RITAL

Information retrieval and natural language processing
Recherche d'information et traitement automatique de la langue

Master 1 DAC, semestre 2

Nicolas Thome

SCIENCES SORBONNE UNIVERSITÉ

ISIR INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE

MLIA Machine Learning & Deep Learning for Information Access

Word Embeddings

Outline

Textual data = different analysis level

- At the **document** scale:
  - ⇒ Several efficient algorithm to classify/categorize
- At the **sentence** scale:
  - Segmentation
  - POS, NER, SRL...
  - ⇒ Sequential approaches (HMM, CRF)
- At the **word** scale:
  - Defining / understanding the word
  - Linking the word with antonym, synonym, etc, ...
  - Binding the word to a lexical field
  - ⇒ Latent semantics / linguistic resources

⇒ How deep learning / representation learning can improve our previous solutions?

- Simplest encoding of text inputs: **one-hot representation**
- Binary vector of vocabulary size $|V|$, with 1 corresponding to term index
- $|V|$ small for chars (~ 10), large for words ($\sim 10^4$), huge for sentences
- Basis for constructing Bag of Word (BoW) Models

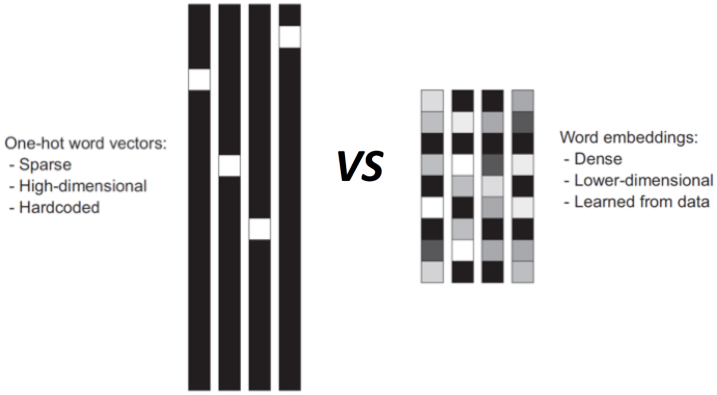
the dog is on the table

- Handcrafted feature used with ML shallow models, *e.g.* kernels methods
- Still very competitive for some NLP tasks, *e.g.* text topic classification
- Can be extended to (bags of) bi-grams for *e.g.* language identification

- Limitation: $\langle\ r(\text{"}motel\text{"})\ ;\ r(\text{"}hotel\text{"})\ \rangle = 0$

motel [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0] AND
hotel [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0] = 0

- Text embedding motivation: extract representation reflecting semantic similarities between text primitives ("Tokens")



- Learn mapping from one-hot encoding to a smaller vectorial space
- **General idea:** representing a word by means of its neighbors

"You shall know a word by the company it keeps"

(J. R. Firth 1957: 11)

- **Distributional Hypothesis:** One of the most successful ideas of modern statistical NLP

Word that appear in similar contexts in text tend to have similar meanings

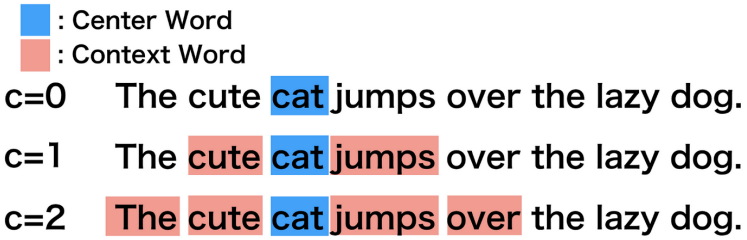
he curtains open and the moon shining in on the barely
ars and the cold , close moon " . And neither of the w
rough the night with the moon shining so brightly , it
made in the light of the moon . It all boils down , wr
surely under a crescent moon , thrilled by ice-white
sun , the seasons of the moon ? Home , alone , Jay pla
m is dazzling snow , the moon has risen full and cold
un and the temple of the moon , driving out of the hug
in the dark and now the moon rises , full and amber a
bird on the shape of the moon over the trees in front
But I could n't see the moon or the stars , only the
rning , with a sliver of moon hanging among the stars
they love the sun , the moon and the stars . None of
the light of an enormous moon . The plash of flowing w
man 's first step on the moon ; various exhibits , aer
the inevitable piece of moon rock . Housing The Airsh
oud obscured part of the moon . The Allied guns behind

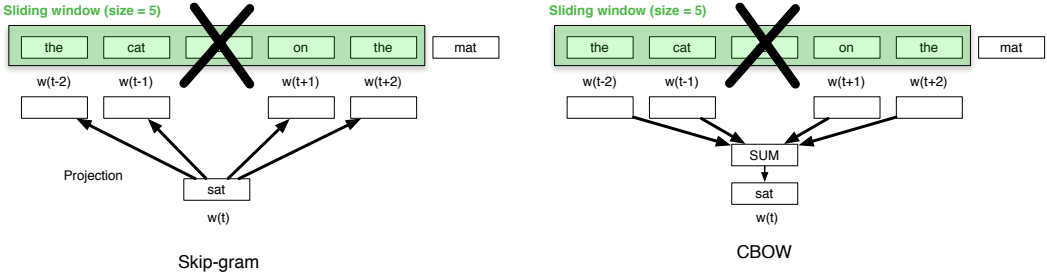
- Simplest historical strategy to represent context: co-occurrence matrices

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

- Dimensionality explosion $\mathcal{O}(|V|^2) \Rightarrow$ memory & statistical robustness
- Use SVD to reduce dimension $\Rightarrow$ dense low-dimensional vector
  - Still, scalability with SVD

- Modern approaches: directly learn low-dim text vectors
- **Word2vec** [MSC+13]: similar words $\Leftrightarrow$ similar contexts
  - Predict surrounding words (context) from central word: **CBoW**
  - Predict context from central word: **Skipgram**
- N.B.: cast unsupervised task as supervised one: "auto-supervision" (more next course) $\neq$ recontruction / ML, *e.g.* SVD



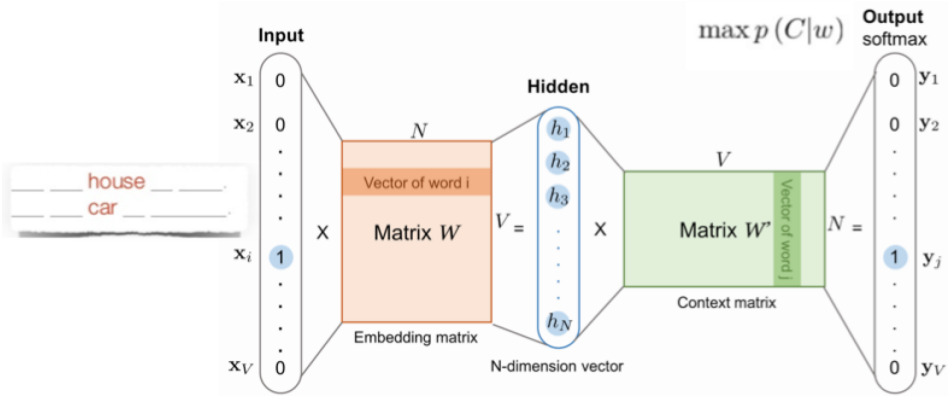


Skip-Gram (& *negative sampling* implementation) : easy learning

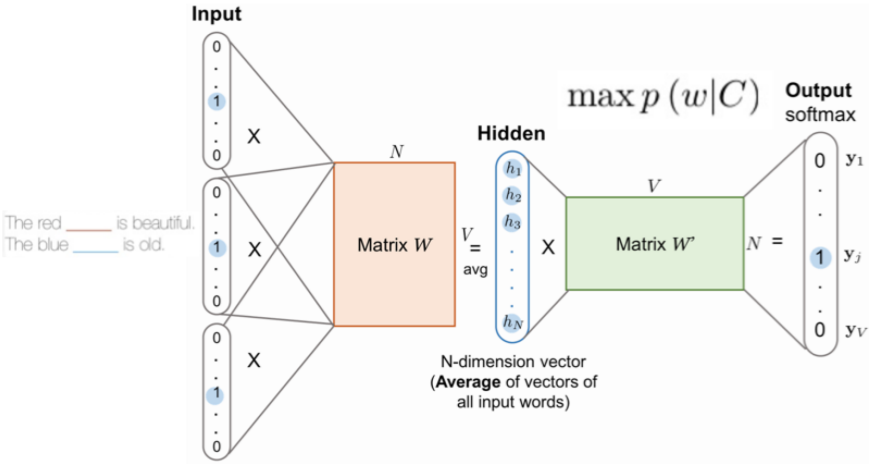
- Local analysis
- *predictive* criterion : estimating missing value
- Sliding window = local context C of word w

Mikolov, Sutskever, Chen, Corrado, Dean, NIPS 2013 (arXiv 2012)
Distributed representations of words and phrases and their compositionality

- From a word $\boldsymbol{w}$, one-hot encoded (size $|V|$) $\Rightarrow$ infer its context C
- Project $\boldsymbol{w}$ into latent space $\boldsymbol{h}$, size $N \Rightarrow$ matrix $\boldsymbol{W}$ (select j^{st} column)
- Project $\boldsymbol{h}$ back into $|V|$ space $\Rightarrow$ matrix $\boldsymbol{W}'$ + soft-max
- Loss function: average cross-entropy for randomly context words



- From a context C one-hot encoded (size $|V|$) $\Rightarrow$ infer central word $\boldsymbol{w}$
- Encode C into latent space $\boldsymbol{h}$ + average (or $\sim$ project avg C $\Leftrightarrow$ BoW)
- Decode $\boldsymbol{h}$ back into $|V|$ space + soft-max
- Loss function: cross-entropy for central word



- Skipgram and CBoW trained with back-prop (see next)
- Soft-max : normalization over a huge vocabulary $|V|$:

$$\mathcal{L}(\boldsymbol{W}) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} log\left(\frac{exp(v'_{w_{t+j}} v_{w_t})}{\sum_i exp(v'_{w_i} v_{w_t})}\right)$$ Options:

  - Hierarchical soft-max
  - Use sigmoid instead + negative sampling (see next)
- CBoW works well for frequent words, Skipgram for rare words

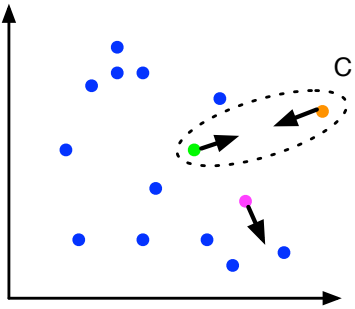
Conclusion

- Unsupervised: can be trained on huge generalist corpus
  - Transfer and fine-tuning possible on specific supervised tasks
    - Word2Vec and Glove $\Leftrightarrow$ VGG or ResNet for vision
    - BUT only one layer transfer
- Extension of Skipgram for sentences Skip-Thought Vectors [KZS+15]
  - predict the surroundings sentences of a given sentence
  - Extended to discriminative learning recently [LL18] $\Rightarrow$ faster training

- Given word w and local contexts C:

$$\text{Idée SG: } \arg\max_{\theta} \prod_{C} \prod_{w \in C} p(C|w;\theta)$$

- $p(D = 1|w_i, w_j; \theta) \Rightarrow$ proba. that w_i and w_j occur in the same context



$$\arg\max_{\theta} \prod_{i,j \in C} p(D = 1|w_i, w_j; \theta) + \underbrace{\prod_{i,j \in \bar{C}} p(D = 0|w_i, w_j; \theta)}_{\text{Negative Sampling}}$$

Goldberg, Levy, arXiv 2014
word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method

Hammer, NN 2002
Generalized Relevance Learning Vector Quantization

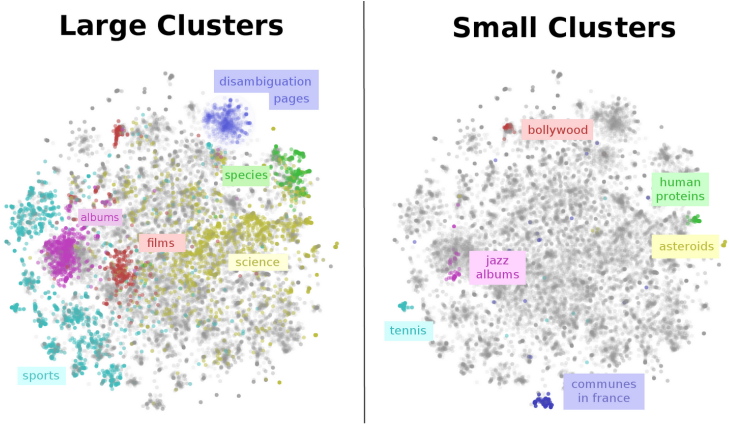
$$\arg\max_{\theta} \prod_{i,j \in C} p(D=1|w_i, w_j; \theta) \underbrace{+ \prod_{i,j \in \bar{C}} p(D=0|w_i, w_j; \theta)}_{\text{Negative Sampling}}$$

- Using logistic function : $p(D=1|w_i, w_j) = \frac{1}{1+\exp(-z_i z_j)}$
- Global log-likelihood : $\arg\max_{Z} \left(\sum_{i,j \in C} \log \sigma(z_i \cdot z_j) + \sum_{i,j \in \bar{C}} \log \sigma(-z_i \cdot z_j) \right)$

σ : fct sigmoide, C : Set of Cooccurences, $\bar{C}$: Set of Non-Cooc

- Stochastic Gradient Descent + triplet loss
- Frequent word subsampling trick : picking words with $p(w_i) = 1 - \sqrt{\frac{t}{freq(w_i)}}, \qquad t = 10^{-5}$

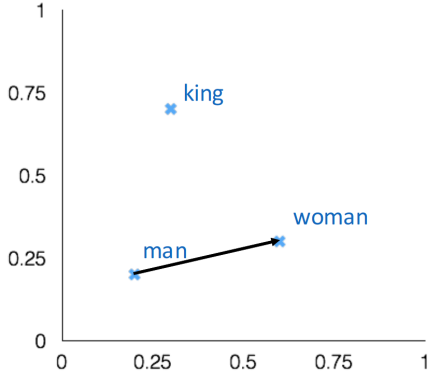
- Visualization of the latent space: t-SNE
- Semantic clustering of concept
- Ex with paragraph word2vec trained on wikipedia: see
  http://colah.github.io/posts/2015-01-Visualizing-Representations/



- Word Vector Analogies: a:b :: c:?, *e.g.* man:woman:: king:?
  $\Rightarrow$ **map the relation between a and b to c**

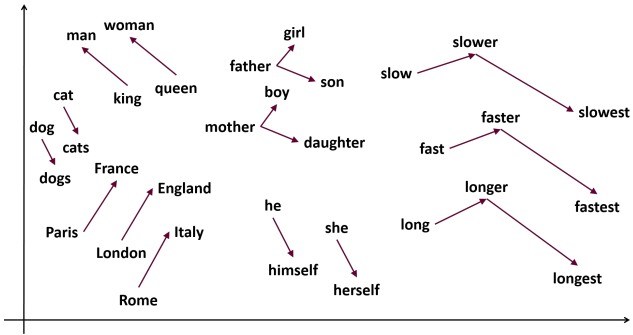
- Assumption: can be done with simple algebraic operations (sum, subtraction): $r(c) + r(b) - r(a)$
- Disentangling in the learning representation space

$$d = \arg\max_i \frac{[r(c) + r(b) - r(a)]^T r(i)}{||r(c) + r(b) - r(a)||}$$



a is to *b* what *c* is to ??? $\Leftrightarrow$ $z_b - z_a + z_c$

Syntactical property (1):



$z_{woman} - z_{man} \approx z_{queen} - z_{king}$

$z_{kings} - z_{king} \approx z_{queens} - z_{kings}$

Query:

$z_{woman} - z_{man} + z_{king} = z_{req}$

Nearest neighbor:

$\underset{i}{\text{argmin}} \|\boldsymbol{z}_{req} - \boldsymbol{z}_i\| = \text{queen}$

a is to b what c is to ??? $\Leftrightarrow$ $z_b - z_a + z_c$

Syntactical property (2): Query:

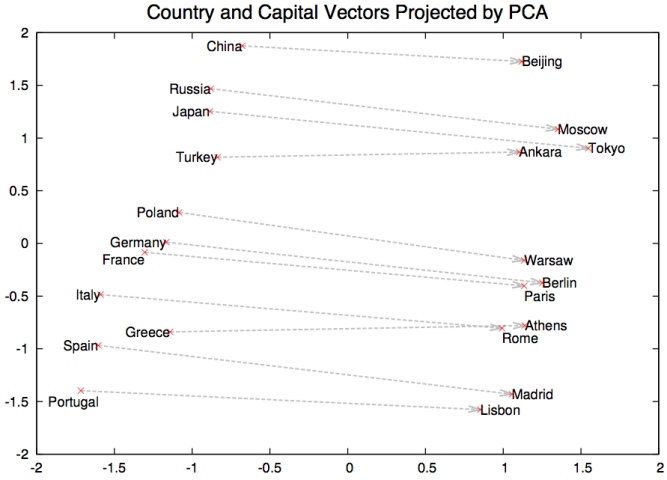
$\boldsymbol{z}_{easy} - \boldsymbol{z}_{easiest} + \boldsymbol{z}_{luckiest} = \boldsymbol{z}_{req}$

Nearest neighbor:

$\underset{i}{\text{argmin}} \, \|\boldsymbol{z}_{req} - \boldsymbol{z}_i\| = \text{lucky}$

a is to *b* what *c* is to ??? $\Leftrightarrow$ $z_b - z_a + z_c$

Semantic Property (1):



a is to b what c is to ??? $\Leftrightarrow$ $z_b - z_a + z_c$

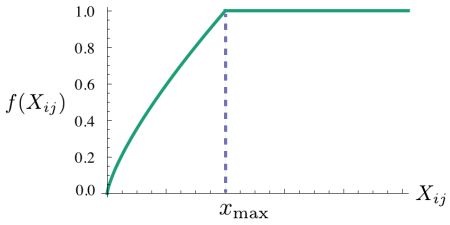
Semantic Property (2)

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De

Table 5: Vector compositionality using element-wise addition. Four closest tokens to the sum of two vectors are shown, using the best Skip-gram model.

- GloVe: Global Vectors for Word Representation
- Based on co-occurrence matrix $\boldsymbol{X}$ $\Rightarrow$ leverage global context
  - Based on ratio of co-occurrence probabilities
- Explicitly enforcing semantic embedding, *i.e.* $\boldsymbol{w}_i^T \boldsymbol{w}_j \approx log(X_{ij})$:

$$J = \sum_{i,j=1}^{|V|} f(X_{ij}) \left(\boldsymbol{w}_i^T \boldsymbol{w}_j - log(X_{ij}) \right)$$



- $\boldsymbol{w}_i$ main word embedding, $\boldsymbol{w}_j$ context embedding ($\boldsymbol{X}$ computed on local windows)
- $f(X_{ij}) = \begin{cases} (\frac{x}{x_{max}})^{\alpha} & \text{if } x \leq x_{max} \\ 1 & \text{otherwise} \end{cases}$

Why is it working?

- Predicting instead of counting... Not the good explanation!
- Learning a local semantics !

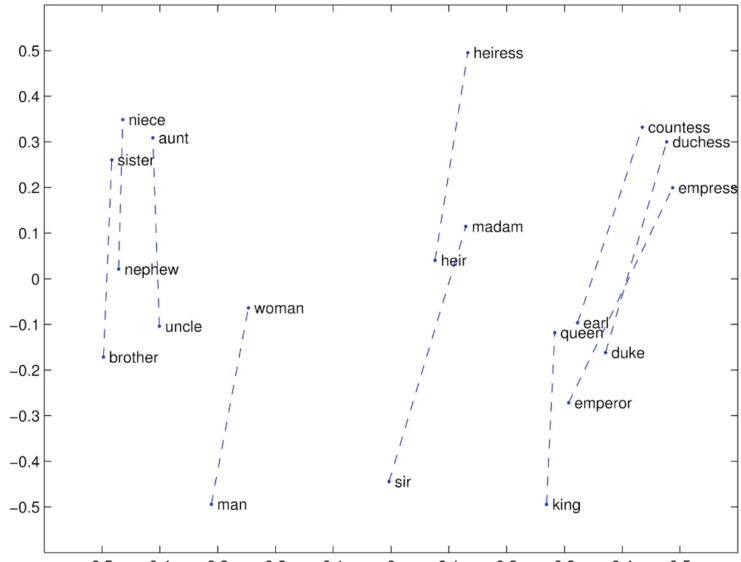
$\Rightarrow$ GloVe $\approx$ PLSA + local context + embedding based implem.

- $X \in \mathbb{R}^{V \times V}$ word co-occurrence matrix
- X_{ij} frequency of word i co-occurring with word j
- $X_i = \sum_k X_{ik}$ total number of occurrences of word i in corpus
- $P_{ij} = P(j|i) = \frac{X_{ij}}{X_i}$ a.k.a. probability of word j occurring within the context of word i
- $w \in \mathbb{R}^z$ a word embedding of dimension z
- $\tilde{w} \in \mathbb{R}^z$ a context word embedding of dimension z

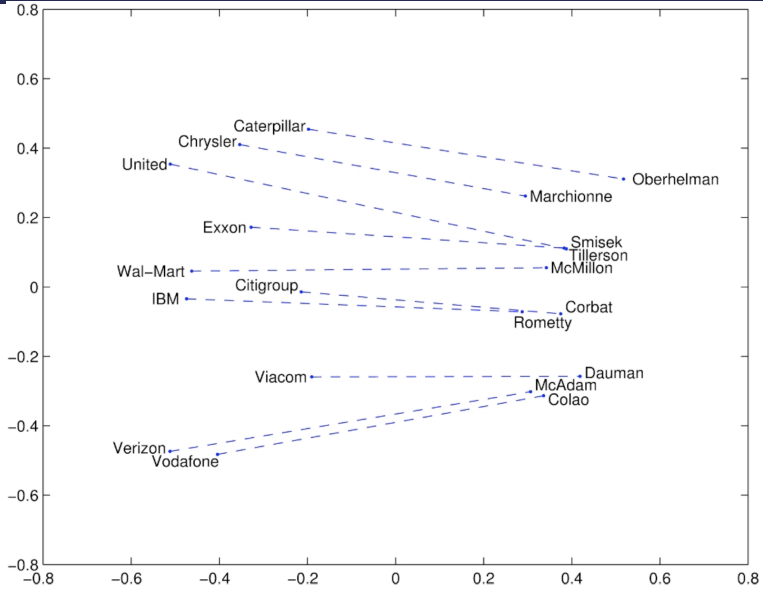
Baroni, Dinu & Kruszewski, ACL 2014
Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors

Pennington, Socher & Manning, EMNLP 2014
Glove: Global Vectors for Word Representation

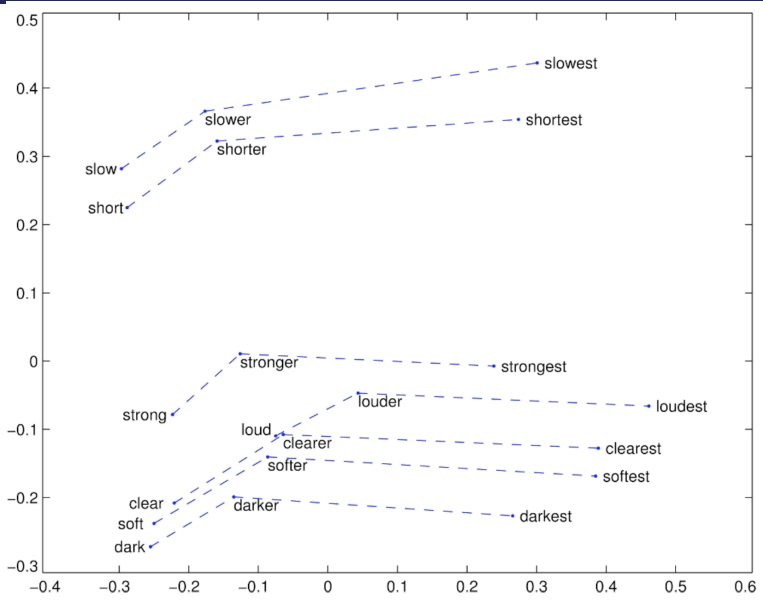
GlovE Analogies Examples: Man - Woman



GlovE Analogies Examples: Company - CEO

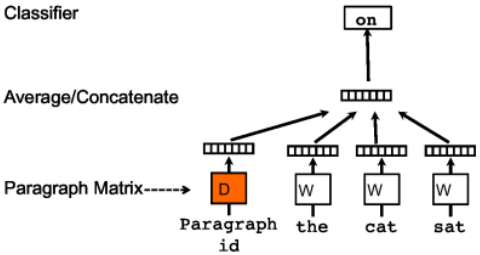


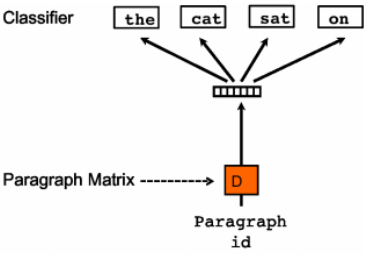
GlovE Analogies Examples: Superlatives



One remaining question: we learned a powerful semantics at the **word level**... How scaling to the **sentence** or **document level**?

One remaining question: we learned a powerful semantics at the **word level**... How scaling to the **sentence** or **document level**?





Q. Le, T. Mikolov, ICML 2014
Distributed representations of sentences and documents

One remaining question: we learned a powerful semantics at the **word level**... How scaling to the **sentence** or **document level**?

Or simple averaging of word embeddings:

- \+ great results on small word groups
- − poor results on larger groups
  - quickly converge to a central abstract point of the latent space

One remaining question: we learned a powerful semantics at the **word level**... How scaling to the **sentence** or **document level**?

1. Aggregate multiple words associated to a single entity
   - *Pointwise Mutual Information* threshold:

$$\text{score}(w_i, w_j) = \frac{\text{count}(w_i w_j) - \delta}{\text{count}(w_i) \times \text{count}(w_j)}.$$

2. Include new terms in the dictionary before running word2vec

Newspapers			
New York	New York Times	Baltimore	Baltimore Sun
San Jose	San Jose Mercury News	Cincinnati	Cincinnati Enquirer
NHL Teams			
Boston	Boston Bruins	Montreal	Montreal Canadiens
Phoenix	Phoenix Coyotes	Nashville	Nashville Predators
NBA Teams			
Detroit	Detroit Pistons	Toronto	Toronto Raptors
Oakland	Golden State Warriors	Memphis	Memphis Grizzlies
Airlines			
Austria	Austrian Airlines	Spain	Spainair
Belgium	Brussels Airlines	Greece	Aegean Airlines
Company executives			
Steve Ballmer	Microsoft	Larry Page	Google
Samuel J. Palmisano	IBM	Werner Vogels	Amazon

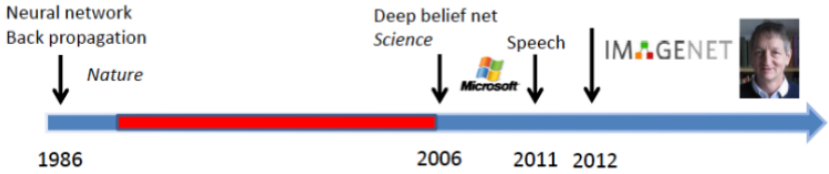
- FastText [JGBM17]
- ELMo [PNI+18]
  - Standard word embeddings (word2Vec, Glove), word embedding context independent, *e.g.* "stick"
  - ELMo: an embedding based on the context it's used in - to both capture the word meaning in that context as well as other contextual information
- BERT [DCLT18]: Bidirectional Encoder Representations from Transformers (next)

Deep Learning in NLP

- Before 90's: linguistic, regex and simple statistical models (Naive Bayes)
- 90's: BoW, SVM and sequence models CRF
- 2000-2010: transition, seminal neural networks works for NLP
- 2010-now: renewal of deep learning & representation learning

- 1990 Matrix factorization :
  - In NLP $\Rightarrow$ SVD / PCA [Deerwester, 1990]
- 2005 First neural architecture for text representation
  - an alternative to PLSA [Keller, 2005]
- 2008 Convolutional Neural architecture for text
  - precursor of modern architectures
  - multi-tasks [Collobert, 2008]
- 2012 The Word2Vec wave
  - Qualitative & cheap word embeddings [Mikolov, 2012]
- 2013 Manifest for representation learning [Bengio, 2013]
- 2015 Seq2seq paradigm [Luong, 2015]
- 2017 Attention & Transformers [Vaswani, 2017]

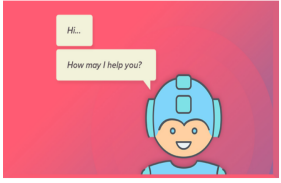
- 90's / 2000's: difficult to train large deep models on existing databases

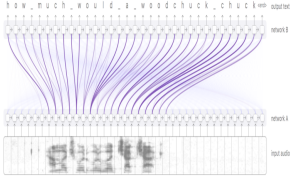


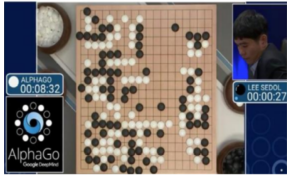
- **ILSVRC'12: the deep revolution**
  ⇒ outstanding success of ConvNets [KSH12]

Rank	Name	Error rate	Description
1	**U. Toronto**	0.15315	Deep learning
2	U. Tokyo	0.26172	Hand-crafted features and learning models. Bottleneck.
3	U. Oxford	0.26979	
4	Xerox/INRIA	0.27058	

- Image classification, speech recognition
- chatbots, translation,
- Games, robotics







- **The formal Neuron**

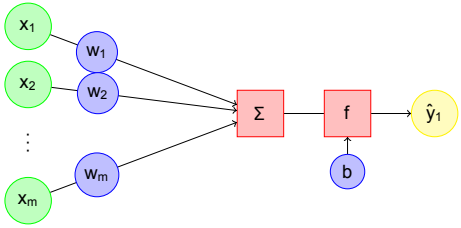
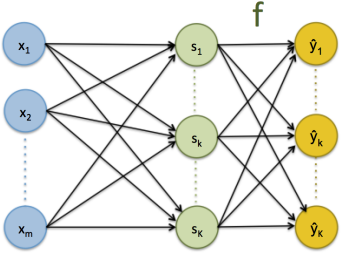


Figure 1: Formal neuron – Credits: R. Herault

x_i: inputs
w_i, b: weights
f: activation function
y: output of the neuron

$$y = f(w^\top x + b)$$

- **Neural Networks:** Stacking several formal neurons ⇒ **Perceptron**

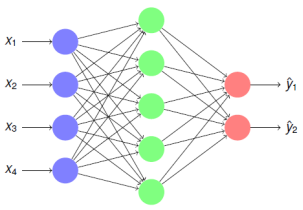


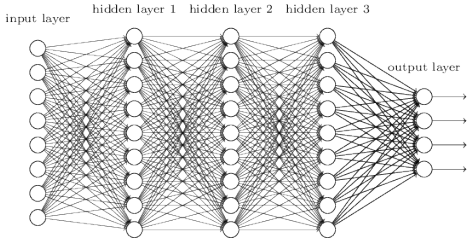
- **Soft-max Activation**:

$$\hat{y}_k = f(s_k) = \frac{e^{s_k}}{\sum_{k'=1}^{K} e^{s_{k'}}}$$

⇒ **Logistic Regression (LR) Model !**

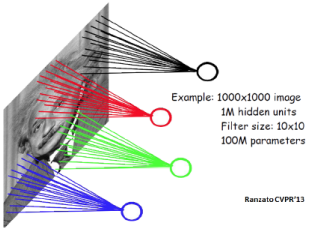
- **Multi-Layer Perceptron (MLP):** Stacking layers of neural networks
  - More complex and rich functions / Logistic Regression (LR)
  - **Neural network with one single hidden layer $\Rightarrow$ universal approximator** [Cyb89]

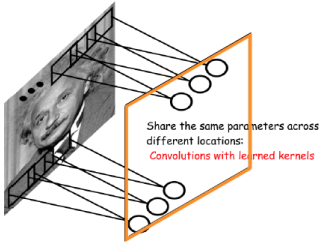




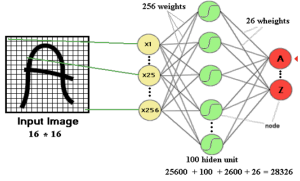
- **Basis of the "deep learning" field**
  - **Hidden layers: intermediate representations from data**
  - **Can be learned with Backpropagation algorithm [Lec85, RHW86]** (chain rule)

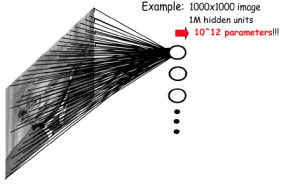
- **ConvNets:** sparse connectivity + shared weights



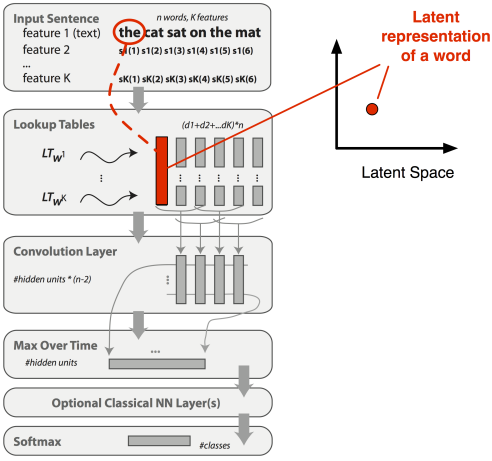


- **Local feature extraction ($\neq$ FCN)**
- Overcome parameter explosion for FCN on images



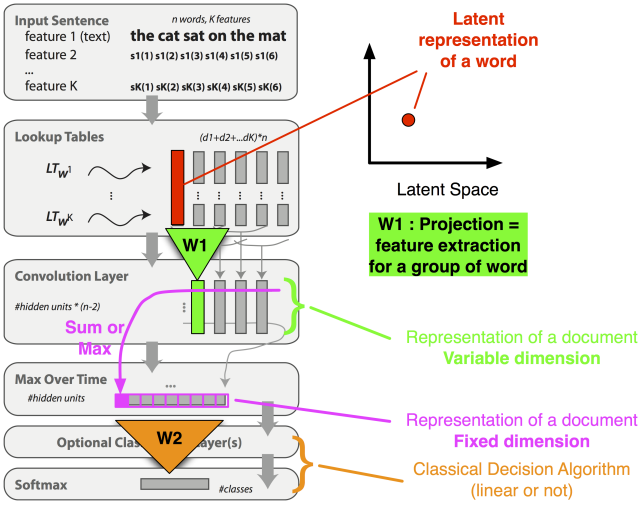


- **Lookup table** concept
  - W_0 matrix
  - $\rightarrow$ **Table of embeddings**
- 2008: state of the art on POS, NER, SRL
- Quite difficult to set...
  - ... But open source + **open embeddings**
- Based on torch... By Collobert



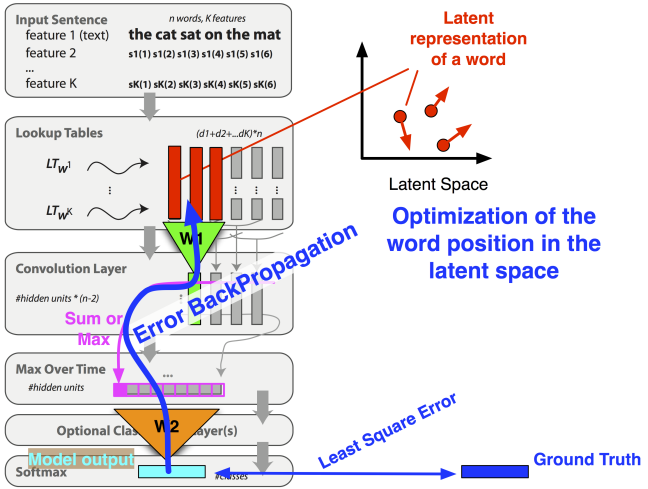
R. Collobert, J. Weston ICML 2008
A unified architecture for natural language processing: Deep neural networks with multitask learning

- **Embedding** matrix W_0
- **Convolutional weights** W_1
  - Limiting number of paramters
- **Max pooling**
  - Single document representation
- **Prediction** W_2
  - Trained for different tasks: POS, NER, *etc*



R. Collobert, J. Weston ICML 2008
A unified architecture for natural language processing: Deep neural networks with multitask learning

- **End-to-end training** with backpropagation
- **All weights W_2, W_1, W_0 optimized during training**
- Based on error on a given target task



R. Collobert, J. Weston ICML 2008
A unified architecture for natural language processing: Deep neural networks with multitask learning

Several important informations

- Embedding are learned **keeping the sentence structure**
- Embeddings benefit from multi-tasks
- Learning is slow...

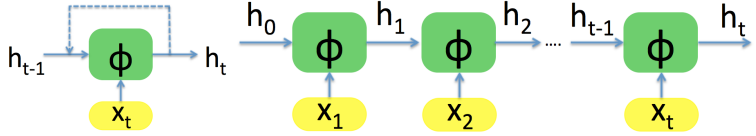
Our embeddings have been trained for about 2 months, over Wikipedia.
`https://ronan.collobert.com/senna/`

- ... But inference is fast & efficient

Task	Benchmark		Performance	Timing (s)
Part of Speech (POS)	(Toutanova et al, 2003)	(Accuracy)	97.29%	3
Chunking (CHK)	CoNLL 2000	(F1)	94.32%	2
Name Entity Recognition (NER)	CoNLL 2003	(F1)	89.59%	2
Semantic Role Labeling (SRL)	CoNLL 2005	(F1)	75.49%	36
Syntactic Parsing (PSG)	Penn Treebank	(F1)	87.92%	74

- + Costly embeddings have been made available to the community

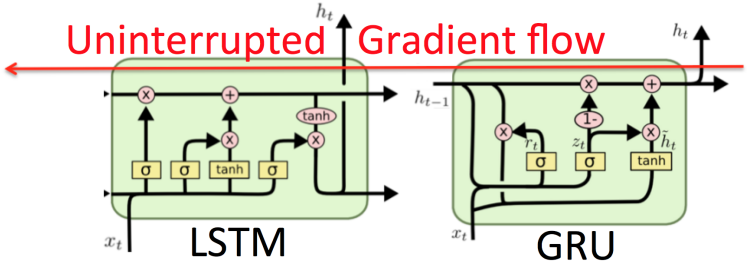
- **RNN Cell:** $\boldsymbol{h}_t = \phi(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}) = f(\mathrm{U}x_t + \mathrm{W}h_{t-1} + b_h)$ [Elm90]
  - $\boldsymbol{h}_t$: **network memory up to time t ⇒ Sequence processing**
  - **Same idea than HMMs, CRFs, but end-to-end training of representation**



Folded RNN Unfolded RNN

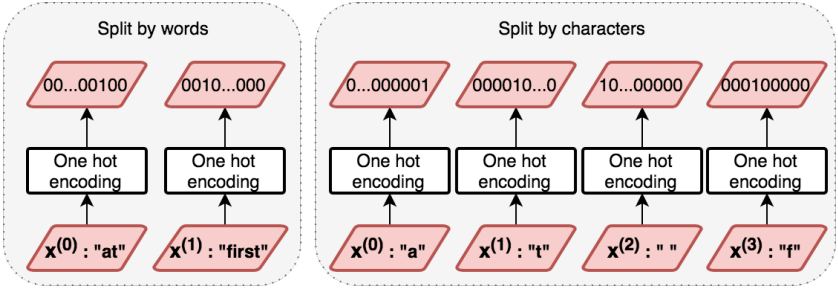
- **Specific architectures for vanishing gradients**

LSTM [HS97],
GRU [CvMG+14]



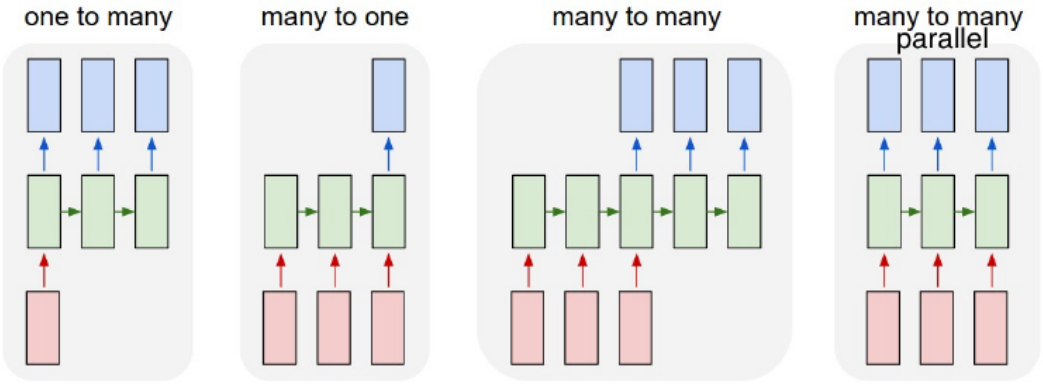
Deep NLP strategy:

1. Extracts text input, "tokens", *e.g.* characters or words
2. One hot encoding of tokens



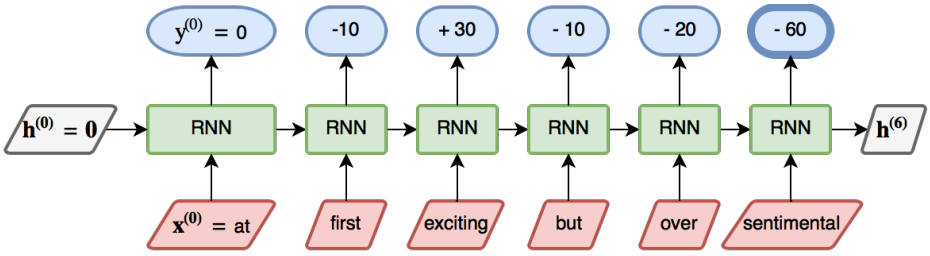
3. Split the text into a "temporal" sequence
4. **RNN to model the temporal structure**
   - Option: use an embedding layer on top of one-hot encoding

Different uses for different applications

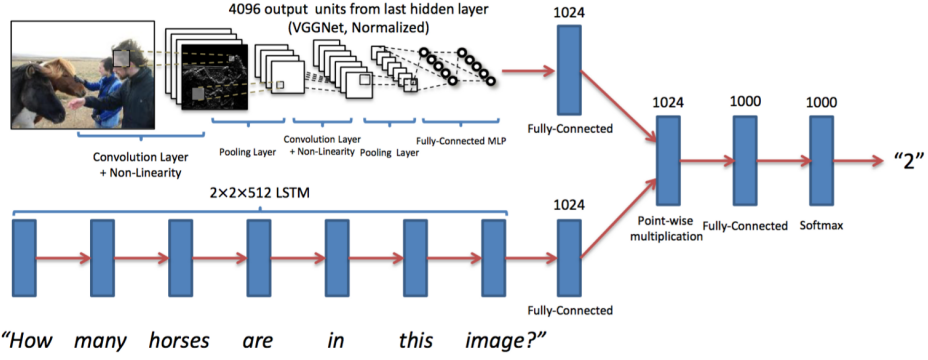


Karpathy's blog `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`

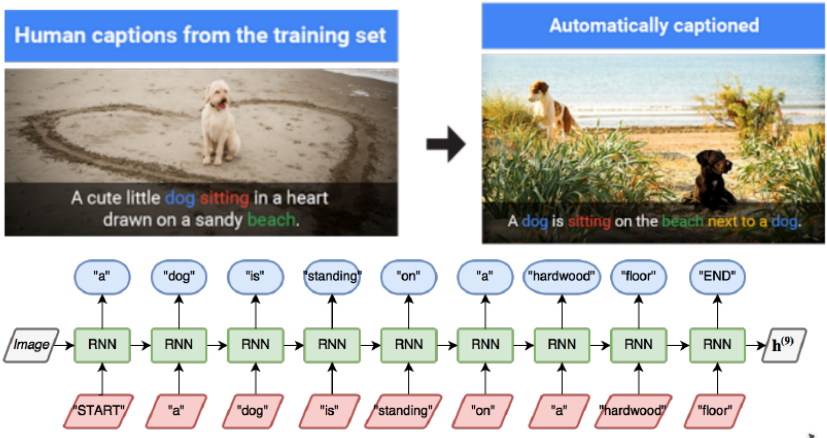
- Sentiment classification
  - Input: "At first exciting but over sentimental"
  - Output: -60 = Bad review



- Question Answering and Visual Question Answering (VQA)

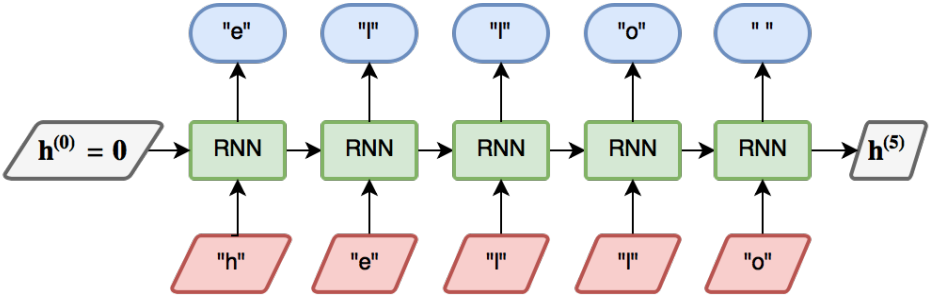


- Image captioning



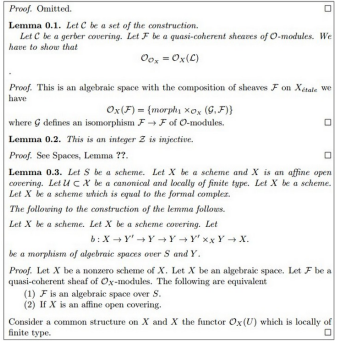
[KL15]

- **Text Generation**, *e.g.* Char-nn
  - Input sequence of characters (Token $\Leftrightarrow$ char)
  - Output: next character
- Many-to-many parallel
  - In practice: many-to-one: predict next character from previous (K) chars



[Kar15]

- Char-nn: applied to raw text, *e.g.* poetry (practical session)
  - Char-nn: learns to correctly spell a given language, although semantic meaning of sentences more challenging
  - Capacity to learn language structural/syntactical rules
    $\Rightarrow$ applications for generating source code, *e.g.* wikipedia pages, XML, Latex, linux source code (C), *etc*
    See here for other examples



Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*
Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y' \times_X Y \to X.$$

be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram

$$\begin{array}{ccc} \mathcal{S} & & \\ \downarrow & & \\ \mathcal{C} & \to & \mathcal{O}_{Y'} \\ & \searrow & \\ & & \mathrm{gor}_s \end{array} \qquad \begin{array}{ccc} & = a' & \\ & \to & \\ = a' & & \\ & \to & a \end{array} \qquad \begin{array}{c} X \\ \downarrow \\ \mathrm{Mor}_{Sets} \quad d(\mathcal{O}_{X_{spa}}, \mathcal{G}) \end{array}$$

$\mathrm{Spec}(K_{\mathcal{G}})$

is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_a. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

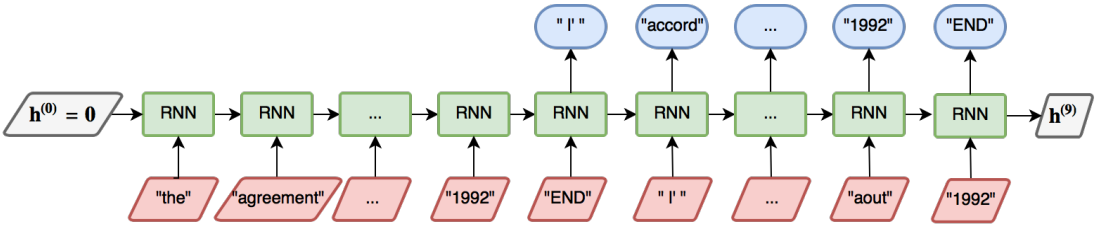
Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced* above we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \to \mathcal{F}_{\overline{y}} \quad \text{-1}(\mathcal{O}_{X_{\acute{e}tale}}) \to \mathcal{O}^*_X(\mathcal{O}_{X_i}(\mathcal{O}^*_{X_i}))$$

is an isomorphism of covering of $\mathcal{O}_{X_i}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

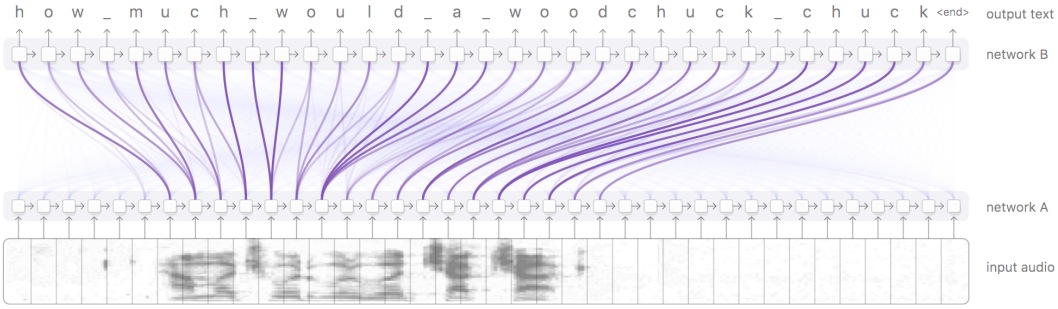
If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_i}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

- **Machine Translation text2text**
  - Input: "The agreement on the European Economic Area was signed in August 1992."
  - Output: "L'accord sur la zone économique européenne a été signé en août 1992."



[BCB14] [OC16]

- **Machine Translation speech2text**
  - Input : Audio mp3 (speech utterance)
  - Output: "How much would a woodchuck chuck"

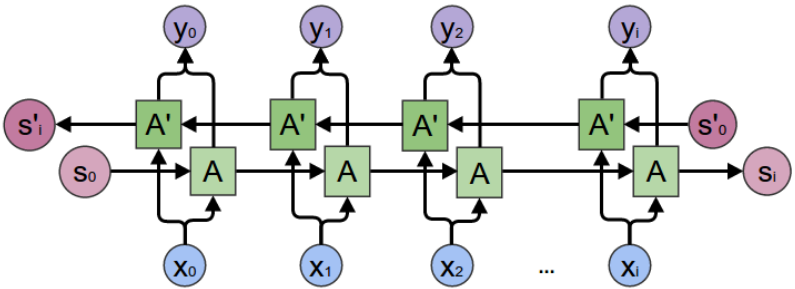


[CJLV15] [OC16]

LSTM

\+ Sequential modeling

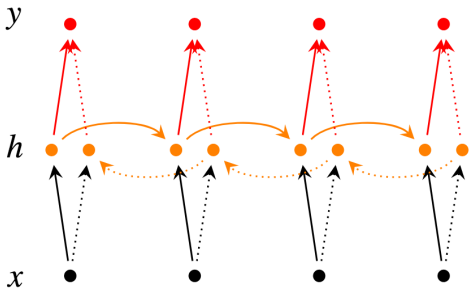
\− Sequential dependencies ! = partial modeling



Bi-dimensional representation $[S_1, S'_1]$ is more powerful representation of the sentence S than each single representation.

Classical notation: $s = [\overrightarrow{s}, \overleftarrow{s}]$

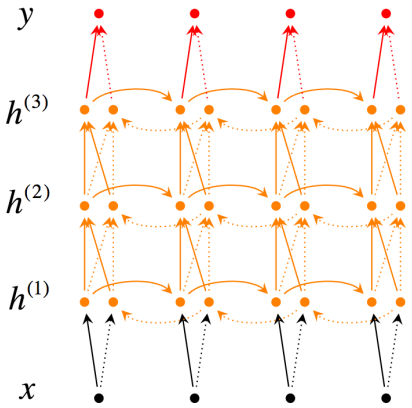
- For classification, incorporate information from words both preceding and following



$$\vec{h}_t = f(\vec{W}x_t + \vec{V}\vec{h}_{t-1} + \vec{b})$$
$$\overleftarrow{h}_t = f(\overleftarrow{W}x_t + \overleftarrow{V}\overleftarrow{h}_{t+1} + \overleftarrow{b})$$
$$y_t = g(U[\vec{h}_t; \overleftarrow{h}_t] + c)$$

$h = [\vec{h}; \overleftarrow{h}]$ now represents (summarizes) the past and future around a single token.

- Deep Bi-directionnal RNNs



$$\overrightarrow{h}_t^{(i)} = f(\overrightarrow{W}^{(i)} h_t^{(i-1)} + \overrightarrow{V}^{(i)} \overrightarrow{h}_{t-1}^{(i)} + \overrightarrow{b}^{(i)})$$

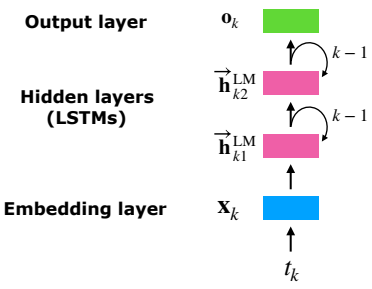
$$\overleftarrow{h}_t^{(i)} = f(\overleftarrow{W}^{(i)} h_t^{(i-1)} + \overleftarrow{V}^{(i)} \overleftarrow{h}_{t+1}^{(i)} + \overleftarrow{b}^{(i)})$$

$$y_t = g(U[\overrightarrow{h}_t^{(L)}; \overleftarrow{h}_t^{(L)}] + c)$$

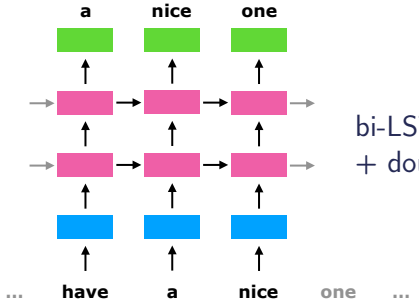
Each memory layer passes an intermediate sequential representation to the next.

Static word embeddings $\Rightarrow$ **Mapping function** = dynamic embeddings

The forward LM architecture



Expanded in the forward direction of *k*



bi-LSTM architecture
+ double hidden layer

M. E. Peters et al.arXiv 2018 Deep contextualized word representations

Static word embeddings ⇒ **Mapping function** = dynamic embeddings

ELMo represents a word t_k as a linear combination of corresponding hidden layers (inc. its embedding)

ELMo is a task specific representation. A down-stream task learns weighting parameters

$$\mathbf{ELMo}_k^{\text{task}} = \gamma^{\text{task}} \times \sum \left\{ s_2^{\text{task}} \times \mathbf{h}_{k2}^{\text{LM}},\ s_1^{\text{task}} \times \mathbf{h}_{k1}^{\text{LM}},\ s_0^{\text{task}} \times \mathbf{h}_{k0}^{\text{LM}} \right.$$

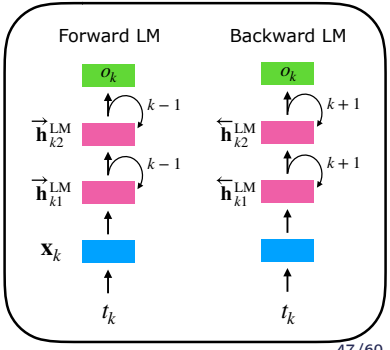
$$\left. ([\mathbf{x}_k ; \mathbf{x}_k]) \right.$$

Concatenate hidden layers

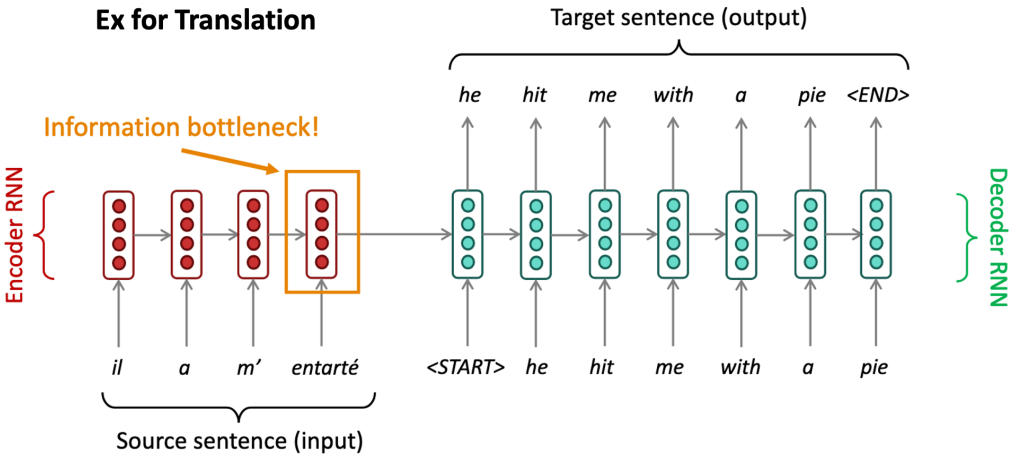
$$[\overrightarrow{\mathbf{h}}_{kj}^{\text{LM}} ; \overleftarrow{\mathbf{h}}_{kj}^{\text{LM}}]$$

Unlike usual word embeddings, ELMo is assigned to every *token* instead of a *type*

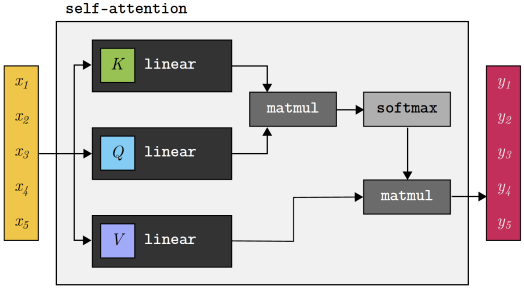
biLMs

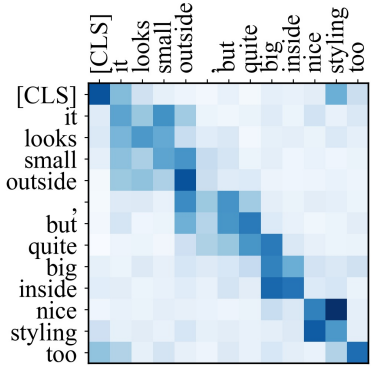


- **Global context:** very important in NLP, **BUT ConvNets:** only local information
- **RNNs:** can in theory encode long-range dependencies **BUT:**
  - Vanishing gradients, bottleneck (last word representation), computation issues (parallelization)



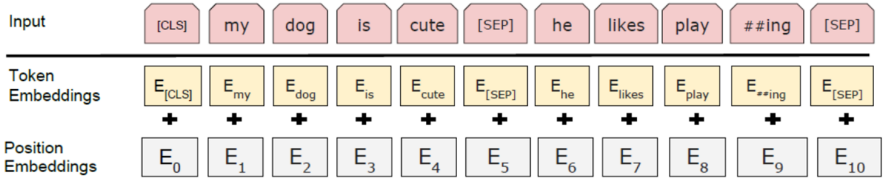
- Transformers: only fully connected layers ($\neq$ ConvNets, RNNs)
- Attention: for a set of input tokens $\boldsymbol{X} := \{\boldsymbol{x}_i\}_{i \in \{1;T\}}$, $\boldsymbol{x}_i \in \mathbb{R}^d$, learn 3 matrix weights:
  - $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_k$, $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_q$, $\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_v$ ($\boldsymbol{K}$ Keys, $\boldsymbol{Q}$ Queries, $\boldsymbol{V}$ Values)
  - $\boldsymbol{A} = \boldsymbol{K}^T\boldsymbol{Q}$ and $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{V}$
    - $\boldsymbol{y}_j = \sum_{i=1}^{T} \boldsymbol{v}_j a_{i,j}$
- **[CLS] token: global description of doc**



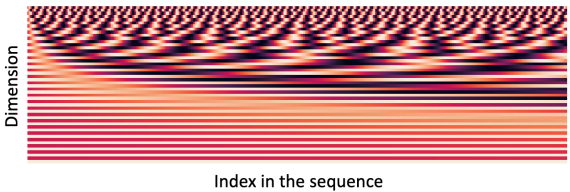


Transformers: positional encoding

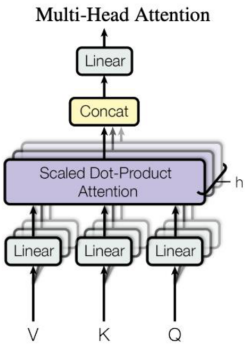
- Fully connected layers: permutation invariant ⇒ no info about each token position!
- Adding positional encoding, handcrafted (sinusoid), or learned



$$p_i = \begin{pmatrix} \sin(i/10000^{2*1/d}) \\ \cos(i/10000^{2*1/d}) \\ \vdots \\ \sin(i/10000^{2*\frac{d}{2}/d}) \\ \cos(i/10000^{2*\frac{d}{2}/d}) \end{pmatrix}$$



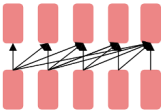
- **High-Level Idea:** Let's perform self-attention multiple times in parallel and combine the results.



[Vaswani et al. 2017]

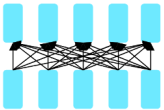
Wizards of the Coast, Artist: Todd Lockwood

Credit: Anna Goldie



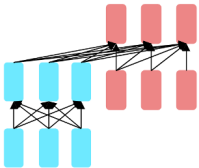
Decoders

- Language models!
- Nice to generate from; can't condition on future words
- **Examples:** GPT-2, GPT-3, LaMDA



Encoders

- Gets bidirectional context – can condition on future!
- Wait, how do we pretrain them?
- **Examples:** BERT and its many variants, e.g. RoBERTa

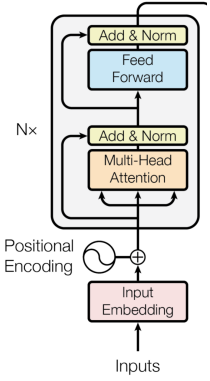


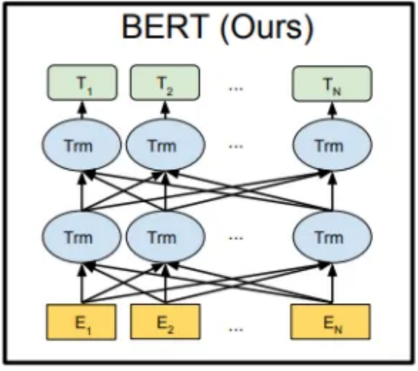
Encoder-Decoders

- Good parts of decoders and encoders?
- What's the best way to pretrain them?
- **Examples:** Transformer, T5, Meena

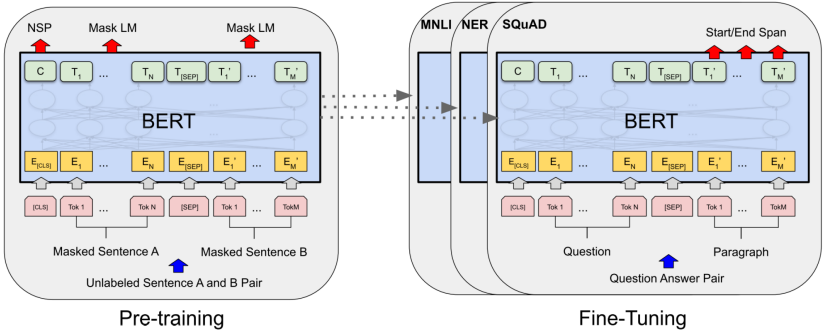
Credit: Anna Goldie

- Transformer block: MHA (+norm) + FFN (+norm), & residual connections
- BERT [DCLT18]: Bidirectional Encoder Representations from Transformer
  - BERT: deep cascade of transformers *Tm*
  - Context-based embedding ($\neq$ Word2vec, Glove), bi-directional embedding ($\neq$ ELMO)

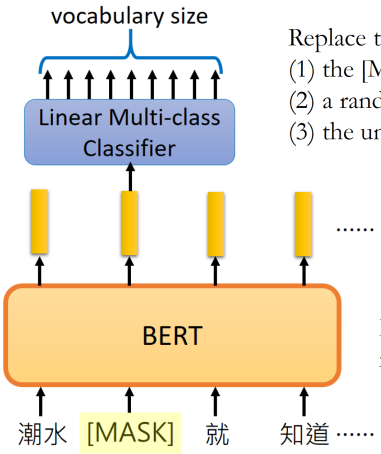




- Pre-training on unlabeled data: huge data ($\sim$ ImageNet in vision)
  - Next Sequence Prediction (NSP)
  - Masked Language Model (MLM)
- Fine-tuning on downstream task
  - **[CLS] token: can be used to provide a global sentience representation**
  - **Leverages learned attention for effective pooling (better than avg, max)**



Credit: Y. Fang

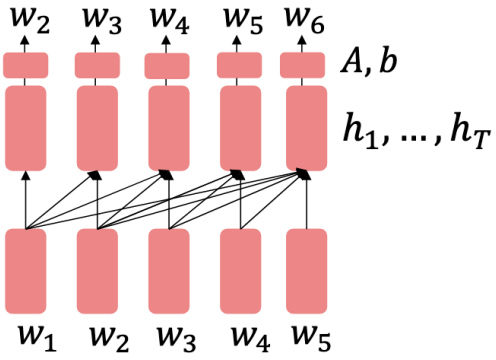


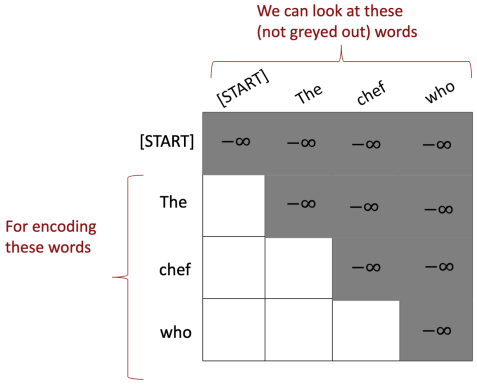
Replace the token with
(1) the [MASK] token 80% of the time.
(2) a random token 10% of the time.
(3) the unchanged i-th token 10% of the time.

Mask 15% of all WordPiece tokens in each sequence at random for prediction.

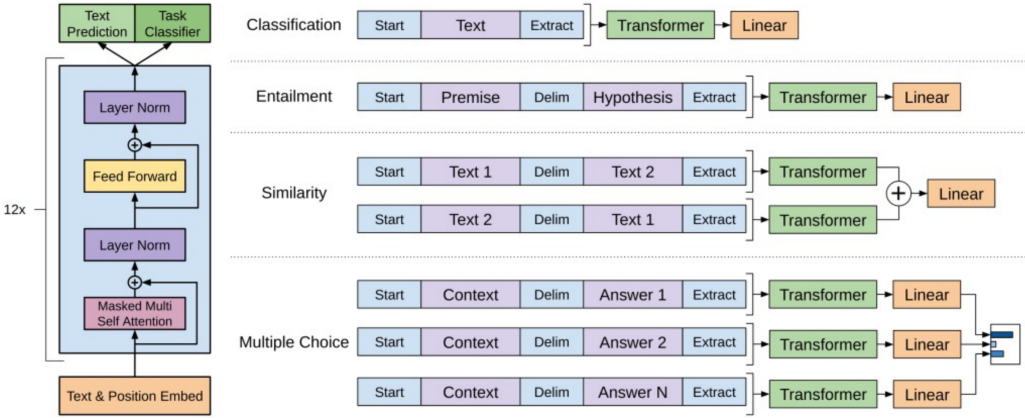
Credit: Y. Fang

- Decoder: generative language models to predict next word: $P_\theta\left(\boldsymbol{w}_t|(\boldsymbol{w}_{t-1},\ldots,\boldsymbol{w}_1)\right)$
- **Issue:** How do we keep the decoder from "cheating" (look ahead and "see" the answer)?
  - **Solution:** Masked Attention: hide (mask) information about future tokens from the model

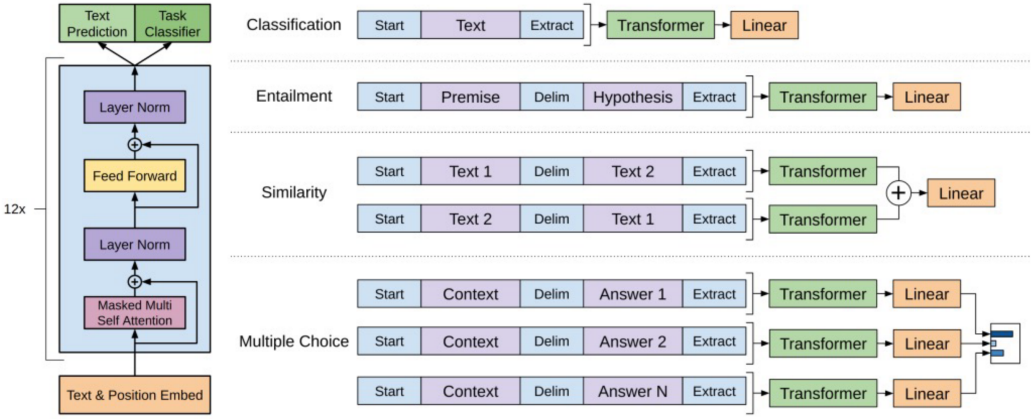




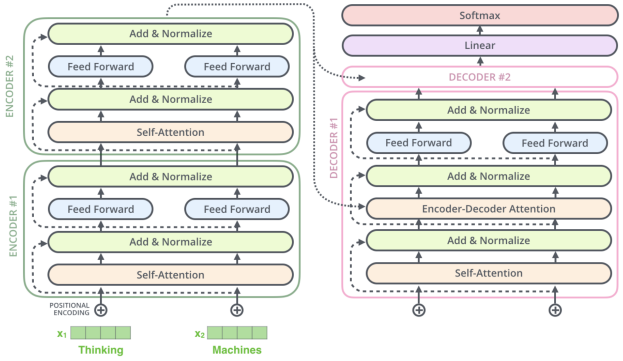
- Generative Pre-trained Transformer (GPT) [RNSS18]
- 12 layers: 768-dimensional hidden states, 3072-dimensional feed-forward hidden layers
- Byte-pair encoding with 40,000 merges, trained on BooksCorpus (over 7000 unique books)
- Contains long spans of contiguous text, for learning long-distance dependencies.

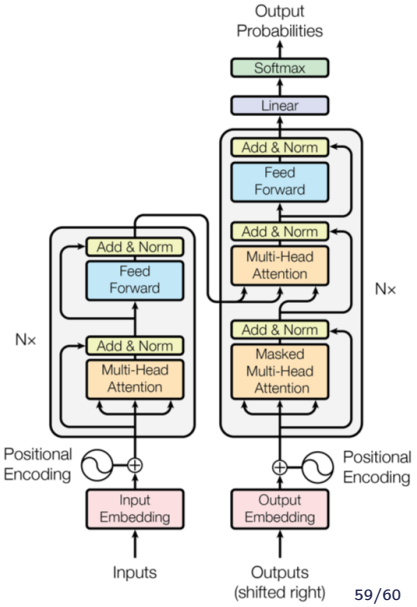


- GPT: Big success in pre-training a decoder
- GPT-2: larger version trained on more data $\Rightarrow$ convincing samples of natural language
- GPT-3: 175 billion parameters (10x more than GPT-2)



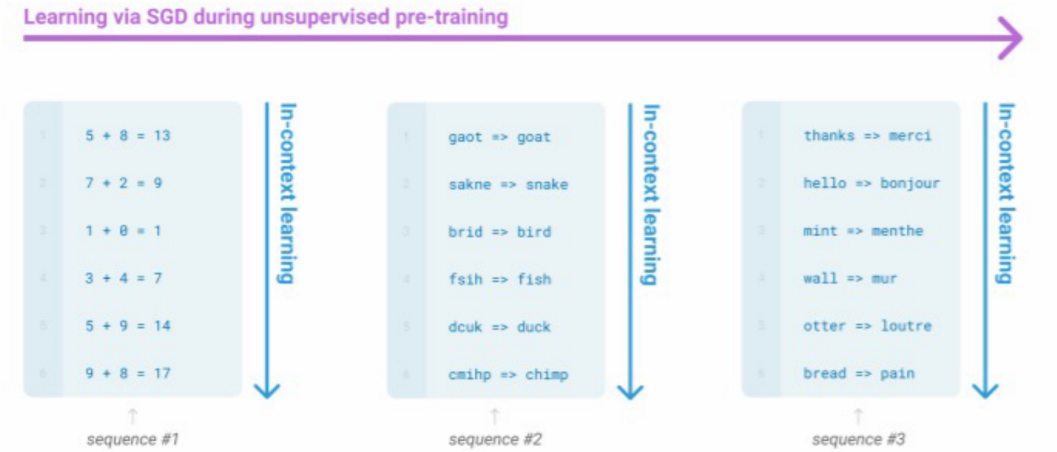
- Encoder+Decoder: Seq2Seq with transformers
- Famous example for translation: "attention is all you need" [VSP+17]
- Cross-attention between encoder and decoder





- Jay Alammar Blog 2018 http://jalammar.github.io/illustrated-transformer/

- Transformers: revolution in NLP
- Mediatic impact, see GPT-3
- Prompt ('in-context') learning: emerging (and not fully understood) behavior



References I

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio.
Neural machine translation by jointly learning to align and translate.
CoRR, abs/1409.0473, 2014.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals.
Listen, attend and spell.
CoRR, abs/1508.01211, 2015.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio.
Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.
cite arxiv:1406.1078Comment: EMNLP 2014.

George Cybenko.
Approximation by superpositions of a sigmoidal function.
Mathematics of control, signals and systems, 2(4):303–314, 1989.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova.
BERT: pre-training of deep bidirectional transformers for language understanding.
CoRR, abs/1810.04805, 2018.

Jeffrey L. Elman.
Finding structure in time.
COGNITIVE SCIENCE, 14(2):179–211, 1990.

References II

Sepp Hochreiter and Jürgen Schmidhuber.
Long short-term memory.
Neural Comput., 9(8):1735–1780, November 1997.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov.
Bag of tricks for efficient text classification.
In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics, 2017.

Andrej Karpathy.
The unreasonable effectiveness of recurrent neural networks, 2015.

Andrej Karpathy and Fei-Fei Li.
Deep visual-semantic alignments for generating image descriptions.
In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3128–3137, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton.
Imagenet classification with deep convolutional neural networks.
In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler.
Skip-thought vectors.
In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3294–3302. Curran Associates, Inc., 2015.

References III

Yann Lecun.
Une procedure d'apprentissage pour reseau a seuil asymmetrique (A learning scheme for asymmetric threshold networks), pages 599–604.
1985.

Lajanugen Logeswaran and Honglak Lee.
An efficient framework for learning sentence representations.
In *In ICLR*, 2018.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean.
Distributed representations of words and phrases and their compositionality.
In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

Chris Olah and Shan Carter.
Attention and augmented recurrent neural networks.
Distill, 2016.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer.
Deep contextualized word representations.
In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *NAACL-HLT*, pages 2227–2237. Association for Computational Linguistics, 2018.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning.
Glove: Global vectors for word representation.
In *In EMNLP*, 2014.

References IV

D.E. Rumelhart, G.E. Hinton, and R.J. Williams.
Learning representations by back-propagating errors.
Nature, 323:533–536, October 1986.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever.
Improving language understanding by generative pre-training.
2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin.
Attention is all you need.
In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017.