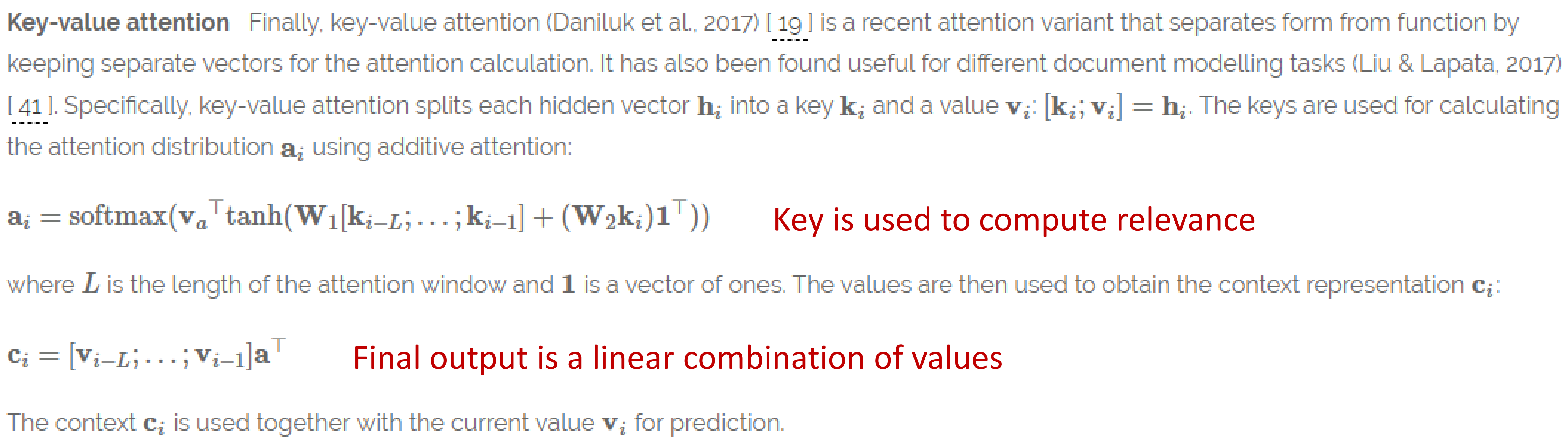
$$\begin{aligned}
&\mathbf{A} = \text{softmax}(\mathbf{V}_a \tanh(\mathbf{W}_a \mathbf{H}^\top)) \\
&\mathbf{C} = \mathbf{A}\mathbf{H}
\end{aligned}$$

In practice, we enforce the following orthogonality constraint to penalize redundancy and encourage diversity in the attention vectors in the form of the squared Frobenius norm:

$$\Omega = \|(\mathbf{A}\mathbf{A}^\top - \mathbf{I}\|_F^2$$

A similar multi-head attention is also used by Vaswani et al. (2017).

Summary



Key-value attention Finally, key-value attention (Daniluk et al., 2017) [19] is a recent attention variant that separates form from function by keeping separate vectors for the attention calculation. It has also been found useful for different document modelling tasks (Liu & Lapata, 2017) [41]. Specifically, key-value attention splits each hidden vector $\mathbf{h}_i$ into a key $\mathbf{k}_i$ and a value $\mathbf{v}_i$: $[\mathbf{k}_i; \mathbf{v}_i] = \mathbf{h}_i$. The keys are used for calculating the attention distribution $\mathbf{a}_i$ using additive attention:

$$\mathbf{a}_i = \text{softmax}(\mathbf{v}_a{}^\top \tanh(\mathbf{W}_1[\mathbf{k}_{i-L}; \ldots; \mathbf{k}_{i-1}] + (\mathbf{W}_2\mathbf{k}_i)\mathbf{1}^\top))$$

Key is used to compute relevance

where L is the length of the attention window and $\mathbf{1}$ is a vector of ones. The values are then used to obtain the context representation $\mathbf{c}_i$:

$$\mathbf{c}_i = [\mathbf{v}_{i-L}; \ldots; \mathbf{v}_{i-1}]\mathbf{a}^\top$$

Final output is a linear combination of values

The context $\mathbf{c}_i$ is used together with the current value $\mathbf{v}_i$ for prediction.

Resources

Resources for attentions and NTMs

- https://distill.pub/2016/augmented-rnns/

- A good read : http://ruder.io/deep-learning-nlp-best-practices/

(lot of notes and tricks for deep learning in NLP)

- Google cache: http://webcache.googleusercontent.com/search?q=cache:t_VgmDEvBo8J:ruder.io/deep-learning-nlp-best-practices/+&cd=3&hl=en&ct=clnk&gl=us
- https://distill.pub/2016/augmented-rnns/#attentional-interfaces

(with very nice animation)

Thanks

Some materials are taken from https://blog.heuritech.com/2016/01/20/attention-mechanism/
https://mchromiak.github.io/articles/2017/Sep/12/Transformer-Attention-is-all-you-need/#.WtstO3puaUl