I want to know what attention is
I want you to show me

Introduction to attention in NLP
(a practitioner's perspective)

Simon Šuster
Oct 2, 2020

Intuition

Given a collection of input representations,
the attention mechanism:

1. finds relevance scores for input representations based on our current point of interest
2. uses the relevance scores to weigh the input representations
3. aggregates those into a single representation

Basic terminology

Given a collection of input representations (**keys**),
the attention mechanism:

1. finds relevance scores for input representations based on our current point of interest (**query**)
2. uses the relevance scores to weigh the input representations (**values**)
3. aggregates those into a single representation (**context vector**)

Two main reasons for using attention:

- to improve model’s performance,
- for interpretability (visual highlights of attention weights to analyse a model’s prediction).

Different foci in literature:

- establishing relevance & compatibility
- memory addressing
- feature selection
- discovering alignment
- interpretability tool

A generalised view of attention

$\boxed{e = f(q, K)}$

f is a compatibility function

q is a query vector, $q \in R^{n(q)}$

K are key vectors, $K \in R^{n(k) \times d(k)}$

"Energy" scores e contain information about the relevance of a key to the query

$\boxed{a = g(e)}$

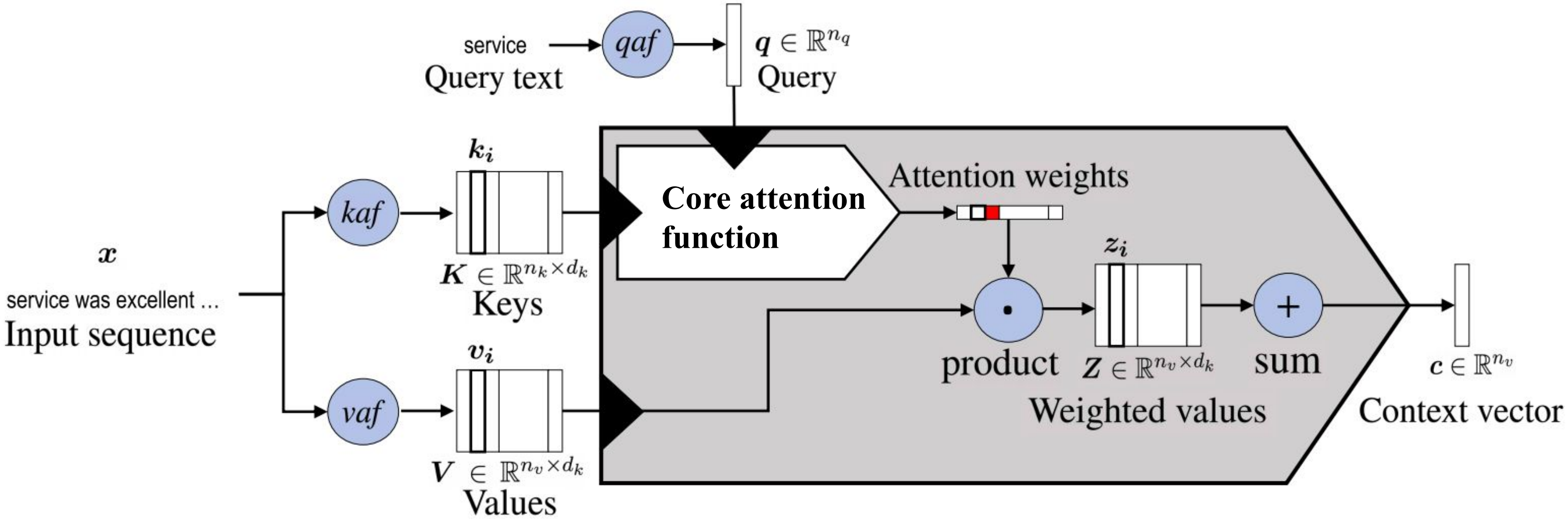
g is a distribution function (commonly softmax)

Attention weights a are the primary outcome of the attention mechanism.
They are applied to the input representation $V \in R^{n(v) \times d(k)}$*, yielding a context vector c:

$$\boxed{c = \sum_{i=1}^{d(k)} a_i v_i}$$

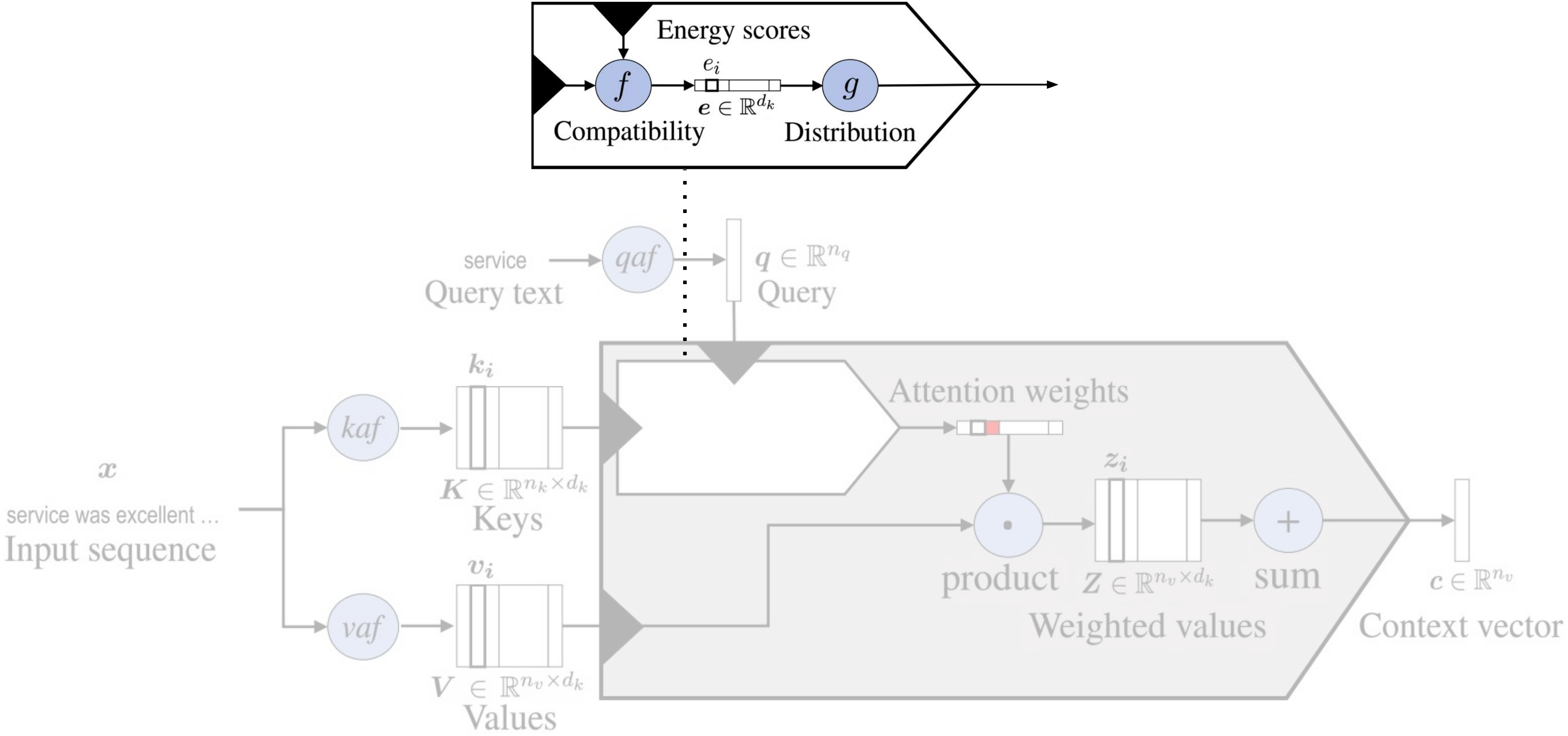
*K and V can be obtained via the same weight matrix.

Diagram of the attention mechanism (1/2)



Galassi et al. (2020)

Diagram of the attention mechanism (2/2)



Galassi et al. (2020)

More on compatibility function $f(q, K)$

Some common approaches:

- $q^\top K$ (dot product)
- $\text{cosine}(q, K)$
- $(q^\top K) / \sqrt{n}$: scaled dot product, e.g. in Transformer; n=key vector dimension, for stability of gradient computation

Parameterised:

- $q^\top W K$ (bilinear or general)
- $\text{act}(q^\top W K + b)$ (MLP)
- $w_{imp}^\top \text{act}(W_1 q + W_2 K + b)$ (additive)
- deep attention, convolution-based attention...

Attention in machine translation

Place of attention in neural machine translation (MT)

Recurrent neural network (RNN) for MT, without attention (Sutskever et al., 2014):

encoder: $h^e_t = f(x_t, h^e_{t-1})$

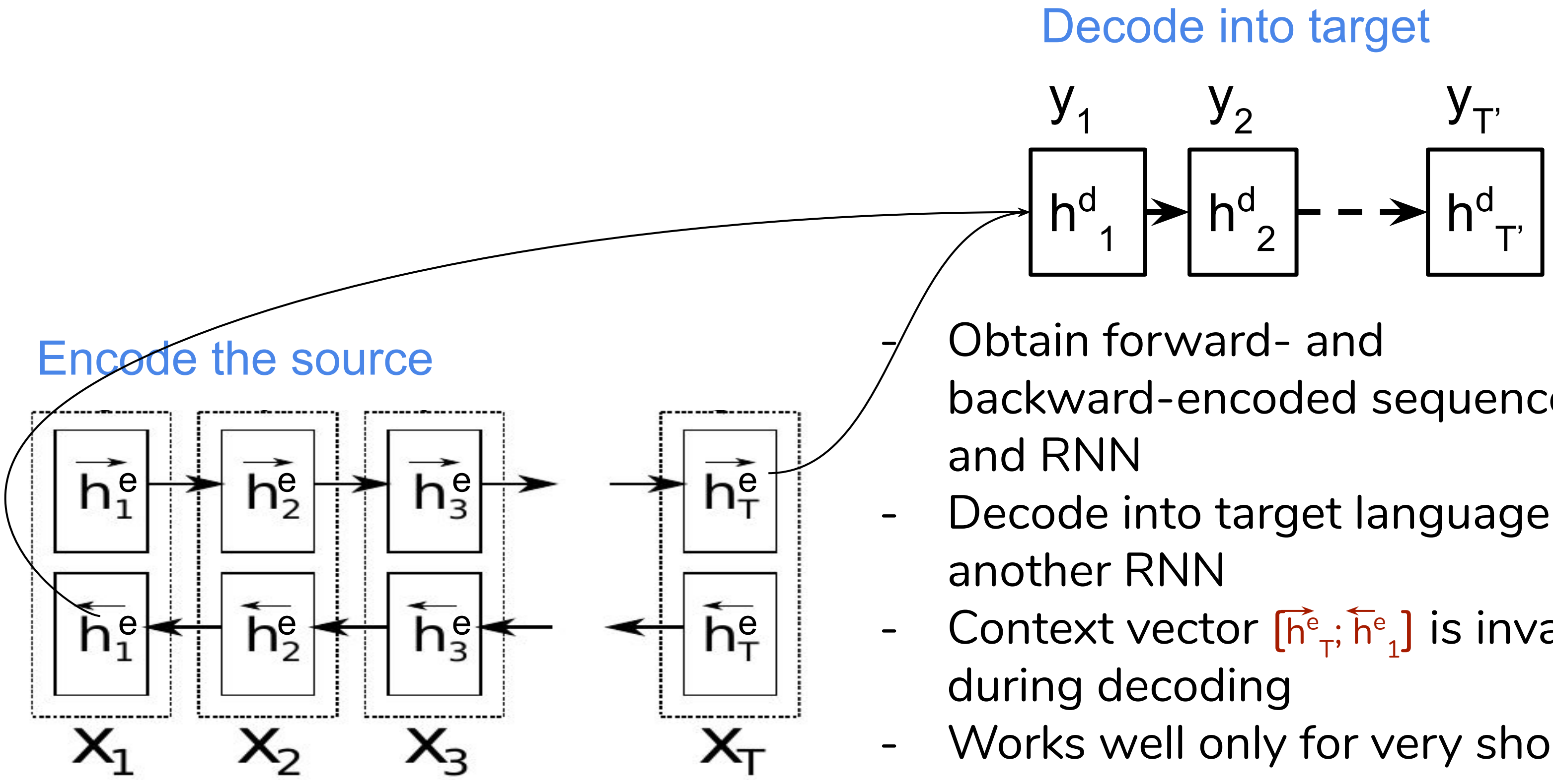
decoder: $h^d_t = f(y_{t-1}, h^d_{t-1}, c)$; $P(y_t|y_{<t}, c) = g(h_t, y_{t-1}, c)$

$c = h^e_T$ (context vector, here set to be encoder's final state)
h^d_t is decoder's newly generated hidden state,
f is a non-linear activation function
g is a non-linear activation function producing valid probabilities

Input sentence is encoded into a single vector c*

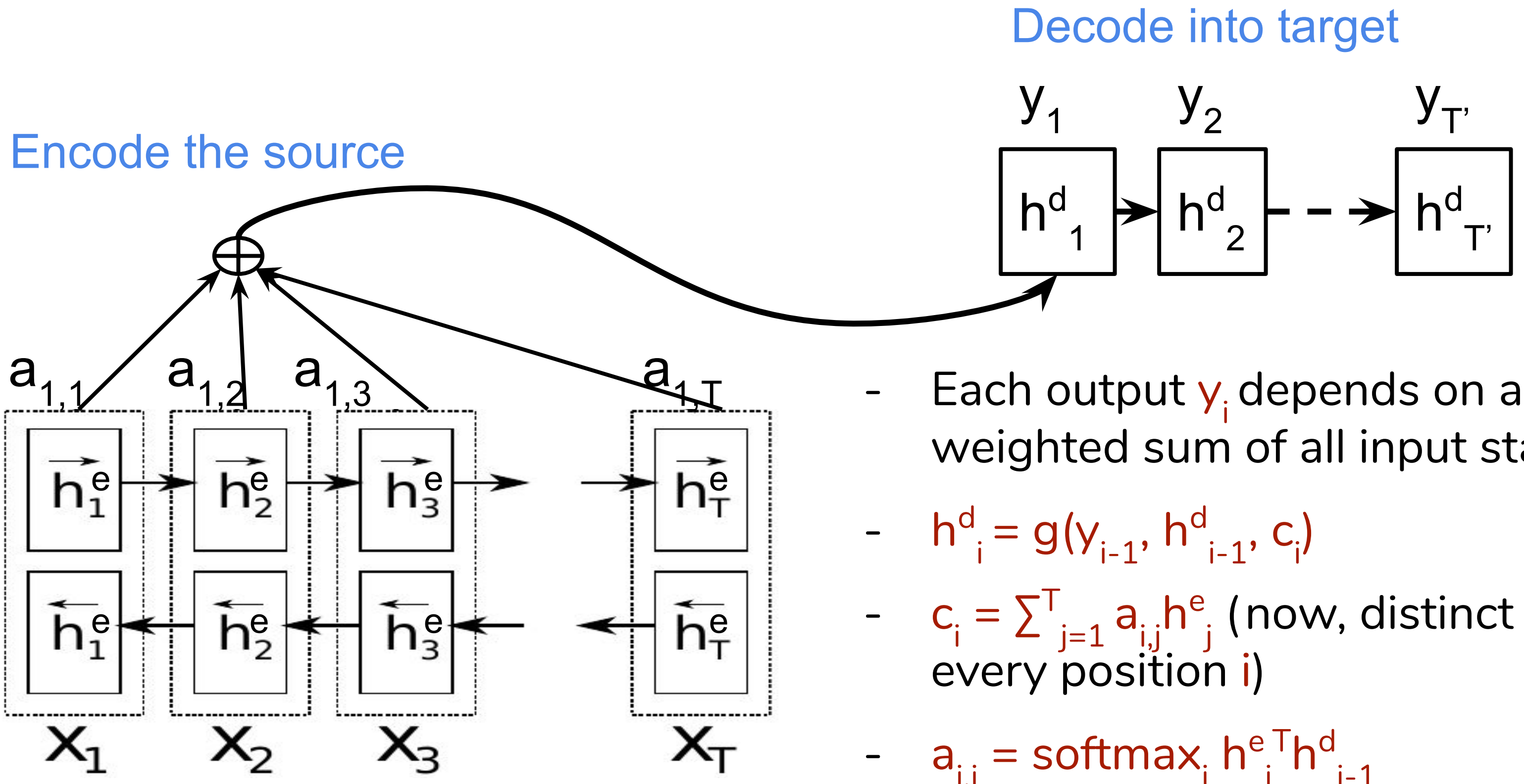
*"*You can't cram the meaning of a whole %&!$# sentence into a single $&!#* vector!*" (R. Mooney)

A neural MT model without attention



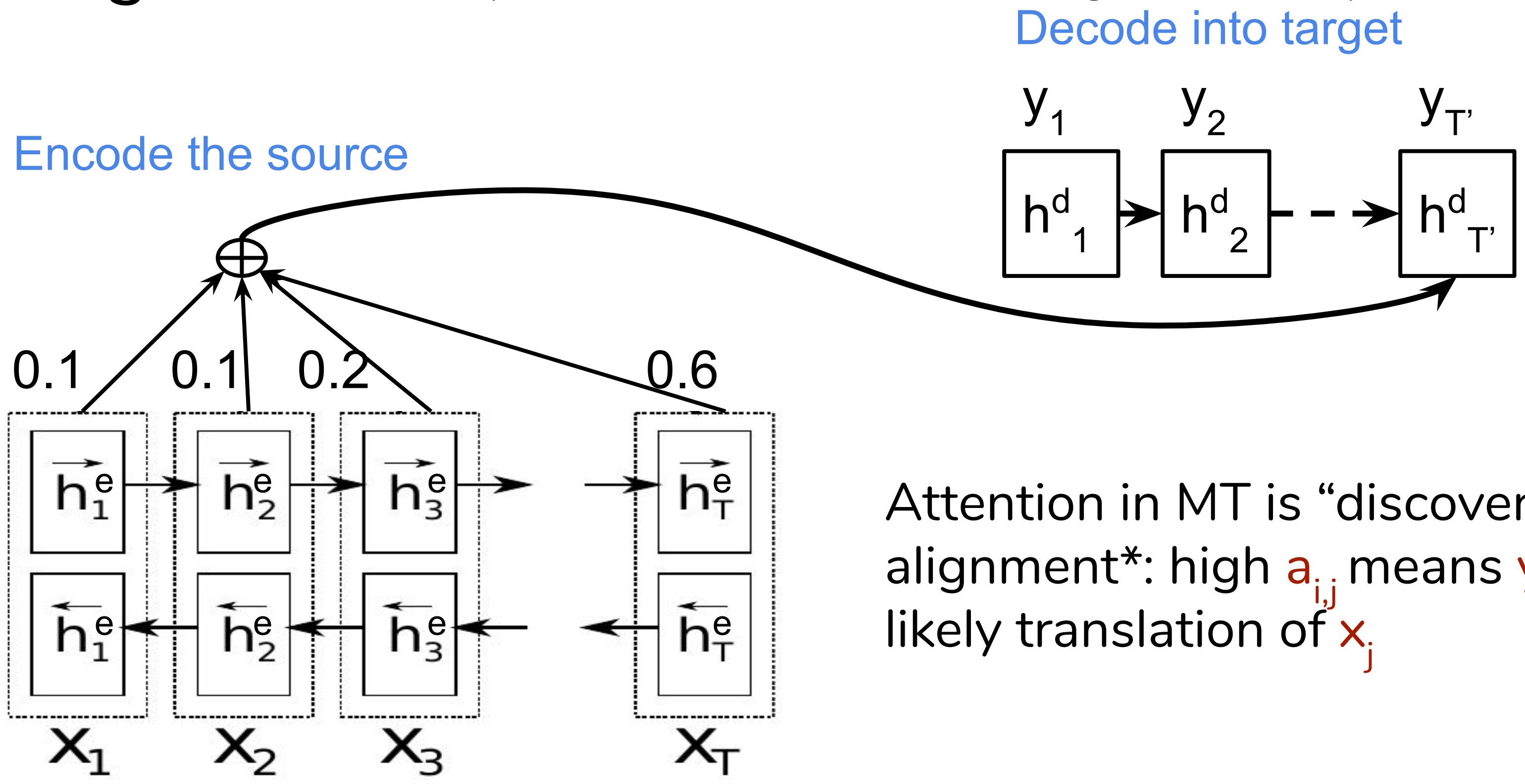
- Obtain forward- and backward-encoded sequence with and RNN
- Decode into target language with another RNN
- Context vector $[\overrightarrow{h}^e_{T'}; \overleftarrow{h}^e_1]$ is invariant during decoding
- Works well only for very short sentences

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



- Each output y_i depends on a weighted sum of all input states
- $h^d_i = g(y_{i-1}, h^d_{i-1}, c_i)$
- $c_i = \sum^T_{j=1} a_{i,j} h^e_j$ (now, distinct c at every position i)
- $a_{i,j} = \text{softmax}_j\, h^{e\,T}_j h^d_{i-1}$
- $(h^e_j = [\overrightarrow{h}^e_j; \overleftarrow{h}^e_j])$

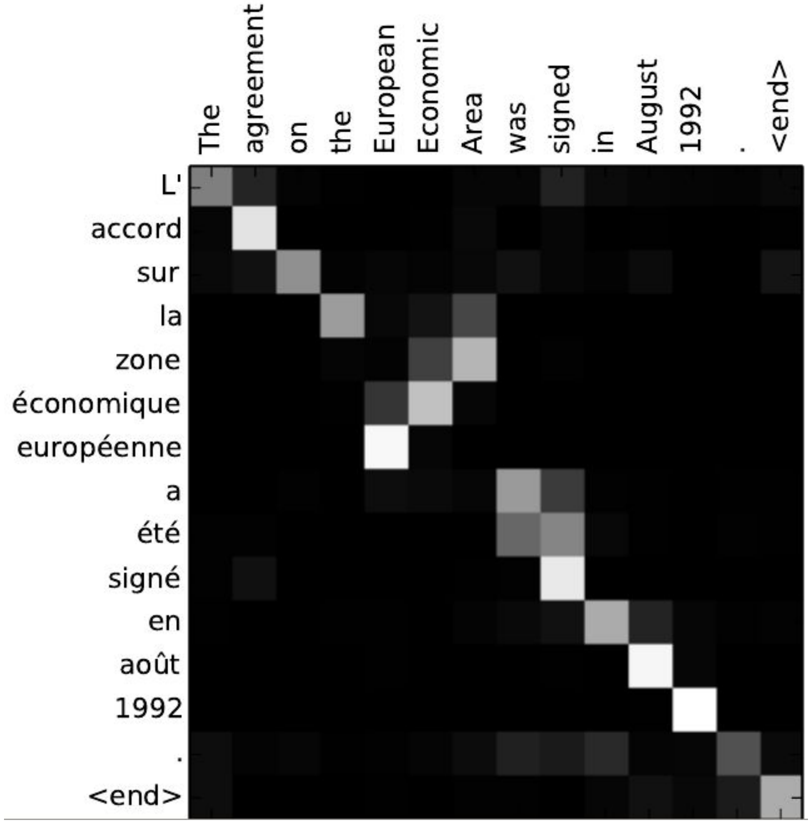
Adding attention (Bahdanau et al. 2014, Luong et al. 2015)



Attention in MT is "discovering" alignment*: high $a_{i,j}$ means y_i is a likely translation of x_j

*Cf. Koehn and Knowles (2017)

Alignment matrix from attention weights
(Bahdanau et al. 2014)



- See word to word translation
- Although attention is a soft alignment, the result is peaky, low-entropy
- Local reordering

A few attention variants

Special case of one input sequence

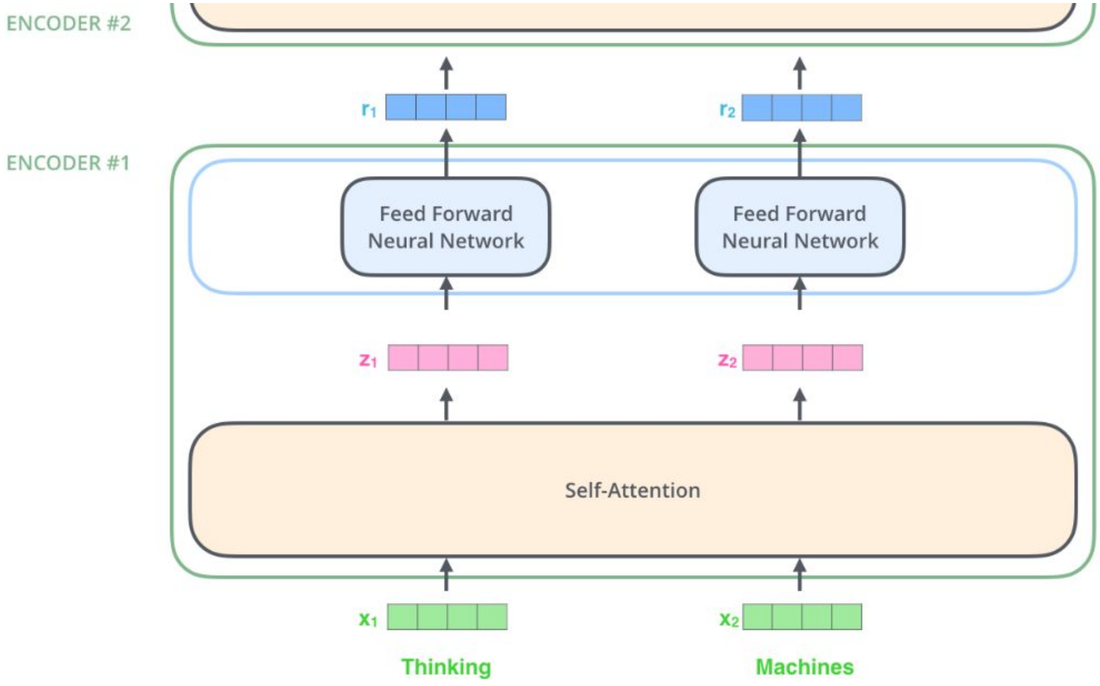
Self-attention:

- $e = f(q, K)$ stays the same, but $x_q \in X_K$
- Relating different positions of the same input to compute its representation
- Intuition: ability to discover lexical relations between tokens

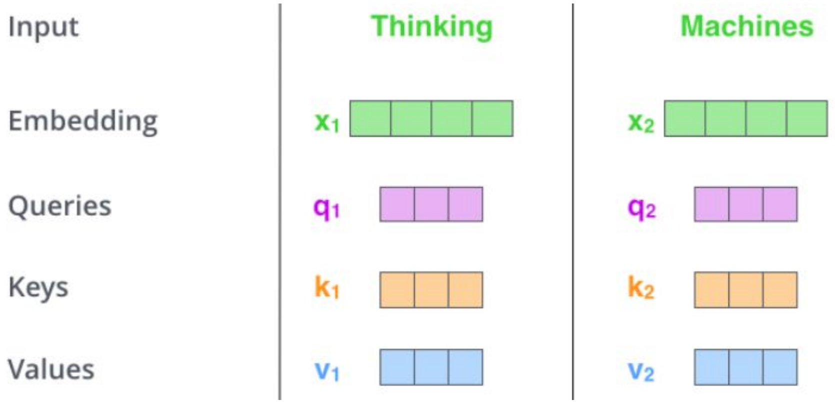
Transformer (Vaswani et al., 2017):

- overcomes sequential computation with an architecture in which “recurrency” is achieved through attention (and positional encoding)
  - self-attention for encoder inputs
  - self-attention for decoder inputs (up to current token)
  - encoder-decoder attention

Attention in Transformer connects different parts of the input



Each word is represented as a key, a query and a value, all with distinct weights



Each of q/k/v’s use multiple weight matrices (“heads”), not only one

Hard attention and biasing the attention distribution

Make a zero-one decision about where to attend (i.e. uses a single sample instead of a distribution)

- harder to train (reinforcement learning)

Other approaches to encouraging sparsity: gumbel softmax, Gaussian noise

While most often we don't have access to attention's target distribution, sometimes knowledge about the desired weight distribution may be available, e.g.

- relevant sentences in a document are somehow marked,
- pre-trained attention weights exist from another task.

Two-way attention and co-attention

Represent the query as a matrix: $Q \in R^{n(q) \times d(q)}$
Energy ("affinity") scores: $E = f(Q, K)$, $E \in R^{d(q) \times d(k)}$

Then the normalisation direction (row- vs. column-wise) on E determines whether we get attention weights for keys or values:

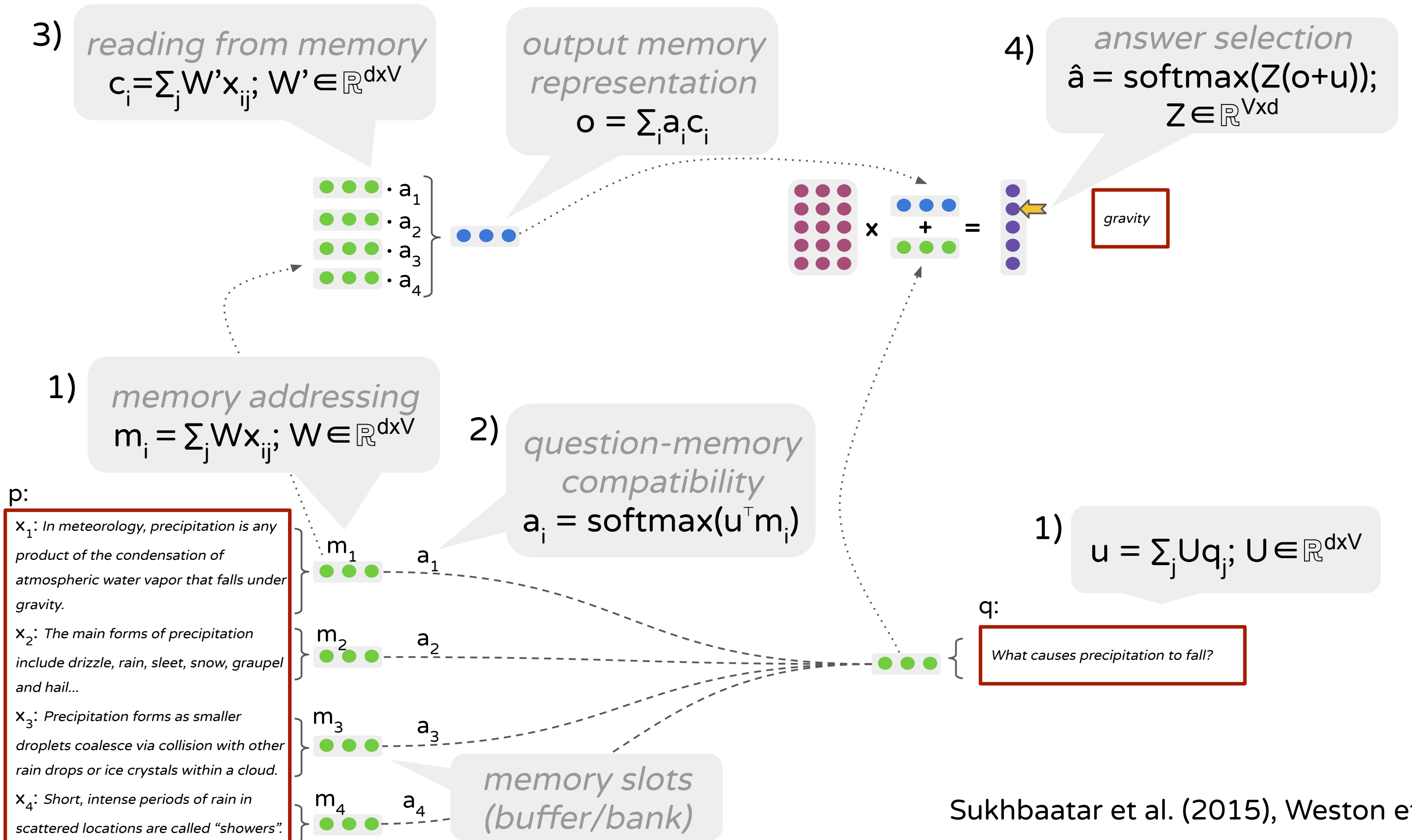
$A^Q = g(E) \in R^{d(q) \times d(k)}$
$A^K = g(E^\top) \in R^{d(k) \times d(q)}$

E.g. instead of representing a sentence with a single vector (say, final LSTM state), have one vector per word

- word-by-word attention in textual entailment (Rocktäschel et al., 2015)
- document word-question word attention in QA (Xiong et al., 2016)

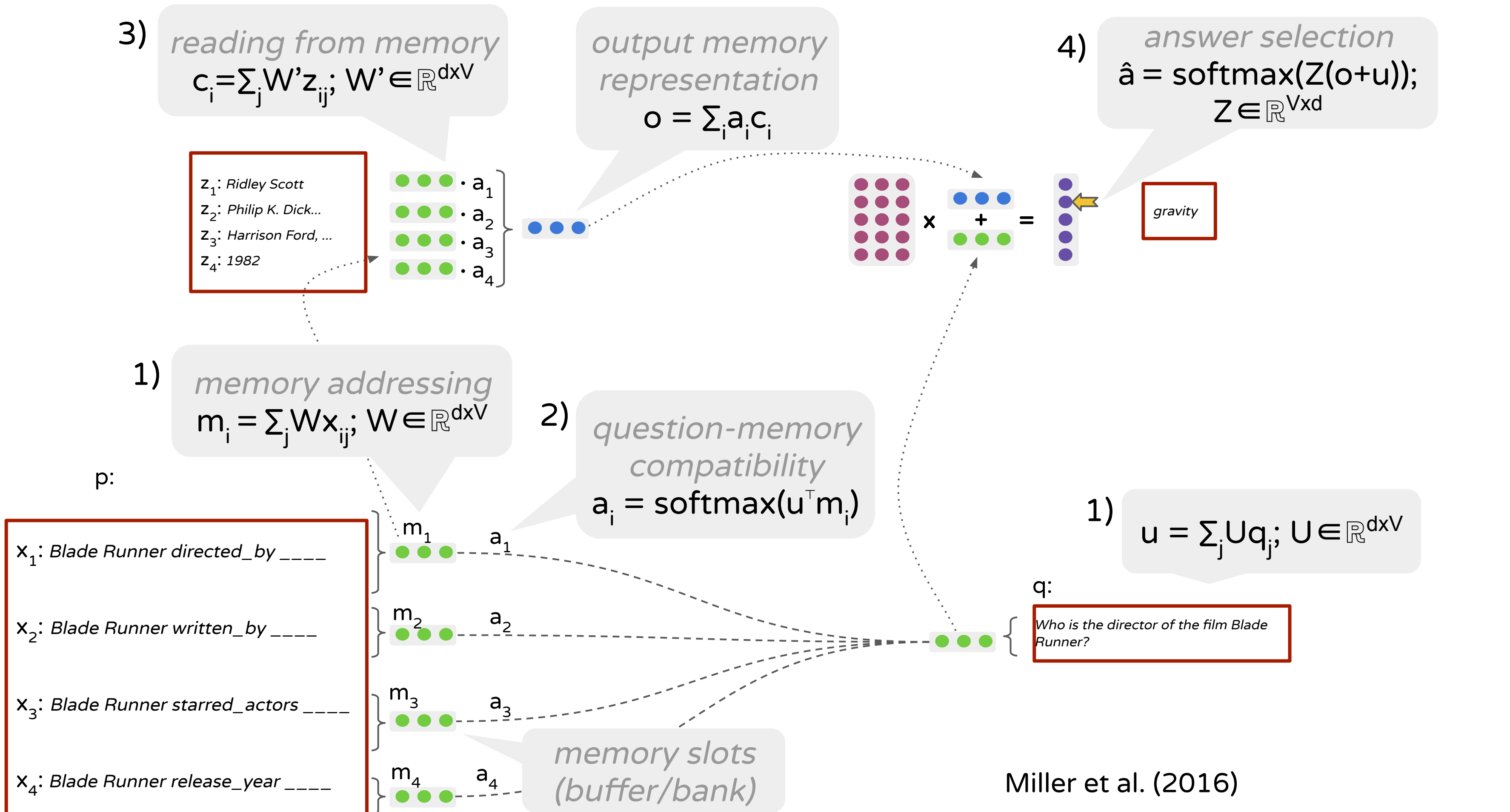
Attention for reading from memory

Attention in a simple memory network



Sukhbaatar et al. (2015), Weston et al. (2015)

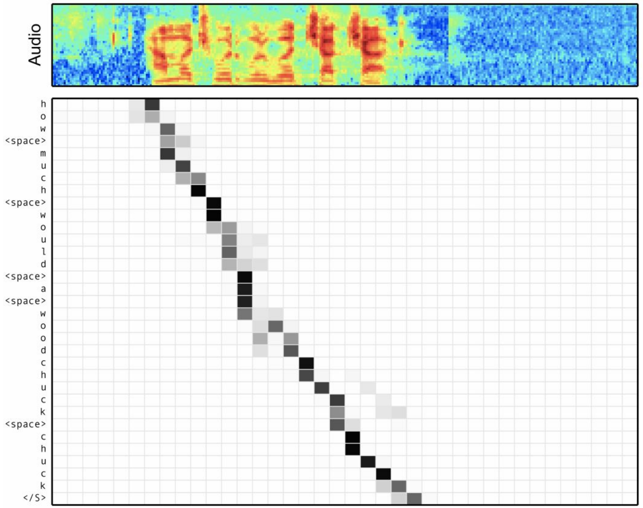
Attention keys and values can be obtained from different inputs

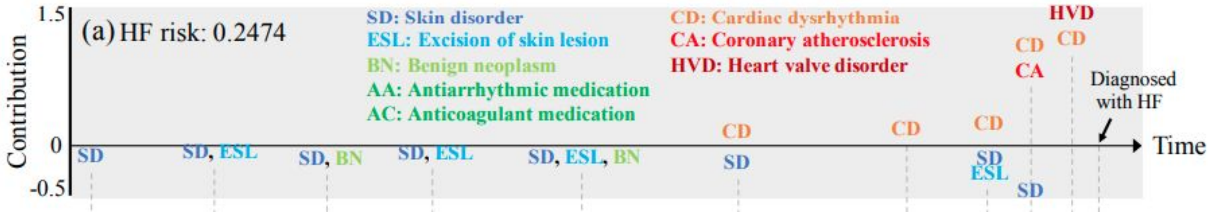


Miller et al. (2016)

Attention in other fields

- Vision
  - image captioning, e.g. Xu et al. (2015)
  - object classification, e.g. Mnih et al. (2014)
- Speech recognition
  - encoding feature vectors from audio frames and decoding into sequence of phonemes (Chorowski et al., 2015)
- Clinical sequential modeling
  - salient medical codes for prediction of heart failure (Choi et al., 2016)





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.
- https://web.stanford.edu/~jurafsky/slp3/10.pdf
- Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1). Cambridge: MIT press. (section 12.4.5.1)
- Galassi, Andrea, Marco Lippi, and Paolo Torroni. "Attention in Natural Language Processing." IEEE Transactions on Neural Networks and Learning Systems (2020): 1–18. Crossref. Web.
- Hu, D. (2019, September). An introductory survey on attention mechanisms in NLP problems. In Proceedings of SAI Intelligent Systems Conference (pp. 432-448). Springer, Cham.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
- Koehn, P., & Knowles, R. (2017). Six challenges for neural machine translation. arXiv preprint arXiv:1706.03872.
- Larochelle, H., & Hinton, G. E. (2010). Learning to combine foveal glimpses with a third-order Boltzmann machine. In Advances in neural information processing systems (pp. 1243-1251).
- Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. In Advances in neural information processing systems (pp. 577-585).
- Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiský, T., & Blunsom, P. (2015). Reasoning about entailment with neural attention. arXiv preprint arXiv:1509.06664.
- Xiong, C., Zhong, V., & Socher, R. (2016). Dynamic coattention networks for question answering. arXiv preprint arXiv:1611.01604.
- Choi, E., Bahadori, M. T., Sun, J., Kulas, J., Schuetz, A., & Stewart, W. (2016). Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In Advances in Neural Information Processing Systems (pp. 3504-3512).