What is attention in NNs?

(with two examples)

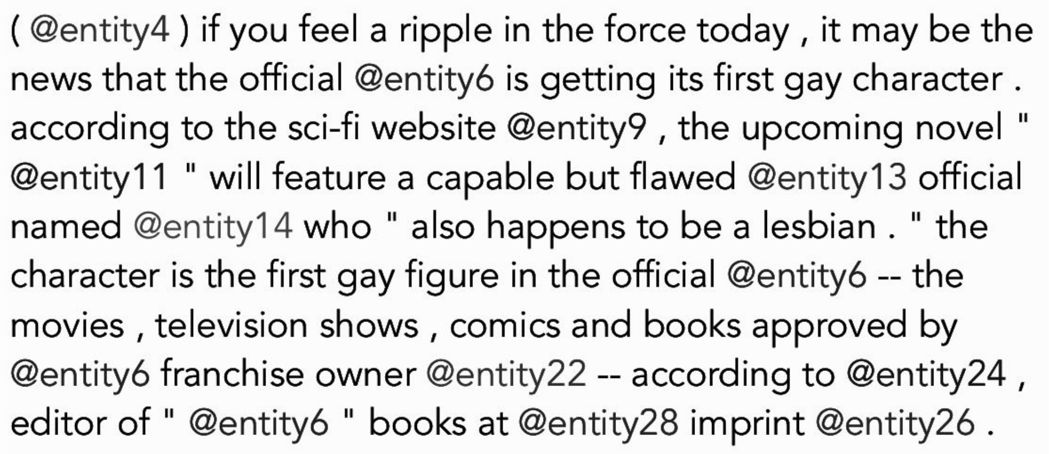
Simon Šuster

27 June ‘17, CLiPS

Machine comprehension

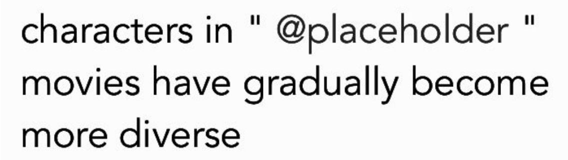
CNN/Daily Mail Cloze Dataset

Passage *p*



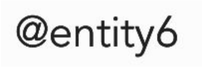
(@entity4) if you feel a ripple in the force today , it may be the news that the official @entity6 is getting its first gay character . according to the sci-fi website @entity9 , the upcoming novel " @entity11 " will feature a capable but flawed @entity13 official named @entity14 who " also happens to be a lesbian . " the character is the first gay figure in the official @entity6 -- the movies , television shows , comics and books approved by @entity6 franchise owner @entity22 -- according to @entity24 , editor of " @entity6 " books at @entity28 imprint @entity26 .

Query *q*



characters in " @placeholder " movies have gradually become more diverse

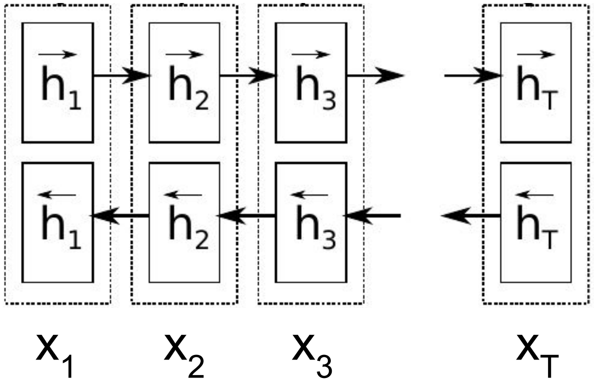
Answer *a*



@entity6

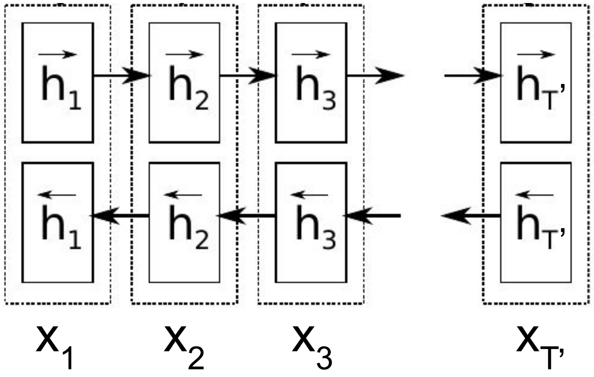
Machine comprehension

Encode passage *p*



$p = \text{concat}(\overrightarrow{h}_T, \overleftarrow{h}_1)$

Encode query *q*

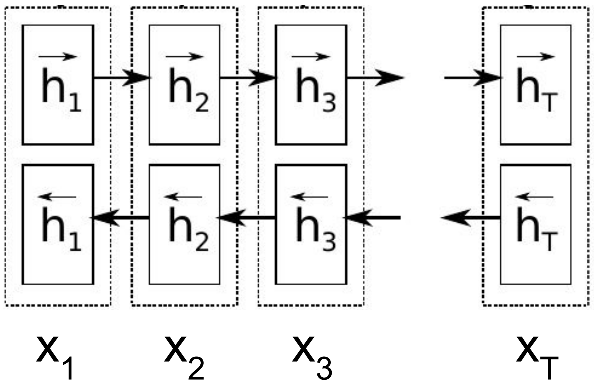


$q = \text{concat}(\overrightarrow{h}_{T'}, \overleftarrow{h}_1)$

Use (any flavor of) RNN to encode passage and query.

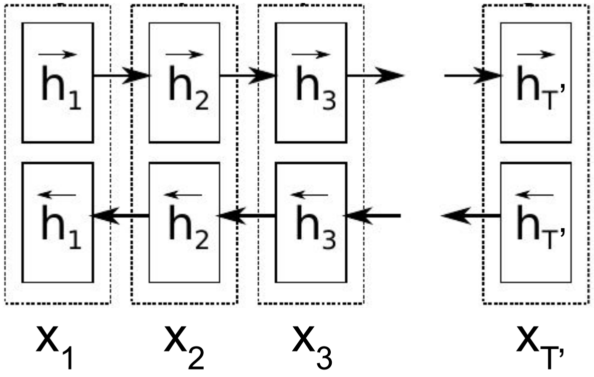
Machine comprehension

Encode passage *p*



$p = \text{concat}(\overrightarrow{h}_T, \overleftarrow{h}_1)$

Encode query *q*



$q = \text{concat}(\overrightarrow{h}_{T'}, \overleftarrow{h}_1)$

Use (any flavor of) RNN to encode passage and query.

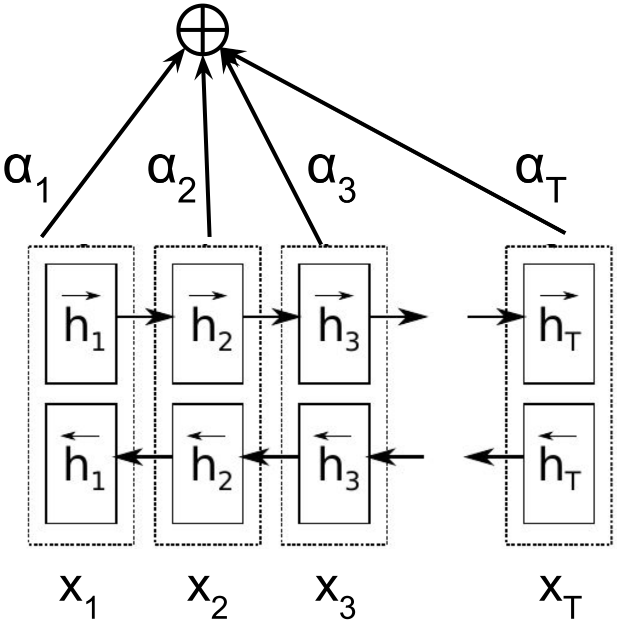
- We could predict an answer directly from p and q.
- But T can be large (documents), which is problematic for RNNs †.

Can we somehow select the information relevant to the query?

- Attentive reader (Chen et al. 2016, Hermann et al. 2015)

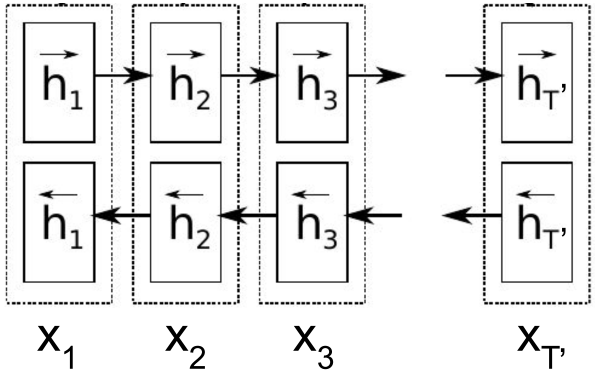
† Depends somewhat on chosen flavor.

Encode passage *p*



$p_i = \text{concat}(\overrightarrow{h}_i, \overleftarrow{h}_i)$

Encode query *q*



$q = \text{concat}(\overrightarrow{h}_{T'}, \overleftarrow{h}_1)$

α are attention weights. They form a probability distribution.

Model gives a prediction by:

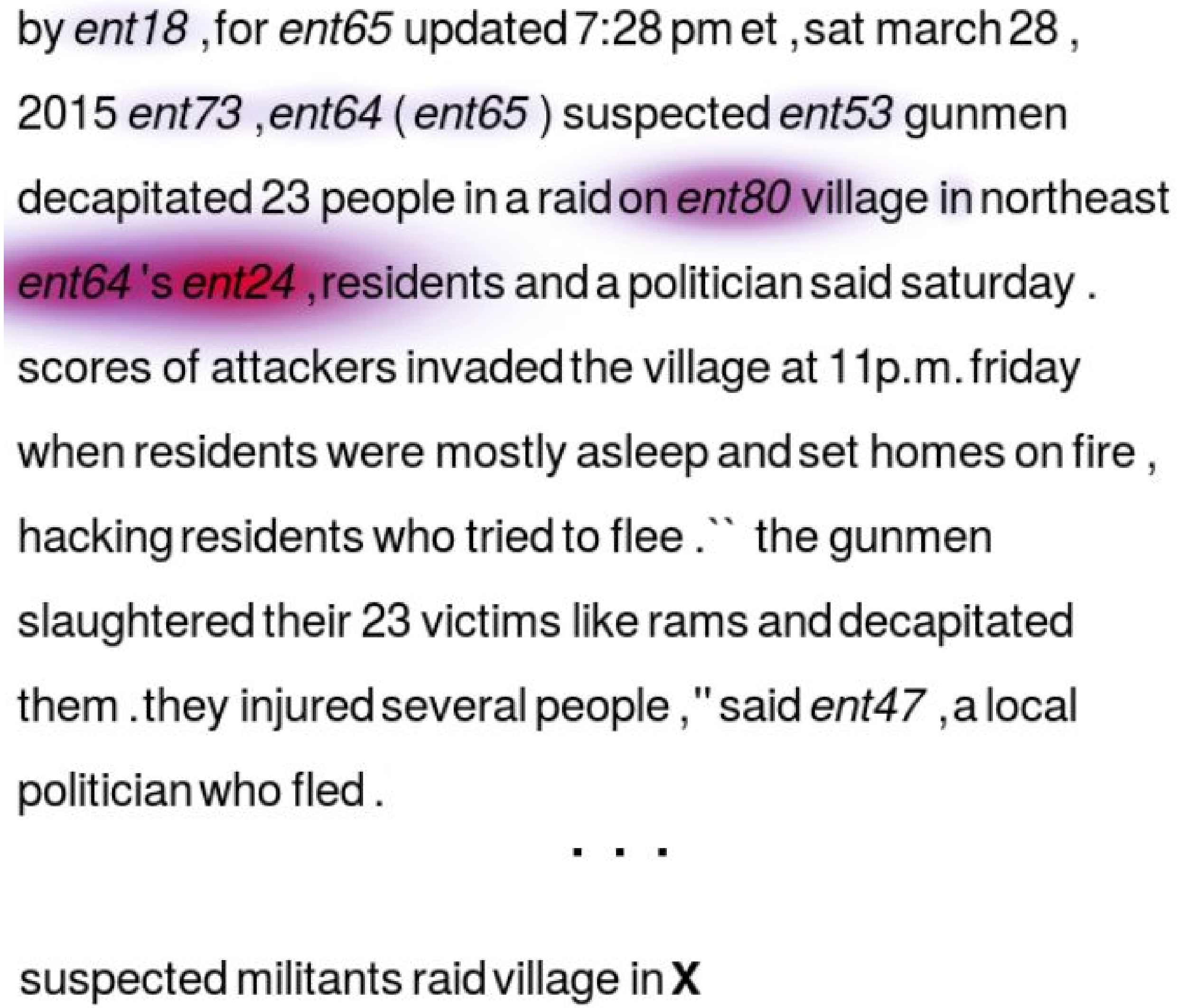
- building the output vector
  $o = \sum_i \alpha_i p_i$
- and predicting the answer
  $a = \text{best\_answer}_{a \in A}(o)$.

Obtaining αs:

- $\alpha_i = \text{softmax}_i\ q^T p_i$
- $\alpha_i = \text{softmax}_i\ q^T W p_i$
- $\alpha_i = \text{MLP}(q, p_i)$

Annotating passage with attention weights

(Hermann et al. 2015)

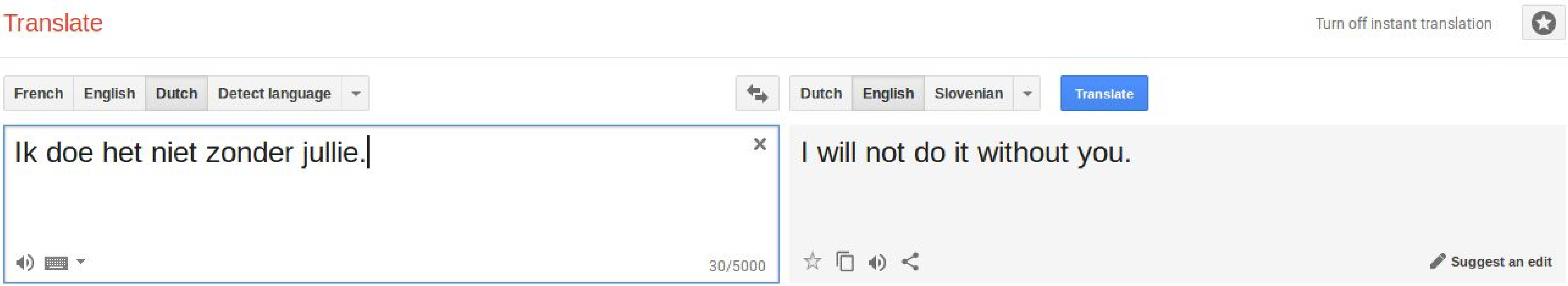


by *ent18* , for *ent65* updated 7:28 pm et , sat march 28 , 2015 *ent73* , *ent64* (*ent65*) suspected *ent53* gunmen decapitated 23 people in a raid on *ent80* village in northeast *ent64* 's *ent24* , residents and a politician said saturday . scores of attackers invaded the village at 11p.m. friday when residents were mostly asleep and set homes on fire , hacking residents who tried to flee . `` the gunmen slaughtered their 23 victims like rams and decapitated them . they injured several people , '' said *ent47* , a local politician who fled .

. . .

suspected militants raid village in **X**

Neural machine translation



Ik doe het niet zonder jullie.

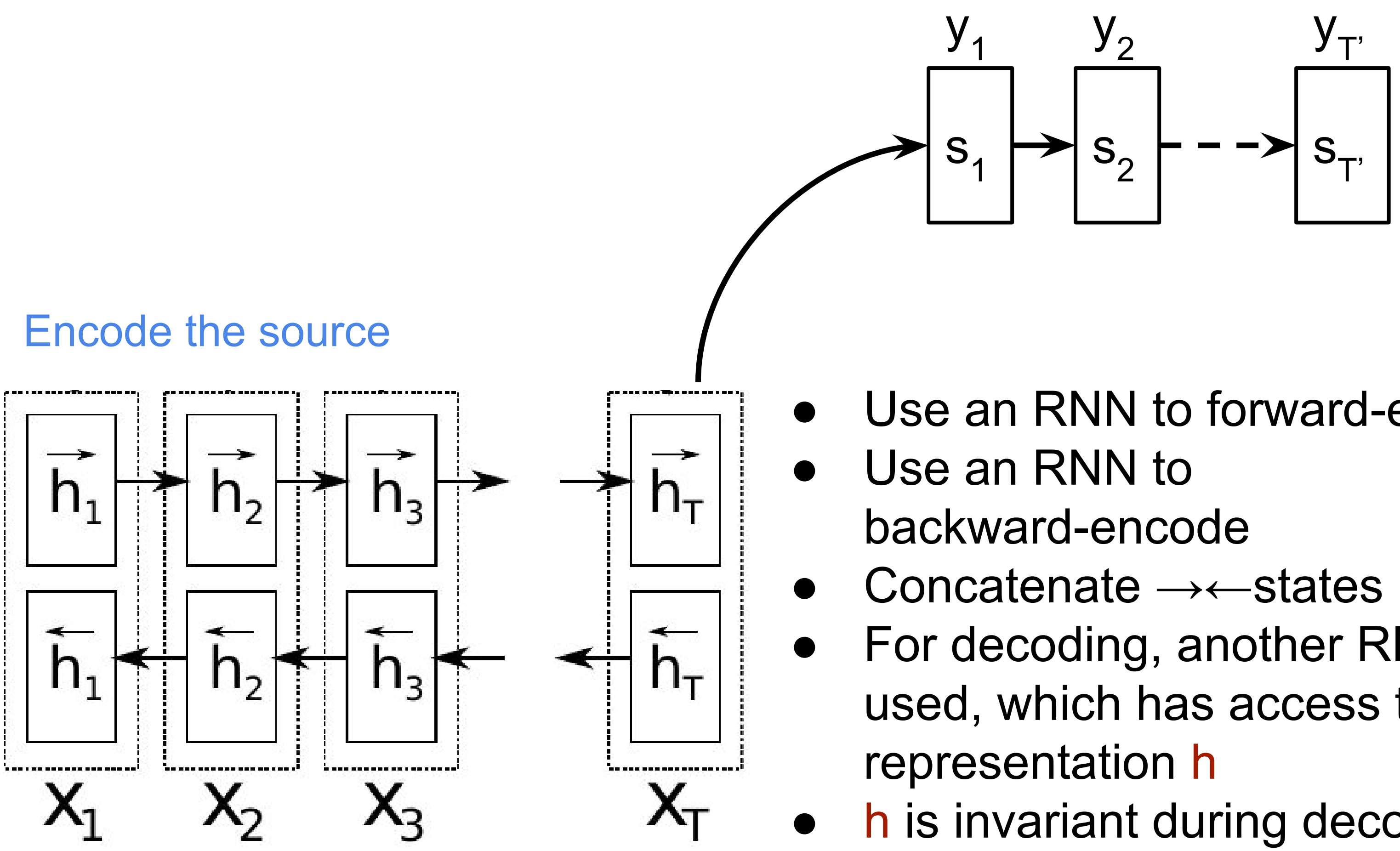
I will not do it without you.

Can model $p(target|source)$ in an end-to-end way

A simple neural MT model

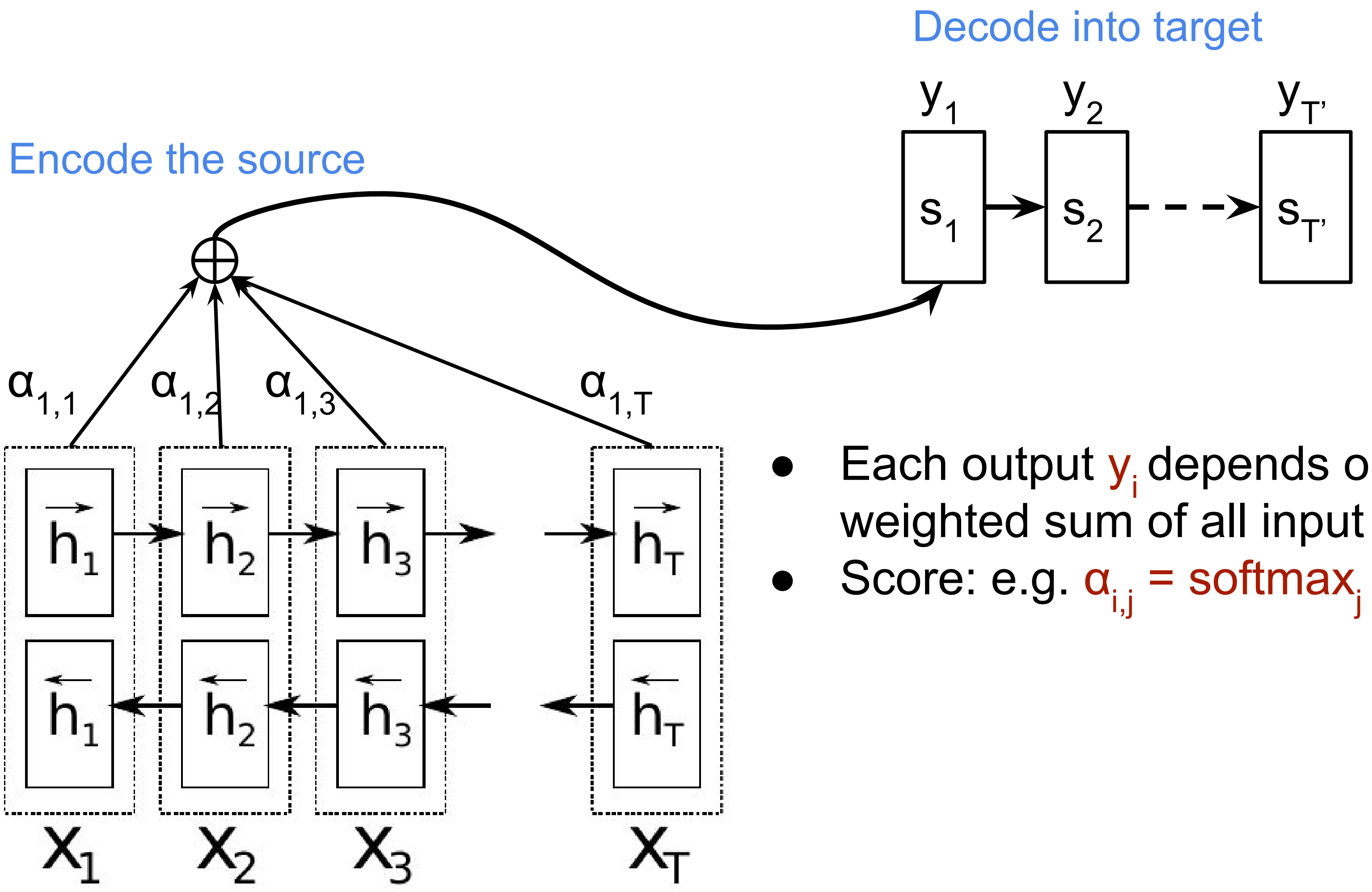
Decode into target

Encode the source



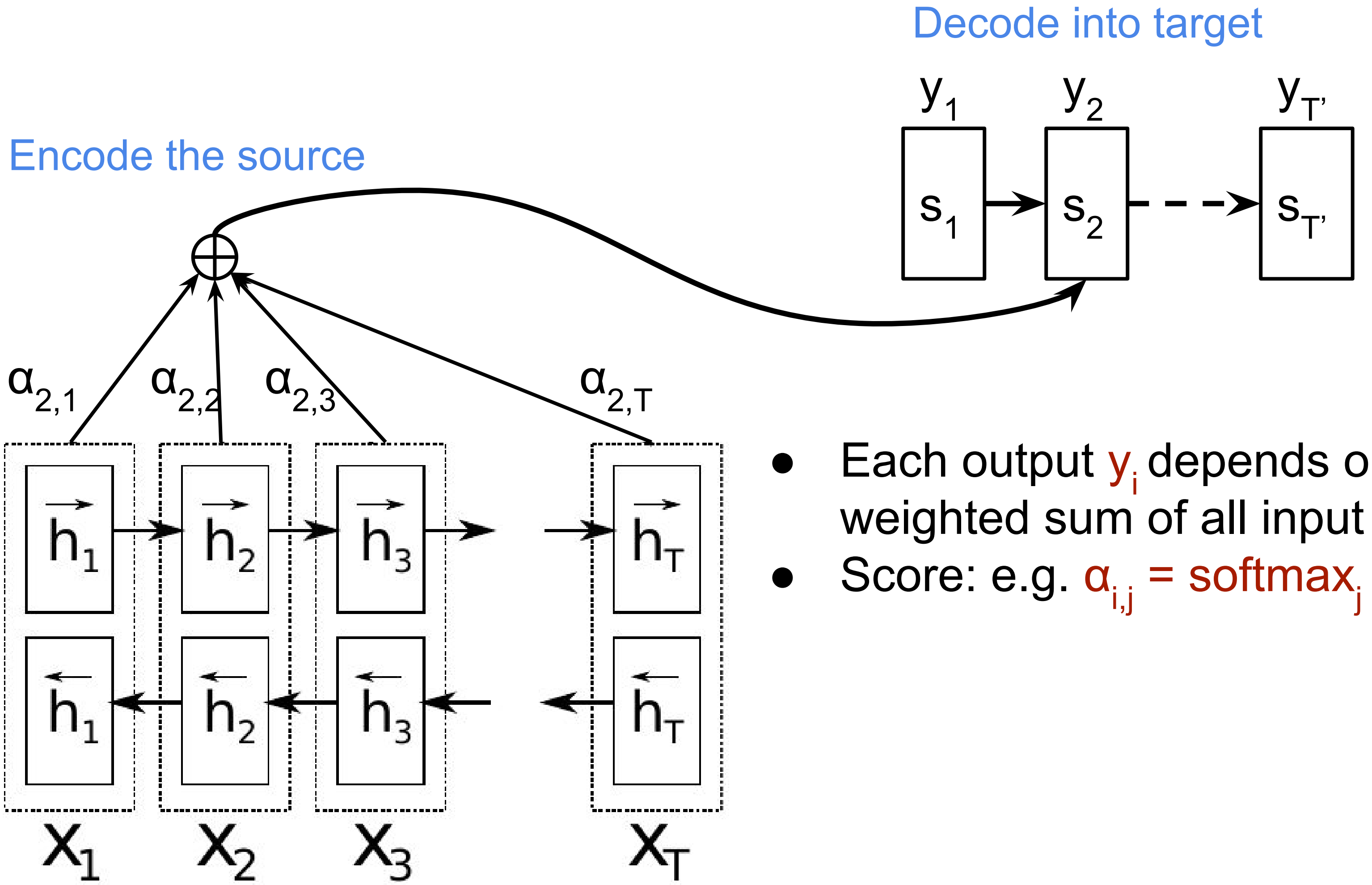
- Use an RNN to forward-encode
- Use an RNN to backward-encode
- Concatenate →←states
- For decoding, another RNN is used, which has access to a representation h
- h is invariant during decoding!
- Will work OK only for very short sentences

Adding attention (Bahdanau et al. 2014, Luong et al. 2015)



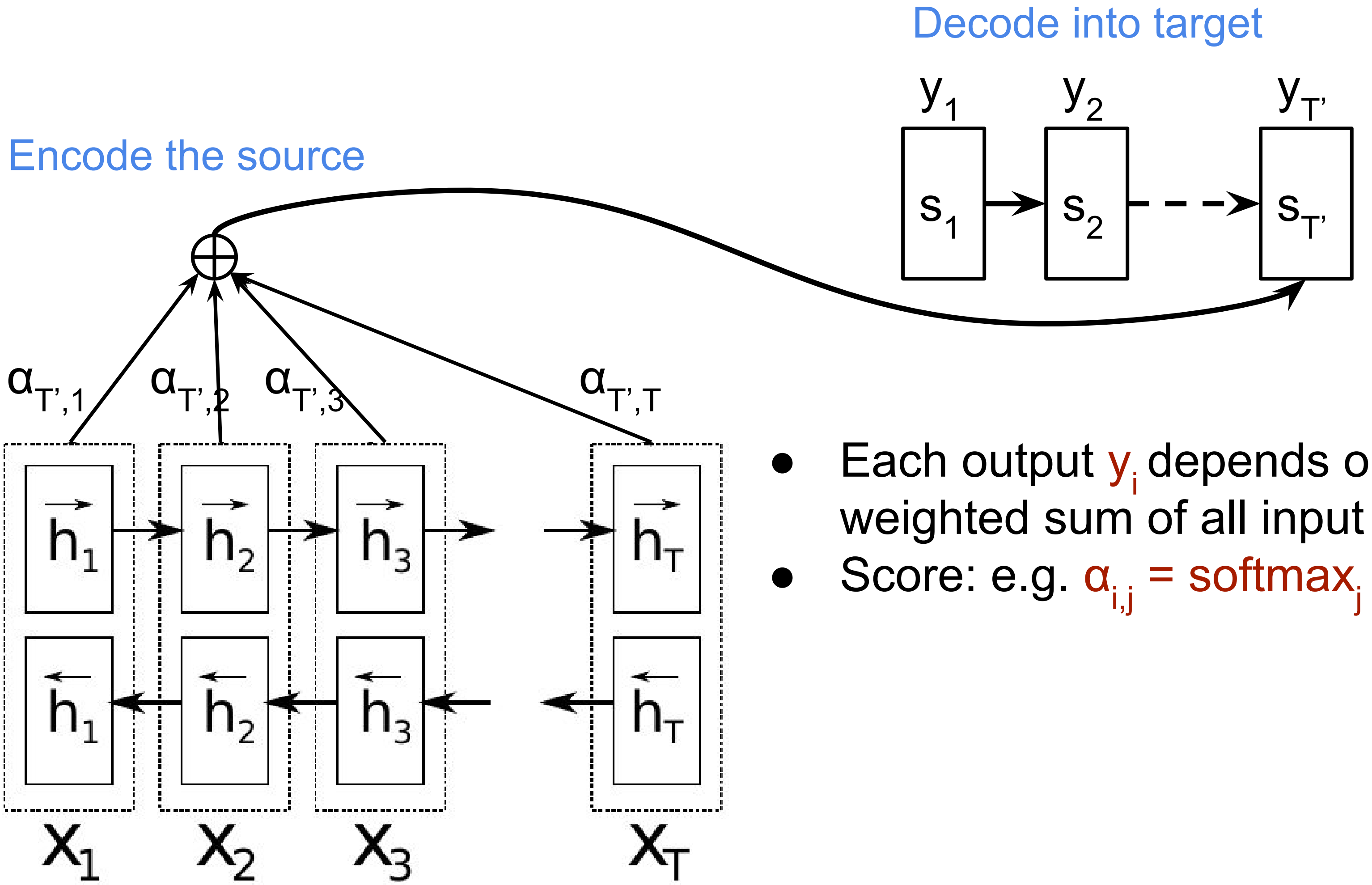
- Each output y_i depends on a weighted sum of all input states
- Score: e.g. $\alpha_{i,j}$ = softmax$_j$ $h_j^T s_i$

Adding attention (Bahdanau et al. 2014, Luong et al. 2015)



- Each output y_i depends on a weighted sum of all input states
- Score: e.g. $\alpha_{i,j}$ = softmax$_j$ $h_j^T s_i$

Adding attention (Bahdanau et al. 2014, Luong et al. 2015)

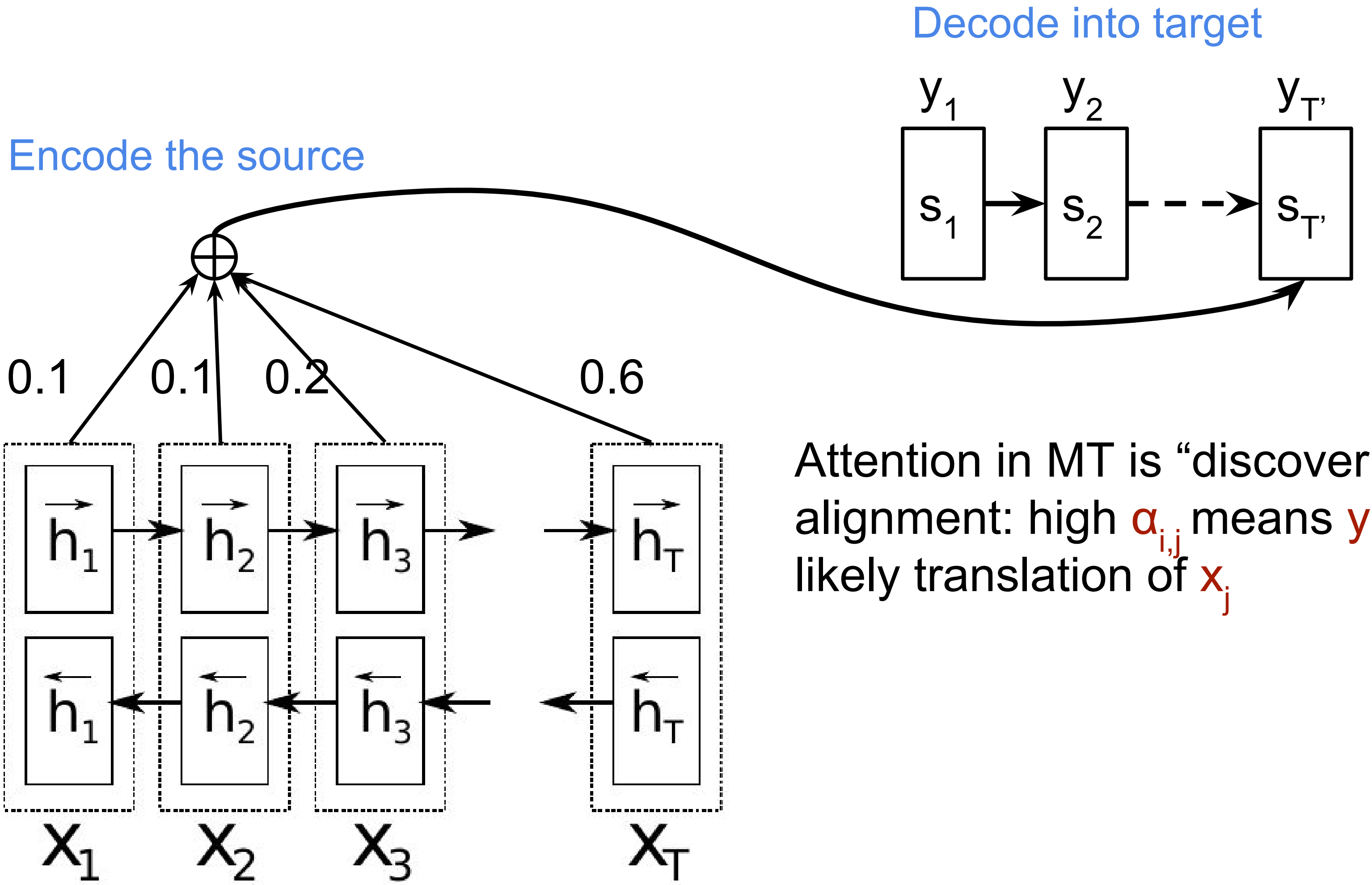


Decode into target

Encode the source

- Each output y_i depends on a weighted sum of all input states
- Score: e.g. $\alpha_{i,j} = \text{softmax}_j\ h_j{}^T s_i$

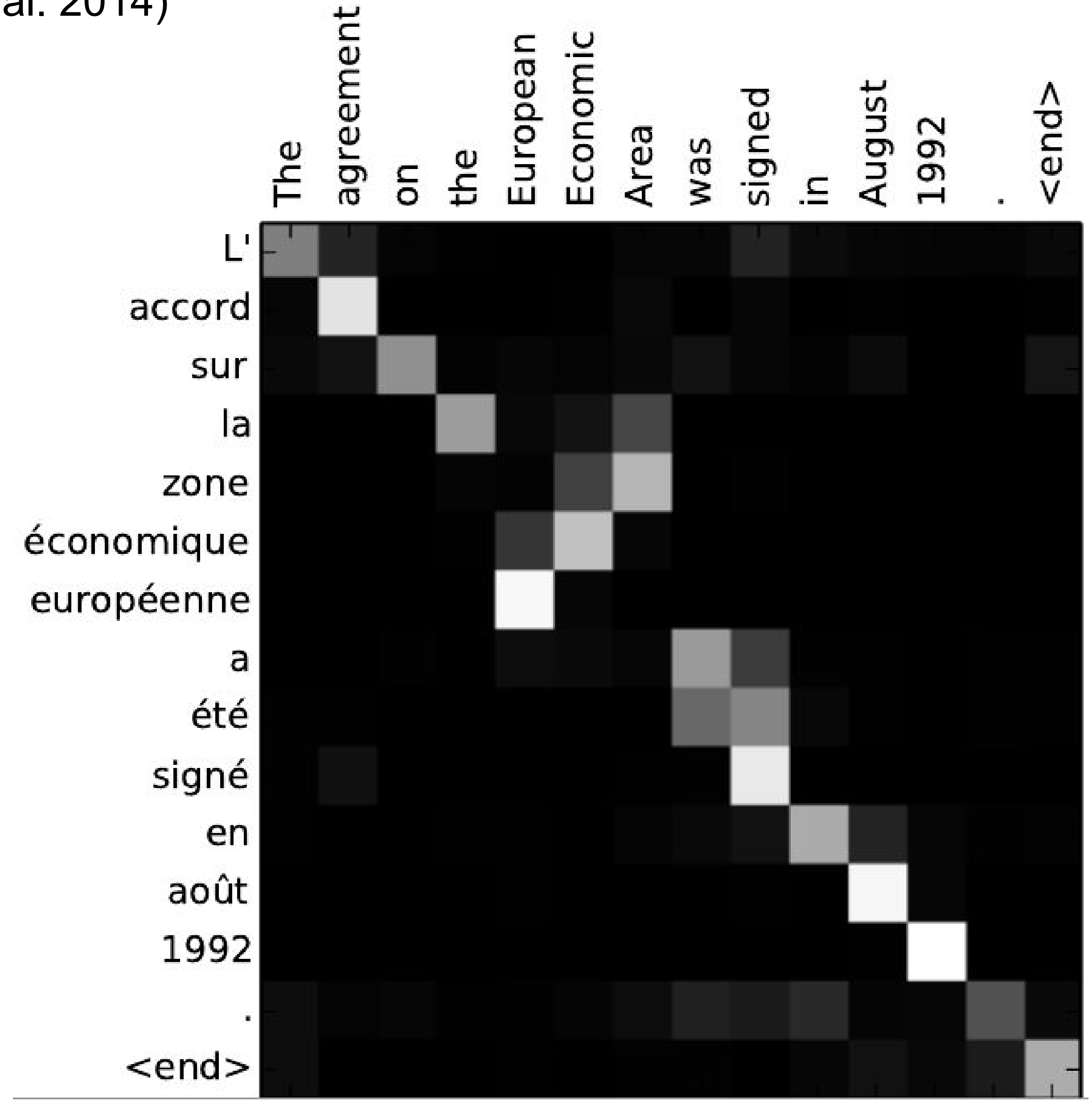
Adding attention (Bahdanau et al. 2014, Luong et al. 2015)



Attention in MT is “discovering” alignment: high $\alpha_{i,j}$ means y_i is a likely translation of x_j

Alignment matrix from attention weights

(Bahdanau et al. 2014)



Useful references

- Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Cho, K. (2015). Natural language understanding with distributed representation. *arXiv preprint arXiv:1511.07916*.
- Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- Manning, C. Lecture 10: Neural Machine Translation and Models with Attention. https://www.youtube.com/watch?v=IxQtK2SjWWM
- Britz, Denny (2016). Attention and Memory in Deep Learning and NLP. http://www.wildml.com/2016/01/attention-and-memory-in-deep-learning-and-nlp/
- Dyer, C. (2017). Lecture 8 - Generating Language with Attention. https://www.youtube.com/watch?v=ah7_mfl7LD0
- Chen, D., Bolton, J., & Manning, C. D. (2016). A thorough examination of the cnn/daily mail reading comprehension task. In ACL.
- Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. In NIPS.