Language Modelling & Language Generation

Vsevolod Dyomkin

prj-nlp-1, 2018-05-17

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

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

LM Applications

* Word choice, predictive typing
* Natural language generation
* Statistical machine translation
* Spelling & grammatical error correction
* OCR, ASR, code breaking, paleolinguistics etc.

Ngram LM

Apply Markov assumption to the word sequence.

If n=3 (trigrams):

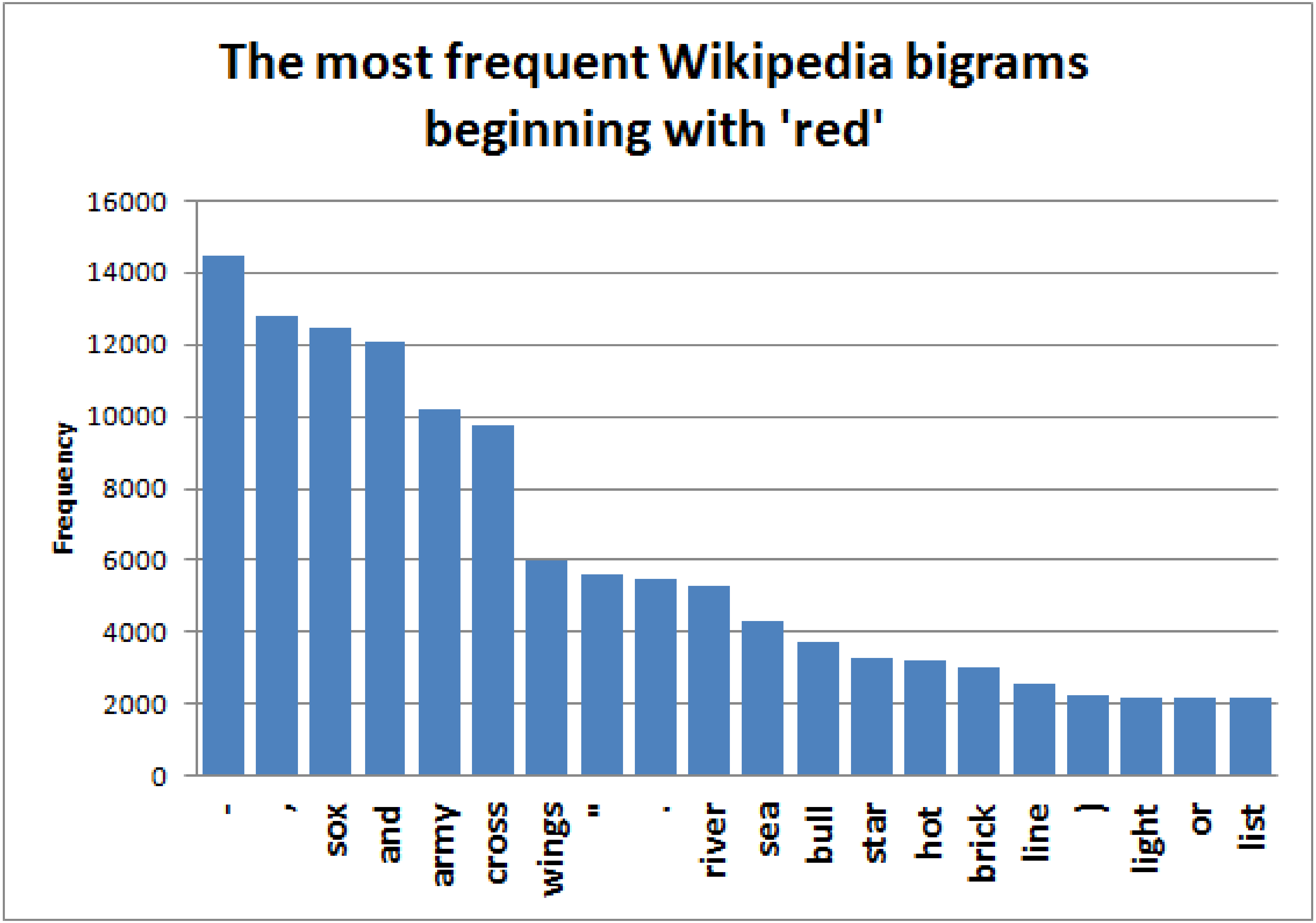
$$P(S) = P(w0) * P(w1|w0) * P(w2|w0\ w1) * P(w3|\cancel{w0}\ w1\ w2) * P(w4|\cancel{w0\ w1}\ w2\ w3)$$

According to the chain rule:

$$P(w2|w0\ w1) = P(w0\ w1\ w2) / P(w0\ w1)$$

We can use MLE

Ngrams Estimation



Ngrams' Problems

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

```
P(S) = P(w0) * P(w1|w0) * P(w2|w0 w1)
     * P(w3|w1 w2) * P(w4|w2 w3)
```

If some of w0-w4 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)

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

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

Other Kinds of Ngrams

* Syntactic (dependency) ngrams
https://research.googleblog.com/2013/05/syntactic-ngrams-over-time.html
* Wildcard/mixed ngrams
* Treelets

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):

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

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

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)?

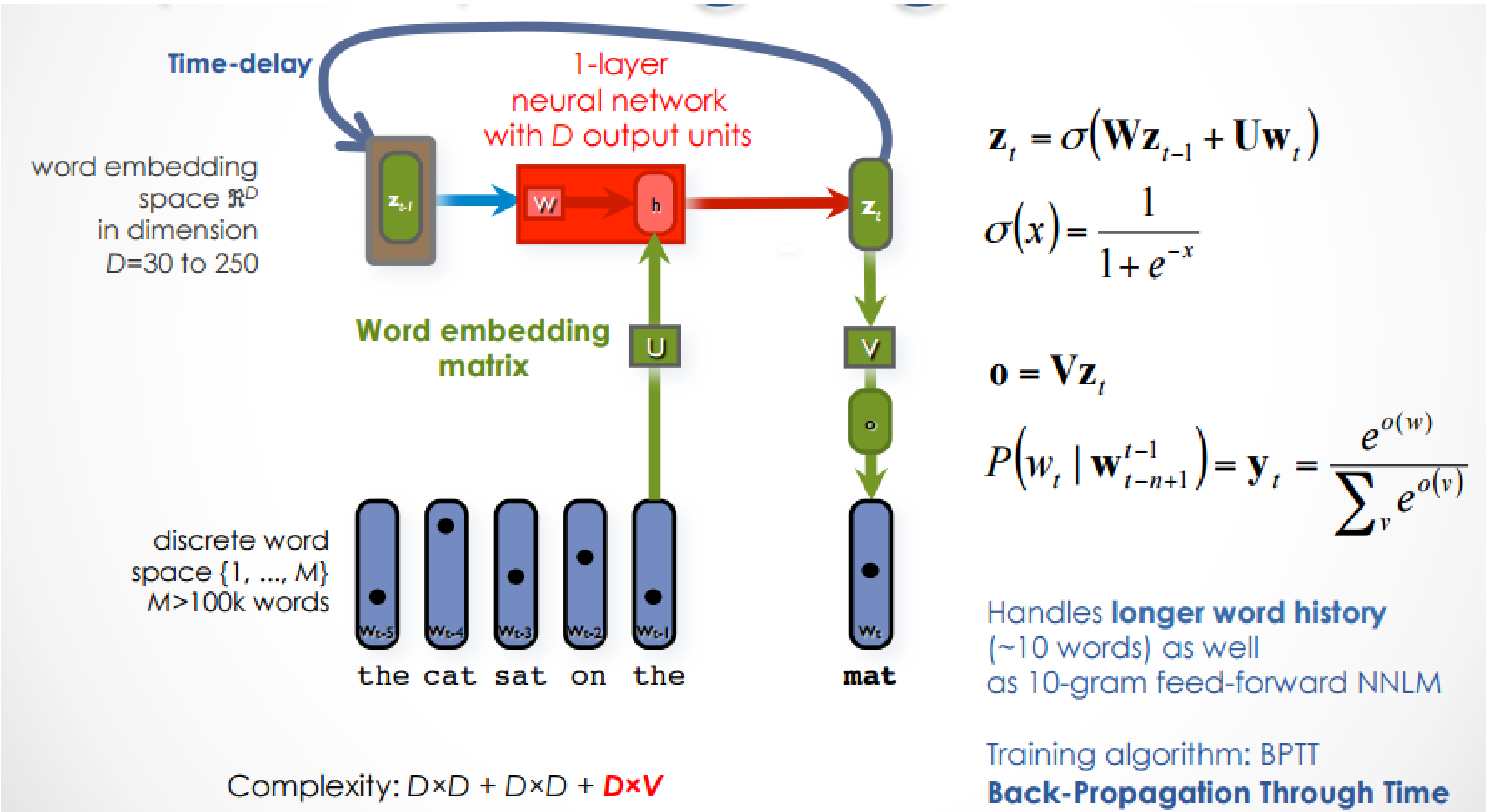
... “The unreasonable effectiveness of Character-level Language Models” http://nbviewer.jupyter.org/gist/yoavg/d76121dfde2618422139

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

Pro: no need for smoothing

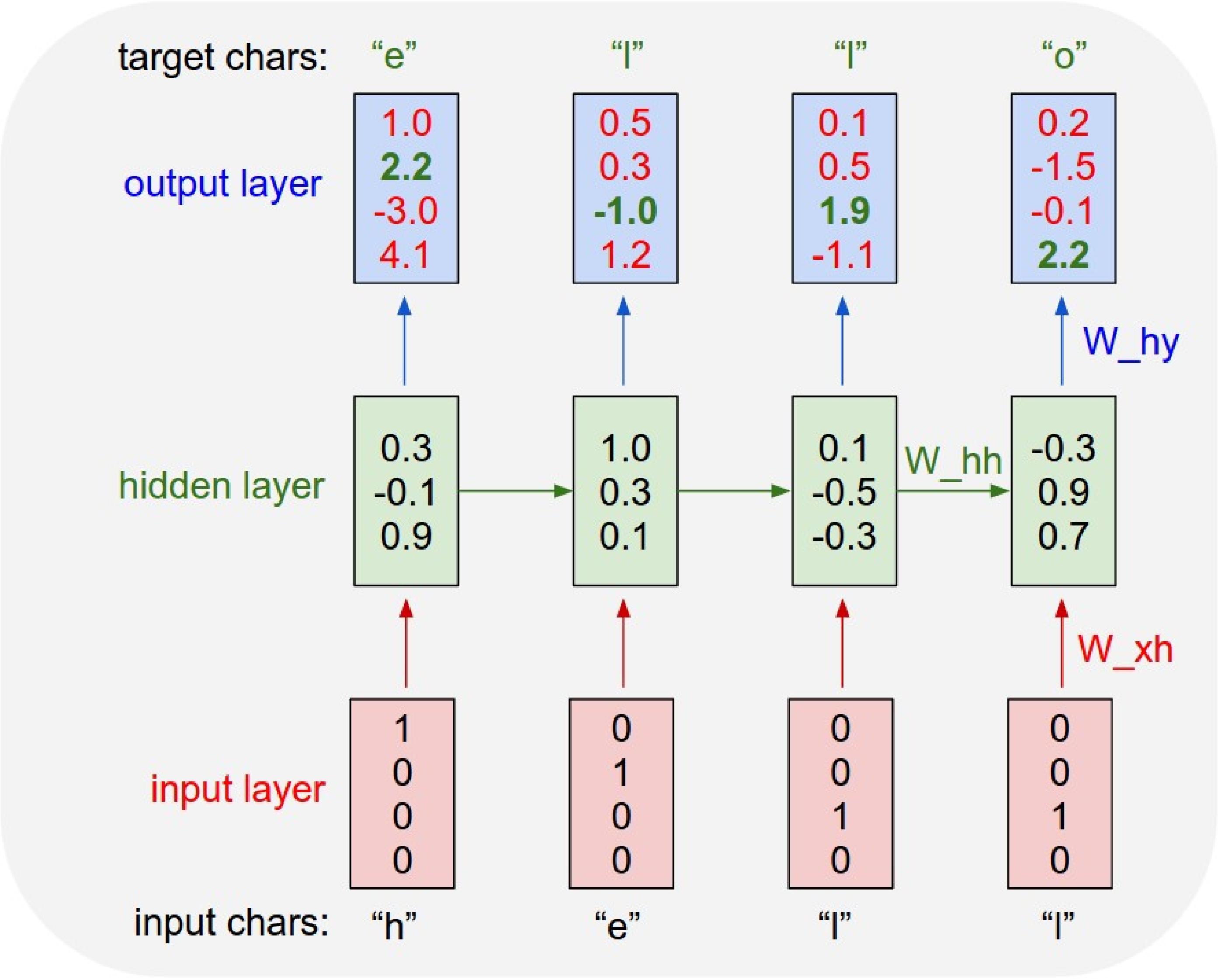
Con: no notion of tokens

Neural LM



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

Neural CharLM



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

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 rerank other model's output

Main approaches:

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

Natural Language Generation (NLG)

* general-purpose
* special-purpose

Applications:

* template-based systems
* data-to-text
* summarization
* simplification
* paraphrasing
* dialogue
* computer-generated verse/poetry

* MT
* GEC
* QA

NLG System Breakdown

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

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: 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/

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

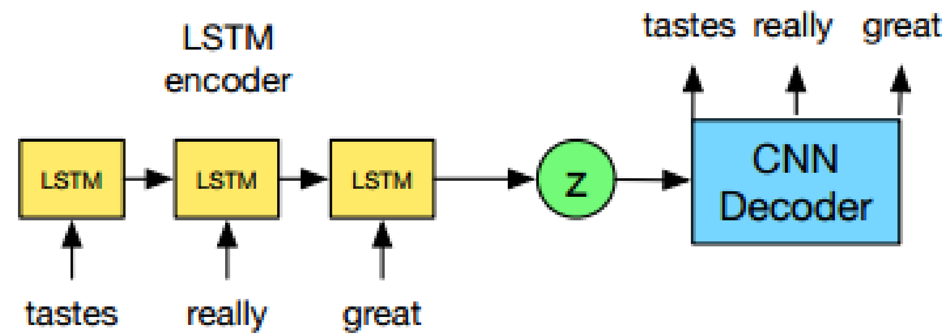
https://ehudreiter.com/2017/10/16/machine-learning-and-rules/

DL Approaches

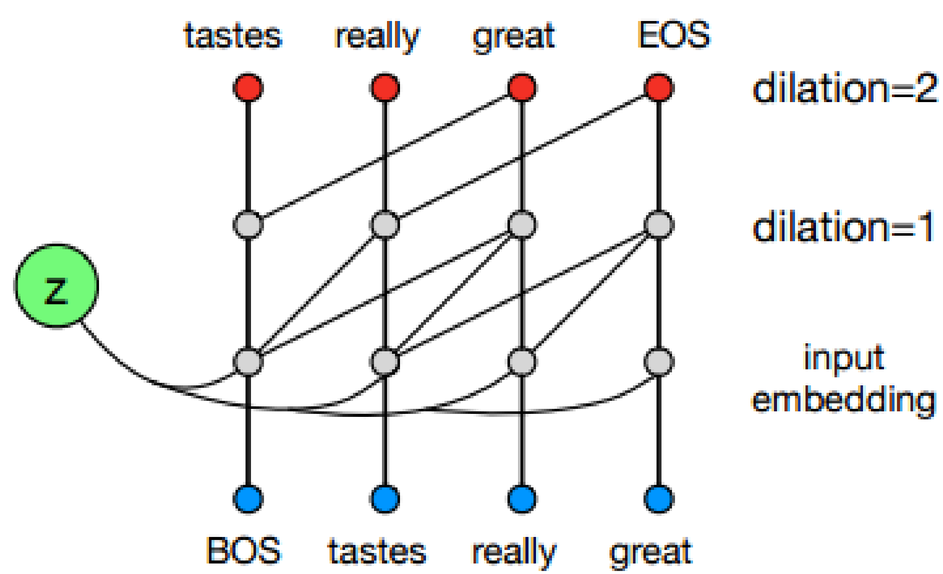
* a plain RNN
* variational autoencoders
* seq2seq
* GANs
* deep re-inforcement learning
* etc.

VAEs

A generation model "framework":
- encoder
- hidden state
- decoder



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



(b) Digram of dilated CNN decoder.

NLG Evaluation

(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/

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

LMs:
http://www.dhgarrette.com/nlpclass/notes/ngrams.pdf
http://www.foldl.me/2014/kneser-ney-smoothing/
http://ofir.io/Neural-Language-Modeling-From-Scratch/
https://slides.com/oleksiysyvokon/lm-advances

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
https://www.salesforce.com/products/einstein/ai-research/tl-dr-reinforced-model-abstractive-summarization/
https://medium.com/@yoav.goldberg/an-adversarial-review-of-adversarial-generation-of-natural-language-409ac3378bd7