Attention based architectures for NLP
Text generation evaluation metrics

M. Vazirgiannis
LIX, Ecole Polytechnique
February 2024

OUTLINE

- Attention based architectures
  - HAN, NMT, ELMO, TRANSFORMER
- Text generation evaluation metrics

Word embeddings

- W2V [mikolov 13…]: CBOW/SKIPGRAM
- Enriching Word Vectors with Subword Information [ACL17]

Context & Senses

- **“a word is defined by “the company it keeps” (Firth, 1957)**
- Words are *ambiguous*:
  - The amount of deposits in the banks decreased by 3.5% in March
  - The trees on bank of the river were offering their shade to the visitors
- We need different vectors to represent all word/token senses

Deep Contextualized Word Representations

Best paper NAACL 2018

transfer learning has been used in vision since 2012 (ImageNet)
it is used in NLP since 2018! [1] [2]

ELMo: Embeddings from Language Models

Initial problem: traditional word vectors map each word to a single, context-independent vector

But some words have more than one sense! (polysemy)

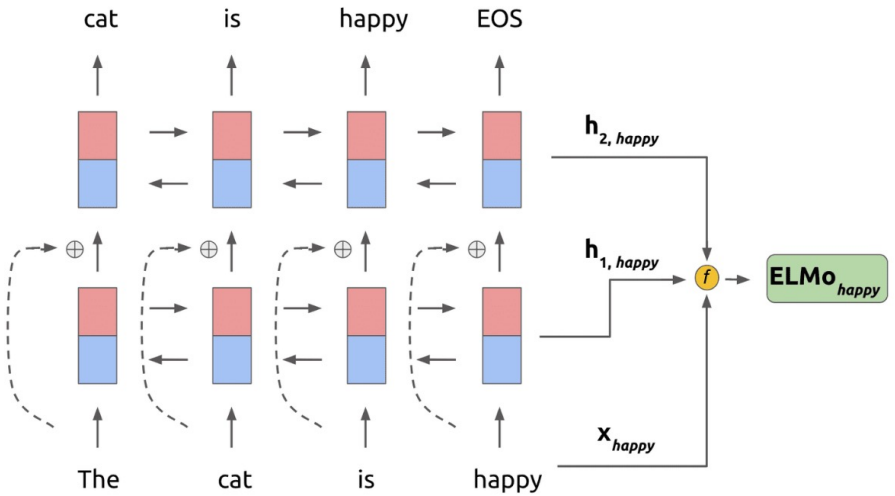
e.g., bank, get, wood, play, mean…

Solution:

- token assigned a representation based on entire input sentence
- use the internal representations of a RNN language model pre-trained on a large dataset

Deep Contextualized Word Representations

- bidirectional LSTM is trained with a coupled language model (LM) on a large text corpus - ELMo (Embeddings from Language Models) representations.

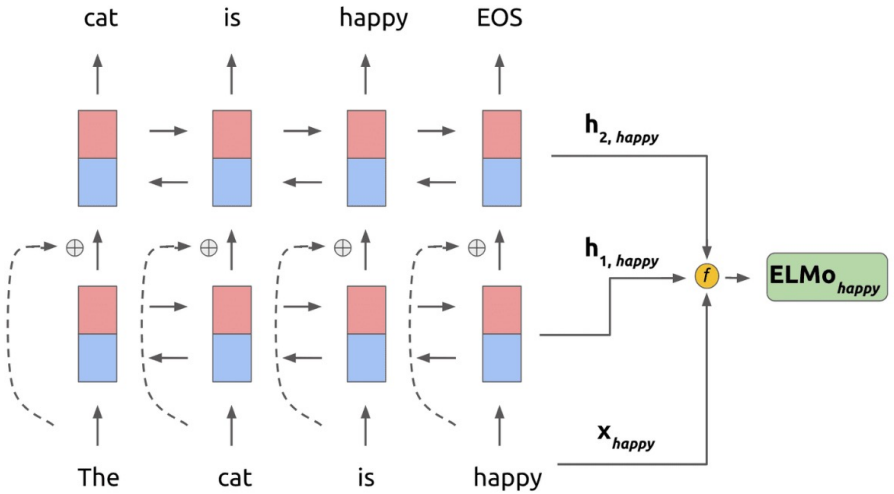


- Forward $p(t_1, t_2 \ldots, t_N) = \prod_{k=1}^{N} p(t_k | t_1, t_2 \ldots, t_N)$
- Bacward $p(t_1, t_2 \ldots, t_N) = \prod_{k=1}^{N} p(t_k | t_{k+1}, t_{k+2} \ldots, t_{k+N})$

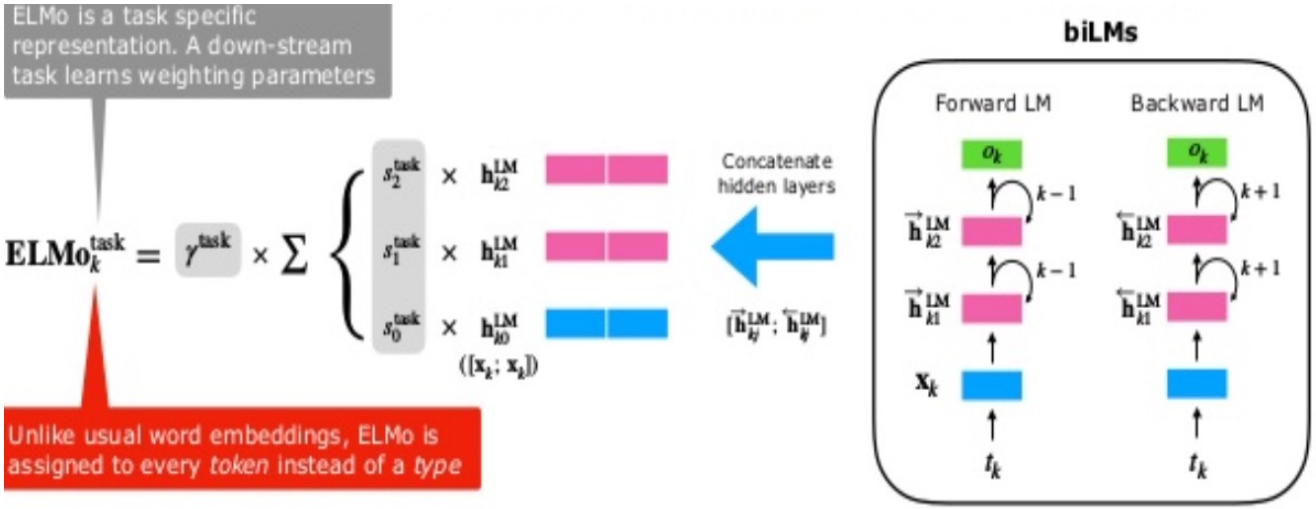
Graphic from: https://www.mihaileric.com/posts/deep-contextualized-word-representations-elmo/

Deep Contextualized Word Representations

- ELMo representations are deep: function of all of the internal layers of the biLM.
  - learn a linear combination of the vectors stacked above each input word for each end task,
  - Combining the internal states in this manner allows for very rich word representations.
  - higher-level LSTM states ʻcontext-dependent aspects of word meaning (e.g., they can be used without modification to perform well on supervised word sense disambiguation tasks)
  - lower level states model aspects of syntax (e.g., they can be used to do part-of-speech tagging).
  - Simultaneously exposing all of these signals is highly beneficial, allowing the learned models select the types of semi-supervision that are most useful for each end task.



Deep Contextualized Word Representations



Semi-supervised approach:

- 1) a deep bidirectional RNN language model is pretrained on a large dataset
- 2) the vector of each word in a given input sentence is computed as a weighted sum of the RNN hidden states
- 1) is *unsupervised*, 2) weights are learned in a *supervised* way on some task-specific dataset

Graphic from: https://www.mihaileric.com/posts/deep-contextualized-word-representations-elmo/

$$\mathbf{ELMo}_k^{task} = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}$$

$h_{k,j}^{LM}$ is the k[th] hidden representation of the j[th] layer of the bi-RNN LM

s^{task} is the softmax weight vector, γ^{task} is a scaling parameter (for optimization)

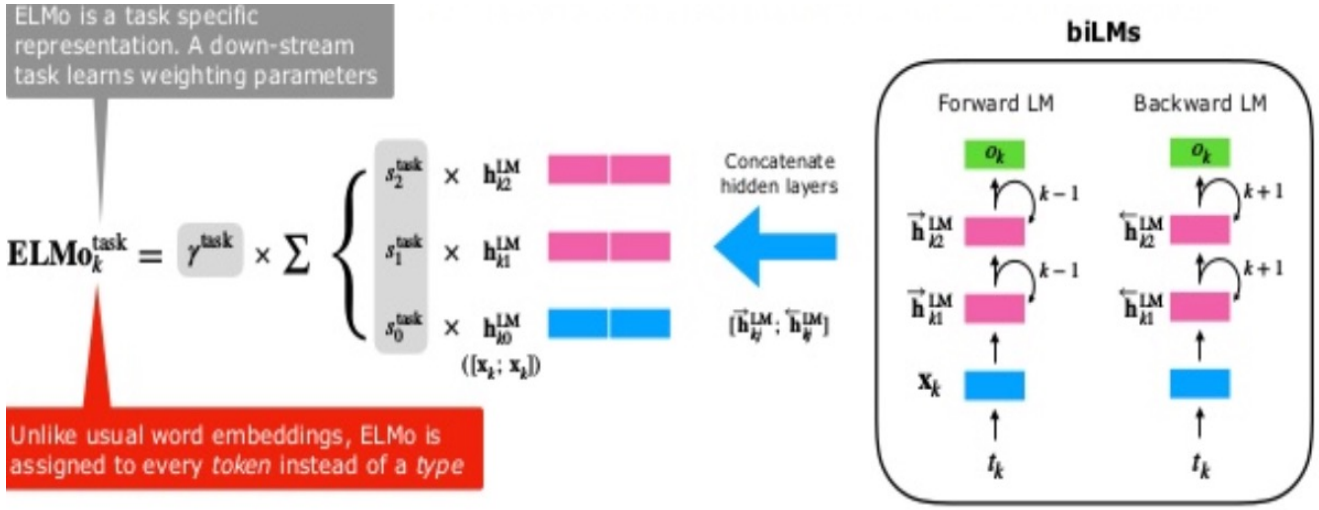
The authors use L=2 in all experiments

Deep Contextualized Word Representations

Use of ELMo in practice, on a task-specific dataset:

- 1) use the pretrained language model in prediction mode and store $h_{k,j}^{LM}$ for each word (each k) and each layer (each j)
- 2) concatenate $ELMo_k^{task}$ with the corresponding input vector* of whatever supervised model is used to solve the task (e.g., RNN, CNN, feed-forward…)
- 3) update s^{task} and γ^{task} with the other parameters of the supervised model during training

*given by word2vec, glove, etc.



Results:

ELMo improves over the baselines on 8 tasks, ranging from question answering to co-reference resolution, sentiment analysis, POS-tagging, and disambiguation

Deep Contextualized Word Representations

TASK	PREVIOUS SOTA		OUR BASELINE	ELMO + BASELINE	INCREASE (ABSOLUTE/ RELATIVE)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

Table 1: Test set comparison of ELMo enhanced neural models with state-of-the-art single model baselines across six benchmark NLP tasks. The performance metric varies across tasks – accuracy for SNLI and SST-5; F_1 for SQuAD, SRL and NER; average F_1 for Coref. Due to the small test sizes for NER and SST-5, we report the mean and standard deviation across five runs with different random seeds. The "increase" column lists both the absolute and relative improvements over our baseline.

SQuAD: Question answering, SNLI: Entailment, SRL: Semantic Role Labelling ("Who did what to whom"), NER: Name entity recognition, SST: Sentiment Analysis, Coref: task of clustering mentions in text that refer to the same underlying real world entities

Deep Contextualized Word Representations

Model	F_1
WordNet 1st Sense Baseline	65.9
Raganato et al. (2017a)	69.9
Iacobacci et al. (2016)	**70.1**
CoVe, First Layer	59.4
CoVe, Second Layer	*64.7*
biLM, First layer	67.4
biLM, Second layer	*69.0*

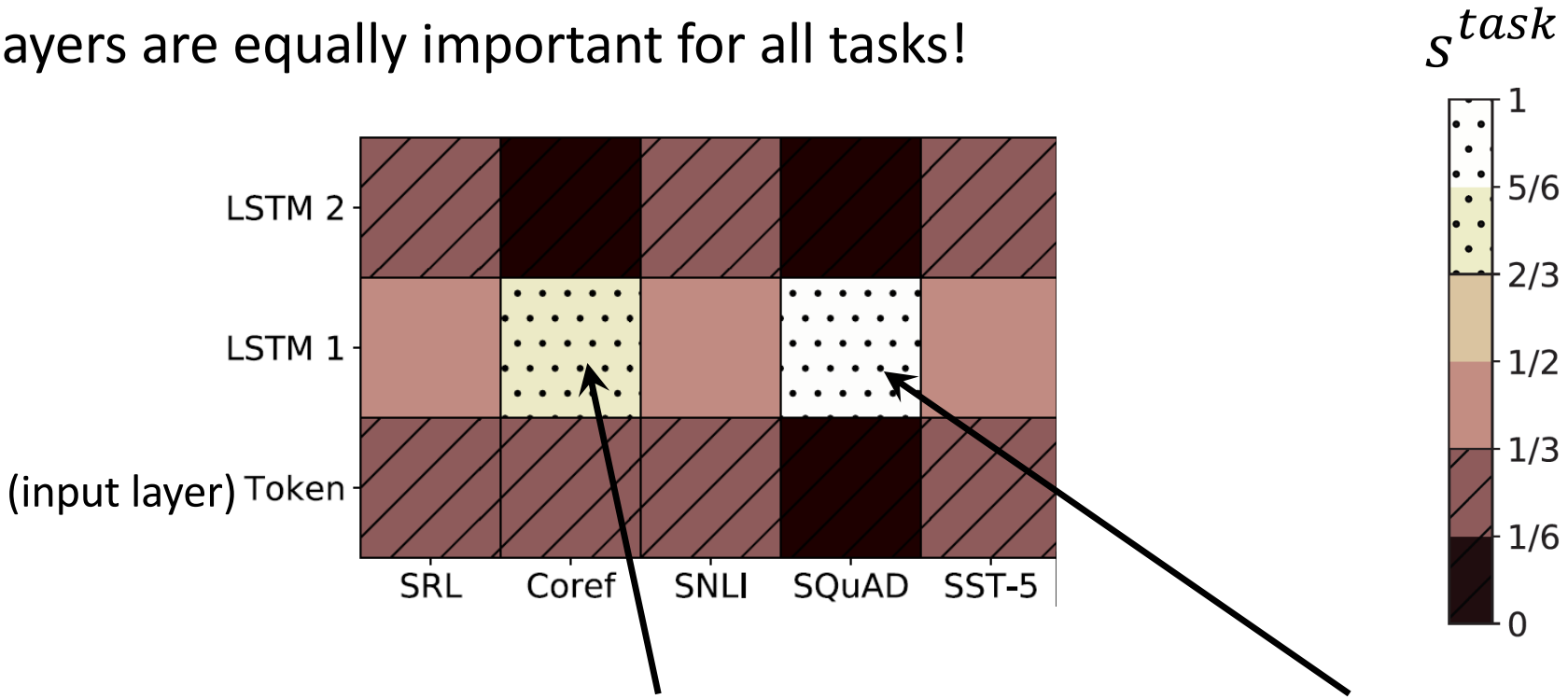
WSD

Model	Acc.
Collobert et al. (2011)	97.3
Ma and Hovy (2016)	97.6
Ling et al. (2015)	**97.8**
CoVe, First Layer	*93.3*
CoVe, Second Layer	92.8
biLM, First Layer	*97.3*
biLM, Second Layer	96.8

POS tagging

Deep Contextualized Word Representations

Not all layers are equally important for all tasks!



The 1st layer clearly dominates for co-reference resolution and question answering
For the other tasks, the weights are more evenly distributed among layers

ATTENTION BASED ARCHITECTURES

Attention is ubiquitous in DL today

Image captioning

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.

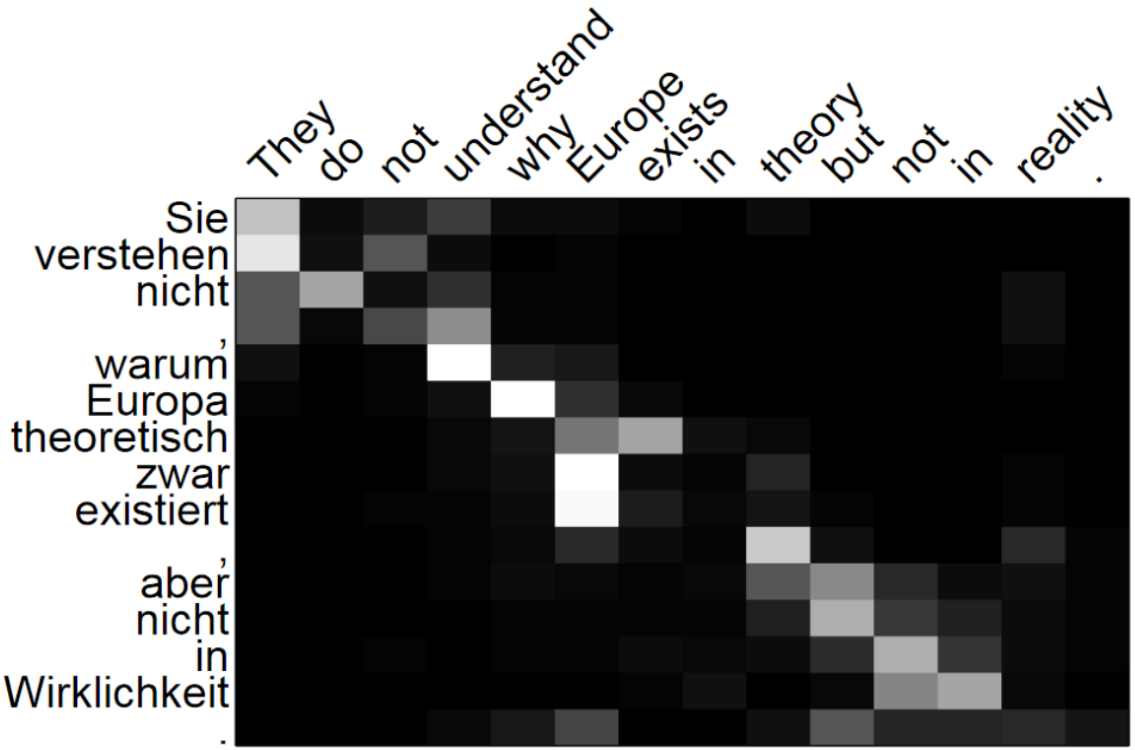
A group of people sitting on a boat in the water.

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

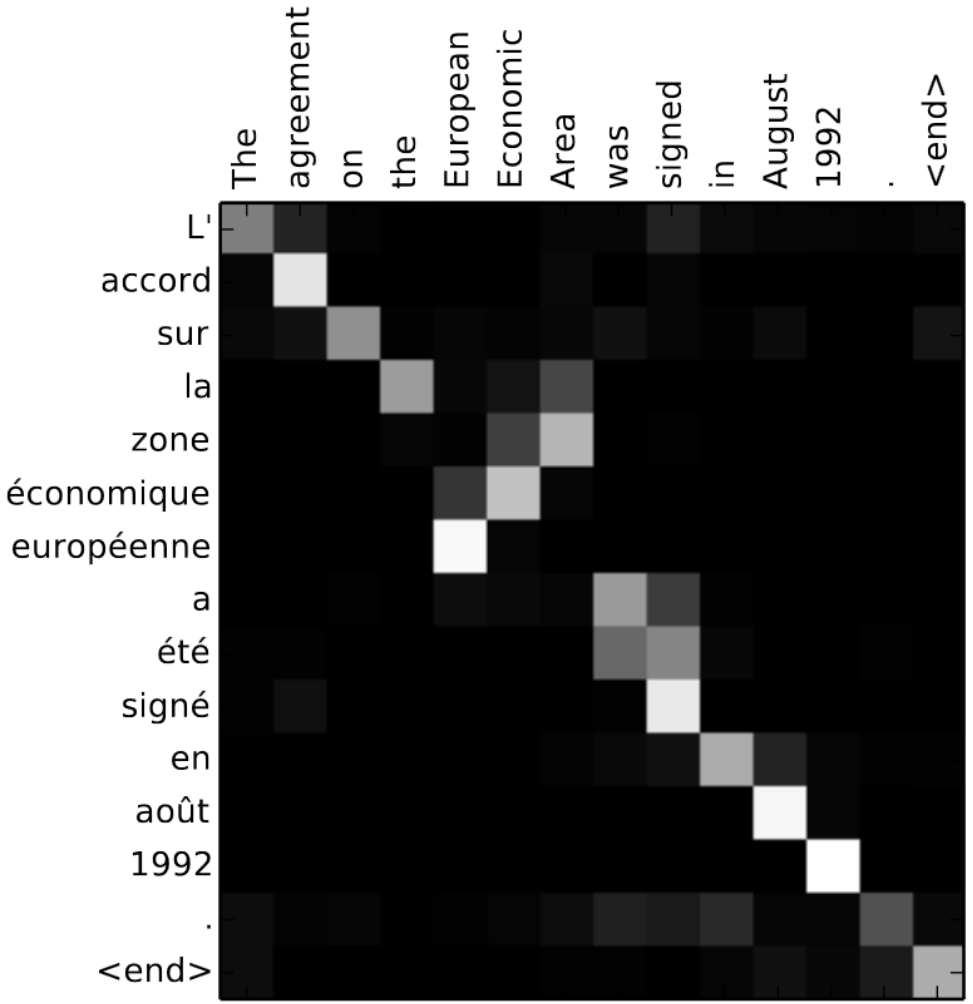
Show, attend and tell: Neural image caption generation with visual attention (Xu et al. 2015)

Attention is ubiquitous in DL today

Neural Machine Translation



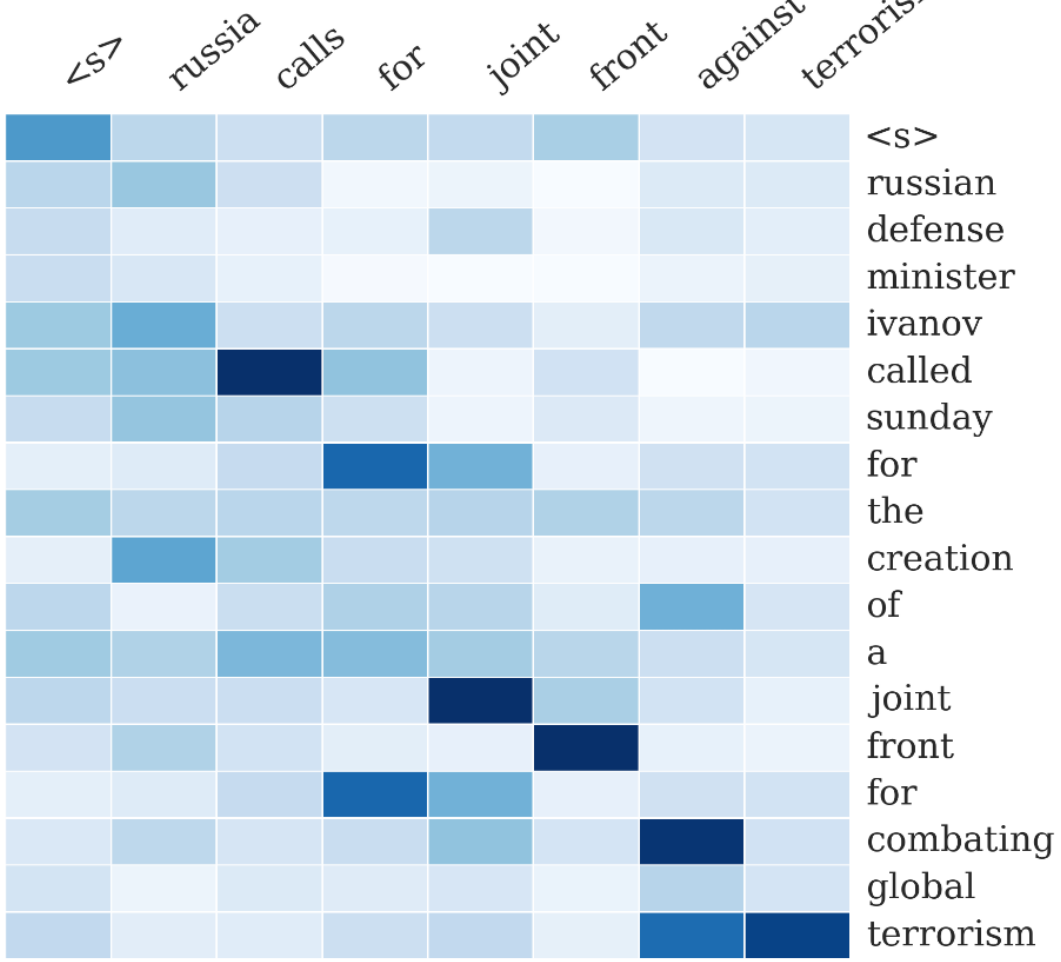
Effective approaches to attention-based neural machine translation (Luong et al. 2015)



Neural machine translation by jointly learning to align and translate (Bahdanau et al. 2014)

Attention is ubiquitous in DL today

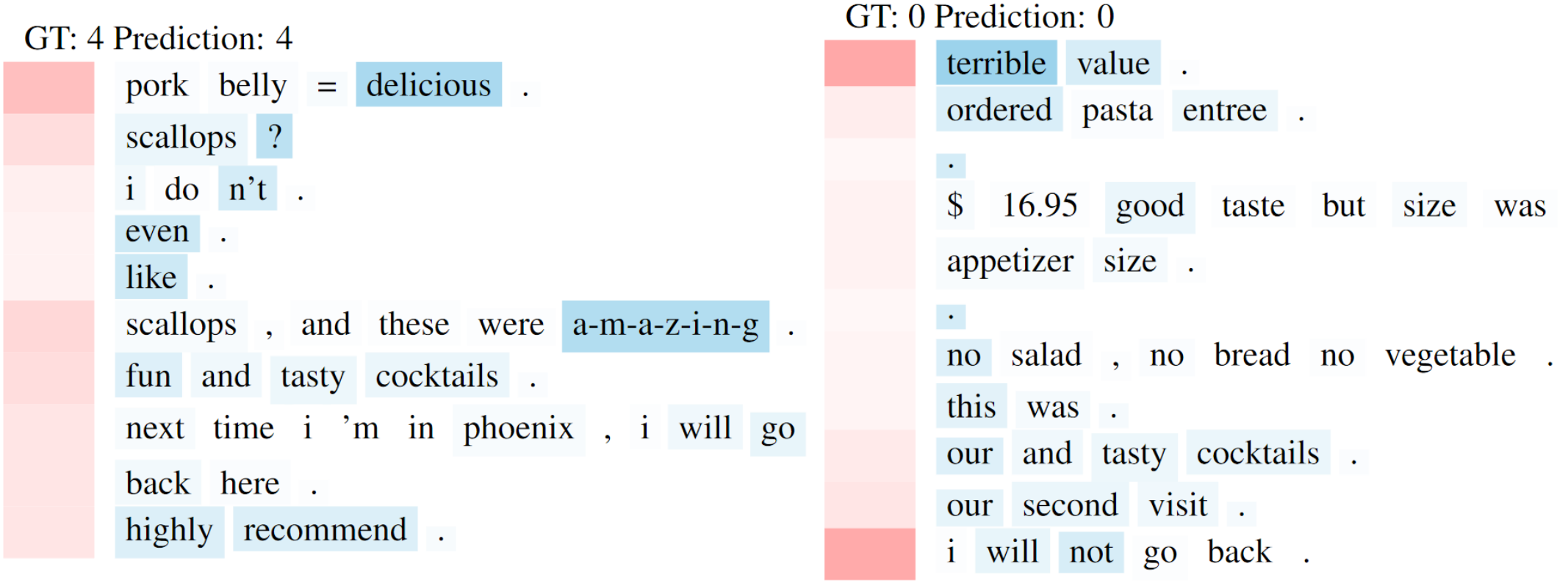
Abstractive summarization



A neural attention model for abstractive sentence summarization (Rush et al. 2015)

Attention is ubiquitous in DL today

Sentiment analysis



Hierarchical attention networks for document classification (Yang et al. 2016)

Encoder-Decoder Architectures

General idea:

encoder **decoder**

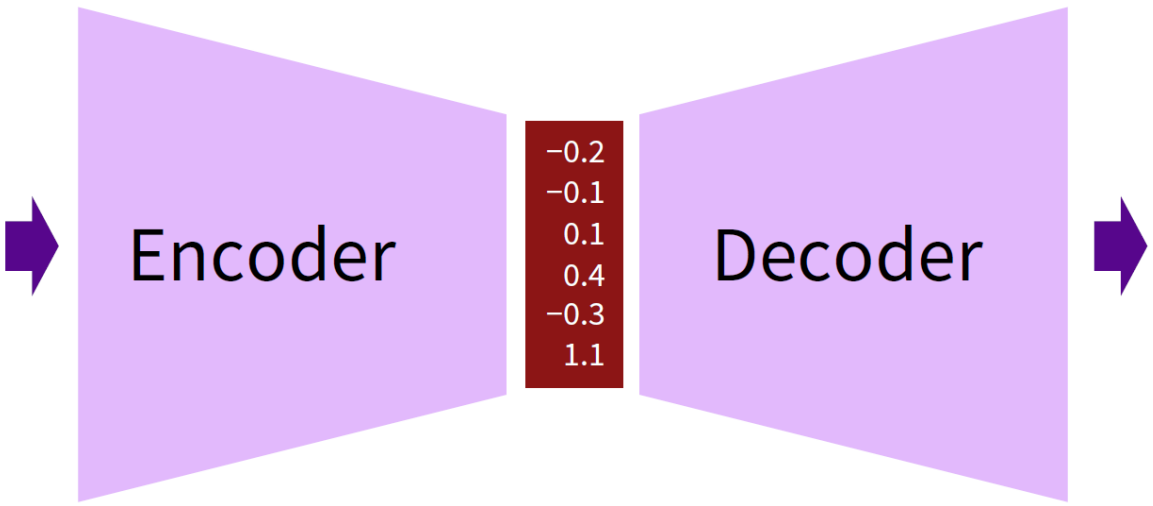
Input -> embedding -> Output

- image
- document
- source language
- …

- caption
- summary
- target language
- …

Known as *sequence-to-sequence* (seq2seq) when input and output are sequences (e.g., NLP applications)

The full architecture is differentiable => end-to-end training



What is attention?

Objective

- Traditional models (e.g., encoder) having to embed the input into a single fixed-length vector (lossy).
- Alternatively information can be kept and stored into multiple vectors.
- information can be retrieved later on (e.g., by the decoder). (Bahdanau et al. 2014)

Quick history:

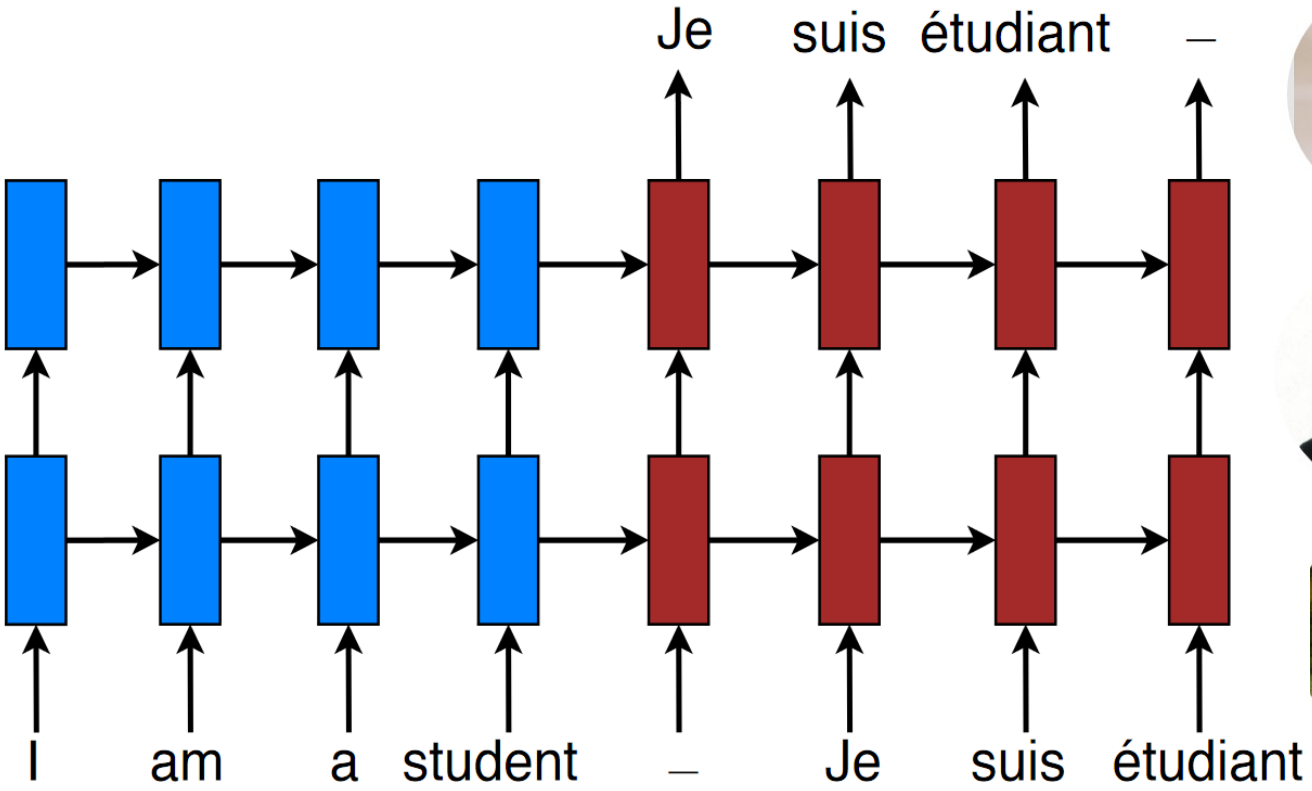
- developed in the context of **encoder-decoder** architectures for neural machine translation (Bahdanau et al. 2014)
- rapidly applied to naturally related tasks like image captioning (Xu et al. 2015) and summarization (*Luong et al. 2015*)
- also proposed for encoders only, e.g. for text classification (Yang et al. 2015) and representation learning (Conneau et al. 2017).

  Known as *self or inner attention* in such cases.

Encoder-Decoder for Neural Machine Translation

Encoder **Decoder**

***Target** sentence (generated)*



***Source** sentence (input)*

Minh-Thang Luong

Research Scientist at Google
Verified email at google.com - Homepage

Deep Learning Natural Language Processing

Hieu Pham

Carnegie Mellon University, Google Brain
Verified email at google.com

Machine Learning

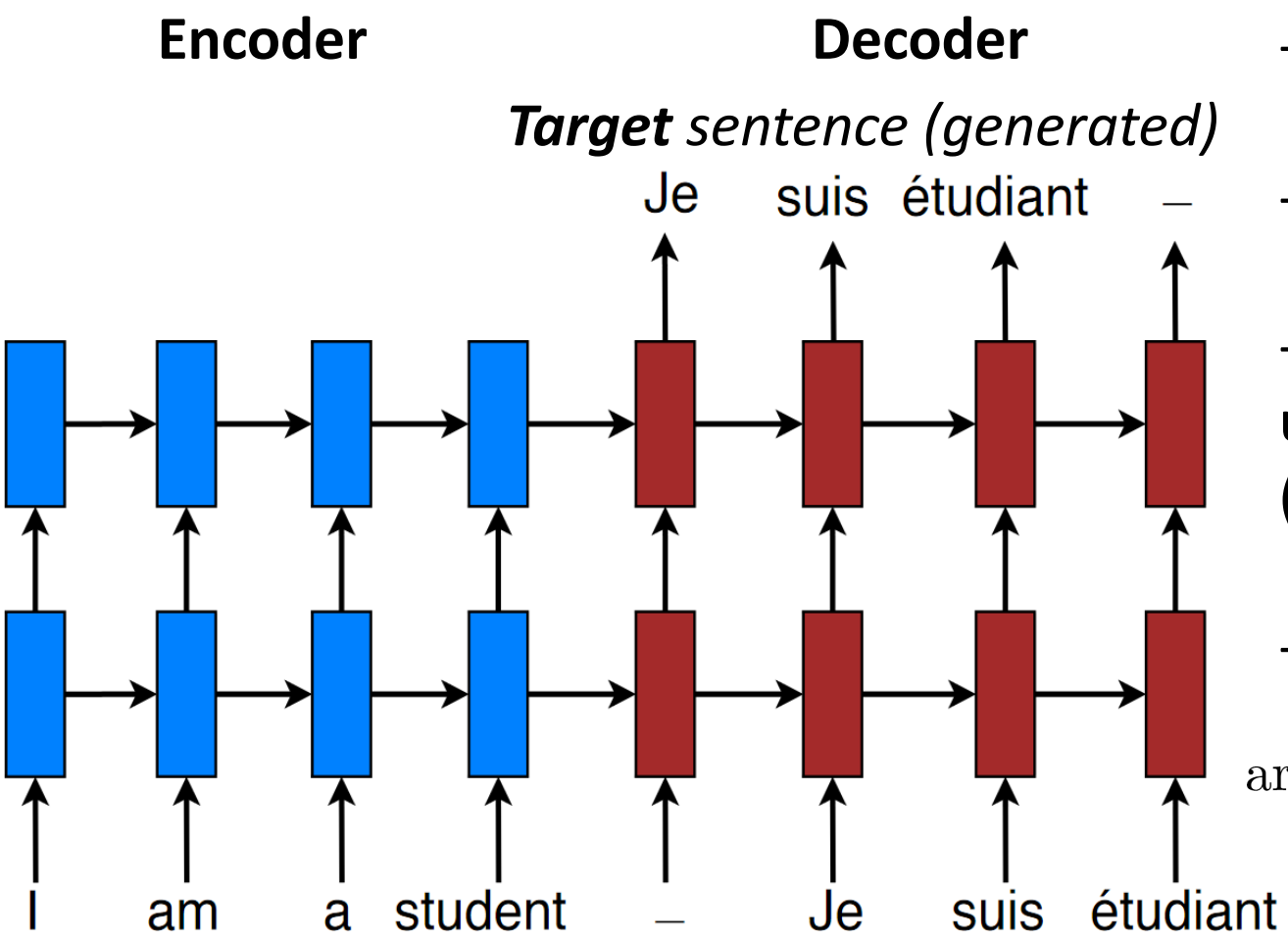
Christopher D Manning

Professor of Computer Science and Linguistics
Verified email at stanford.edu - Homepage

Natural Language Processing

Effective approaches to Attention-Based Neural Machine Translation (2015)

Encoder-Decoder for Neural Machine Translation



- source: $\left(x_1, \ldots, x_{T_x}\right)$

- target: $\left(y_1, \ldots, y_{T_y}\right)$

- Both encoder & decoder are unidirectional deep RNNs (a.k.a. *stacking* RNNs)

- Training objective:

$$\text{argmax}_\theta \left\{ \sum_{(x,y)\in \text{corpus}} \log\, p(y|x;\theta) \right\}$$

Luong et al. (2015)

Encoder-Decoder for Neural Machine Translation

Encoder: usually: CNN, stacking RNN* with LSTM or GRU units…
* unidirectional (Luong et al. 2015) or
bidirectional (Bahdanau et al. 2014).

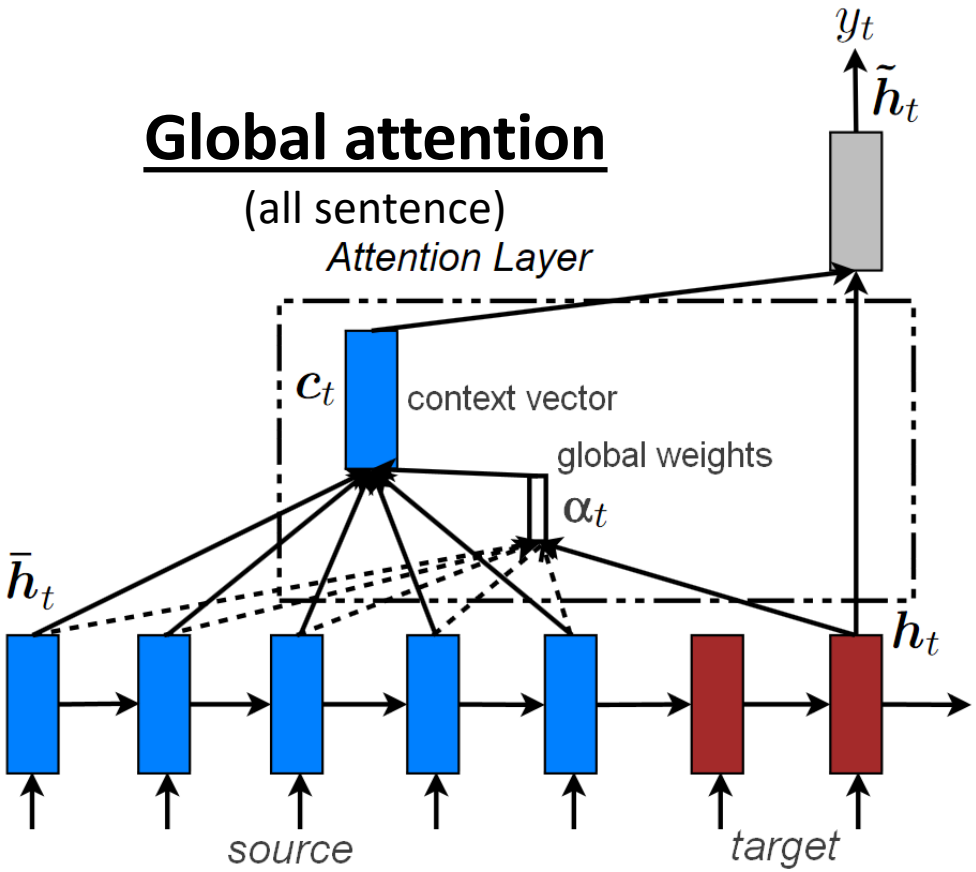
Decoder: unidirectional *RNN* (well suited to text generation), best if deep.

Generates the target sentence $\left(y_1, \cdots, y_{T_y}\right)$ one word at a time:

$$P\left[y_t | \{y_1, ..., y_{t-1}\}, c_t\right] = \text{softmax}\left(W_s \tilde{h}_t\right)$$

Luong et al. (2015)

Encoder-**Decoder** for Neural Machine Translation



$$P\left[y_t|\{y_1,\ldots,y_{t-1}\},c_t\right]=\mathrm{softmax}\left(W_s\tilde{h}_t\right)$$

attentional hidden state

$$\tilde{h}_t=\tanh\left(W_c\left[c_t;h_t\right]\right)$$

context vector

decoder hidden state

$$c_t=\sum_{i=1}^{T_x}\alpha_{t,i}\bar{h}_i$$

i^{th} encoder hidden state

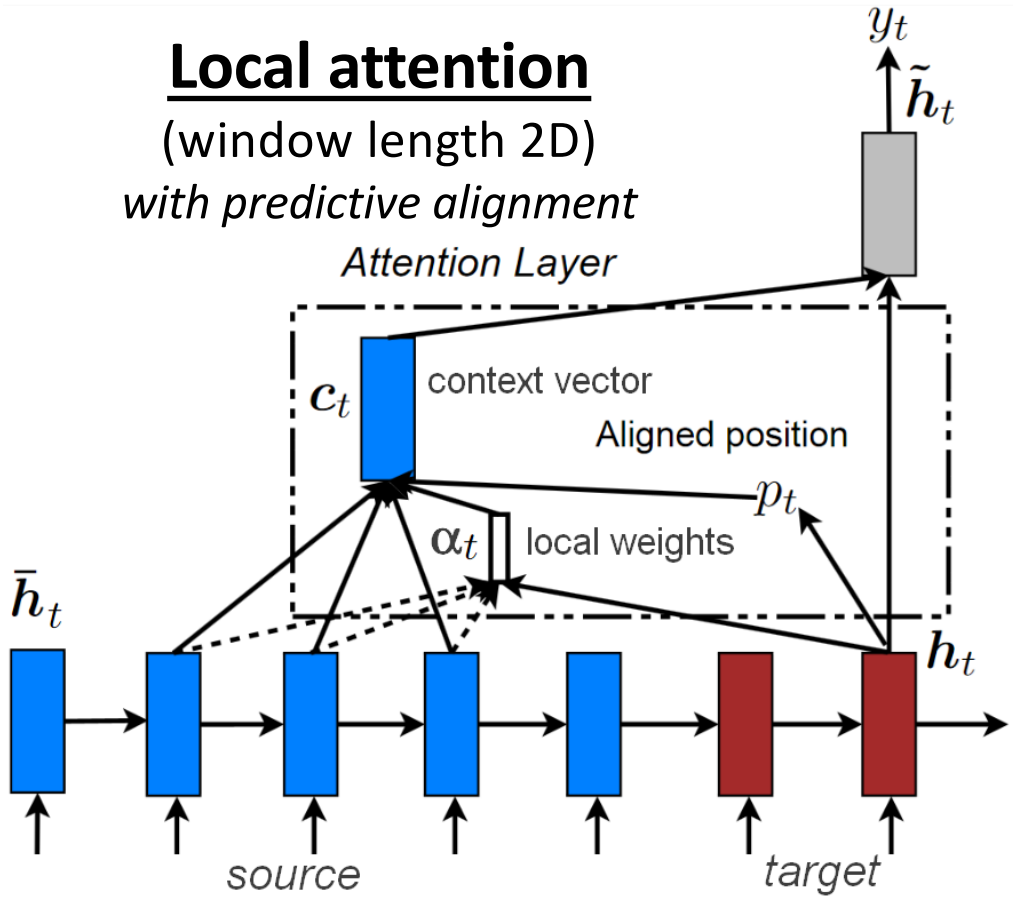
alignment vector

$$\alpha_{t,i}=\frac{\exp\left(\mathrm{score}\left(h_t,\bar{h}_i\right)\right)}{\sum_{i'=1}^{T_x}\exp\left(\mathrm{score}\left(h_t,\bar{h}_{i'}\right)\right)}$$

$$\mathrm{score}\left(h_t,\bar{h}_i\right)=h_t^{\top}\bar{h}_i$$

Luong et al. (2015)

Encoder-**Decoder** for Neural Machine Translation



$$P\left[y_t|\{y_1,...,y_{t-1}\},c_t\right] = \text{softmax}\left(W_s\tilde{h}_t\right)$$

attentional hidden state

$$\tilde{h}_t = \tanh\left(W_c\left[c_t; h_t\right]\right)$$

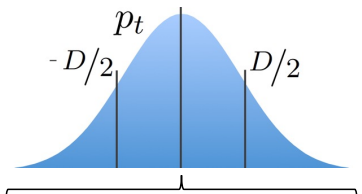
decoder hidden state

$$p_t = T_x \cdot \sigma\left(v_p^\top \tanh(W_p h_t)\right)$$

context vector

$$c_t = \sum_{i=p_t-D}^{p_t+D} \alpha_{t,i}\bar{h}_i$$

i^{th} encoder hidden state



alignment vector

$$\alpha_{t,i} = \frac{\exp\left(\text{score}(h_t,\bar{h}_i)\right)}{\sum_{i'=p_t-D}^{p_t+D}\exp\left(\text{score}(h_t,\bar{h}_{i'})\right)}\exp\left(-\frac{(i-p_t)^2}{2(D/2)^2}\right)$$

$$\text{score}(h_t,\bar{h}_i) = h_t^\top W_\alpha \bar{h}_i$$

Luong et al. (2015)

Encoder-Decoder for Neural Machine Translation

Facts:

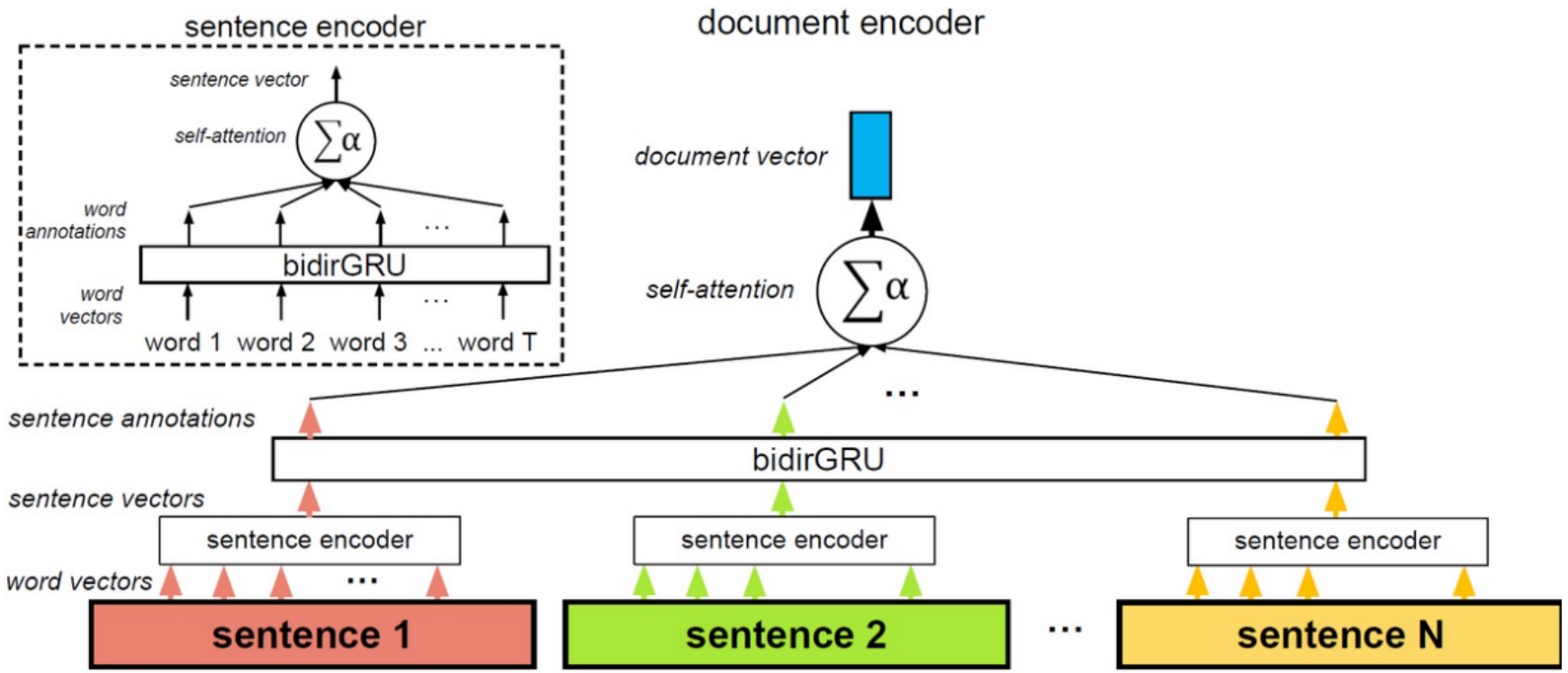
- Tested on the English <-> German task (WMT'14 dataset)
- 4.5M sentence pairs
- Encoder and decoder RNNs feature 4 layers of stacking and 1000-dimensional hidden states
- Window of size *D=10* for local attention
- Trained for 12 epochs (total of 7-10 days on a single GPU, at 1K target words/s)
- New state-of-the-art performance

Lessons learned:

- Local attention with predictive alignment gives better results than global attention
- Dot product ($\text{score}(h_t, \bar{h}_i) = h_t^\top \bar{h}_i$) works well for global attention
- The general formulation ($\text{score}(h_t, \bar{h}_i) = h_t^\top W_\alpha \bar{h}_i$) is better for local attention

Luong et al. (2015)

Self-attention for RNN encoders



Self-attention for RNN encoders

- Input is a sentence $(x_1, \ldots, x_T)$
- We're only interested in getting an embedding s of the sentence for some downstream task (e.g., classification)

encoder hidden state

context vector

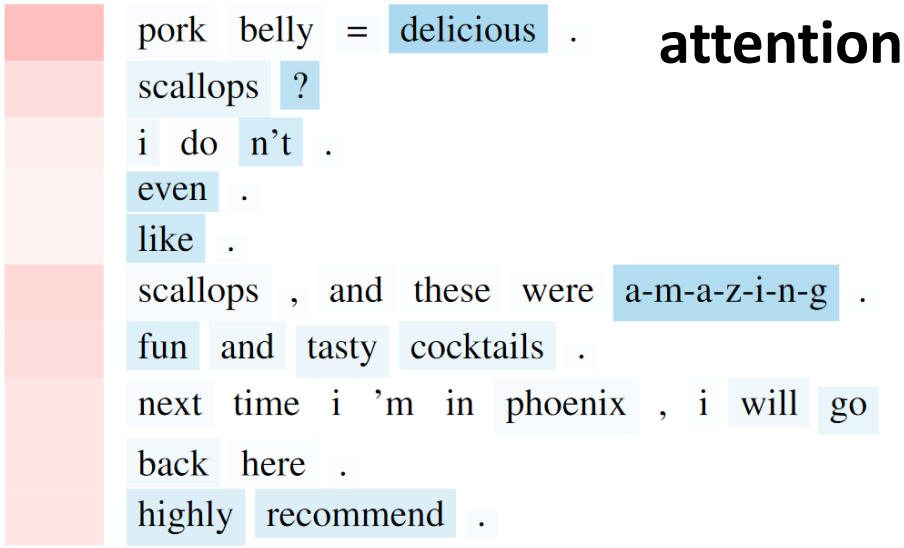
$$u_t = \tanh(W h_t)$$

$$\alpha_t = \frac{\exp(\text{score}(u_t, u))}{\sum_{t'=1}^{T} \exp(\text{score}(u_{t'}, u))}$$

$$s = \sum_{t=1}^{T} \alpha_t h_t$$

Where $\text{score}(u_t, u) = u_t^\top u$

The same process can be repeated over the sentence vectors s -> **hierarchical attention**



Example of self attention

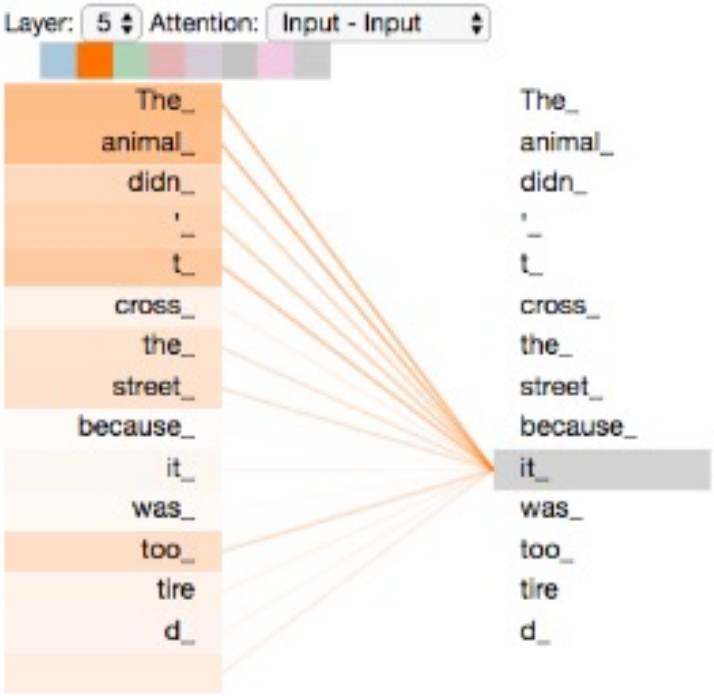
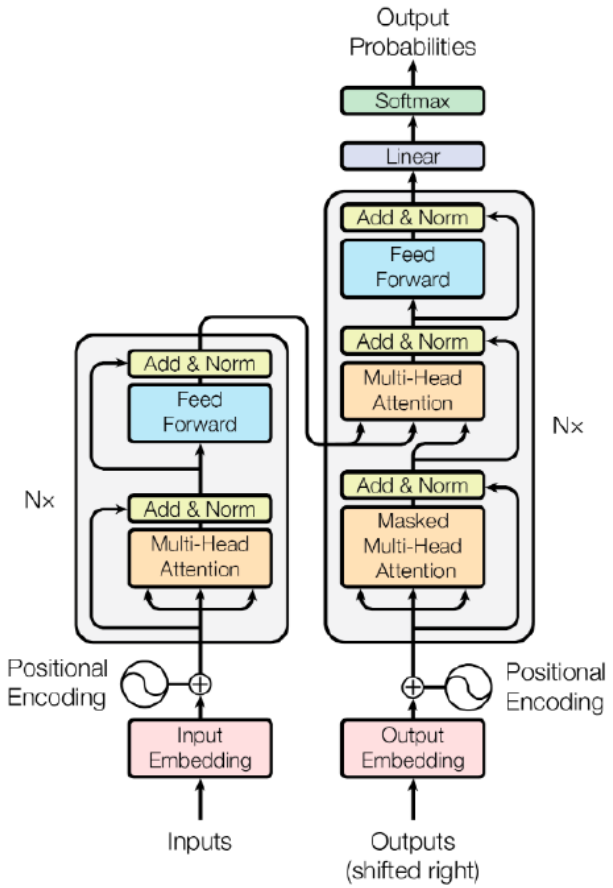


Figure: As we are encoding the word "it" in encoder #5 (the top encoder in the stack), part of the attention mechanism was focusing on "The Animal", and baked a part of its representation into the encoding of "it".

Attention is All You Need - Transformer

Transformer

- A model that follows the encoder-decoder structure, where the input tokens are mapped to a sequence of continuous representations, and these representations are consumed by the decoder which generates a sequence of outputs.
- Does not use recurrent neural networks or convolutional neural networks. Instead, it only uses dense and attention layers.



Attention Is All You Need, Ashish Vaswani, et. Al. https://arxiv.org/abs/1706.03762, Cited by 108083!!

Attention

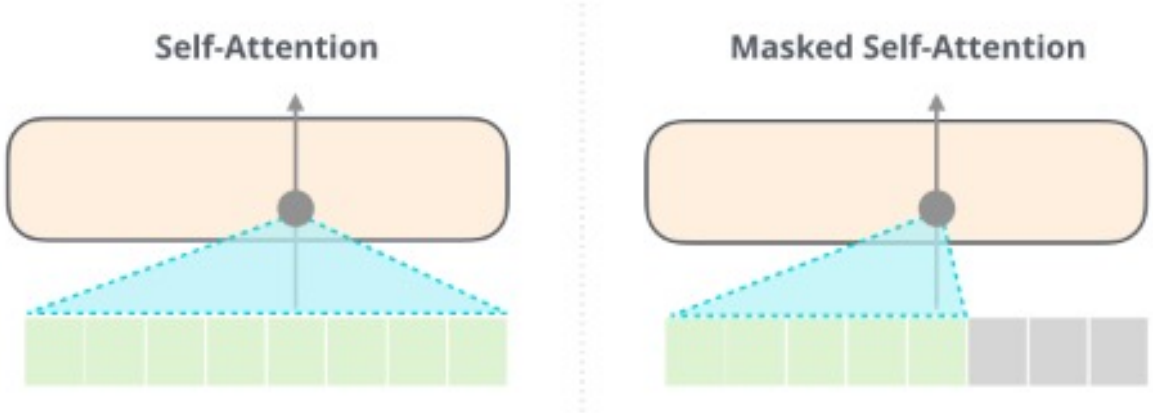


Figure: Self-attention of Encoder vs. Decoder.[1]

- Encoder-Decoder Attention (source-target alignment)

1) https://jalammar.github.io/illustrated-gpt2/

Transformer Architecture

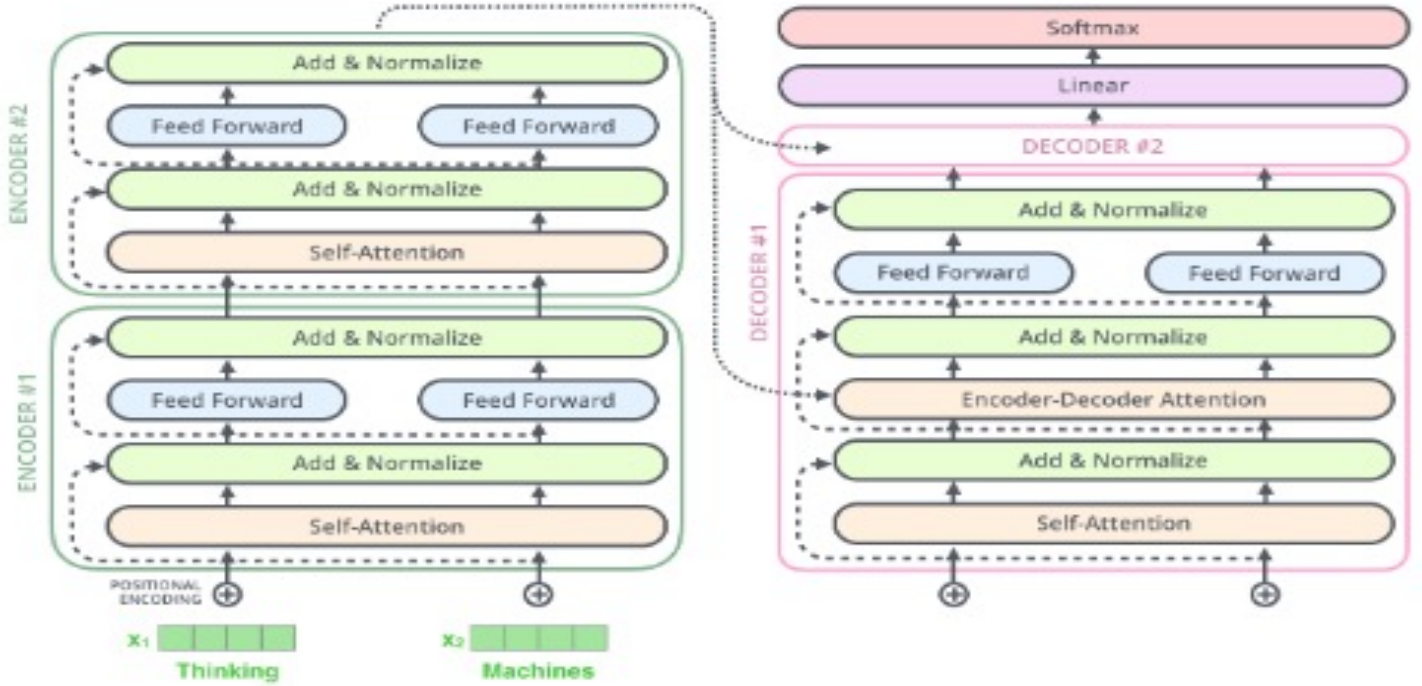


Figure: Transformer architecture.[1,2]

1) https://jalammar.github.io/illustrated-transformer/
2) http://nlp.seas.harvard.edu/annotated-transformer/

Postitional encoding

$$PE(pos, 2i) = sin(pos/10000^{2i/d_{model}})$$
$$PE(pos, 2i+1) = cos(pos/10000^{2i/d_{model}})$$

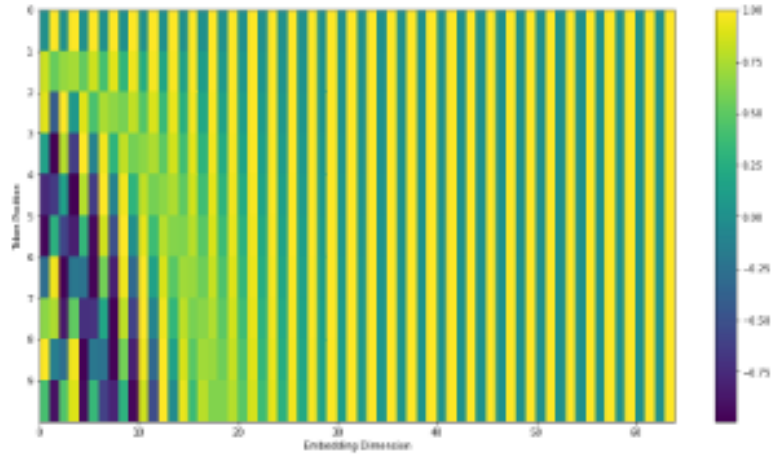


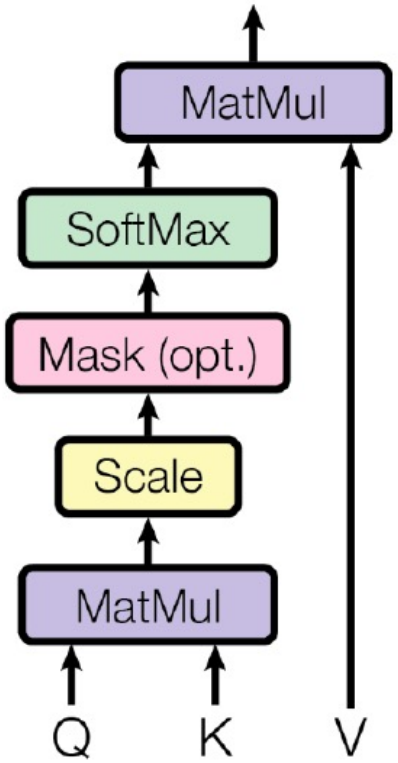
Figure: Sinusoid positional encoding.[1]

• RNN • Parallel computing • Learned positional embeddings

1) https://machinelearningmastery.com/a-gentle-introduction-to-positional-encoding-in-transformer-models-

Transformer – Scaled dot product attention

- A query Q_i representing the token at position i attend to a sequence of tokens (represented by K): $Q_i K^T$.
- The result scaled by $\sqrt{d_k}$ (dimension of Q_i) is passed to a softmax function to compute a score for each of the tokens: $softmax(\frac{Q_i K^T}{\sqrt{d_k}})$
- A weighted sum of the Value vectors is calculated as the new representation of the token at position i: $softmax(\frac{Q_i K^T}{\sqrt{d_k}})V$
- Queries are stacked in one matrix Q, the output of the attention layer becomes: $softmax(\frac{QK^T}{\sqrt{d_k}})V$



Query - Key - Value

computations of vector representation of tokens “thinking” and “machines” IN CONTEXT via attention.

- W^Q, W^K, W^V are trainable matrices
- q,k,v vectors corresponding to the tokens in the context.
- For “thinking”: q1*k1 is the self similarity for this word which is normalized and softmaxed. The result multiplied by v_1.

- z1 = sum (q1, k1, v1) that represent “thinking” in the context of “thinking”).
- Similar process happens among “Thinining” and “Machines” resulting in the z2 vector that represents after all computations the vector that corresponds to “thinking” in the context of “machines”
- The W^* matrices represent attention weights among the tokens in the context.. ?

Zij representation of word i to the head j – then we MERGE and give them to the next layer.

- K, W^iV are intialised randomly and are trainable – i stand for the head
- X1 : [“Thinking” embedding: positional embedding of “Thinking”]
- q1 = X1 * W^Q, k1 = X1 * W^K, v1 = X1 * W^V
- qi*kj : attention among tokens i and j. Then normalized by softmax
- z1 = sum(v1j): representation of token 1 in context, j: words in context
- zi ‘s concatenated and passed to the next attention layers…

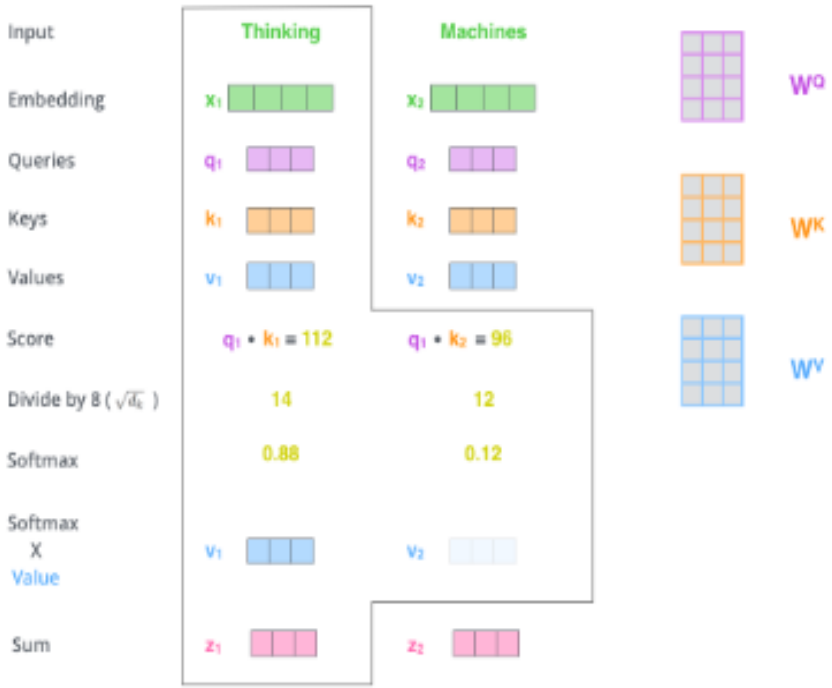


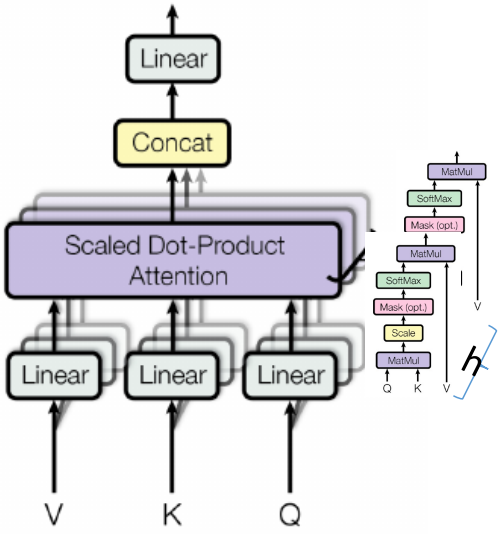
Figure: Self-attention calculation.

- Multi-Head Attention: $\{W_0^Q, W_0^K, W_0^V\}, \{W_1^Q, W_1^K, W_1^V\}, \ldots$

Transformer – Multihead Attention

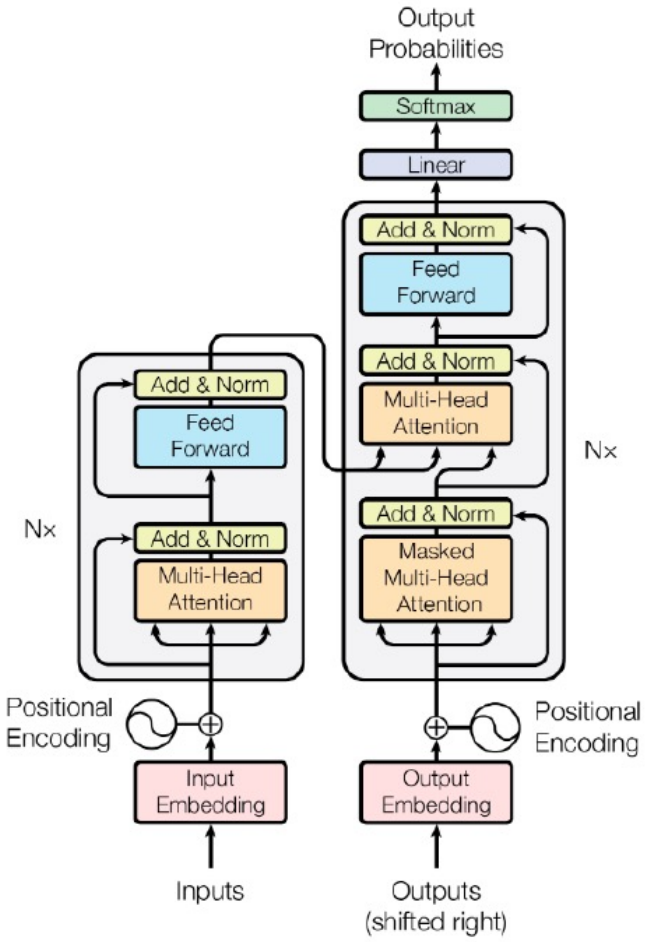
- Q, K and V are linearly projected h times.
- Scaled dot-product attention is applied h times on the projections to produce h heads:
  $head_j = Attention(QW_j^Q, KW_j^K, VW_j^V)$
- heads are concatenated and linearly projected to produce the new representation:

$$MultiHead(Q, K, V) = Concat(head_i, ..., head_h)W$$



Transformers- Attention

- **Self-attention:** Applied in the encoder and in the decoder:
  - In the encoder: Q, K and V come from the previous encoder sub-layer.
  - In the decoder: Q, K and V come from the previous decoder layer and a mask is applied to prevent the decoder from attending future positions.
- **Encoder-decoder attention:** Applied only in the decoder. Q come from the previous decoder layer. K and V come from the output representations of the encoder.



Transformer - experiments

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost

Similarity Metrics for Text Generation Evaluation

Goal: Automatically evaluate semantic equivalence between candidate and reference text

Text Generation tasks

- Summarization
- Machine Translation
- Dialog Response Generation
- Paraphrasing
- Question Generation
- ……

Similarity Metrics for Text Generation Evaluation

- **N-gram Overlap Metrics**
  - Precision, Recall, F-Measure
  - BLEU
  - ROUGE
  - METEOR
- **Alignment-based Metrics**
  - Word Mover's Distance
  - Sentence Mover's similarity
  - Bertscore & Moverscore
- **Supervised Metrics**
  - RUSE
  - BLEURT

Precision, Recall, F-Measure

SYSTEM A: Israeli officials ~~responsibility~~ ~~of~~ airport ~~safety~~

REFERENCE: Israeli officials are responsible for airport security

- Precision

$$\frac{\textit{correct}}{\textit{output-length}} = \frac{3}{6} = 50\%$$

- Recall

$$\frac{\textit{correct}}{\textit{reference-length}} = \frac{3}{7} = 43\%$$

- F-measure

$$\frac{\textit{precision} \times \textit{recall}}{(\textit{precision} + \textit{recall})/2} = \frac{.5 \times .43}{(.5 + .43)/2} = 46\%$$

BLEU (Bilingual Evaluation Understudy)

- N-gram precision between candidate and reference text : *How much the n-grams in the candidate text appeared in the reference text*
- Compute precision for n-grams of size 1 to 4
- Add brevity penalty (for too short texts)

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}$$

$$BLEU = min(1, \frac{output-length}{reference-length})(\prod_{n=1}^{4} p_n)^{\frac{1}{4}}$$

"Bleu: a method for automatic evaluation of machine translation." (2002)

BLEU (Bilingual Evaluation Understudy)

SYSTEM A: Israeli officials (2-GRAM MATCH) responsibility of airport (1-GRAM MATCH) safety

REFERENCE: Israeli officials are responsible for airport security

SYSTEM B: airport security (2-GRAM MATCH) Israeli officials are responsible (4-GRAM MATCH)

Metric	System A	System B
precision (1gram)	3/6	6/6
precision (2gram)	1/5	4/5
precision (3gram)	0/4	2/4
precision (4gram)	0/3	1/3
brevity penalty	6/7	6/7
BLEU	0%	52%

ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

- N-gram recall between candidate and reference text : *How much the n-grams in the reference text appeared in the candidate text*
- ROUGE-L uses longest common subsequence
- ROUGE-S measures skip-bigram based co-occurrence statistics
- ROUGE-SU measures skip-bigram based and unigram-based co-occurrence statistics
- ROUGE-N generalises the above

$$\text{ROUGE-N} = \frac{\sum_{S \in \{ReferemceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \quad (1)$$

"ROUGE: A Package for Automatic Evaluation of Summaries" (2004)

ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

The quick brown fox jumped over the lazy dog. 

The quick brown dog jumped over the lazy fox. 

HIGH ROUGE L F score: 77

Semantically Inaccurate

The quick brown fox jumped over the lazy dog. 

The fast wood-coloured fox hopped over the lethargic dog. 

LOWER ROUGE L F score: 55

Semantically Accurate

METEOR
(Metric for Evaluation of Translation with Explicit Ordering)

- Enables flexible matching: *matching stems and synonyms*

$$Fmean = \frac{10PR}{R + 9P}$$

Partial credit for

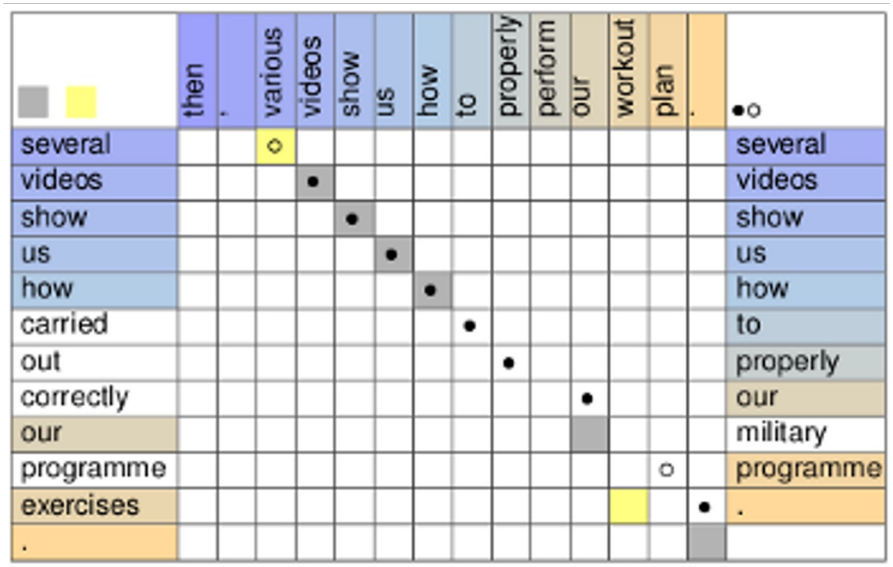
- Harmonic mean of unigram precision and recall with recall weighted higher than precision
- Fragmentation penalty captures how well-ordered the matched words are

$$Score = Fmean * (1 - Penalty)$$

"METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments" (2005)

METEOR
(Metric for Evaluation of Translation with Explicit Ordering)

- Yellow: synonym/stem (matched using the Porter stemmer or a synonym dictionary - rule-based)
- Grey: exact match
- Black dot - exact match ,
- white dot - synonym/stem.



Segment 2001

P:	0.650	vs	0.855	:	0.205
R:	0.578	vs	0.689	:	0.111
Frag:	0.522	vs	0.472	:	-0.051
Score:	0.281	vs	0.375	:	0.094

Word Mover's Distance

- Text dissimilarity = *Minimum amount of distance the embedded words of one document need to "travel" to reach the embedded words of another document*
- Based on the well-known concept of Earth Mover's Distance
- Uses the knowledge encoded within the word embedding space

$$\min_{\mathbf{T} \geq 0} \sum_{i,j=1}^{n} \mathbf{T}_{ij} c(i,j)$$

$$\text{subject to:} \sum_{j=1}^{n} \mathbf{T}_{ij} = d_i \quad \forall i \in \{1, \ldots, n\}$$

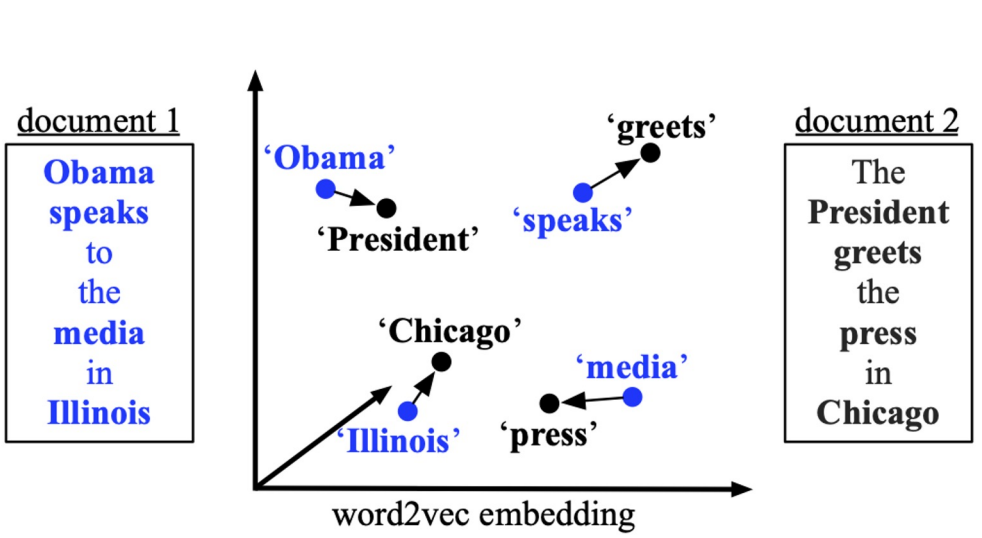
$$\sum_{i=1}^{n} \mathbf{T}_{ij} = d'_j \quad \forall j \in \{1, \ldots, n\}.$$

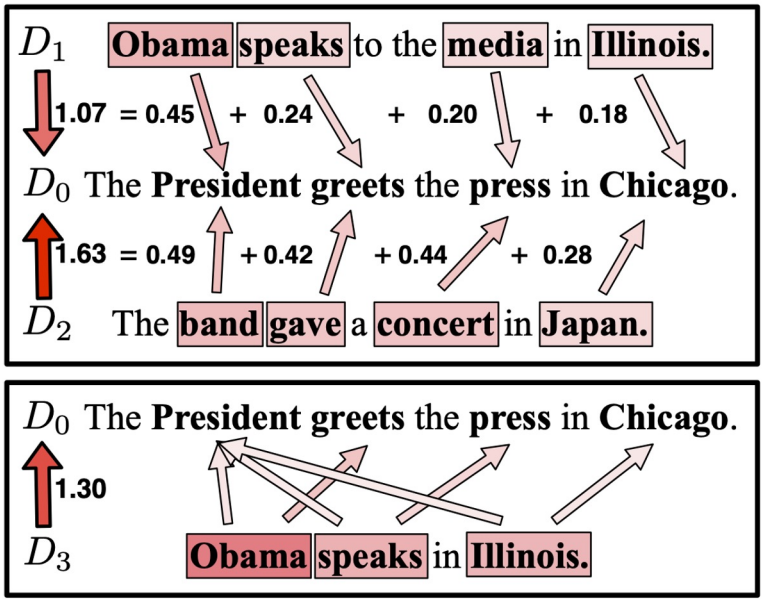
c (i, j) : *cost* associated with "traveling" from word i to word j

d and *d'* : nBOW representation of two text documents in the *(n – 1)*-simplex

"From Word Embeddings To Document Distances" (2015)

Word Mover's Distance



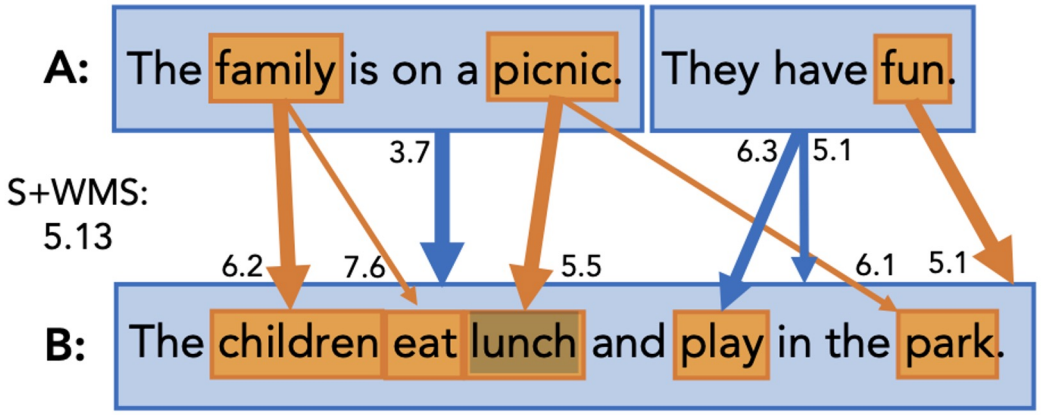


All non-stop words (***bold***) of both documents are embedded into a *word2vec* space. The distance between the two documents is the minimum cumulative distance that all words in document 1 need to travel to exactly match document 2.

(Top:) The components of the WMD metric between a query D0 and two sentences D1 , D2 (with equal BOW distance). The arrows represent flow between two words and are labeled with their distance contribution. *(Bottom:)* The flow between two sentences D3 and D0 with different numbers of words. This mismatch causes the WMD to move words to multiple similar words.

Sentence Mover's Similarity

- Based on Word Mover's Distance
- Represents each document as sentences or both words and sentences
- Weight each sentence according to its length



This metric finds the minimal cost of "moving" both the word embeddings (orange) and the sentence embeddings (blue) in Document A to those in Document B. An arrow's width is the proportion of the embedding's weight being moved, and its label is the Euclidean distance. Here we show only the highest weighted connections.

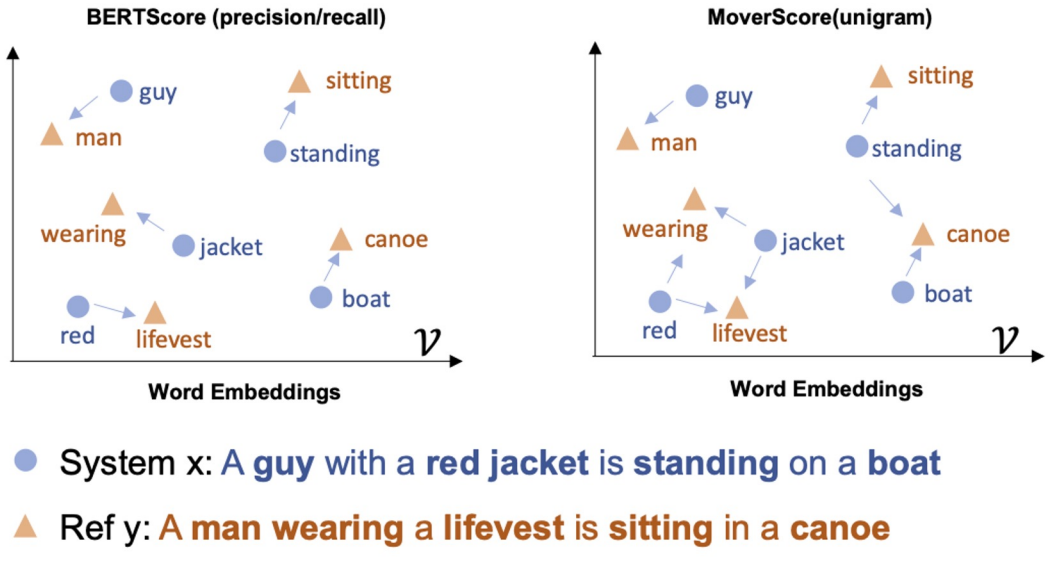
"Sentence Mover's Similarity: Automatic Evaluation for Multi-Sentence Texts" (2019)

BertScore

- A non-optimized MoverScore
- Greedy matching instead of optimal matching
- Hard alignments instead of soft alignments

MoverScore

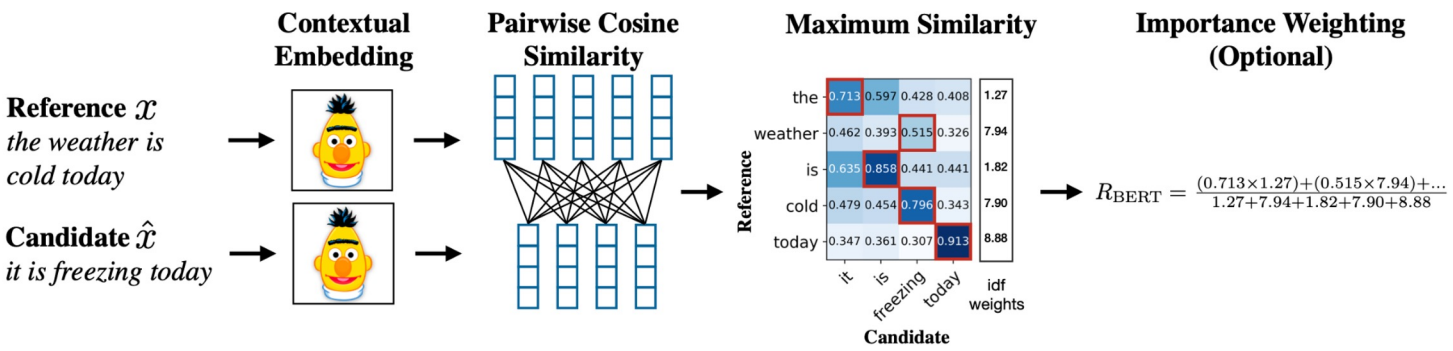
- Based on Word Mover's Distance and Sentence Mover's Similarity
- Leverages pretrained contextual embeddings (ELMO and BERT)



"BertScore: Evaluating Text Generation with BERT" (2019)

"MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover" Distance (2019)

BertScore



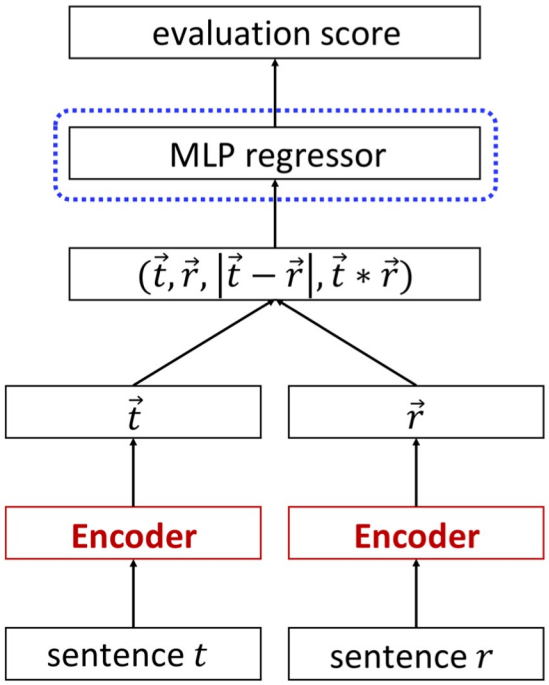
reference sentence $x = \langle x_1, \ldots, x_k \rangle$ candidate sentence $\hat{x} = \langle \hat{x}_1, \ldots, \hat{x}_l \rangle$

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \quad , \quad P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \quad , \quad F_{\text{BERT}} = 2\frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \quad .$$

- Importance weighting $R_{\text{BERT}} = \frac{\sum_{x_i \in x} \text{idf}(x_i) \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j}{\sum_{x_i \in x} \text{idf}(x_i)}$
- Baseline Rescaling $\hat{R}_{\text{BERT}} = \frac{R_{\text{BERT}} - b}{1 - b}$

RUSE
(Regressor Using Sentence Embeddings)

- Learned metric - *supervised*
- Encode sentences with universal sentence embeddings
- Supervised end-to-end regression model
- Trained on human ratings (evaluation score)

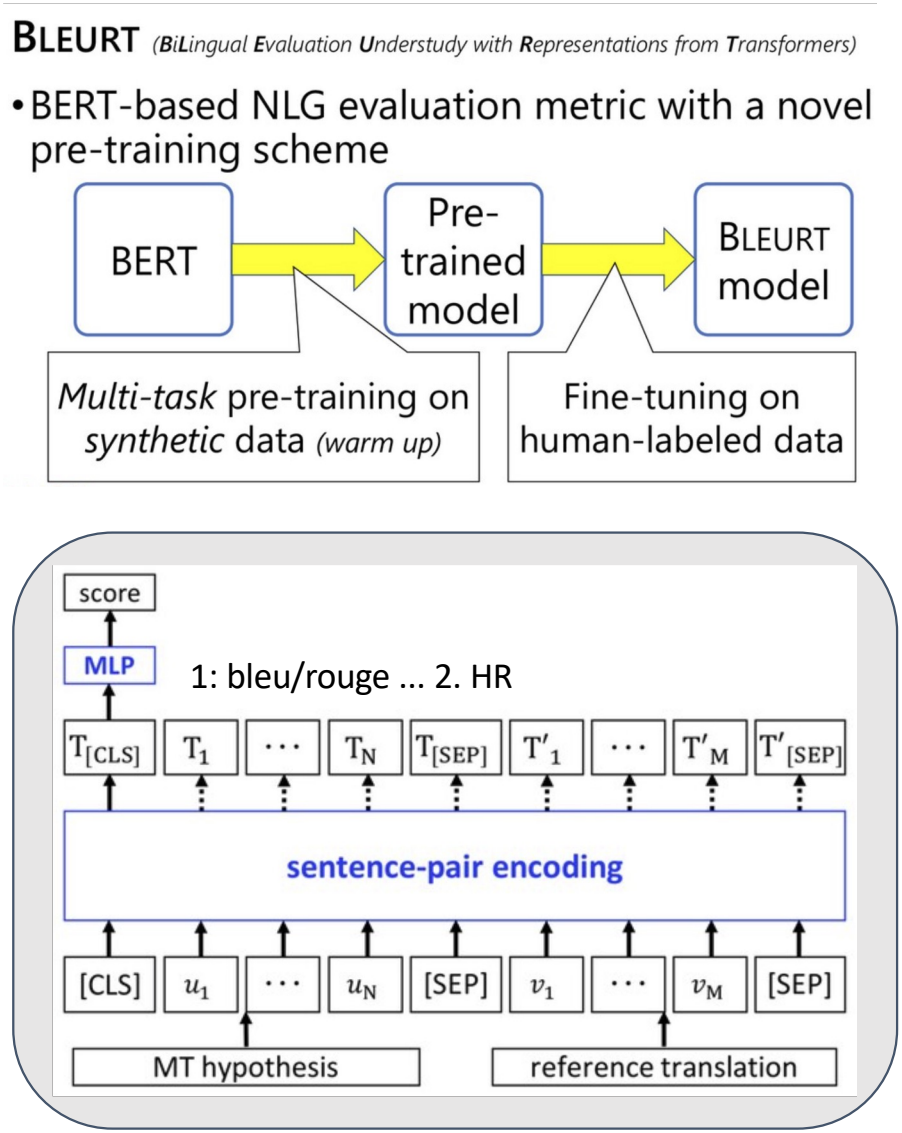


"RUSE: Regressor Using Sentence Embeddings for Automatic Machine Translation Evaluation" (2018)

BLEURT

- Learned metric - *supervised*
- Regression model using BERT embeddings
- Pretraining on synthetic data using automatic metrics (BLEU, ROUGE, BertScore) as labels
- Further fine-tuning with reference text and human ratings (HR)
- Combines expressivity and robustness

"BLEURT: Learning Robust Metrics for Text Generation" (2020)



Evaluation of Similarity Metrics

Similarity metrics are evaluated by its **correlation** with **human judgements**

An example : Text summarization

Correlation

- Pearson r : measure of linear correlation

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

- Spearman ρ : measure of rank correlation

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

Human judgements

- Pyramid score: how many important semantic content units in the reference summaries are covered by the system summary
- Responsiveness score: how well a summary responds to the overall quality combining both content and linguistic quality

Pyramid Score : A Human Annotated Metric for Summarization Evaluation

A1. The industrial espionage case involving GM and VW began with the hiring of Jose Ignacio Lopez, an employee of GM subsidiary Adam Opel, by VW as a production director.

B3. However, he left GM for VW under circumstances, which along with ensuing events, were described by a German judge as "potentially the biggest-ever case of industrial espionage".

C6. He left GM for VW *in March, 1993.*

D6. The issue stems from the alleged recruitment of GM's eccentric and visionary Basque-born procurement chief Jose Ignacio Lopez de Arriortura and seven of Lopez's buisness colleagues.

E1. On March 16, 1993, with Japanese car import quotas to Europe expiring in two years, renowned cost-cutter, Agnacio Lopez De Arriortua, left his job as head of purchasing at General Motor's Opel, Germany, to become Volkswagen's Purchasing and Production director.

F3. In March 1993, Lopez and seven other GM executives moved to VW overnight.

SCU1 (w=6): *Lopez left GM for VW*
A1. the hiring of Jose Ignacio Lopez, an employee of GM ... by VW
B3. he left GM for VW
C6. He left GM for VW
D6. recruitment of GM's ... Jose Ignacio Lopez
E1. Agnacio Lopez De Arriortua, left his job ... at General Motor's Opel ... to become Volkswagen's ... director
F3. Lopez ... GM ... moved to VW

SCU2 (w =3) *Lopez changes employers in March 1993*
C6. in March, 1993
E1. On March 16, 1993
F3. In March 1993

- Summary content units (SCUs):
- units of meaning, annotated by experts
- notes information that is repeated across different reference summaries for the same input
- Weights are associated with each SCU indicating the number of summaries in which it appeared
- Example: Emergence of two SCUs from six human abstracts

Pyramid Score : A Human Annotated Metric for Summarization Evaluation

- SCUs are partitioned in a pyramid based on their weights
- In descending tiers, SCUs become less important informationally
- For an optimal summary, an SCU from tier (n-1) should not be expressed if all the SCUs in tier n have not been expressed
- Informativeness of a new summary : ratio of the sums of the weights of its SCUs to the weight of an optimal summary with the same number of SCUs

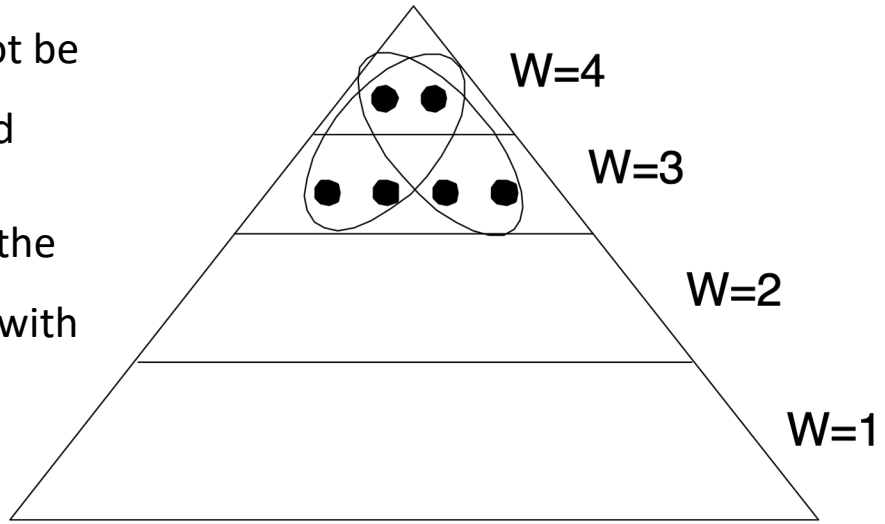


Fig. 1. Two of six optimal summaries with 4 SCUs

The pyramid method: incorporating human content selection variation in summarization evaluation (2007)

Experimental Results - Unsupervised Metrics

TAC 2008 and TAC 2009 : summarization datasets with news articles

Setting	Metrics	TAC-2008 Responsiveness r	TAC-2008 Responsiveness ρ	TAC-2008 Pyramid r	TAC-2008 Pyramid ρ	TAC-2009 Responsiveness r	TAC-2009 Responsiveness ρ	TAC-2009 Pyramid r	TAC-2009 Pyramid ρ
BASELINES	S^3_{best} (*)	0.715	0.595	0.754	0.652	0.738	**0.595**	**0.842**	**0.731**
	ROUGE-1	0.703	0.578	0.747	0.632	0.704	0.565	0.808	0.692
	ROUGE-2	0.695	0.572	0.718	0.635	0.727	0.583	0.803	0.694
	BERTSCORE-F1	0.724	0.594	0.750	0.649	0.739	0.580	0.823	0.703
SENT-MOVER	SMD + W2V	0.583	0.469	0.603	0.488	0.577	0.465	0.670	0.560
	SMD + ELMO + PMEANS	0.631	0.472	0.631	0.499	0.663	0.498	0.726	0.568
	SMD + BERT + PMEANS	0.658	0.530	0.664	0.550	0.670	0.518	0.731	0.580
	SMD + BERT + MNLI + PMEANS	0.662	0.525	0.666	0.552	0.667	0.506	0.723	0.563
WORD-MOVER	WMD-1 + W2V	0.669	0.549	0.665	0.588	0.698	0.520	0.740	0.647
	WMD-1 + ELMO + PMEANS	0.707	0.554	0.726	0.601	0.736	0.553	0.813	0.672
	WMD-1 + BERT + PMEANS	0.729	0.595	0.755	0.660	0.742	0.581	0.825	0.690
	WMD-1 + BERT + MNLI + PMEANS	**0.736**	**0.604**	**0.760**	**0.672**	**0.754**	0.594	0.831	0.701
	WMD-2 + BERT + MNLI + PMEANS	0.734	0.601	0.752	0.663	0.753	0.586	0.825	0.694

Table 2: Pearson r and Spearman ρ correlations with summary-level human judgments on TAC 2008 and 2009.

Experimental Results - Supervised Metrics

Tides 2003 and WMT18 : machine translation datasets

System ID	Correlation
BLEU	0.817
NIST	0.892
Precision	0.752
Recall	0.941
F1	0.948
Fmean	0.952
METEOR	0.964

Table 1: Pearson *r* correlations with human judgments over the Chinese portion of the Tides 2003 dataset

model	cs-en τ / DA	de-en τ / DA	et-en τ / DA
sentBLEU	20.0 / 22.5	31.6 / 41.5	26.0 / 28.2
BERTscore w/ BERT	29.5 / 40.0	39.9 / 53.8	34.7 / 39.0
BERTscore w/ roBERTa	31.2 / 41.1	42.2 / 55.5	37.0 / 40.3
Meteor++	22.4 / 26.8	34.7 / 45.7	29.7 / 32.9
RUSE	27.0 / 34.5	36.1 / 49.8	32.9 / 36.8
YiSi1	23.5 / 31.7	35.5 / 48.8	30.2 / 35.1
YiSi1 SRL 18	23.3 / 31.5	34.3 / 48.3	29.8 / 34.5
BLEURTbase -pre	33.0 / 39.0	41.5 / 54.6	38.2 / 39.6
BLEURTbase	34.5 / **42.9**	43.5 / 55.6	39.2 / 40.5
BLEURT -pre	34.5 / 42.1	42.7 / 55.4	39.2 / 40.6
BLEURT	**35.6** / 42.3	**44.2 / 56.7**	**40.0 / 41.4**

Table 2: Kendall Tau and Pearson *r* correlations with human judgments on the WMT18 Metrics Shared Task

Summary

- **N-gram Overlap Metrics**
  - Precision, Recall, F-Measure
  - BLEU
  - ROUGE
  - METEOR
  - Fast and easy to compute
  - Surface level matching - no representations/semantics
- **Alignment-based Metrics**
  - Word Mover's Distance
  - Sentence Mover's similarity
  - Bertscore & Moverscore
  - More reliable performance
  - Hyper-Parameter free and highly interpretable
  - Higher computation cost
- **Supervised Metrics**
  - RUSE
  - BLEURT
  - Best correlation with human ratings
  - Exigence of training data

References

- Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311-318).
- Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." In *Text summarization branches out*, pp. 74-81. 2004.
- Banerjee, S., & Lavie, A. (2005, June). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization* (pp. 65-72).
- Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015, June). From word embeddings to document distances. In *International conference on machine learning* (pp. 957-966). PMLR.
- Clark, E., Celikyilmaz, A., & Smith, N. A. (2019, July). Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 2748-2760).
- Zhao, W., Peyrard, M., Liu, F., Gao, Y., Meyer, C. M., & Eger, S. (2019, November). MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 563-578).

References

- Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.
- Shimanaka, H., Kajiwara, T., & Komachi, M. (2018, October). Ruse: Regressor using sentence embeddings for automatic machine translation evaluation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers* (pp. 751-758).
- Sellam, T., Das, D., & Parikh, A. (2020, July). BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7881-7892).
- Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.
- Celikyilmaz, A., Clark, E., & Gao, J. (2020). Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.
- Nenkova, Ani, Rebecca Passonneau, and Kathleen McKeown. "The pyramid method: Incorporating human content selection variation in summarization evaluation." *ACM Transactions on Speech and Language Processing (TSLP)* 4.2 (2007): 4-es.

THANK YOU

Acknowledgements

References (4)

Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025 (2015).

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

Peters, Matthew E., et al. "Deep contextualized word representations." arXiv preprint arXiv:1802.05365 (2018).

Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

References

- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

References

- Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A Neural Probabilistic Language Model. The Journal of Machine Learning Research, 3, 1137–1155. http://doi.org/10.1162/153244303322533223
- Mikolov, T., Corrado, G., Chen, K., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. Proceedings of the International Conference on Learning Representations (ICLR 2013), 1–12.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. NIPS, 1–9.
- Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing. Proceedings of the 25th International Conference on Machine Learning - ICML '08, 20(1), 160–167. http://doi.org/10.1145/1390156.1390177
- Kim, Y., Jernite, Y., Sontag, D., & Rush, A. M. (2016). Character-Aware Neural Language Models. AAAI. Retrieved from http://arxiv.org/abs/1508.06615
- Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., & Wu, Y. (2016). Exploring the Limits of Language Modeling. Retrieved from http://arxiv.org/abs/1602.02410
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (almost) from Scratch. Journal of Machine Learning Research, 12 (Aug), 2493–2537. Retrieved from http://arxiv.org/abs/1103.0398
- Chen, W., Grangier, D., & Auli, M. (2015). Strategies for Training Large Vocabulary Neural Language Models, 12. Retrieved from http://arxiv.org/abs/1512.04906

References

- Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving Distributional Similarity with Lessons Learned from Word Embeddings. Transactions of the Association for Computational Linguistics, 3, 211–225. Retrieved from https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/570
- Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 1532–1543. http://doi.org/10.3115/v1/D14-1162
- Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. ACL, 238–247. http://doi.org/10.3115/v1/P14-1023
- Levy, O., & Goldberg, Y. (2014). Neural Word Embedding as Implicit Matrix Factorization. Advances in Neural Information Processing Systems (NIPS), 2177–2185. Retrieved from http://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization
- Hamilton, W. L., Clark, K., Leskovec, J., & Jurafsky, D. (2016). Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Retrieved from http://arxiv.org/abs/1606.02820
- Hamilton, W. L., Leskovec, J., & Jurafsky, D. (2016). Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. arXiv Preprint arXiv:1605.09096.