

Generation

Vsevolod Dyomkin
prj-nlp 2018-05-16

Natural Language Generation (NLG)

- * general-purpose
- * special-purpose
- * word-level
- * text-level
- * book-level

Applications:

- * data-to-text
- * simplification
- * summarization
- * paraphrasing
- * dialogue
- * computer-generated verse/poetry
- * MT
- * GEC
- * QA

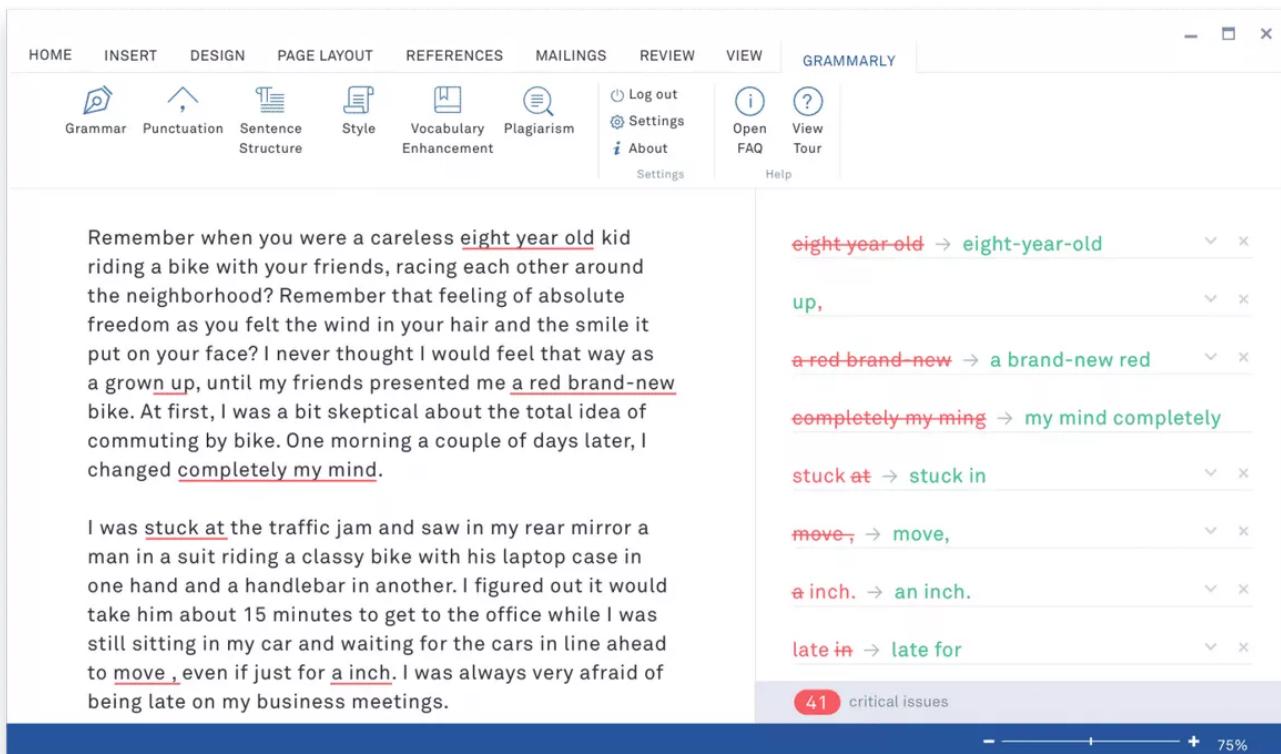
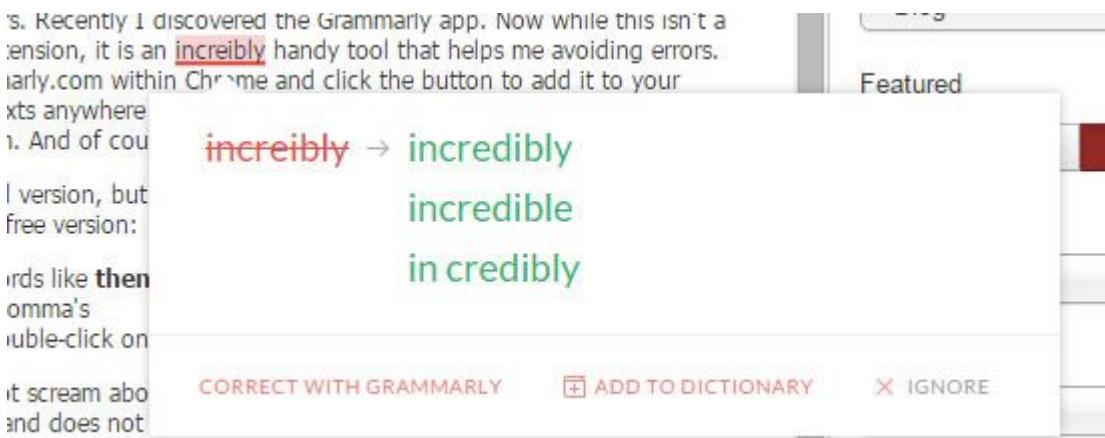
Levels of NLG

- Level 1: Simple Fill-in-the-Blank Systems
- Level 2: Script/Rule-based Systems
- Level 3: Word-Level Grammatical Functions
- Level 4: Dynamically Creating Sentences
- Level 5: Dynamically Creating Documents

<https://ehudreiter.com/2016/12/18/nlg-vs-templates/>

Example: Grammarly

- * word choice
 - * phrase rewriting
 - * ... text rewriting

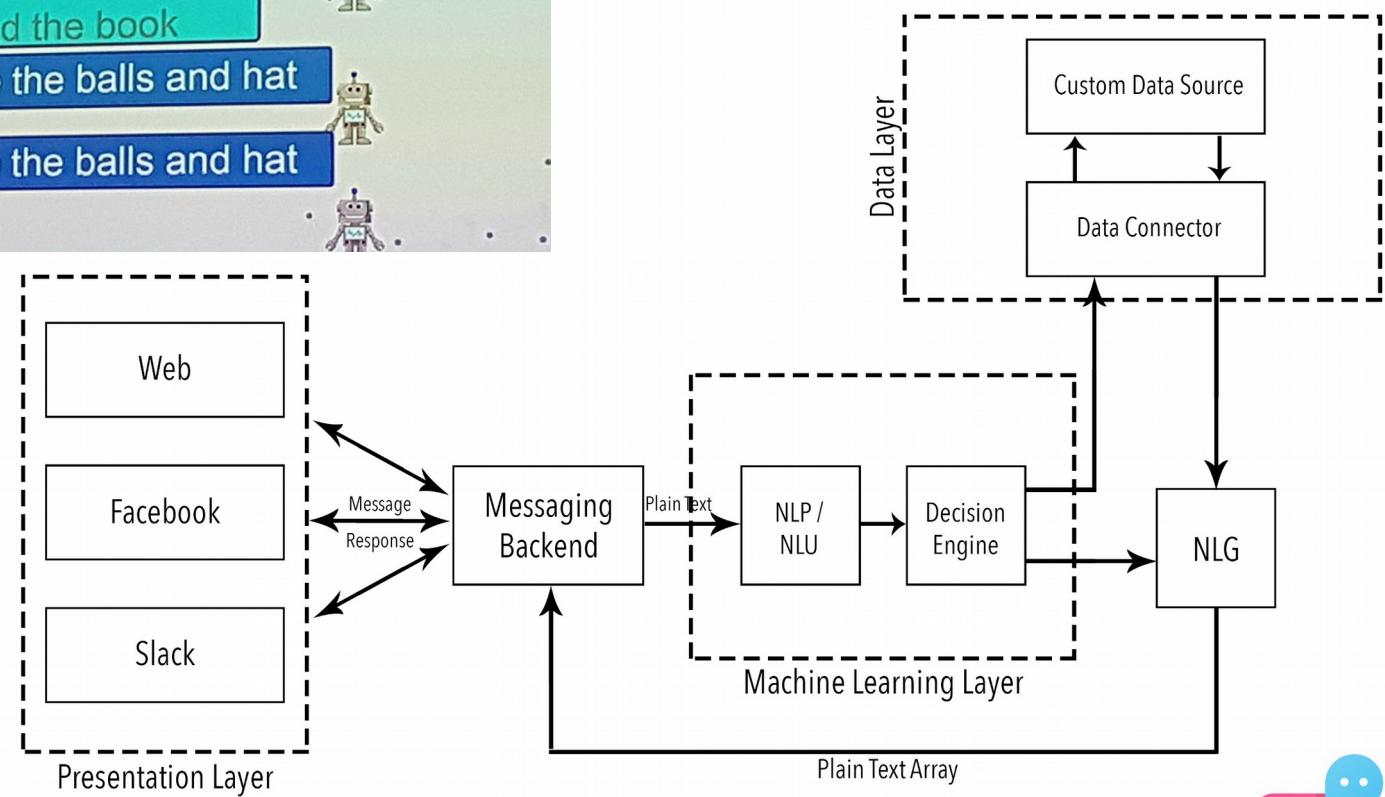
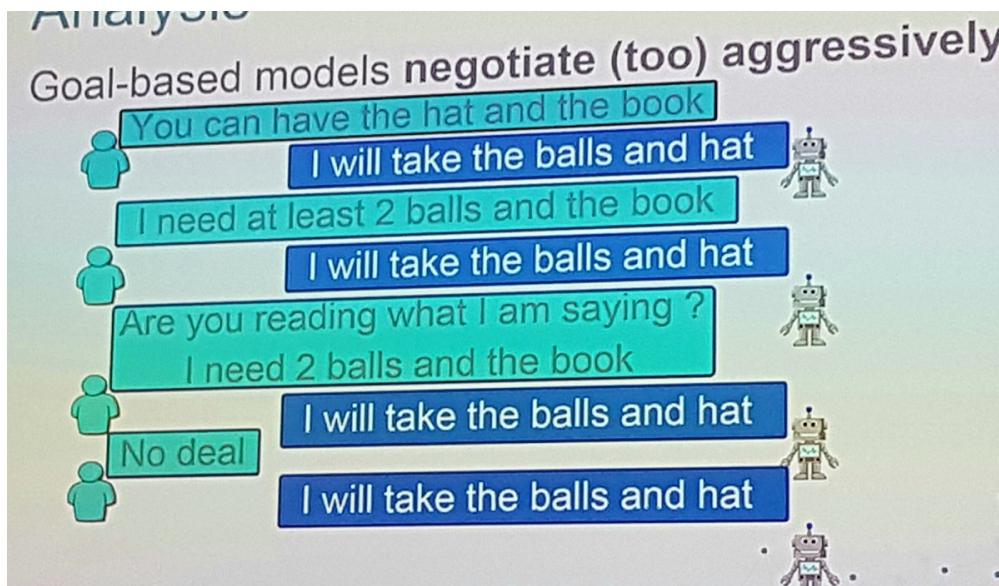


Example: data-to-text

- * weather
- * sports
- * stocks
- * news

<https://www.aclweb.org/anthology/W18-6504>

Example: chatbots



<https://github.com/facebookresearch/end-to-end-negotiator>



Example: books

A screenshot of a Twitter post from user @tinysubversions. The post features a profile picture of a man with glasses and a beard, the name 'Darius Kazemi' in bold black text, and the handle '@tinysubversions'. To the right of the name is a small blue Twitter logo. The tweet text reads: 'Hey, who wants to join me in NaNoGenMo: spend the month writing code that generates a 50k word novel, share the novel & the code at the end'. Below the tweet are engagement metrics: 179 likes and 7:00 PM - Nov 1, 2013. To the right of the timestamp is a small info icon. At the bottom of the post is a blue button with a speech bubble icon and the text '148 people are talking about this', followed by a right-pointing arrow.

Titles:

- * Webster's Slovak – English Thesaurus Dictionary for \$28.95
- * The 2007-2012 World Outlook for Wood Toilet Seats for \$795
- * The World Market for Rubber Sheath Contraceptives (Condoms): A 2007 Global Trade Perspective for \$325
- * Ellis-van Creveld Syndrome – A Bibliography and Dictionary for Physicians, Patients, and Genome Researchers for \$28.95
- * Webster's English to Haitian Creole Crossword Puzzles: Level 1 For \$14.95

<https://singularityhub.com/2012/12/13/patented-book-writing-system-lets-one-professor-create-hundreds-of-thousands-of-amazon-books-and-counting/>

Example: abstractive summarization

Article	novell inc. chief executive officer eric schmidt has been named chairman of the internet search-engine company google .
Human summary	novell ceo named google chairman
Textsum	novell chief executive named to head internet company

<https://rare-technologies.com/text-summarization-in-python-extractive-vs-abstractive-techniques-revisited/>

NLG Evaluation

Need to capture quality & diversity

(Best) Real-World Task-Based (Extrinsic)

(Good) Laboratory Task-Based or Real-
World Human Ratings

(OK) Laboratory Human Ratings

(Worst) Metrics

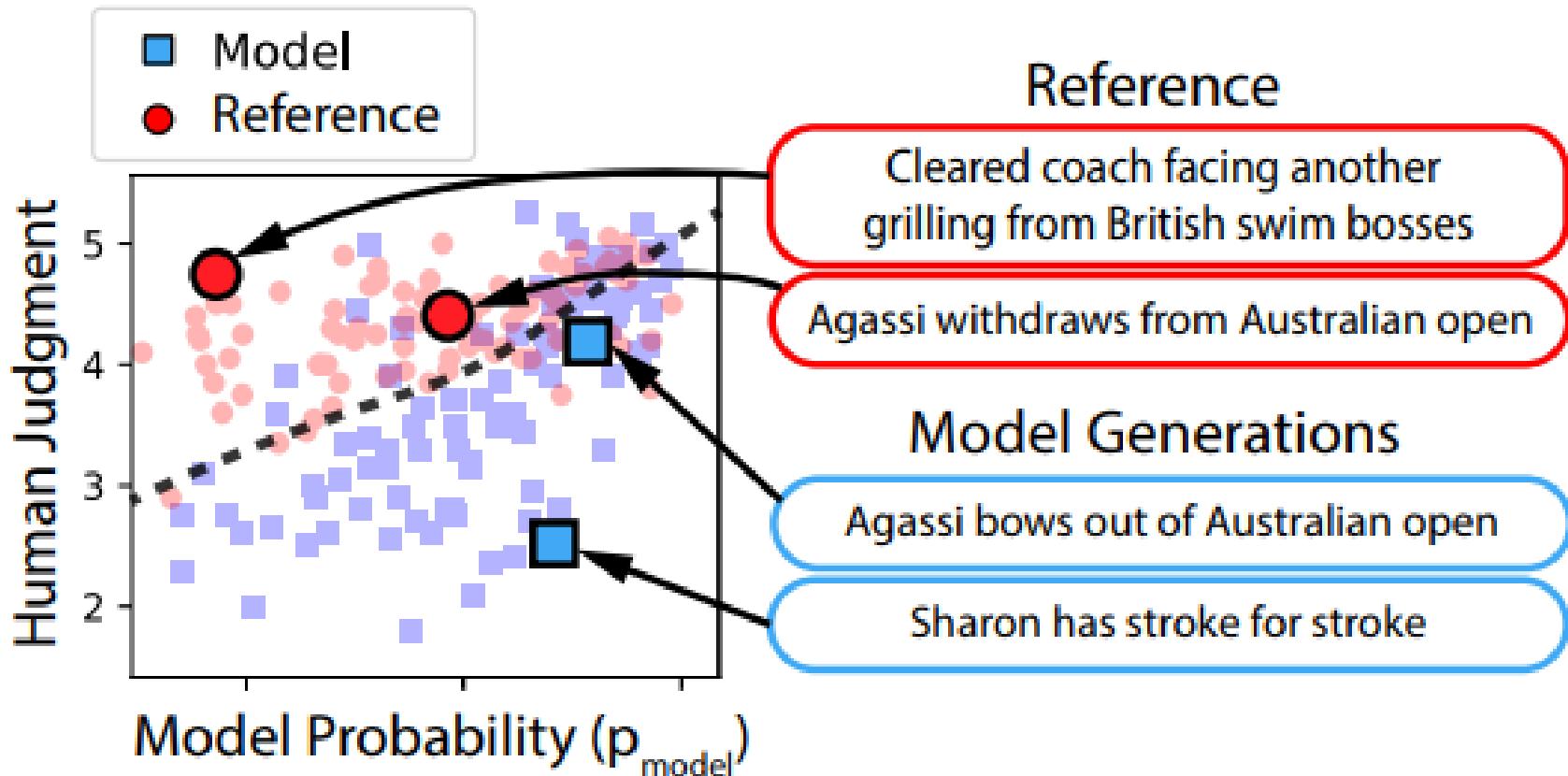
[https://ehudreiter.com/2017/01/19/types-
of-nlg-evaluation/](https://ehudreiter.com/2017/01/19/types-of-nlg-evaluation/)

Metrics

- * BLEU
- * ROUGE
- * METEOR
- * Perplexity

NLG Evaluation: HUSE

Combine human evaluation & perplexity



<https://arxiv.org/pdf/1904.02792.pdf>

Classic Approach to NLG

- 1) Content determination
- 2) Document structuring
- 3) Aggregation
- 4) Lexical choice
- 5) Referring expression generation
- 6) Realization

Hybrid Approaches

- * overgenerate than select

<https://aclanthology.info/pdf/P/P98/P98-1116.pdf>

(an example using AMR)

- * ML choosers embedded in a rule-based framework

<https://aclanthology.info/pdf/J/J17/J17-1001.pdf>

<https://ehudreiter.com/2017/10/16/machine-learning-and-rules/>

DL Approaches

- * a plain RNN
- * variational autoencoders
- * seq2seq
- * transformers

Also, maybe:

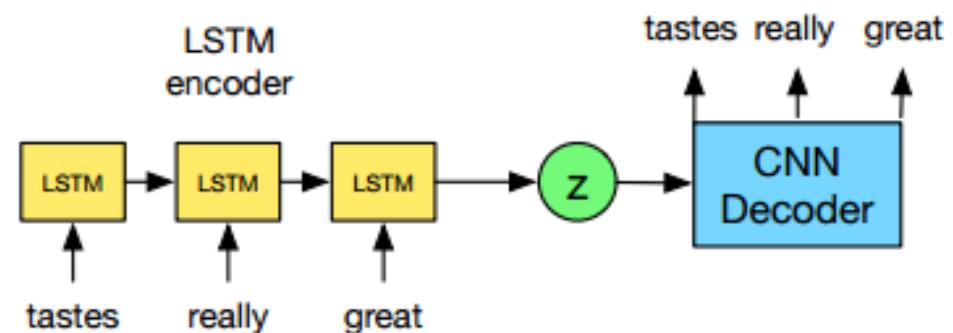
- * GANs
- * deep re-inforcement learning

VAEs

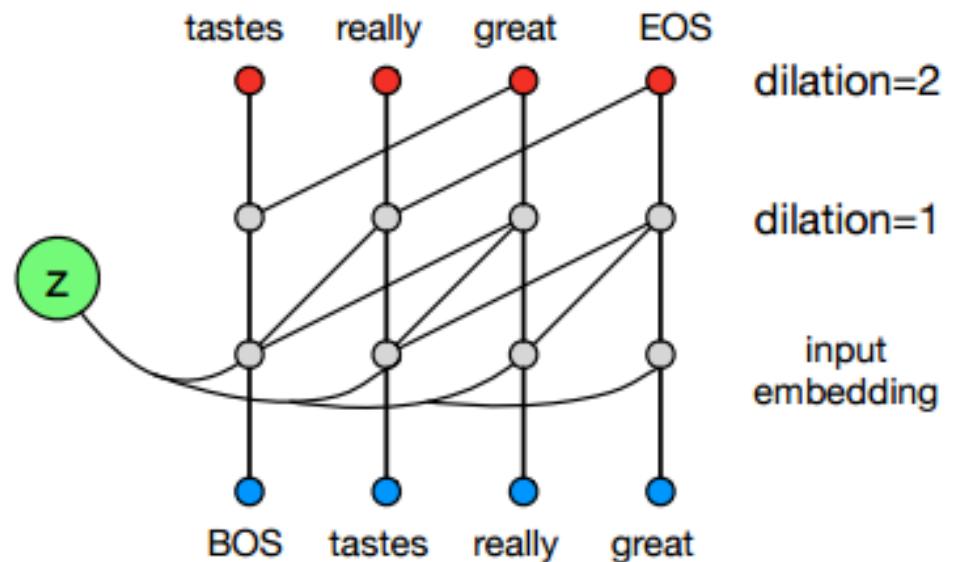
A generation model

“framework”:

- encoder
- hidden state
- decoder



(a) VAE training graph using a dilated CNN decoder.



(b) Diagram of dilated CNN decoder.

Language Modelling Task

Question: what is the probability of a sequence of words (sentence/paragraph/text)?

And why do we need it?

For the sentence

the dog barks STOP

we would have

$$\begin{aligned} p(\text{the dog barks STOP}) &= q(\text{the}|*, *) \\ &\quad \times q(\text{dog}|*, \text{the}) \\ &\quad \times q(\text{barks}|\text{the}, \text{dog}) \\ &\quad \times q(\text{STOP}|\text{dog}, \text{barks}) \end{aligned}$$

LM Applications

- * Word choice, predictive typing
- * NLG
- * Statistical machine translation
- * Spelling & grammatical error correction
- * OCR, ASR, code breaking, paleolinguistics etc.
- * transfer learning

Ngram LM

Apply Markov assumption to the word sequence.

If $n=3$ (trigrams):

$$\begin{aligned} P(S) = & P(w_0) * P(w_1 | w_0) * P(w_2 | w_0 \ w_1) \\ & * P(w_3 | w_0 \ w_1 \ w_2) * P(w_4 | \cancel{w_0} \ \cancel{w_1} \ w_2 \ w_3) \end{aligned}$$

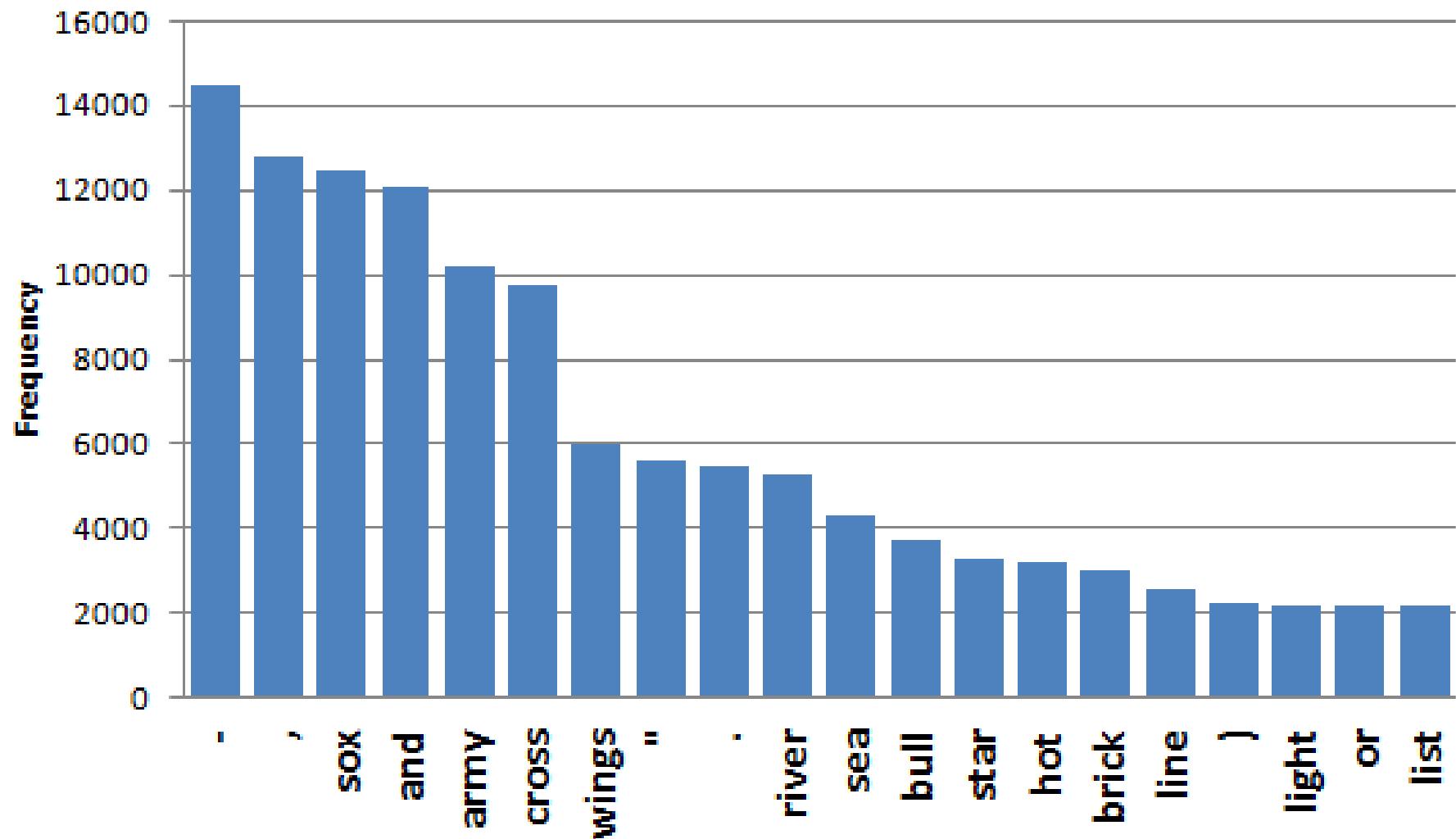
According to the chain rule:

$$P(w_2 | w_0 \ w_1) = P(w_0 \ w_1 \ w_2) / P(w_0 \ w_1)$$

We can use MLE

Ngrams Estimation

The most frequent Wikipedia bigrams
beginning with 'red'



Ngrams' Problems

- * Need big corpus for MLE
- * Number of ngrams $\sim O(e^n)$ (n-ngram rank)
- * Sparsity (problem of UNKs):

$$P(S) = P(w_0) * P(w_1 | w_0) * P(w_2 | w_0 \ w_1) \\ * P(w_3 | w_1 \ w_2) * P(w_4 | w_2 \ w_3)$$

If some of w_0-w_4 are UNK $P(S) = 0!$

Ngrams Smoothing

- * Laplace smoothing

Ngrams Smoothing

- * Laplace smoothing
- * Naive +1 smoothing

Ngrams Smoothing

- * Laplace smoothing
- * Naive +1 smoothing
- * Good-Turing smoothing, Katz smoothing
- * Knesser-Ney smoothing:
 - a discounting interpolation
 - (using lower-order ngrams)

$$P_{KN}(w_i \mid w_{i-1}) = \frac{\max(c(w_{i-1}w_i) - \delta, 0)}{\sum_{w'} c(w_{i-1}w')} + \lambda \frac{|\{w_{i-1} : c(w_{i-1}, w_i) > 0\}|}{|\{w_{j-1} : c(w_{j-1}, w_j) > 0\}|}$$

$$\lambda(w_{i-1}) = \frac{\delta}{c(w_{i-1})} |\{w' : c(w_{i-1}, w') > 0\}|$$

Ngrams Implementation

- * cut-off
- * efficient storage (binary trees, perfect hash-tables, ...)
- * quantization
- * efficient estimation (MapReduce)

LM Software:

- * BerkeleyLM
- * KenLM

https://kheafield.com/papers/stanford/crawl_paper.pdf

LMs Evaluation

Intrinsic evaluation –
perplexity (a measure of surprise /per word):

$$2^{H(p)} = 2^{-\sum_x p(x) \log_2 p(x)}$$

$$PP(s_1, s_2, \dots) = (\sum_i |s_i|) \sqrt{\frac{1}{\prod_i p(s_i)}}$$

A corpus-based measure. Current corpus – 1B
word benchmark (<http://arxiv.org/abs/1312.3005>)

Extrinsic evaluation also necessary

SOTA Perplexity

MODEL	TEST PERPLEXITY
SIGMOID-RNN-2048 (JI ET AL., 2015A)	68.3
INTERPOLATED KN 5-GRAM, 1.1B N-GRAMS (CHELBA ET AL., 2013)	67.6
SPARSE NON-NEGATIVE MATRIX LM (SHAZEER ET AL., 2015)	52.9
RNN-1024 + MAXENT 9-GRAM FEATURES (CHELBA ET AL., 2013)	51.3
LSTM-512-512	54.1
LSTM-1024-512	48.2
LSTM-2048-512	43.7
LSTM-8192-2048 (NO DROPOUT)	37.9
LSTM-8192-2048 (50% DROPOUT)	32.2
2-LAYER LSTM-8192-1024 (BIG LSTM)	30.6
BIG LSTM+CNN INPUTS	30.0
BIG LSTM+CNN INPUTS + CNN SOFTMAX	39.8
BIG LSTM+CNN INPUTS + CNN SOFTMAX + 128-DIM CORRECTION	35.8
BIG LSTM+CNN INPUTS + CHAR LSTM PREDICTIONS	47.9

<https://arxiv.org/pdf/1602.02410.pdf>

Character LM

What if we use characters instead of words
(for ngrams or as input to the NN)?

... “The unreasonable effectiveness of
Character-level Language Models”

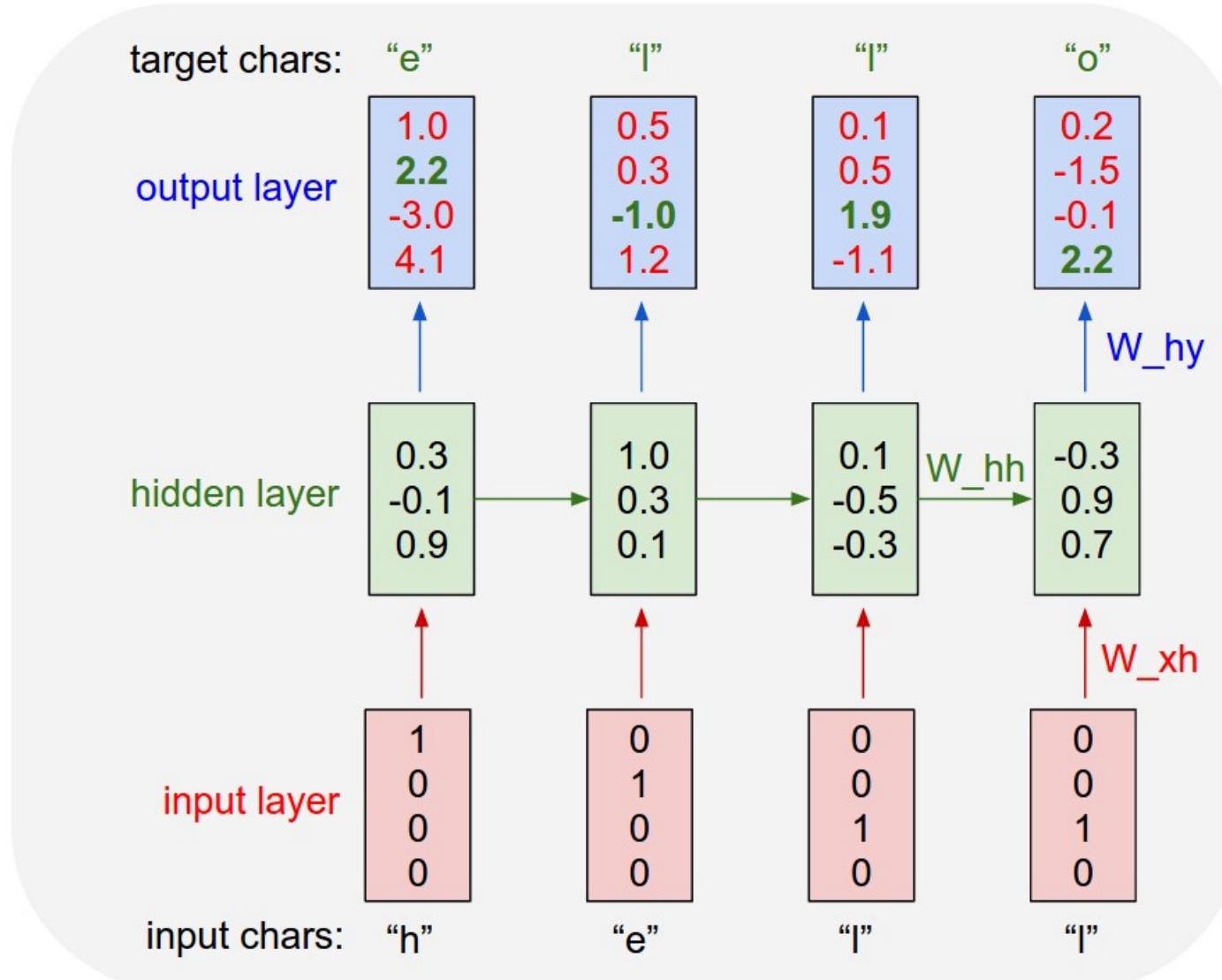
<http://nbviewer.jupyter.org/gist/yoavg/d76121dfde2618422139>

For ngram-based models, as number of tokens
is small, order may be quite large (10-20-
100?)

Pro: no need for smoothing

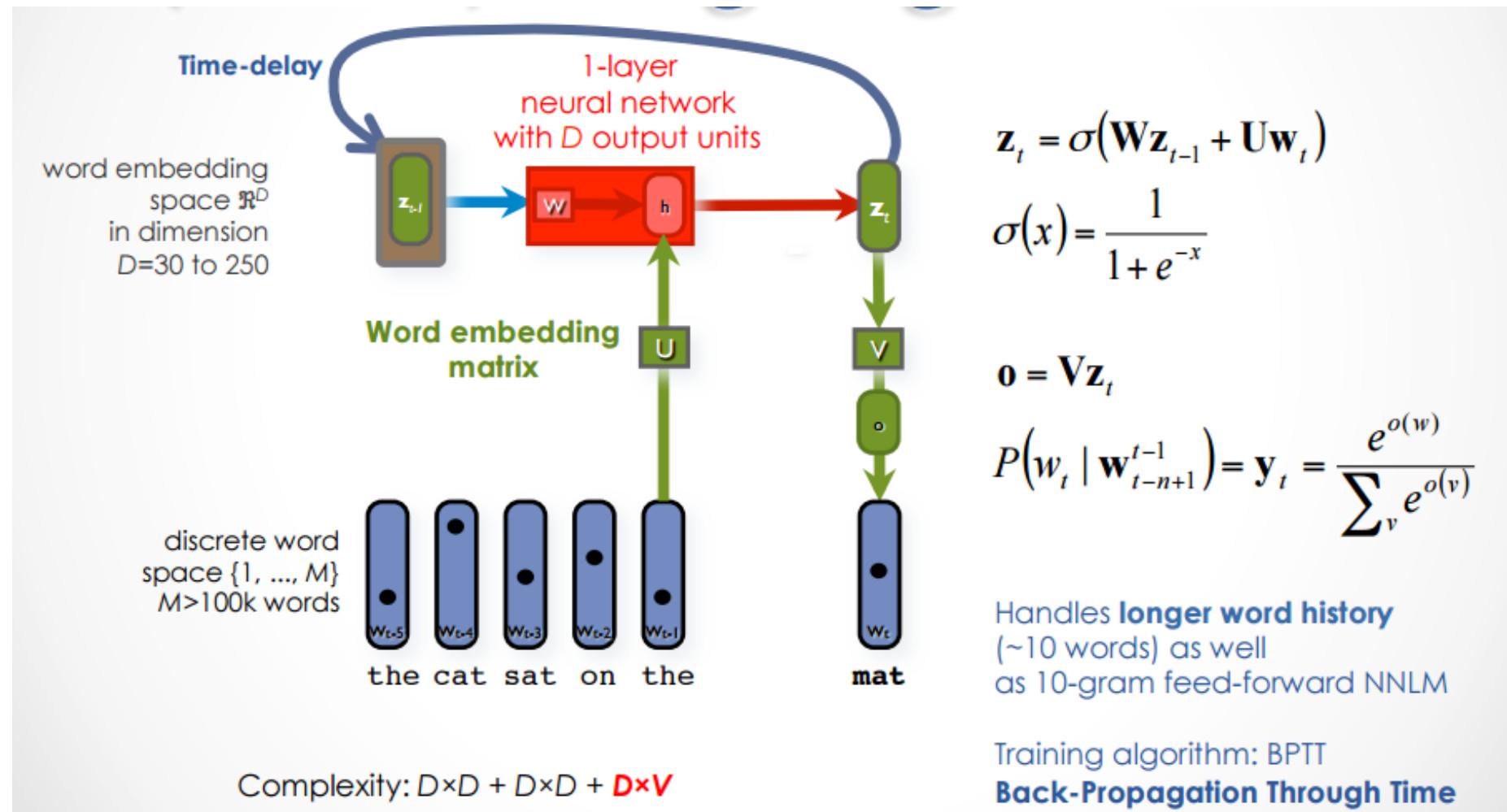
Con: no notion of tokens

Neural CharLM



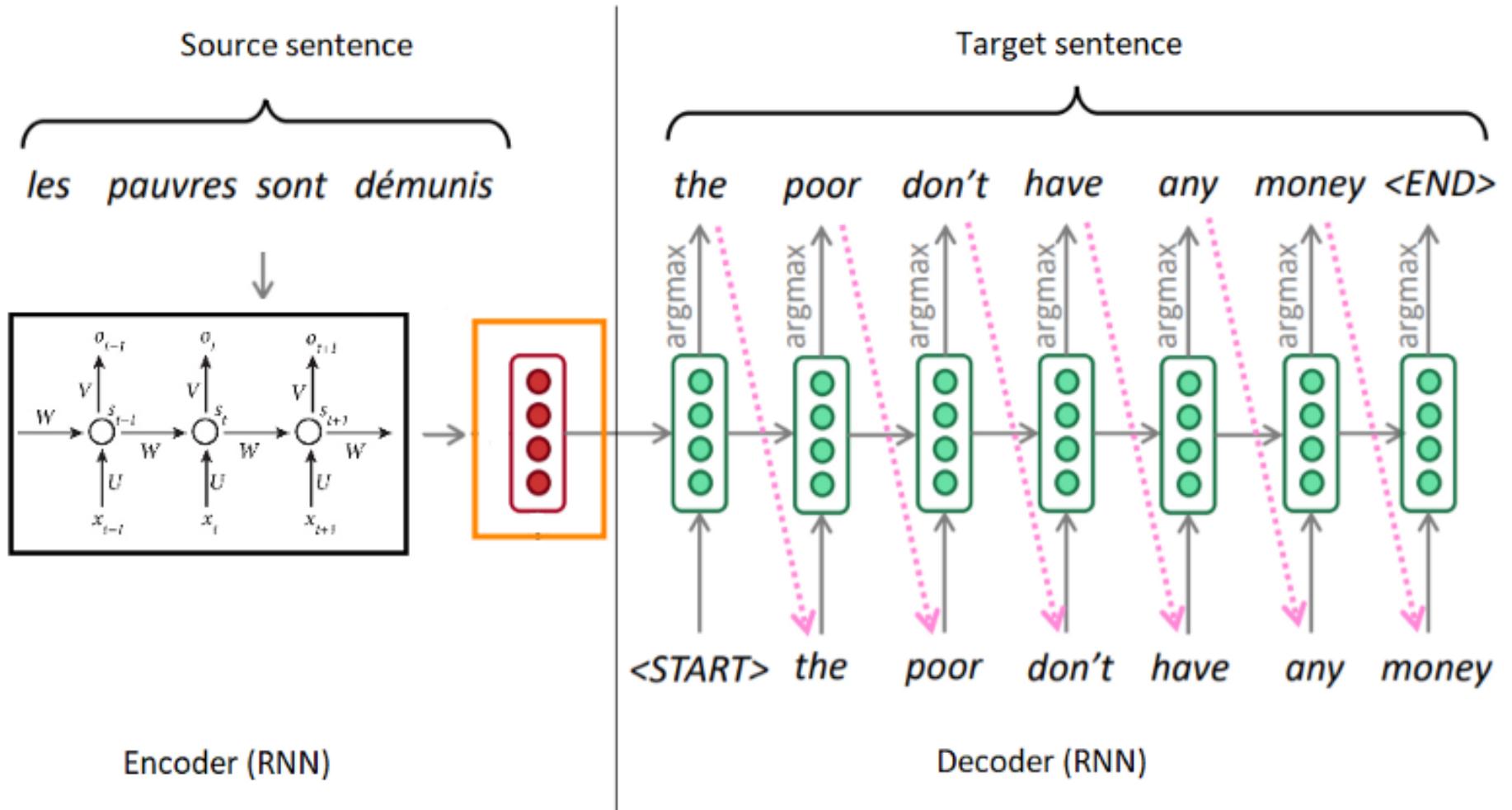
<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

Neural LM



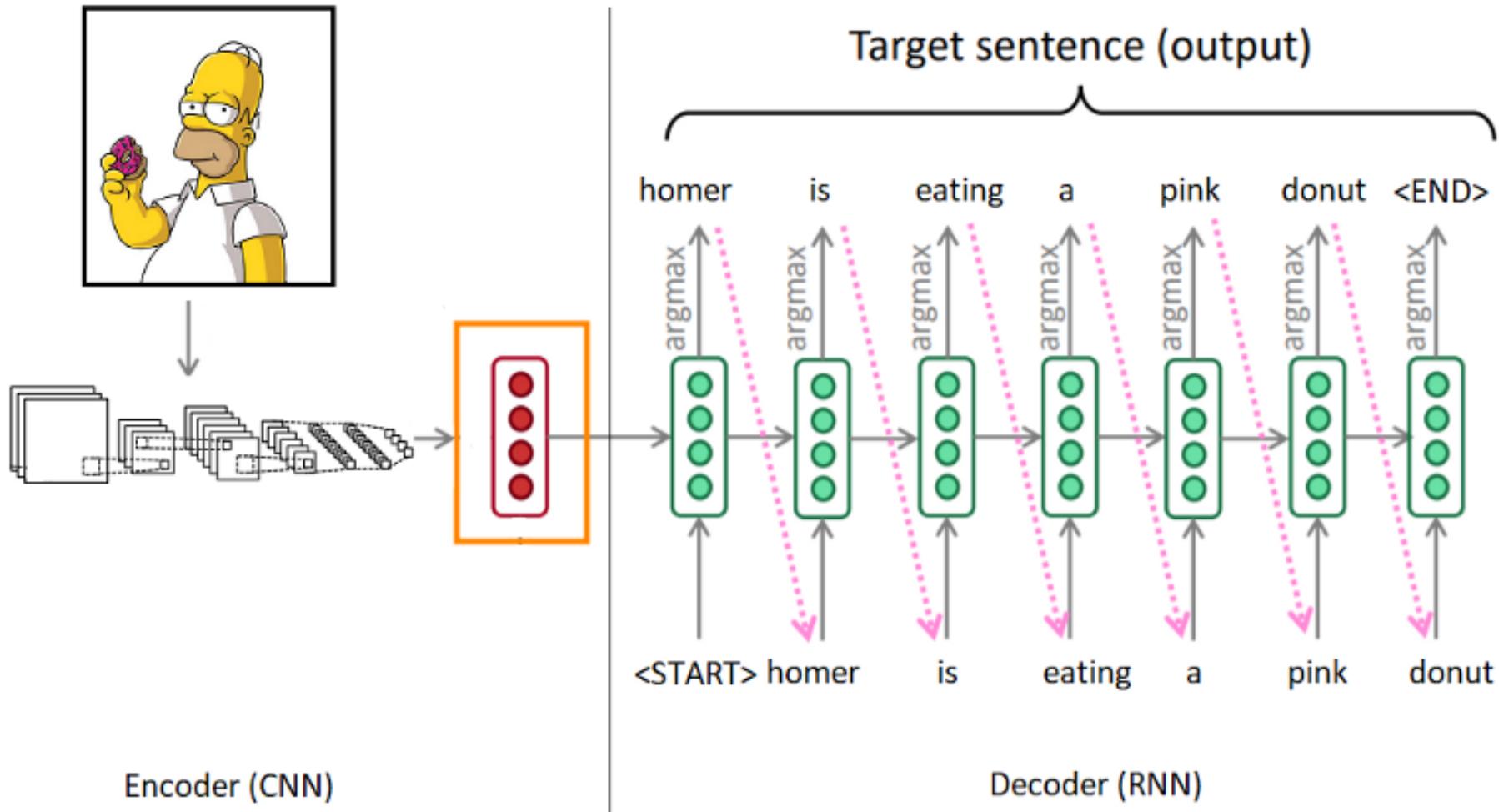
<http://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf>

seq2seq

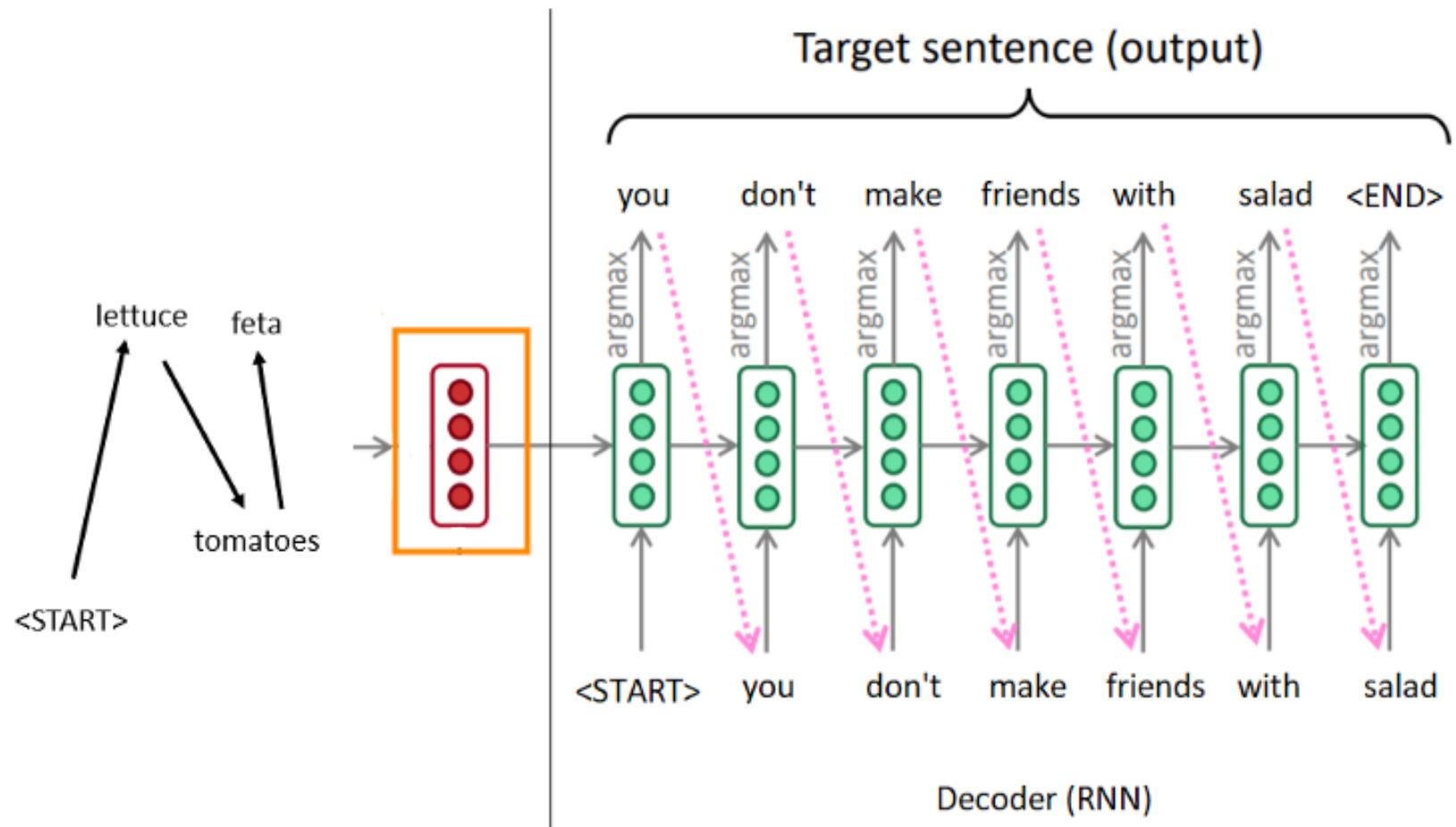


<https://medium.com/phrasee/neural-text-generation-generating-text-using-conditional-language-models-a37b69c7cd4b>

seq2seq variants: image captioning

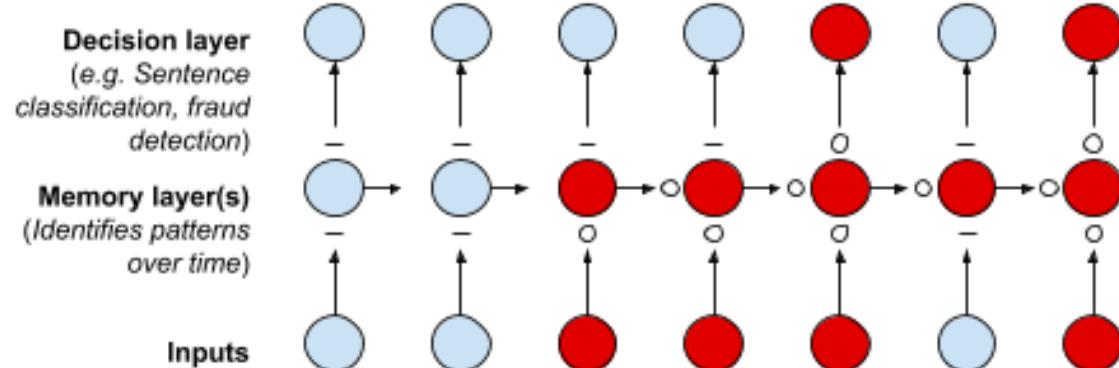


seq2seq variants: guided generation

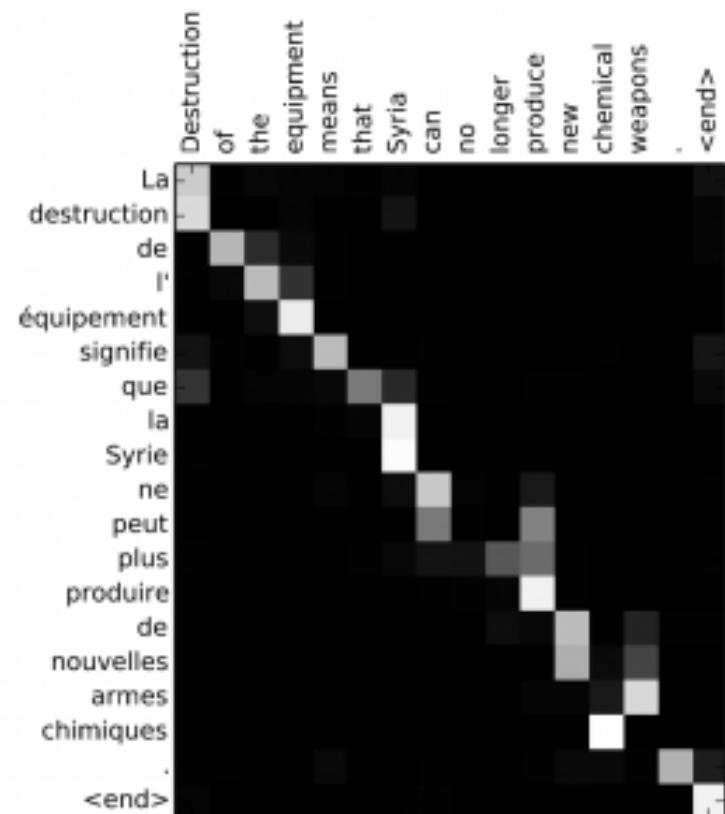
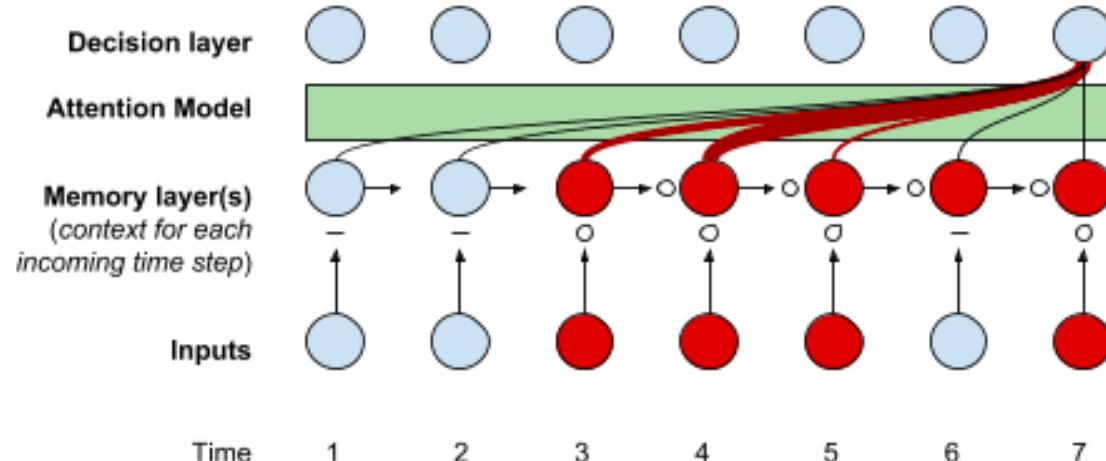


Attention

Recurrent Networks

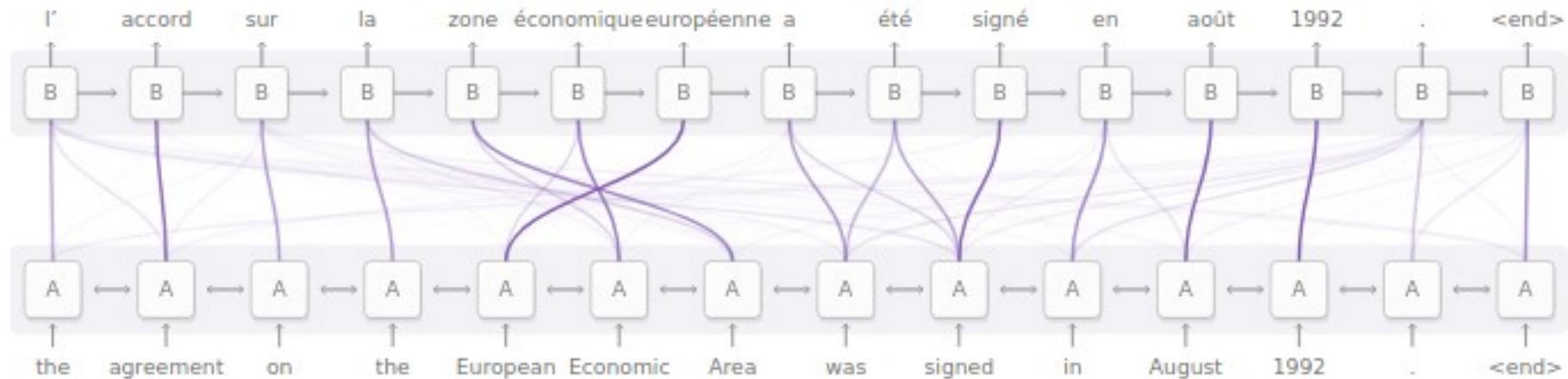
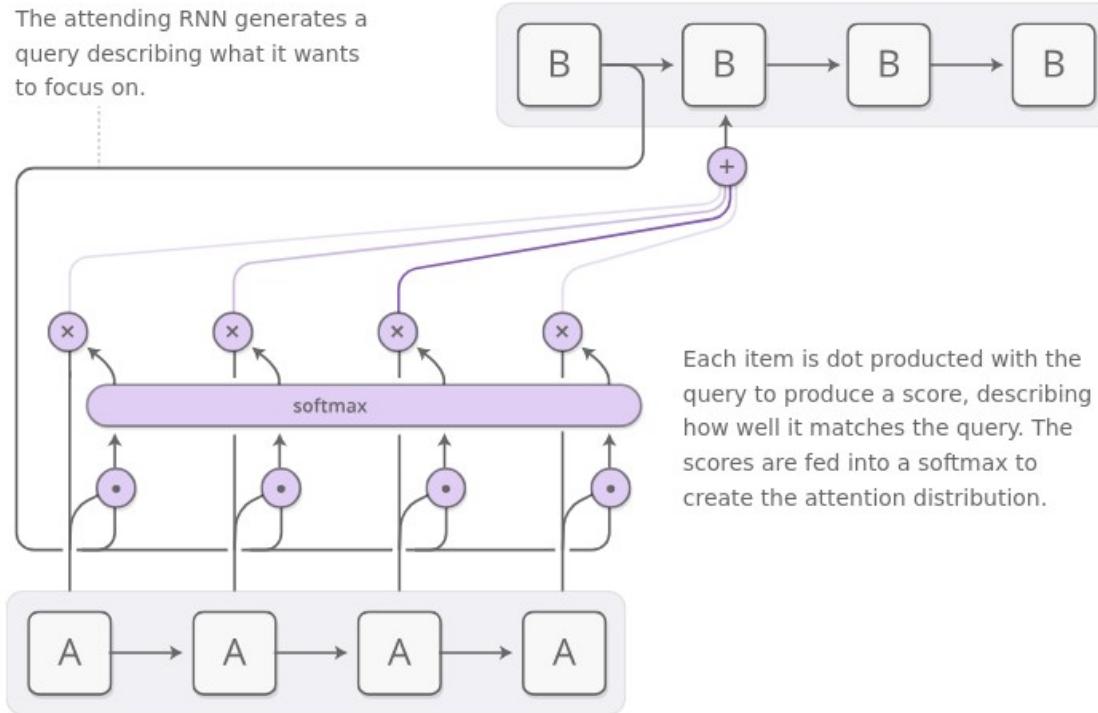


Attention Mechanism



<https://skymind.ai/wiki/attention-mechanism-memory-network>

Attention Mechanism



LMs Recap

LMs may be used both in classification and generation tasks:

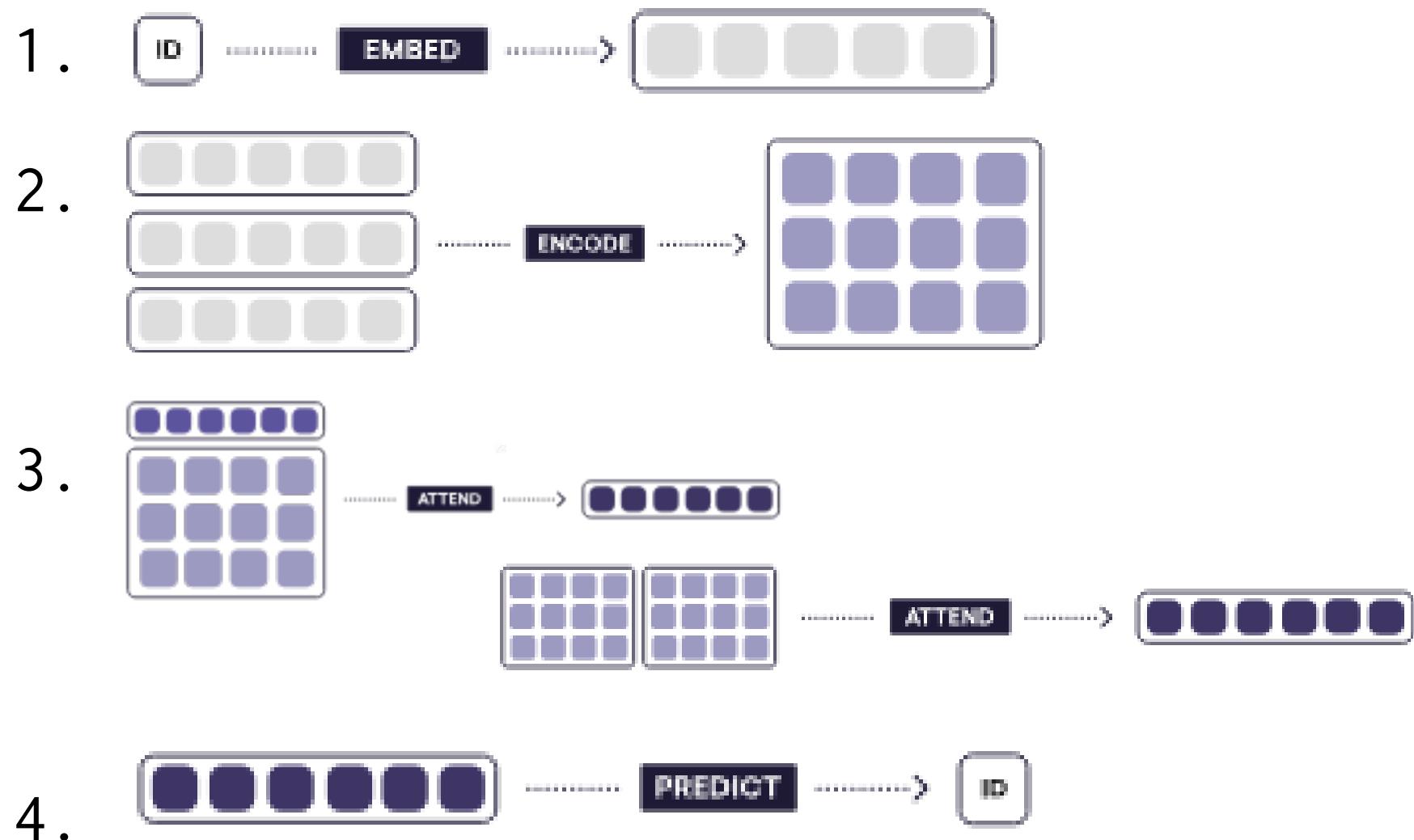
- * in classification they can be combined with a domain model
- * in generation: sample from the model or re-rank other model's output

Main approaches:

- * charLMs
- * smoothed ngrams
- * neural language models (SOTA)
- * but other variants are also possible
(grammars, topic models...)

The “DL Formula”

<https://explosion.ai/blog/deep-learning-formula-nlp>
Embed, encode, attend, predict



NLG Recap

- * NLG - the pinnacle of NLP
- * Allows for many approaches.
A good area to utilize DL strong points.
- * But evaluation is complicated
(+ lack of quality resources)

Read More

NLG:

<https://ehudreiter.com>

<https://arxiv.org/pdf/1509.00685.pdf>

<https://aclweb.org/anthology/J/J12/J12-1006.pdf>

<https://www.youtube.com/watch?v=9zKuYvjFFS8>

LMs:

<http://www.dhgarrette.com/nlpclass/notes/ngrams.pdf>

<http://www.foldl.me/2014/kneser-ney-smoothing/>

NNs:

<http://www.wildml.com/2016/01/attention-and-memory-in-deep-learning-and-nlp/>

<https://medium.com/@yoav.goldberg/an-adversarial-review-of-adversarial-generation-of-natural-language-409ac3378bd7>

<https://medium.com/@hyponymous/paper-summary-neural-machine-translation-by-jointly-learning-to-align-and-translate-84970177e08c>

<http://ofir.io/Neural-Language-Modeling-From-Scratch/>

<https://slides.com/oleksiysyvokon/lm-advances>

<https://medium.com/@adityathiruvengadam/transformer-architecture-attention-is-all-you-need-aecd9f50d09>