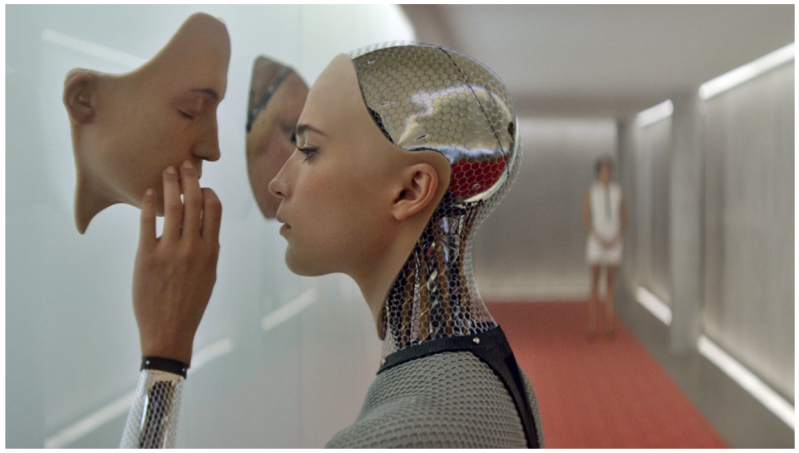


Getting to Machine Reading:

Word Representations, Attention, Self-Attention, and Pretrained Language Models in NLP

Jay Urbain, PhD

urbain@msoe.edu

Professor, EECS, Milwaukee School of Engineering

co-Director, Biomedical Informatics, CTSI SE Wisconsin

Topics

Topics

- NLP Intro
- Prologue
- Word Representations
- RNNs
- Attention
- Bidirectional LSTM with Attention Flow
- Transformer
- Self-Attention
- Demo
- Pretrained language models: Elmo, ULMFit, GPT, BERT, GPT-2

NLP Intro

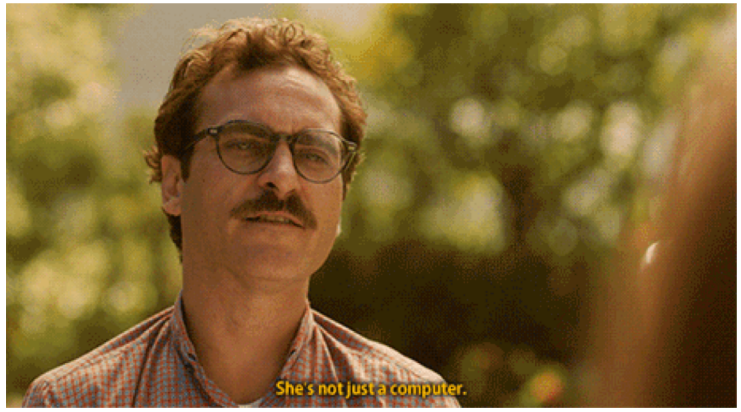
WHAT IS A NATURAL LANGUAGE PROCESSING (NLP)?

"Natural language processing (NLP) is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data. "

Wikipedia

GOAL: for computers to process or "*understand*" natural language in order to perform tasks that are useful such as question answering

- Add structure to unstructured text
- Full understanding for now is unsolved



NLP APPLICATIONS

- Keyword search
- Text classification
- Named entity recognition
- Machine translation
- Question answering, chatbot dialog systems
- Much more...

CLINICAL & TRANSLATIONAL RESEARCH INFORMATICS

Patient Deidentification

https://cis.ctsi.mcw.edu/deid/

2019-04-04 15:13:09.055

Date offset:

10

Patient name:

Input text:

Jay Urbain, jay.urbain@gmail.com, born December 6, 2156 is an elderly caucasian male suffering from illusions of grandeur and LBP. He is married to Kimberly Urbain, who is much better looking. Patient father, Francis Urbain has a history of CAD and DM. Jay has been prescribed meloxicam, and venti americano. He lives at 9050 N. Tennyson Dr., Disturbia, WI with his wife and golden retriever Mel. You can reach him at 414-745-5102.

Data Format Pretty Print

Submit

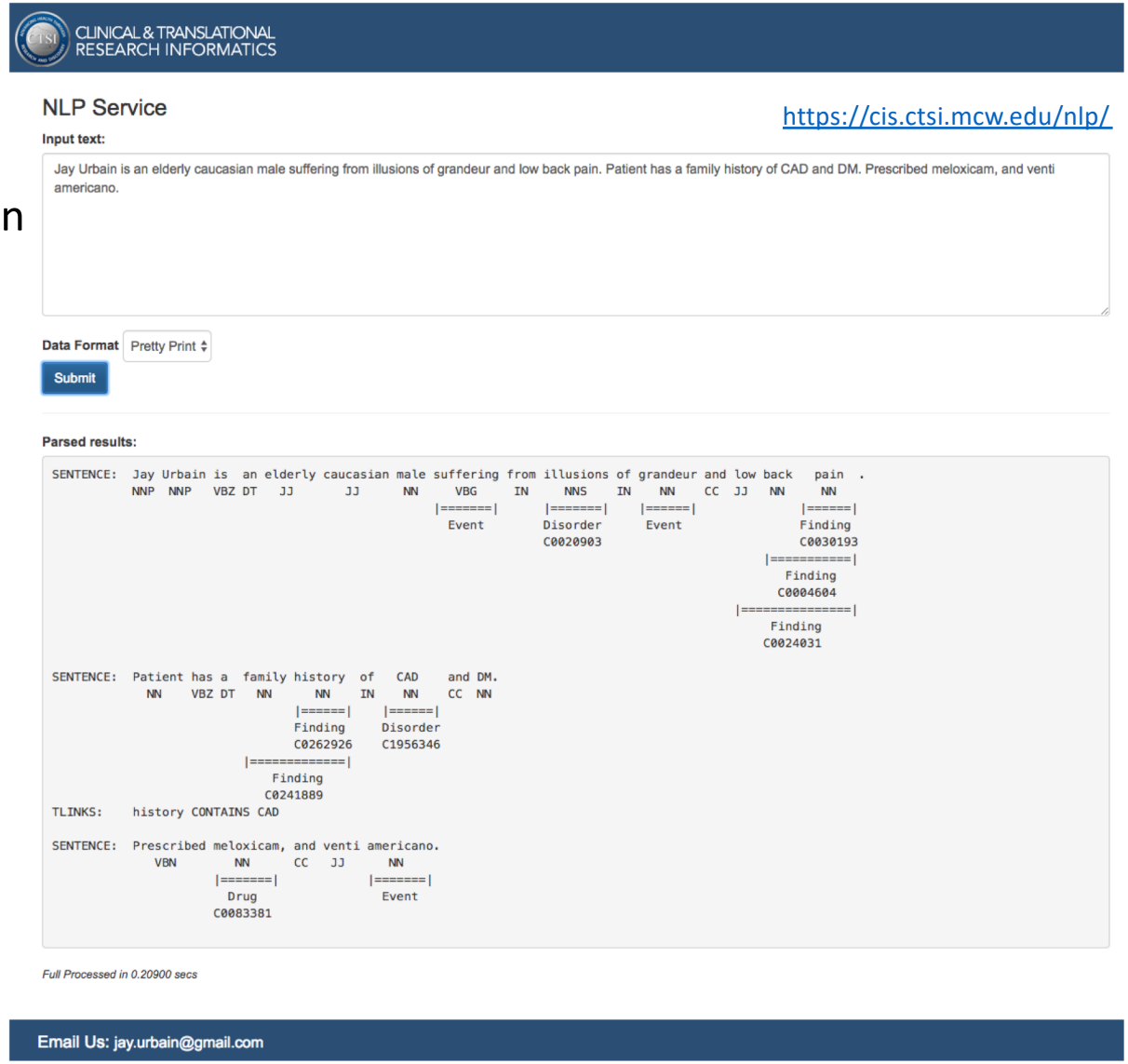
Parsed results:

[PERSON] [PERSON], [xxx@xxx.xxx] , born [12_16_2156] is an elderly caucasian male suffering from illusions of grandeur and LBP. He is married to [PERSON] [PERSON], who is much better looking. Patient father, [PERSON] [PERSON] has a history of CAD and DM. [PERSON] has been prescribed meloxicam, and venti americano. He lives at [xxxxx x. xxxxx] Dr., Disturbia, WI with his wife and golden retriever [PERSON]. You can reach him at [xxx_xxx_xxxx] .

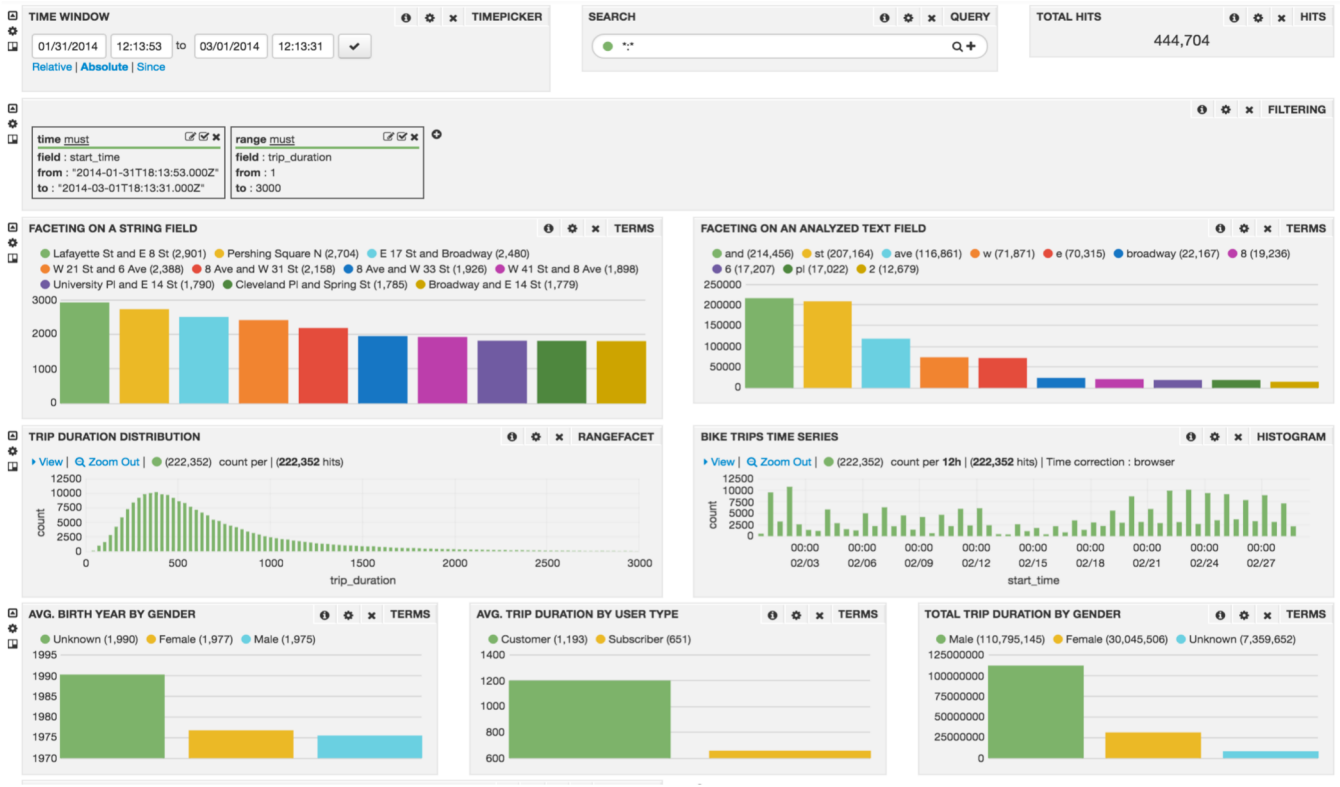
Processed in 0.24000 secs

Email Us: jay.urbain@gmail.com

Clinical Named Entity Identification



Search driven discovery: aggregating statistics from named entities



WHY IS NLP HARD?
… and why Deep Learning may help

Language: the ultimate UI

Where is A Bug's Life playing in Milwaukee?

A Bug's Life is playing at the Downer Theater.

When is it playing there?

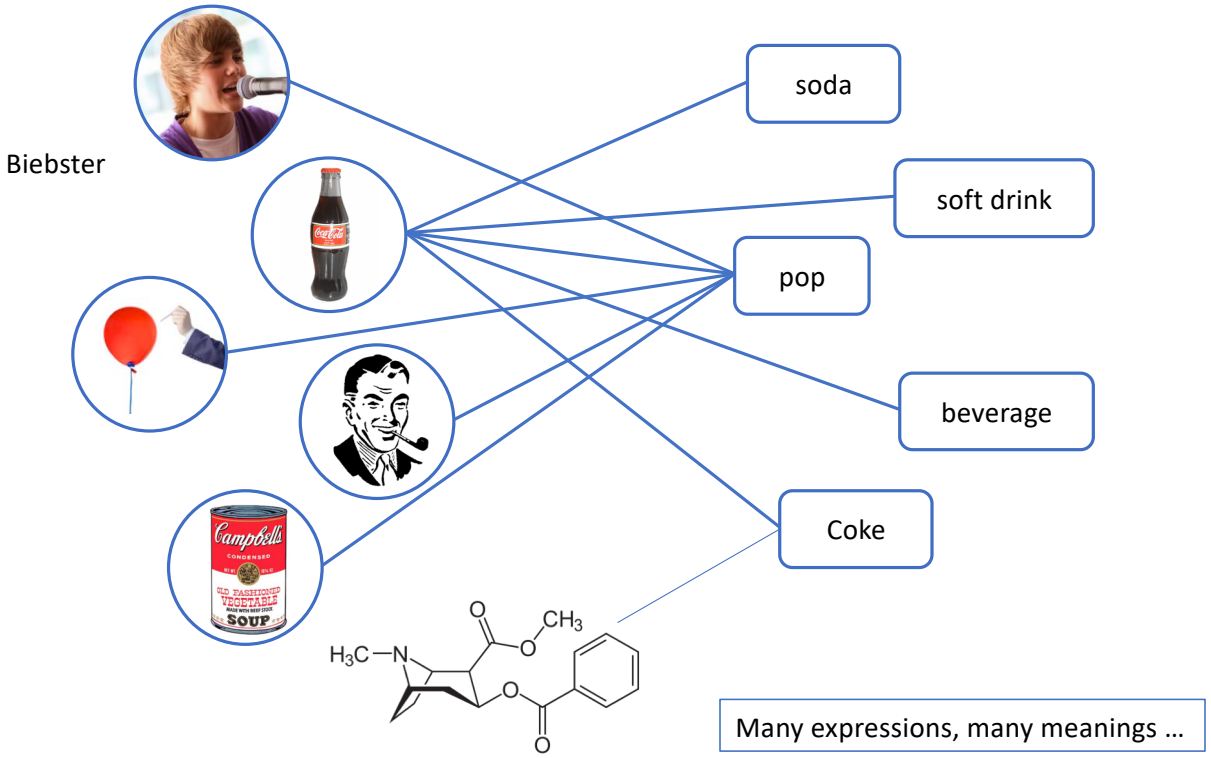
It's playing at 2pm, 5pm, and 8pm.

OK. I'd like 1 adult and 2 children for the first show. How much would that cost?

But we need domain knowledge, discourse knowledge, world knowledge

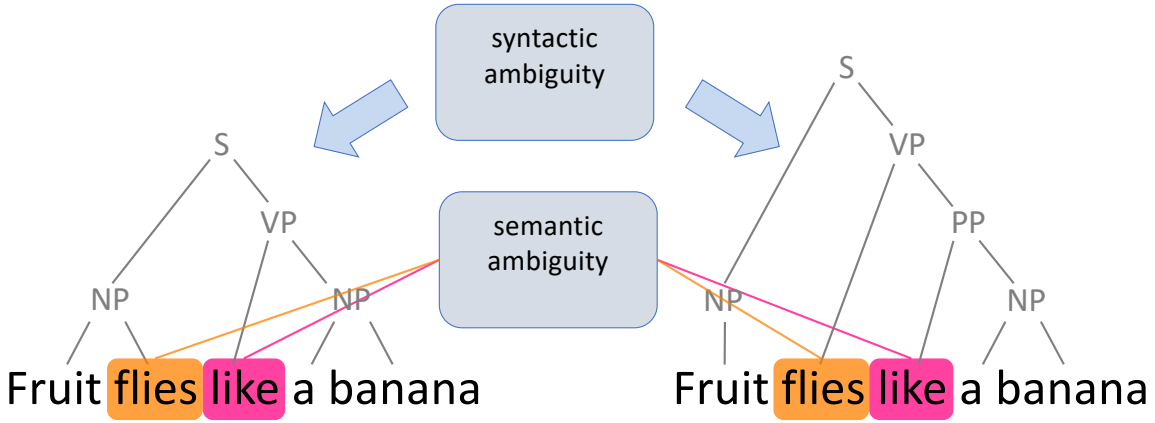
Meanings and expressions

Big obstacle: relation between meanings and expressions is not one-to-one



Many expressions, many meanings …

Syntactic & semantic ambiguity

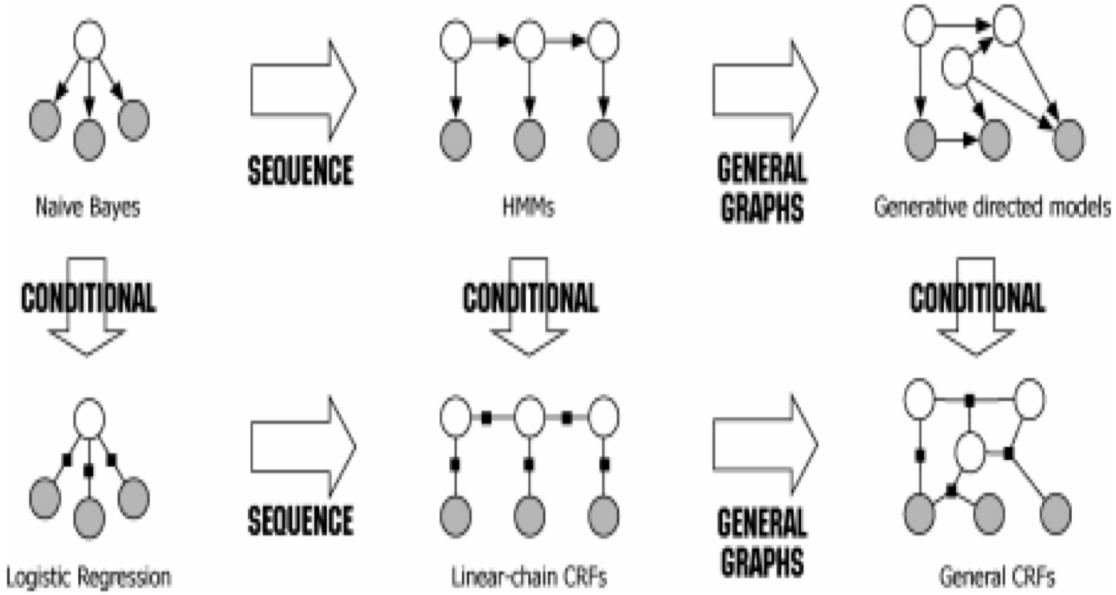


["like" is a remarkable word: it can be used as a noun, verb, adverb, adjective, preposition, particle, conjunction, or interjection.]

photos from worth1000.com

Traditional NLP use “shallow” statistical machine learning models: HMM, MEMM, CRF

- http://homepages.inf.ed.ac.uk/csutton/publications/crftut-fnt.pdf



Traditional Models: HMM, MEMM, CRF

- http://homepages.inf.ed.ac.uk/csutton/publications/crftut-fnt.pdf
- Linear chain CRF for sequence class, Named entity CRF features

Definition 2.2. Let Y, X be random vectors, $\theta = \{\theta_k\} \in \Re^K$ be a parameter vector, and $\mathcal{F} = \{f_k(y, y', \mathbf{x}_t)\}_{k=1}^K$ be a set of real-valued feature functions. Then a *linear-chain conditional random field* is a distribution $p(\mathbf{y}|\mathbf{x})$ that takes the form:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\}, \qquad (2.18)$$

where $Z(\mathbf{x})$ is an input-dependent normalization function

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\}. \qquad (2.19)$$

W=v	$w_t = v$	$\forall v \in \mathcal{V}$
T=j	part-of-speech tag for w_t is j (as determined by an automatic tagger)	$\forall$POS tags j
P=I–j	w_t is part of a phrase with syntactic type j (as determined by an automatic chunker)	
Capitalized	w_t matches `[A-Z][a-z]+`	
Allcaps	w_t matches `[A-Z][A-Z]+`	
EndsInDot	w_t matches `[^\.]+.*\.`	
	w_t contains a dash	
	w_t matches `[A-Z]+[a-z]+[A-Z]+[a-z]`	
Acro	w_t matches `[A-Z][A-Z\\.]*\\.[A-Z\\.]*`	
Stopword	w_t appears in a hand-built list of stop words	
CountryCapital	w_t appears in list of capitals of countries	
⋮	many other lexicons and regular expressions	
$q_k(\mathbf{x}, t+\delta)$ for all k and $\delta \in [-1, 1]$		

Custom solutions for each domain, and for each application.
Difficulty capturing long-range dependencies, context, and semantics

NLP: With deep Learning, reality is getting closer to vision

LEARNING REPRESENTATIONS WITH DEEP LEARNING

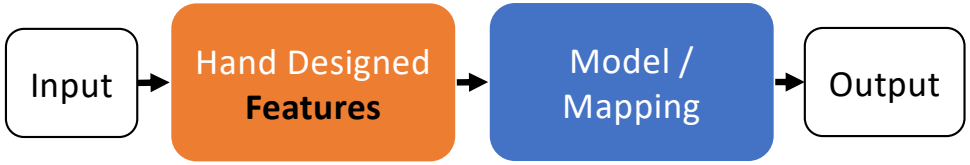
Power of Deep learning =
Learning representations/features

- The traditional model of pattern recognition (since the late 50's)
  - Fixed/engineered features (or fixed kernel) + trainable classifier

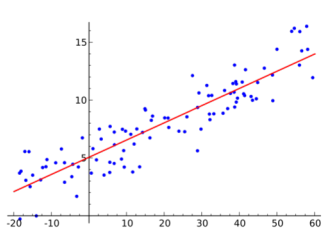
- End-to-end learning / Feature learning / Deep learning
  - Trainable features (or kernel) + trainable classifier

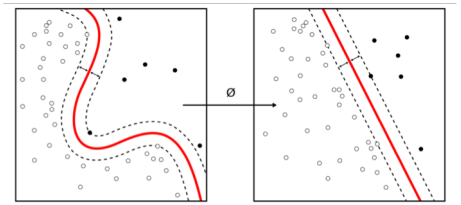
Difference in Workflow

Classic Machine Learning [1990 : now]

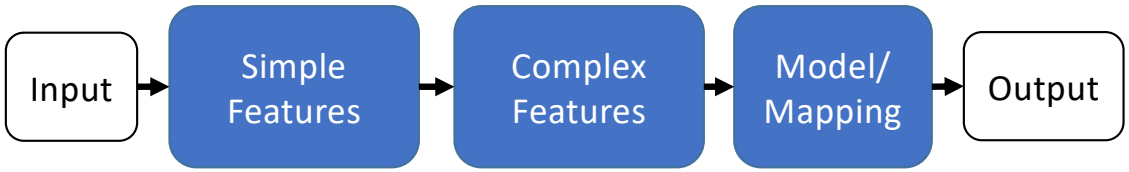


Examples [Regression and SVMs]

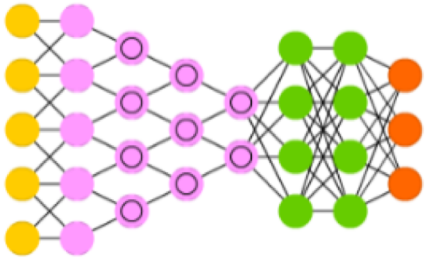




Deep/End-to-End Learning [2012 : now]



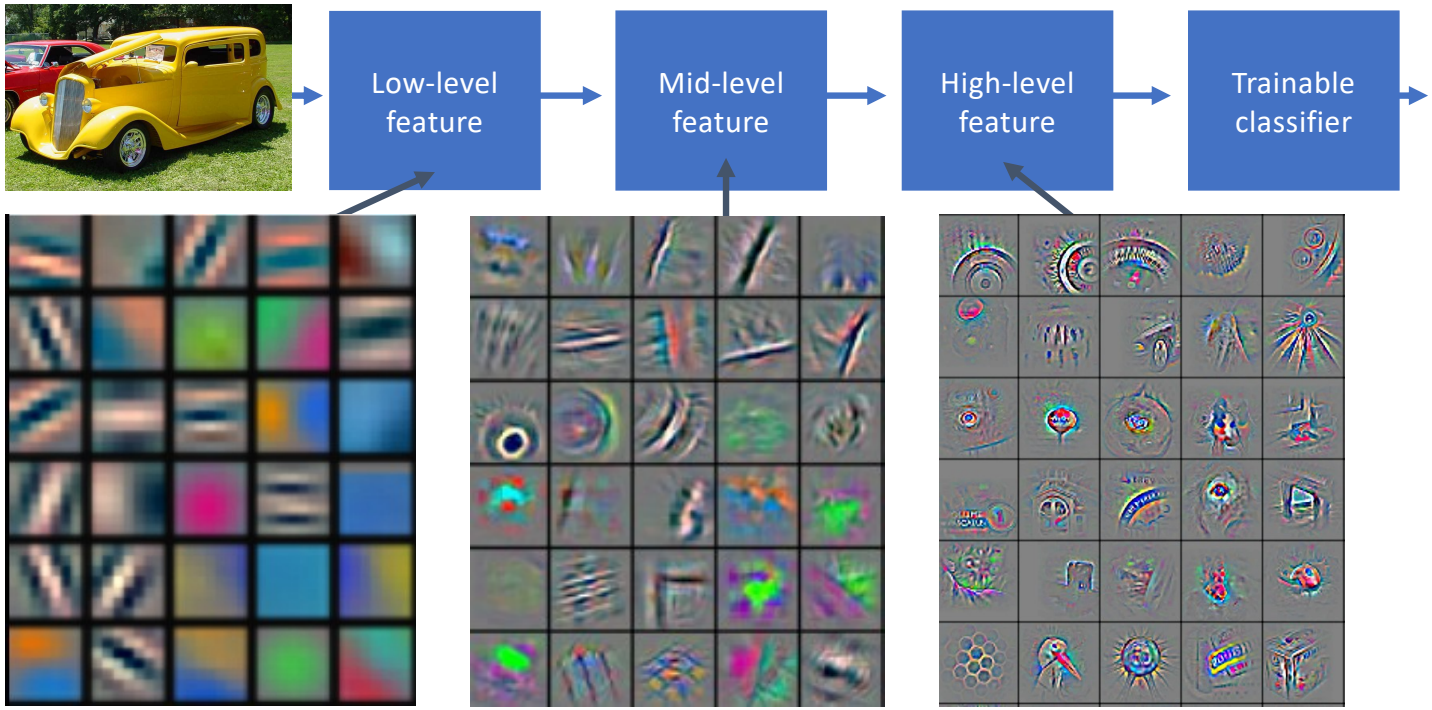
Example [Conv Net]



Machine learning workflow shifts from engineering features for “shallow” models
to architecting deep learning models with the ability to learn hierarchical representations of features

Deep learning = learning hierarchical representations

It's deep if it has more than one stage of non-linear feature transformation

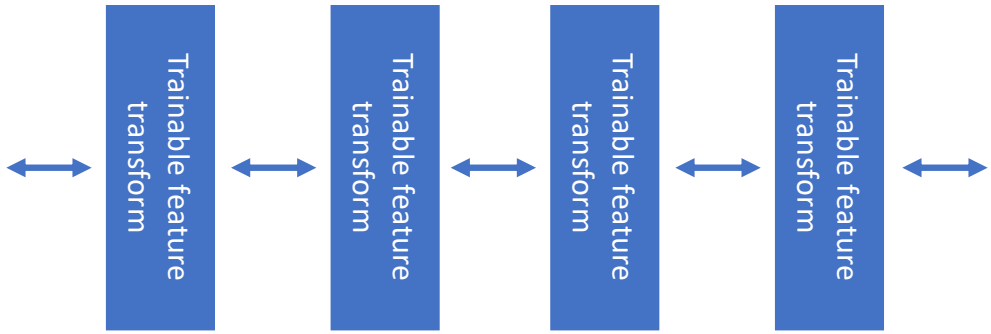


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Trainable feature hierarchy

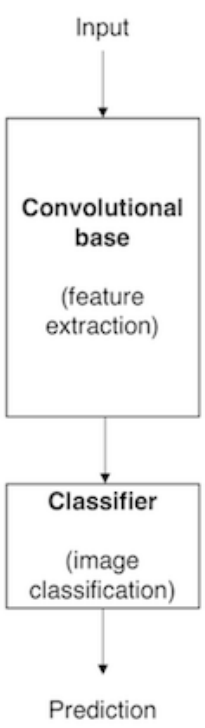
Hierarchy of representations with increasing level of abstraction. Each stage is a kind of trainable feature transform

- Image recognition
  - Pixel → edge → motif → part → object
- Text
  - Character → word → word group → clause → sentence → story/semantic understanding
- Speech
  - Sample → spectral band → sound → … → phone → phoneme → word



Transfer Learning + Fine Tuning – NLP?

- Norm for computer vision tasks, work of Razavian et al (2014).
- Reduces the training time and data needed to achieve custom ML task.
- Replace final layers of pre-trained CNN model (ImageNet), with one or more custom layers.
- Continue training with custom data set:
  - Freeze original weights, fine-tune, etc.



MACHINE READING PROJECT PROLOGUE

Prologue – Medical Record Machine “Translation”

- work with Bradley Crotty, MD

COLONOSCOPY Procedure Note. Date of Procedure: [1_21_2016]. Primary Physician: [PERSON] [PERSON] [PERSON], MD. Attending Physician: [PERSON] [PERSON], MD. Fellow:None. Indications: Colon cancer screening for family history of colon cancer Colon cancer screening for family history of polyps.. Previous COLONOSCOPY: Yes. Date: [8_16_2007]. Medications Administered: Agents given by the anesthesia service during MAC. Procedure Details: The patient was placed in the left lateral position and monitored continuously with ECG tracing, pulse oximetry monitoring and direct observations. Medications were administered incrementally over the course of the procedure to achieve an adequate level of moderate sedation. After anorectal examination was performed, the Olympus CF H180 was inserted into the rectum and advanced under direct vision to the terminal ileum. The procedure was considered not difficult. During withdrawal examination, the final quality of the prep was good.. Bowel Prep Scale Right Colon: Grade 2 (minor amount of residual staining, small fragments of stool, and/or opaque liquid, but mucosa of colon segment is seen well) . Bowel Prep Scale for Transverse Colon: Grade 3 (entire mucosa of colon segment seen well, with no residual staining, small fragments of stool, or opaque liquid) . Bowel Prep Scale for Left Colon: Grade 2 (minor amount of residual staining, small fragments of stool, and/or opaque liquid, but mucosa of colon segment is seen well) Additional rinsing and suctioning (600 ml) were necessary to obtain adequate views. A careful inspection was made as the colonoscope was withdrawn, which did include a retroflexed evaluation of the rectum. Findings and interventions are described below. Appropriate photo documentation was obtained. Overall [PATIENT] L [PATIENT] did tolerate the procedure well, without undue discomfort, hypotension or desaturation. At the completion of the procedure she was transferred from the endoscopy suite to be recovered in the FSC observation area per protocol and was discharged when criteria was met. After adequate recovery from sedation, she was discharged to home, with appropriate plans for follow up in place. Scope in time: 0741 Cecum reached time: 0747. Scope out time: 0759. Findings:. Findings for the Anorectal: No mass. Findings for the Terminal Ileum: Normal ileal mucosa. Findings for the Cecum: Normal mucosa. Findings for the Ascending Colon: Normal mucosa. Findings for the Transverse Colon: 1 polyp 2 mm in size, sessile, removed by cold biopsy and sent for pathology. Findings for the Descending Colon: Normal mucosa. Findings for the Sigmoid Colon: Normal mucosa. Findings for the Rectum: 2 polyps 4 mm in size, sessile, removed by hot biopsy and sent for pathology Hypertrophied anal PAPillae Small anal fissure. Interventions: 1 polyp removed by cold biopsy 2 polyps removed by hot biopsy. Complications: None. Adverse Events: No. Impression: 1. Multiple, 3 colon polyps, removed by cold biopsy and hot biopsy and sent for pathology 2. Family history of colon cancer 3. Family history of polyps 4. Hypertrophied anal PAPillae 5. Anal fissure 6. Otherwise normal COLONOSCOPY to the terminal ileum. Recommendations: ? Avoid aspirin and NSAIDs for 14 days ? Await pathology results ?. Dietary recommendations: High fiber diet. ? Anusol HC or similar ointment for discomfort form anal fissure/PAPillitis ?. Repeat COLONOSCOPY in 5 years ? For the patient's next COLONOSCOPY, a 4 liter, split dose prep should be used. In addition, the patient should start a low residue diet two days before the exam and be on a clear liquid diet the day before the exam. ? Would use large caliber colonoscope ? Follow up with primary care physician. Histopathologic Diagnosis: A. Transverse colon polyp, polypectomy: ? Tubular adenoma. B. Rectal polyps, polypectomy: ? Tubular adenoma (x1). ? Hyperplastic polyp (x1).. Qualify for GI Quality project: Yes #@Screening@# Extent of Examination: #@Complete@# Pathology: #@Adenomatous change@# Appendiceal orifice Ileum Rectal polyp Rectal polyp Anal PAPillitis and small fissure @ 6 o'clock [PERSON] [PERSON] [PERSON], MD, FACP

Translation

Date of procedure:

- 1_21_2016

Indications for colon cancer screening:

- Family history of polyps

Complications:

- The procedure was considered not difficult

Findings for the transverse colon:

- 1 polyp 2mm
- Tubular adenoma
- Hyperplastic polyp

NSAIDS should be avoided:

- 14 days

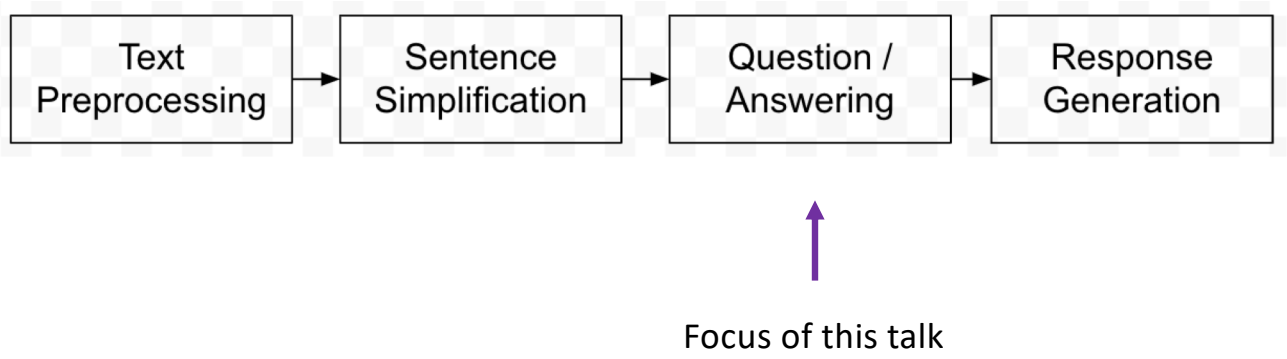
Recommendations:

- High fiber diet

What kind of natural language task is this?

- Translation
- Summarization
- Question/answering
  - Question answering is a direct form of purposeful communication.

"Translation" of medical records becomes a summary of answers to specific questions.



NLP Challenges and Future

- One of the biggest challenges in natural language processing (NLP) is the shortage of training data. *Especially healthcare!*
- Because NLP is a diversified field with many distinct tasks, most task-specific datasets contain only a few thousand or a few hundred thousand human-labeled training examples.
- Modern deep learning-based NLP models see benefits from much larger amounts of data, improving when trained on millions, or billions, of annotated training examples.
- To help close this gap in data, researchers have developed a variety of techniques for training general purpose language representation models using the enormous amount of unannotated text on the web (known as pre-training).

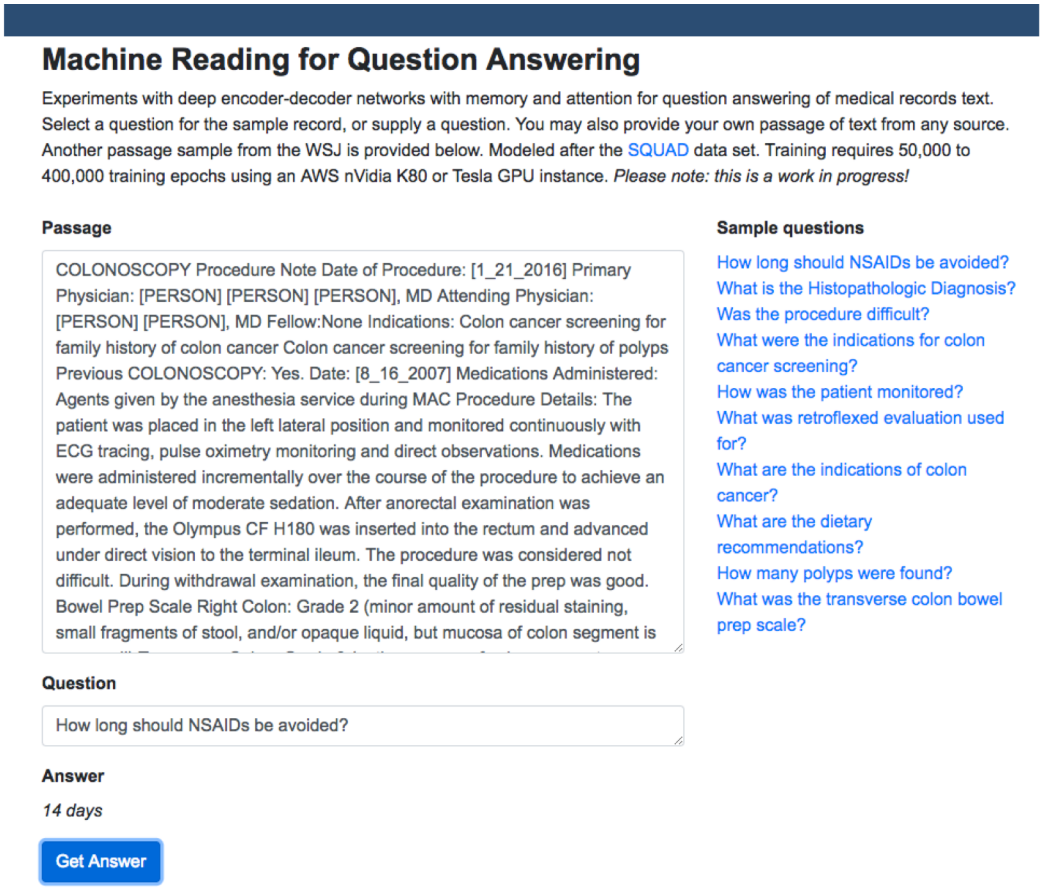
Approach

- Major advances using deep learning models for NLP in the last 1 to 2 years.
  - *Note: look at the dates of papers cited in this presentation.*
- Deep learning requires large datasets.
  - Medical record data hard to obtain in large volumes.

1. Build a deep learning model on large question/answering datasets created from Wikipedia, WSJ, CNN, etc.?
2. Given such a model, fine tune (transfer learning, domain adaptation) healthcare?

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
https://nlp.stanford.edu/pubs/rajpurkar2016squad.pdf

Question Answering



http://ec2-54-163-221-189.compute-1.amazonaws.com:8080/

WORD REPRESENTATIONS

ONE-HOT ENCODING

- Words are categorical, they have no *ordinal* relationship. So they can't just be numerically encoded.
- For each word:
  - Numerically encode each word: {red:0, green:1, blue:2}
  - Generate vector of zeros the length of all possible words/categories
  - Set the index value for the word in the vector

red,	green,	blue
1,	0,	0
0,	1,	0
0,	0,	1

The Distributional Hypothesis (Firth, 1957)

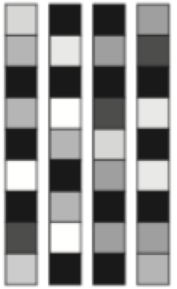
- ‘You can tell a word by the company it keeps’

- *The cat sat on the mat*
- *The dog sat on the mat*
- *The elephant sat on the mat*
- *The quickly sat on the mat*

WORD EMBEDDINGS

- One-hot results in *sparse vector representation* (mostly zeros, hi dimensions).
- Word embeddings provide *dense vector representations*.
- Idea is to "embed" words into a lower-dimensionality space.
- The dimensions of this space are typically defined by word context, i.e., semantically similar words are embedded near each other.
- Popular algorithms:
  - Word2Vec: Skip-gram or CBOW
  - Point-wise mutual information (PMI)
  - GLoVE
  - FastText

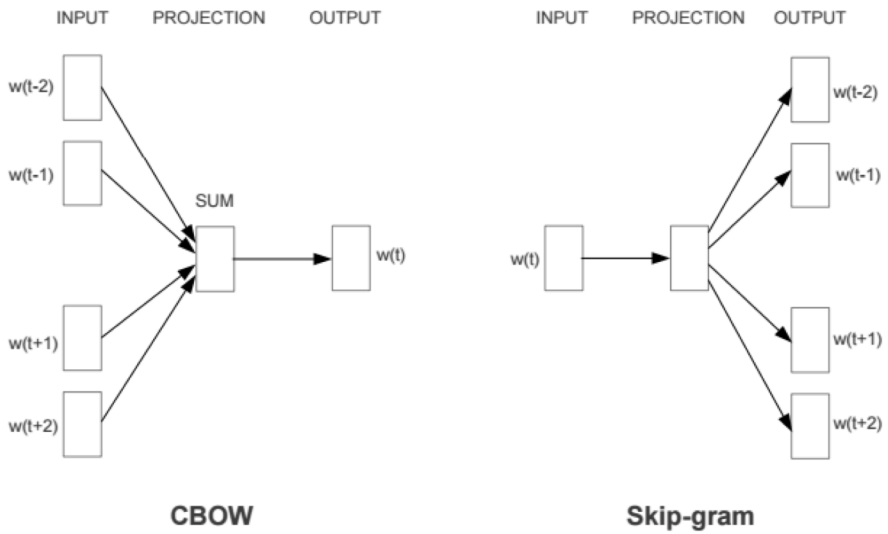
One-hot word vectors:
- Sparse
- High-dimensional
- Hardcoded



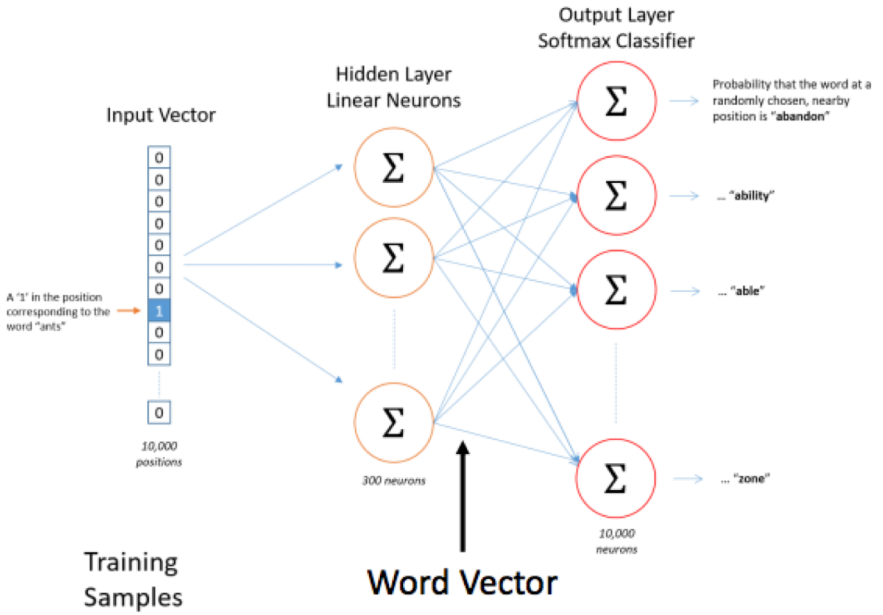
Word embeddings:
- Dense
- Lower-dimensional
- Learned from data

Embeddings: Word2Vec

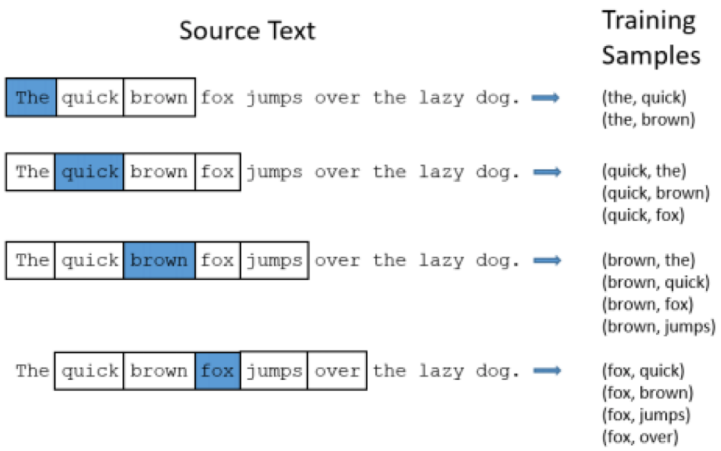
- **CBOW** model predict missing word (focus word) using context (surrounding words).
- **Skip gram** model predicts context based on the word in focus.
- Context is a fixed number of words to the left and right of the word in focus.
- Maximize average log probability of context words co-occurring with focus words.

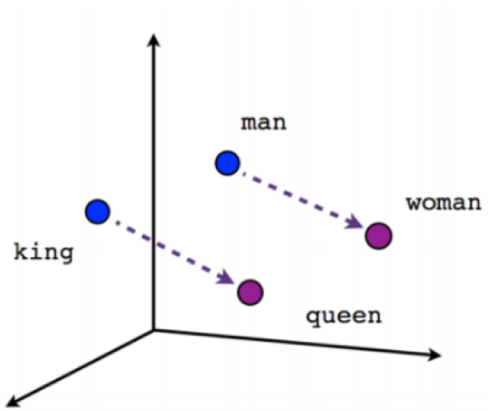


Word Embeddings (Word2Vec)

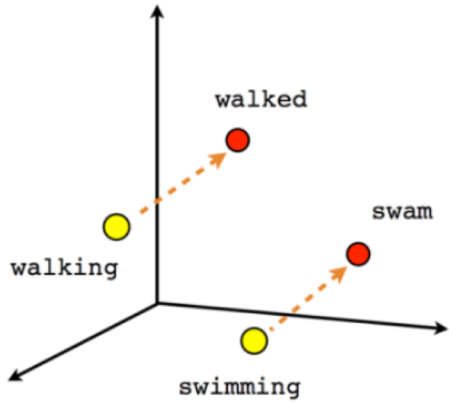


Skip Gram Model:

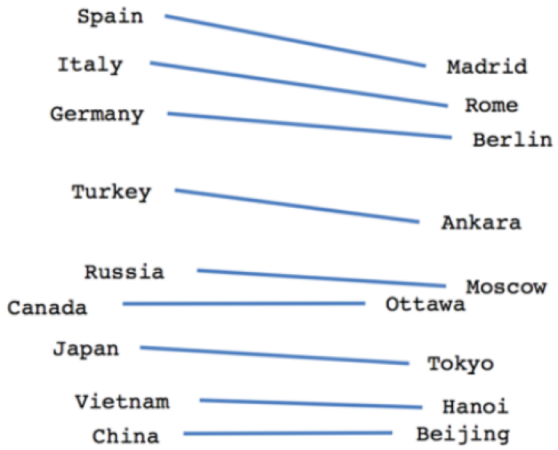




Male-Female

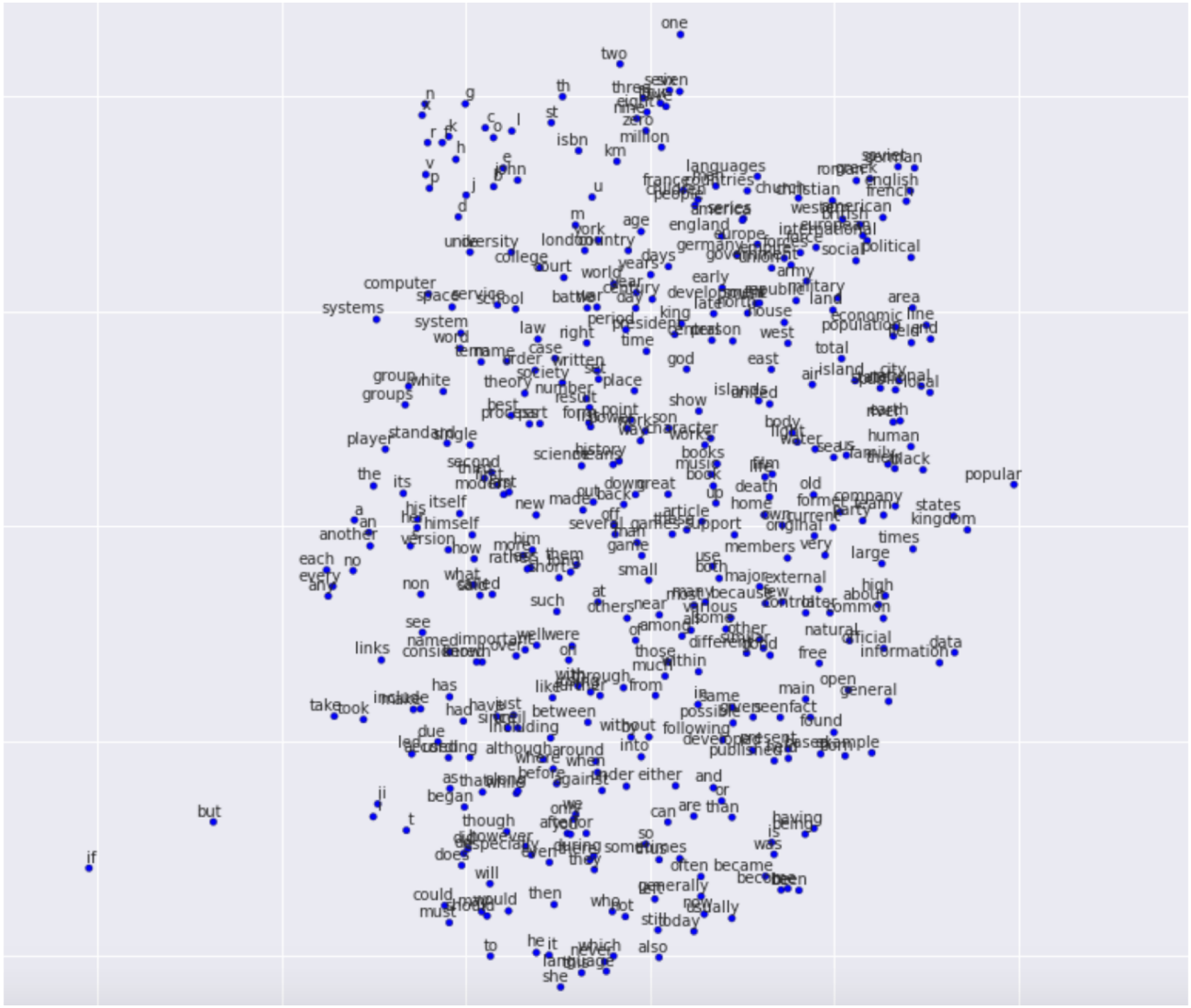


Verb tense



Country-Capital

https://www.tensorflow.org/tutorials/representation/word2vec


one
two
isbn
million
computer
systems
but
if
popular

Pointwise Mutual Information

- Normalized skip gram probability
- Log probability of context word, x, co-occurring with target word y.

$$\mathrm{pmi}(x;y) \equiv \log \frac{p(x,y)}{p(x)p(y)}$$

- Urbain, J., Bushnee, G., Knudson, Kowalski, G., P., Taylor, B. "Distributional Semantic Concept Models for Entity Relation Discovery," NAACL VSM-NLP (National Association of Computational Linguistics workshop on Vector Space Modeling in Natural Language Processing), June 2015.
- Levy, O., and Goldberg, Y., Neural Word Embedding as Implicit Matrix Factorization, NIPS, 2014.
- Urbain, J. A Distributed Dimensional Indexing Model for Information Retrieval and Extraction. NLM/NIBIB Workshop on Natural Language Processing: State of the Art, Future Directions and Applications for Enhancing Clinical Decision-Making, (April 23-24, 2012).

GloVe – Global Vectors for Word Representation

- Count-based model learns word vectors by doing dimensionality reduction on a word co-occurrence counts matrix.
- Factorize this matrix to yield a lower-dimensional word feature matrix.
- Use a reconstruction loss to capture most of the variance.
- Each row now yields a vector representation for each word.

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i} = \frac{X_{ij}}{\sum_k X_{ik}},$$

X_{ij} = number of times word j occurs in the context of word i.

$$\hat{J} = \sum_{i,j} f(X_{ij})(w_i^T \tilde{w}_j - \log X_{ij})^2$$

J. Pennington, R. Socher, and C. Manning, GloVe: Global Vectors for Word Representation, http://www.aclweb.org/anthology/D14-1162

FastText

- Each word is represented as a bag of character n-grams.
- Words are represented as the sum of their character n-grams.
- Captures morphology of words.
- Provides a representation for rare and out-of-vocabulary words.
- Very fast.
- Outperforms Word2Vec and GloVe in most benchmarks.

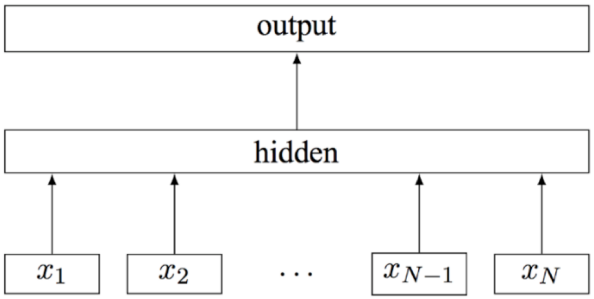


Figure 1: Model architecture of `fastText` for a sentence with N ngram features $x_1, \ldots, x_N$. The features are embedded and averaged to form the hidden variable.

Image taken from the original paper

Bojanowski, P., Grave, E., Jouln, A, Mikolov, T. Enhancing Wore Vectors with Subword Information, 2017.

Selected word embedding: GloVe + Character Embedding

- Embedding for each word *w* was created by concatenating pretrained *p1=300* GloVe vector (Pennington et al., 2014) and the word's character embedding.
- GloVe word vectors are fixed during training.
- All the out-of-vocabulary words are mapped to a token, whose embedding is trainable with random initialization.
- Character embedding:
  - Each character is represented as a trainable vector of dimension *p2 = 200*, meaning each word can be viewed as the concatenation of the embedding vectors for each of its characters.
  - The length of each word is either truncated or padded to 16.
  - Take maximum value of each row of this matrix to get a fixed-size vector representation of each word.
  - Finally, the output of a given word *x* from this layer is the concatenation [xw; xc] ∈ Rp1+p2 , where xw and xc are the word embedding and the convolution output of character embedding of x respectively.
- Adapted from Seo et al., 2016, and Srivastava et al., 2015).

Problems with word embeddings?

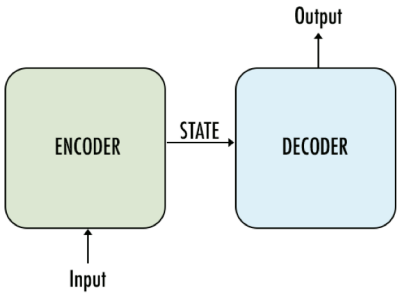
Synonyms:

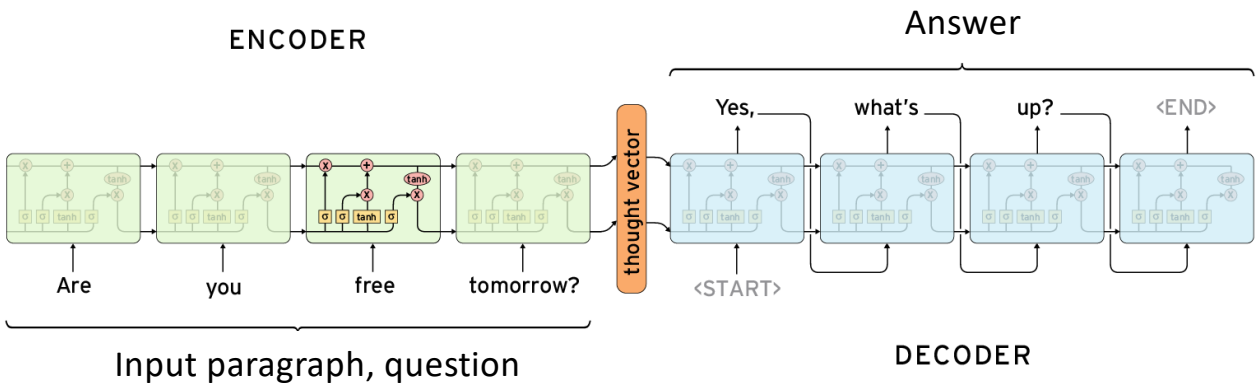
- River *bank*, *Bank* shot, *bank* deposit all have the same word representations.

- *… more on learning general models of language later.*

Encoder-Decoder, RNNs, LSTMs

Encoder-Decoder

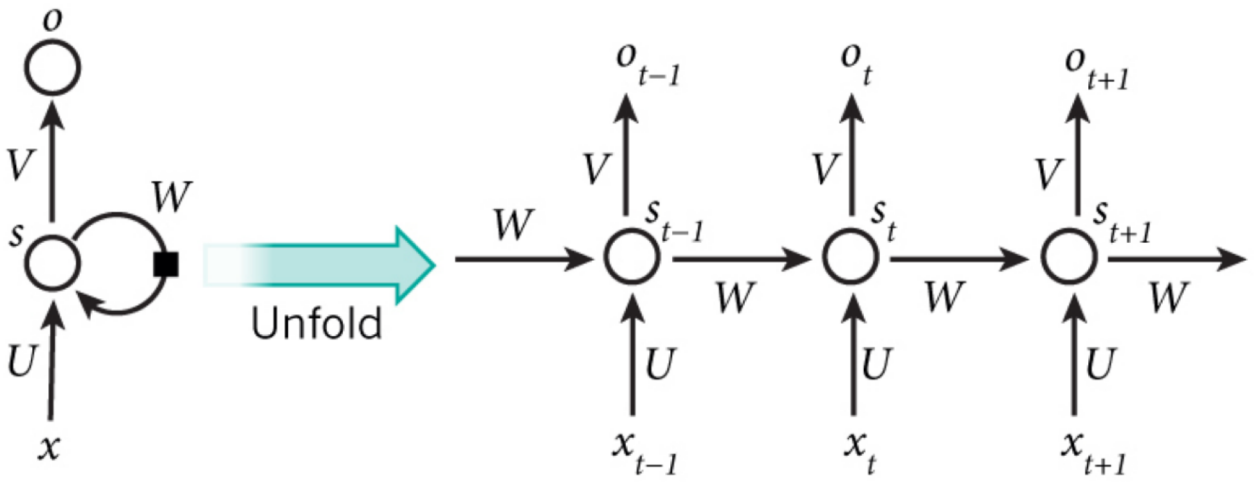




Recurrent Neural Network (RNN)

- RNNs have connections between units along a sequence.
- RNN's use the hidden state from the last time step and the input at the current step to make predictions.
- Allow RNN's can capture dynamic temporal behavior for a time sequence.

- Must run network for each step of the encoder and decoder – slow.
- Context from input words can have a long distance to travel.



Vanishing error gradient

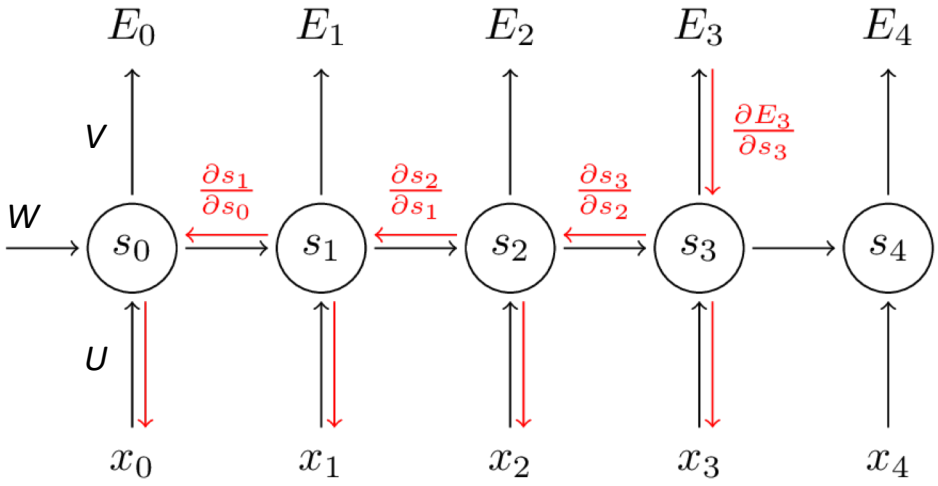
RNNs have difficulties learning long-range dependencies – interactions between words that are several steps apart.

Problem - the meaning of an English sentence is often determined by words that aren't very close: "The man who wore a wig on his head went inside". The sentence is really about a man going inside, not about the wig.

- Learning involves calculating the gradients of the Error *E* with respect to parameters (weights) *U, V*, & *W* using stochastic gradient descent.

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial s_k} \frac{\partial s_k}{\partial W}$$

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \left(\prod_{j=k+1}^{3} \frac{\partial s_j}{\partial s_{j-1}} \right) \frac{\partial s_k}{\partial W}$$



Backpropagation through Time

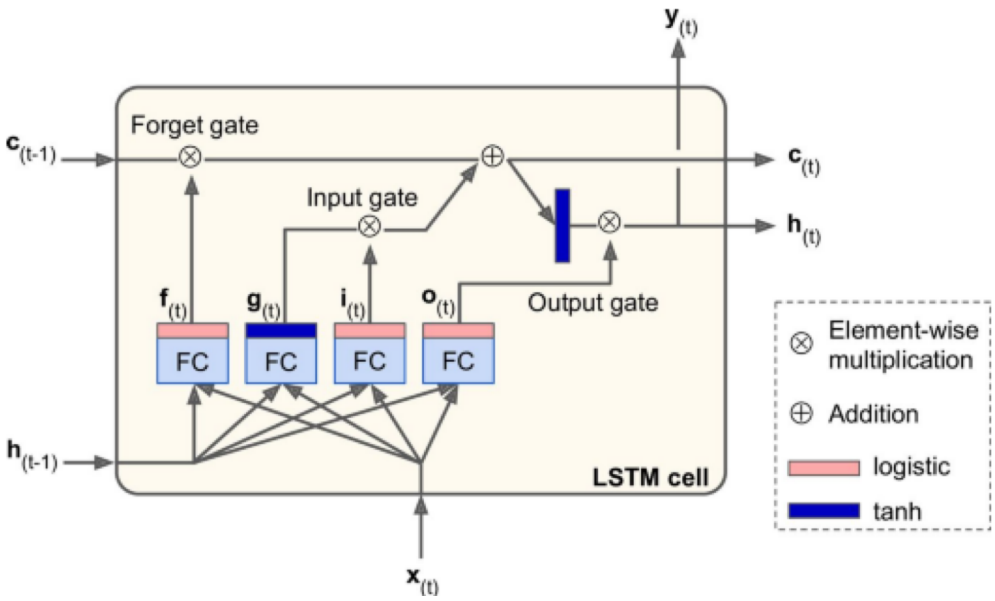
```
def bptt(self, x, y):
    T = len(y)
    # Perform forward propagation
    o, s = self.forward_propagation(x)
    # We accumulate the gradients in these variables
    dLdU = np.zeros(self.U.shape)
    dLdV = np.zeros(self.V.shape)
    dLdW = np.zeros(self.W.shape)
    delta_o = o
    delta_o[np.arange(len(y)), y] -= 1.
    # For each output backwards...
    for t in np.arange(T)[::-1]:
        dLdV += np.outer(delta_o[t], s[t].T)
        # Initial delta calculation: dL/dz
        delta_t = self.V.T.dot(delta_o[t]) * (1 - (s[t] ** 2))
        # Backpropagation through time (for at most self.bptt_truncate steps)
        for bptt_step in np.arange(max(0, t-self.bptt_truncate), t+1)[::-1]:
            # print &amp;quot;Backpropagation step t=%d bptt step=%d &amp;quot; % (t, bptt_step)
            # Add to gradients at each previous step
            dLdW += np.outer(delta_t, s[bptt_step-1])
            dLdU[:,x[bptt_step]] += delta_t
            # Update delta for next step dL/dz at t-1
            delta_t = self.W.T.dot(delta_t) * (1 - s[bptt_step-1] ** 2)
    return [dLdU, dLdV, dLdW]
```

LONG SHORT TERM MEMORY

- LSTM = Long Short Term Memory
  - Variant of RNN
  - Reduces vanishing gradient problem
  - LSTMs can learn “very deep” tasks that require memories of words N words ago

Long Short Term (LSTM) Cell

- Forget gate – what parts of the memory cell should be 'forgotten'
- Input gate – what parts of the input should be applied to the memory cell
- Output gate - what parts of the memory cell should be exposed in the output of the cell



$$\mathbf{i}_{(t)} = \sigma(\mathbf{W}_{xi}{}^{T} \cdot \mathbf{x}_{(t)} + \mathbf{W}_{hi}{}^{T} \cdot \mathbf{h}_{(t-1)} + \mathbf{b}_{i})$$

$$\mathbf{f}_{(t)} = \sigma(\mathbf{W}_{xf}{}^{T} \cdot \mathbf{x}_{(t)} + \mathbf{W}_{hf}{}^{T} \cdot \mathbf{h}_{(t-1)} + \mathbf{b}_{f})$$

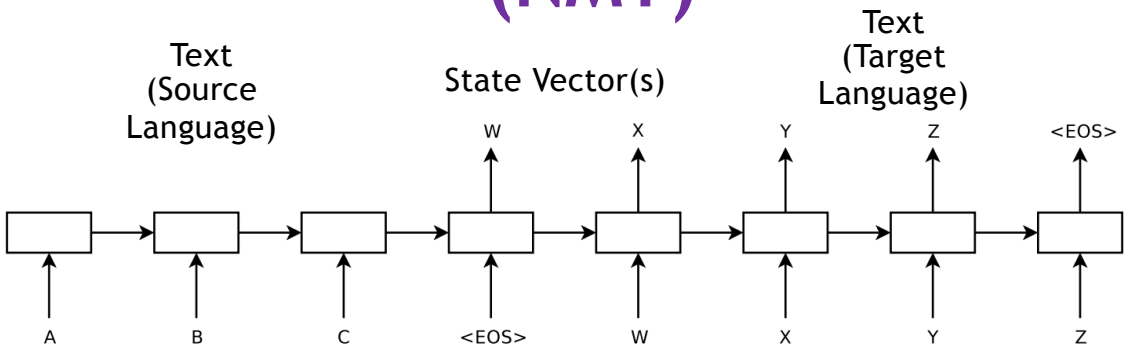
$$\mathbf{o}_{(t)} = \sigma(\mathbf{W}_{xo}{}^{T} \cdot \mathbf{x}_{(t)} + \mathbf{W}_{ho}{}^{T} \cdot \mathbf{h}_{(t-1)} + \mathbf{b}_{o})$$

$$\mathbf{g}_{(t)} = \tanh(\mathbf{W}_{xg}{}^{T} \cdot \mathbf{x}_{(t)} + \mathbf{W}_{hg}{}^{T} \cdot \mathbf{h}_{(t-1)} + \mathbf{b}_{g})$$

$$\mathbf{c}_{(t)} = \mathbf{f}_{(t)} \otimes \mathbf{c}_{(t-1)} + \mathbf{i}_{(t)} \otimes \mathbf{g}_{(t)}$$

$$\mathbf{y}_{(t)} = \mathbf{h}_{(t)} = \mathbf{o}_{(t)} \otimes \tanh(\mathbf{c}_{(t)})$$

Encoder Decoder for Neural Machine Translation (NMT)



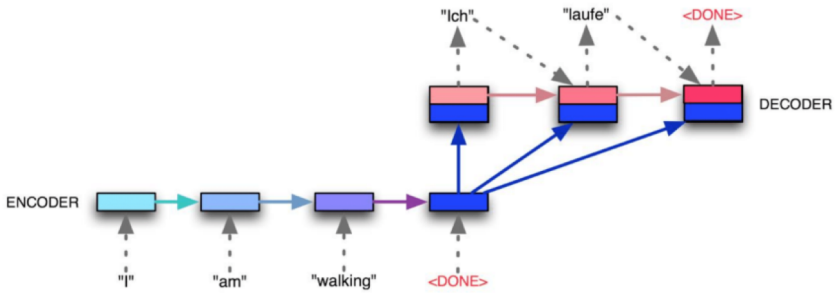
Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, *Yonghui Wu, et al., Technical Report, 2016.*
http://arxiv.org/abs/1609.08144

Sequence to Sequence Learning with Neural Networks, *Ilya Sutskever, Oriol Vinyals, Quoc V. Le. Advances in Neural Information Processing Systems, 2014.*
https://arxiv.org/pdf/1409.3215.pdf

Le, Q. V., & Schuster, M. (2016). A neural network for machine translation, at production scale. *Google research blog*.
https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html

Encoder-Decoder Architecture Sequence-to-Sequence Model - Neural Machine Translation

- Encoder RNN encodes input sequence into a fixed size vector, and then is passed repeatedly to dec

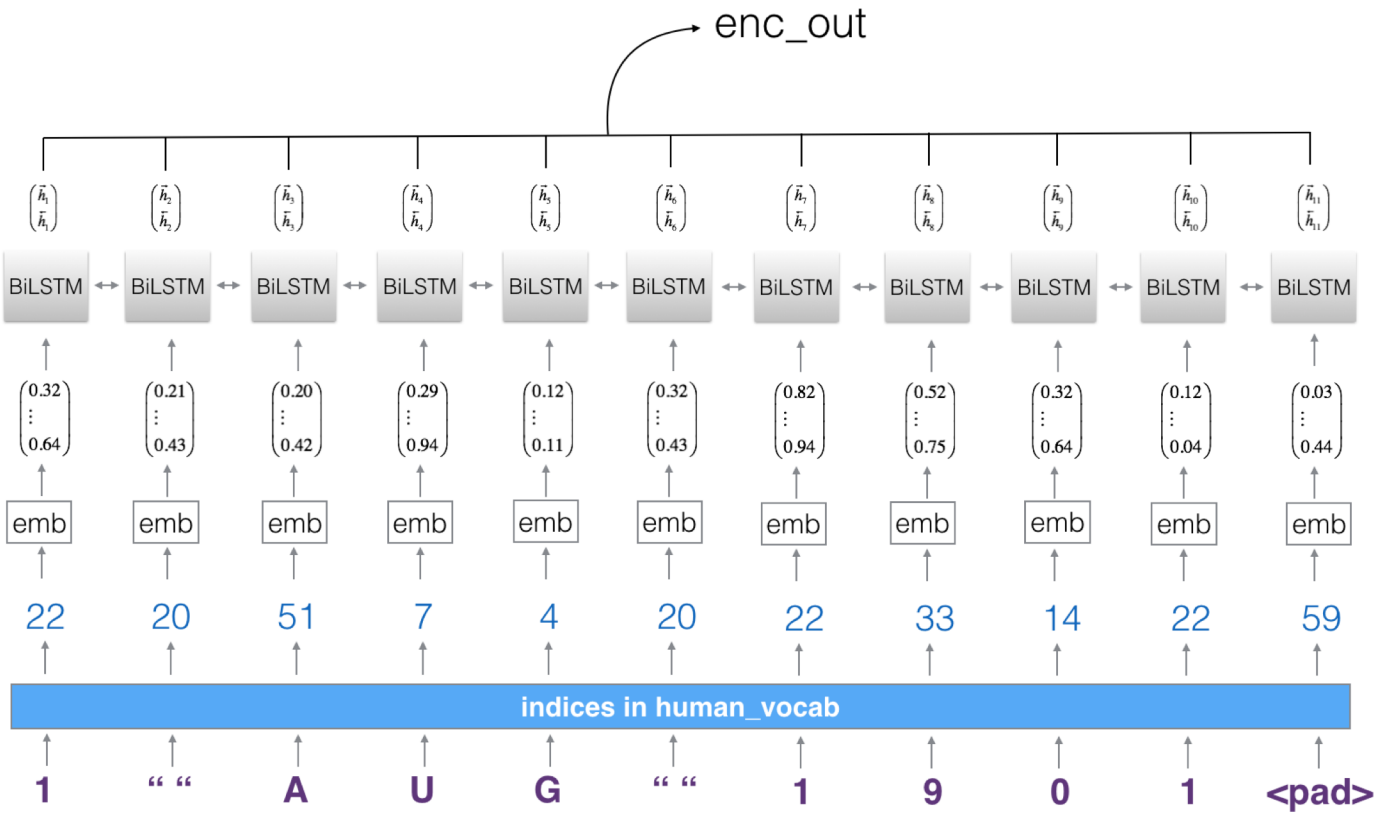


Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, *Yonghui Wu, et al., Technical Report, 2016.*
http://arxiv.org/abs/1609.08144

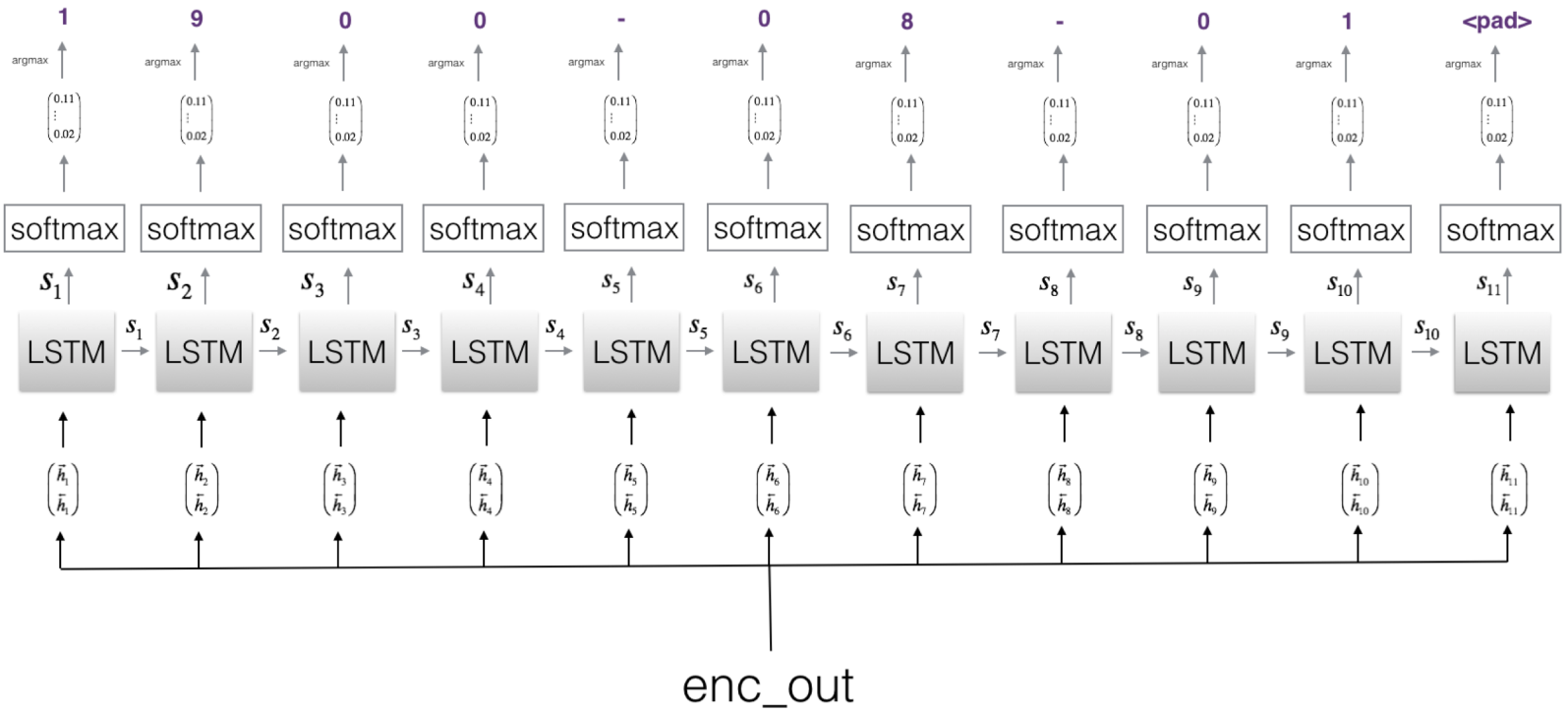
Sequence to Sequence Learning with Neural Networks, *Ilya Sutskever, Oriol Vinyals, Quoc V. Le. Advances in Neural Information Processing Systems, 2014.*
https://arxiv.org/pdf/1409.3215.pdf

Le, Q. V., & Schuster, M. (2016). A neural network for machine translation, at production scale. *Google research blog*.
https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html

Encoder

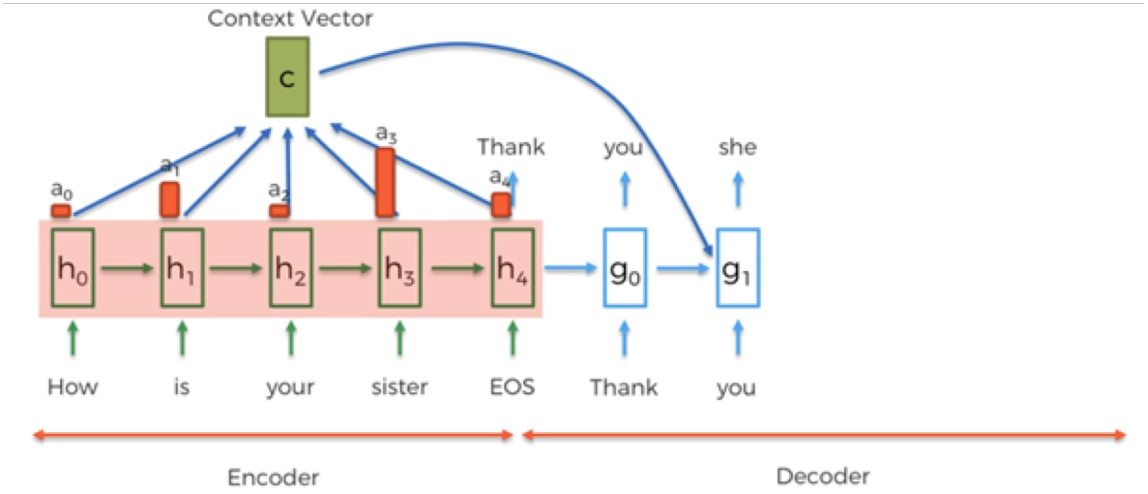

enc_out
BiLSTM
0.32 0.64
0.21 0.43
0.20 0.42
0.29 0.94
0.12 0.11
0.32 0.43
0.82 0.94
0.52 0.75
0.32 0.64
0.12 0.04
0.03 0.44
emb
22 20 51 7 4 20 22 33 14 22 59
indices in human_vocab
1 " " A U G " " 1 9 0 1 <pad>

Decoder



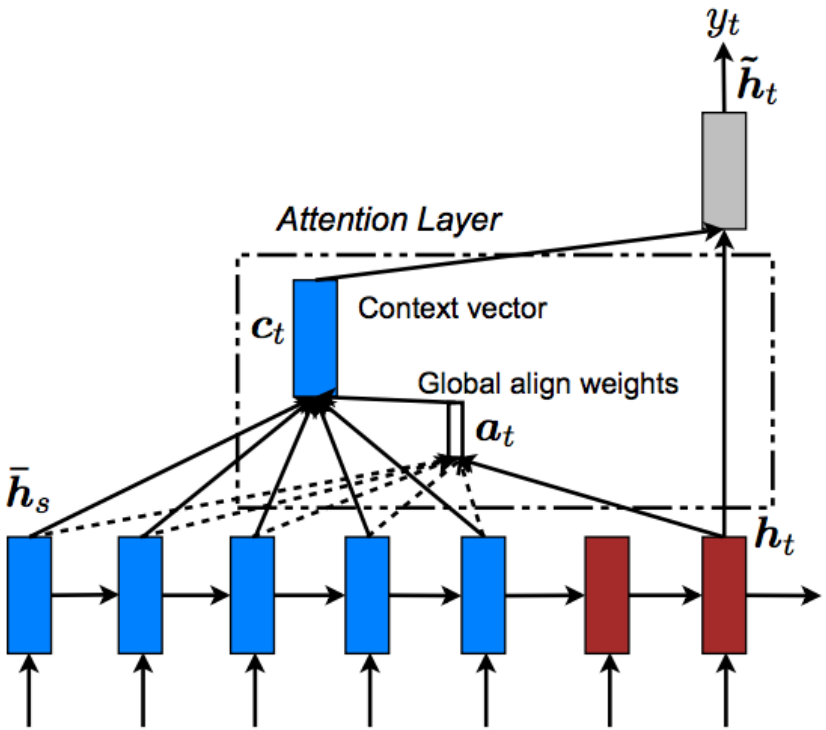
Attention

Attention

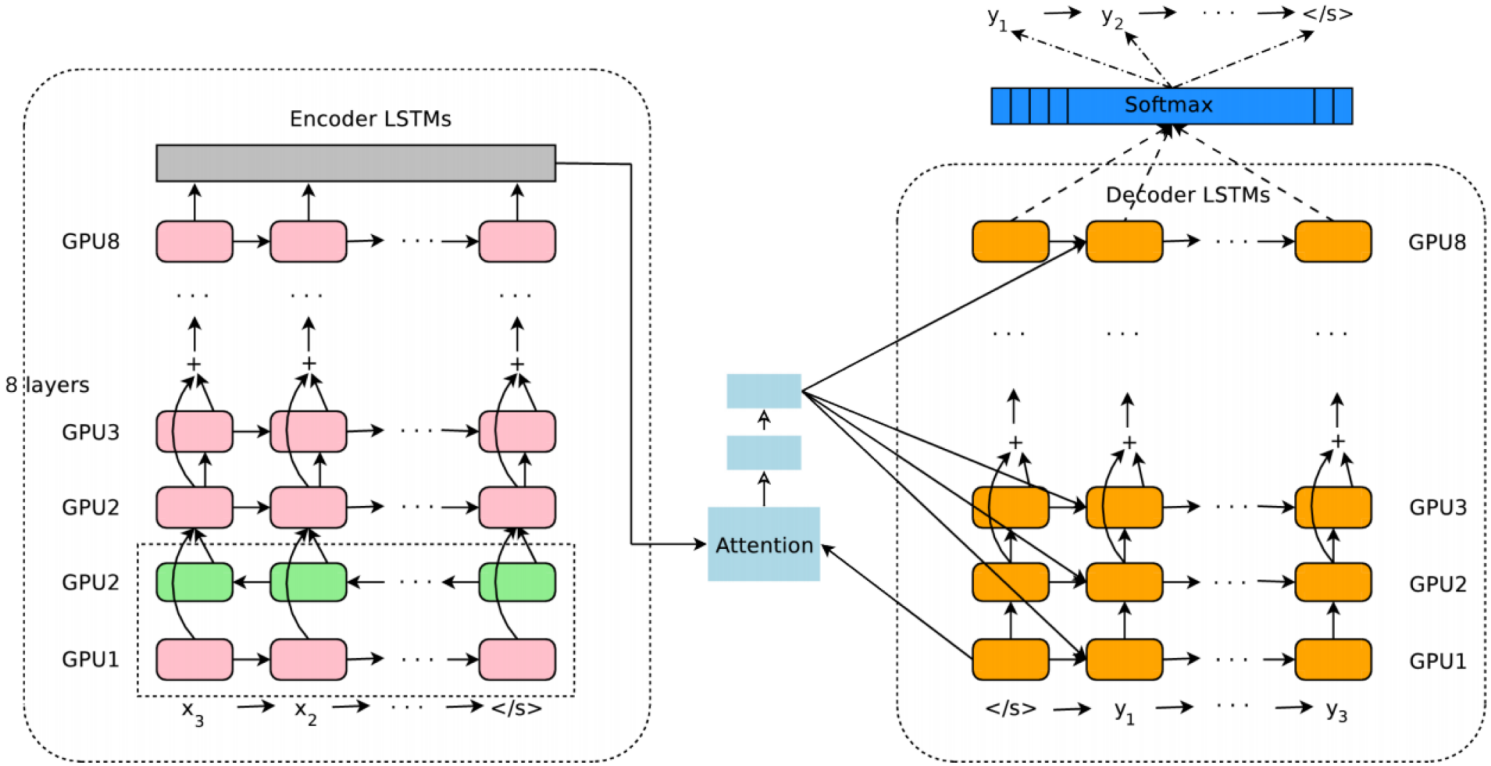

Context Vector
c
a_0
a_1
a_2
a_3
a_4
Thank
you
she
h_0
h_1
h_2
h_3
h_4
g_0
g_1
How
is
your
sister
EOS
Thank
you
Encoder
Decoder

Attention Mechanism

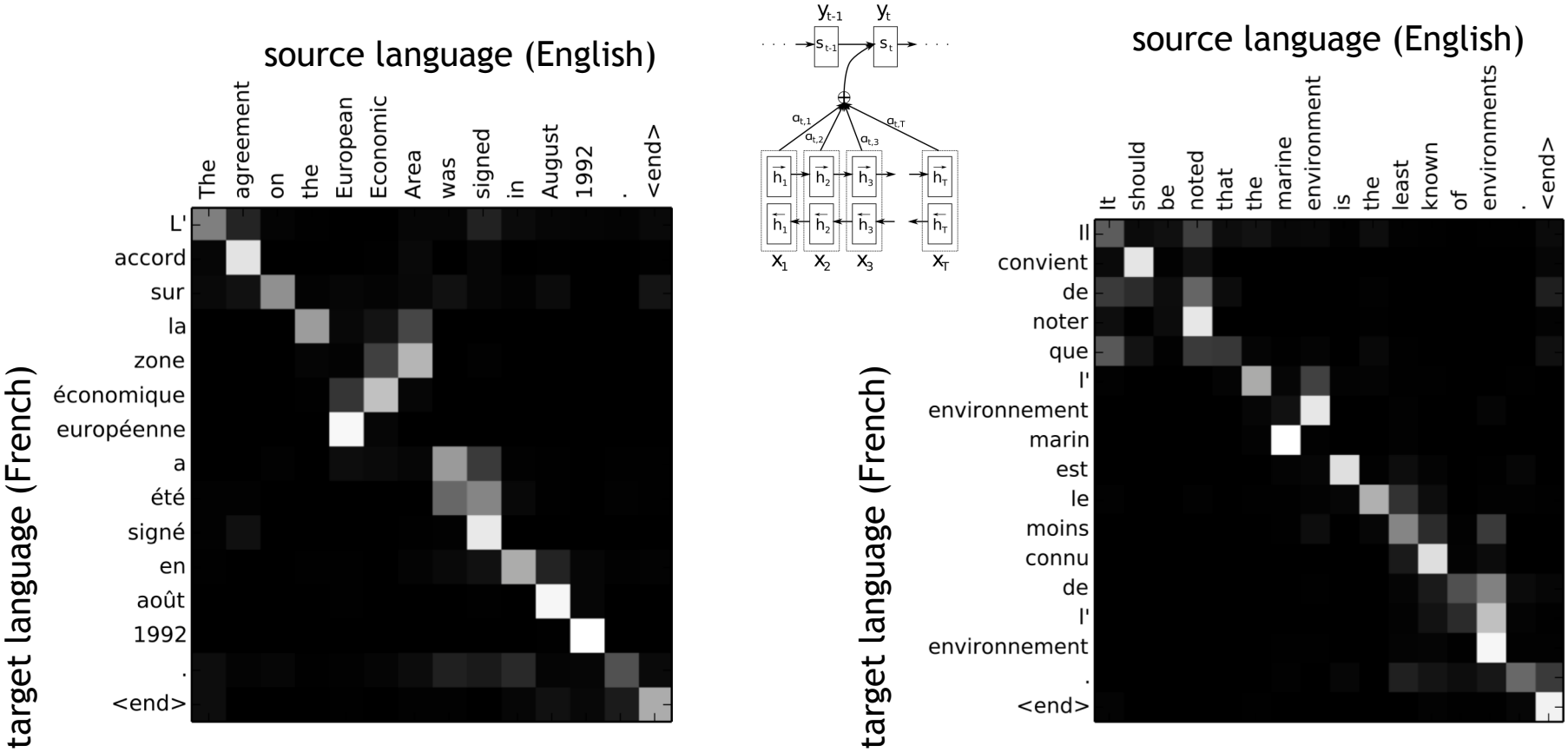
Implementation



Encoder Decoder



ATTENTION MAPS IN TRANSLATION



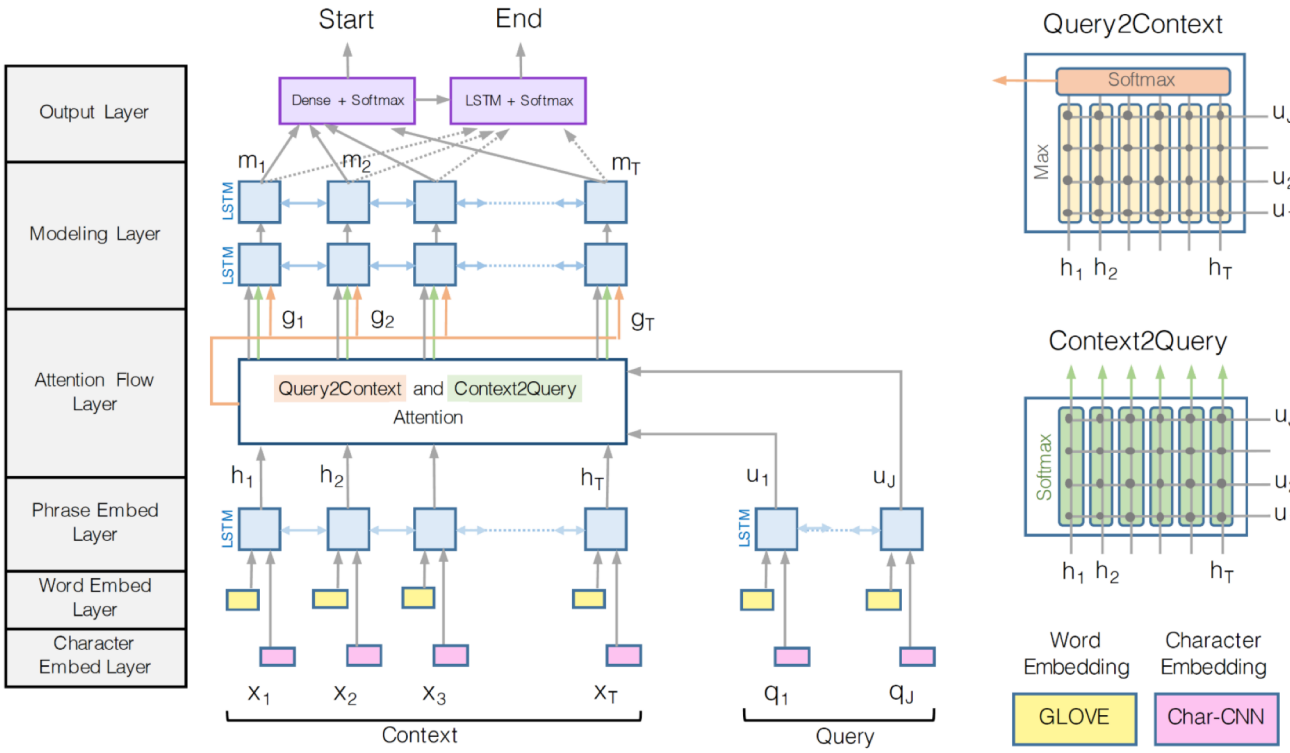
Bahdanau et al. "Neural Machine Translation by Jointly Learning to Align and Translate", 2014

Model 1: Bidirectional LSTM + Attention + Combined GloVe Embedding with character embedding

Model 1: Bi-directional LSTM + attention + word/character embedding

- Answering a query about a given paragraph requires modeling complex interactions between the context and the query.
- Use attention to focus on a small portion of the context and summarize it with a fixed-size vector, couple attentions temporally.
- Use a multi-stage hierarchical process that represents the context at different levels of granularity.
- Use bidirectional attention mechanism to obtain a query-aware context representation without early summarization.
- Incorporate a novel embedding integrating GloVe embeddings with character embeddings.
- Experimental evaluations showed the model achieving state-of-the-art results in Stanford Question Answering Dataset (SQuAD) and CNN/DailyMail cloze test.

BI-DIRECTIONAL ATTENTION FLOW FOR MACHINE COMPREHENSION,
Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hananneh Hajishirzi
https://arxiv.org/pdf/1611.01603.pdf , https://allenai.github.io/bi-att-flow/

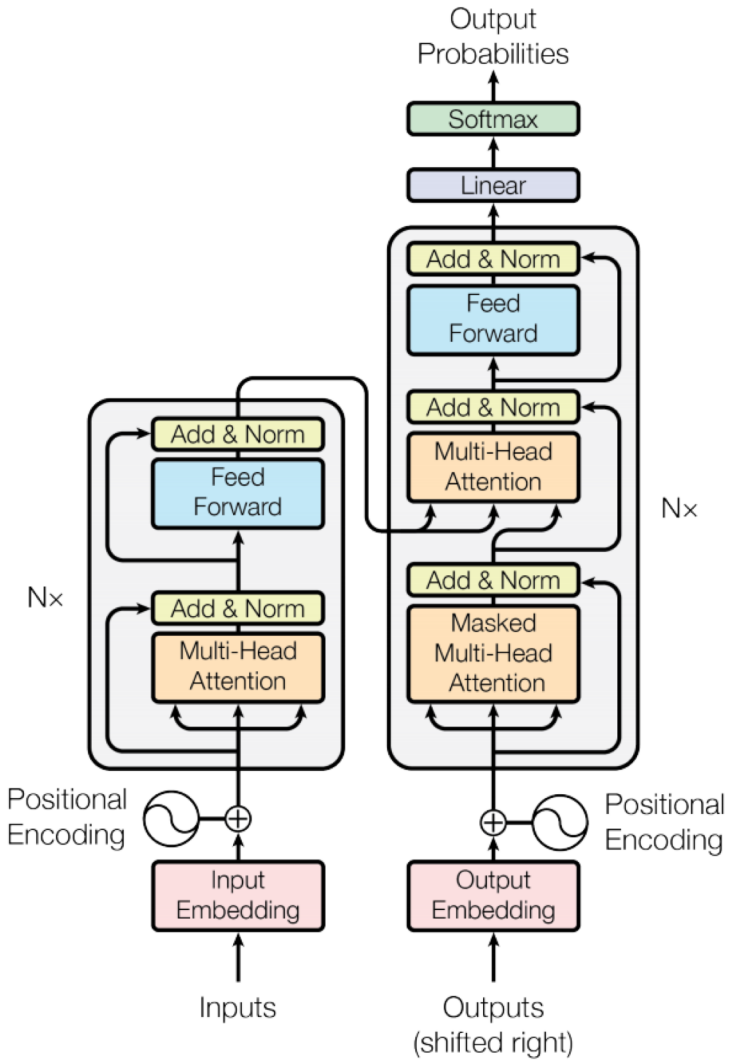


BI-DIRECTIONAL ATTENTION FLOW FOR MACHINE COMPREHENSION,
Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hananneh Hajishirzi
https://arxiv.org/pdf/1611.01603.pdf
https://allenai.github.io/bi-att-flow/

Model 2: Transformer (includes Attention + Self-Attention) + Combined GloVe Embedding with character embedding

Transformer Network
With attention, self-attention

Just using attention they (Google) outperformed the conventional models.



Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). **Attention is all you need**. In *Advances in Neural Information Processing Systems*(pp. 5998-6008).

Transformer - 1

- Feed both the input and output sentences at the same time.
- The outputs initially can be filled with anything, the model ignores whatever you fill into that.
- Uses the entire input sentence and output sentence to predict the next word in a single go.
- Once the next word is predicted, it replaces the current word in output sequence.
- The model only considers output up until that point and ignores what is ahead of it.
- Continues until a complete sentence is formed.

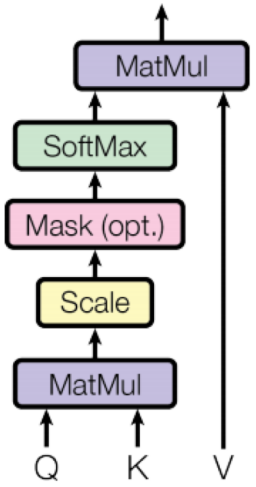
Transformer – 2: Use of Attention

- Uses multi-head attention *(MHA)*.
- Basically, attention is lookup-table that has a large number of values for some other values.
- You query and the table by key and it returns the closest to it.
- Consists of 3 values, ***key, value*** and ***query.***
- Large number of keys, basically 1-dimensional vectors in *n-dimensional* space, where each key has some corresponding value.
- The attention is additive and also uses dot-product. The reason for the formula used can be explained much more effectively using visualizations.
- Look at the image below and you will get a better idea:

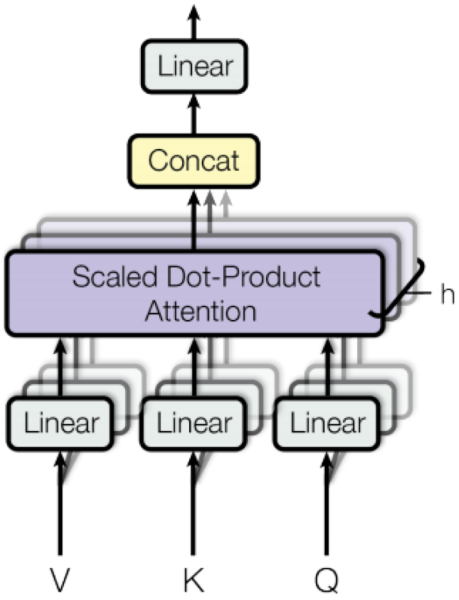
Attention

- Rather than using one attention over the text, apply 8 *(in paper)* attention heads.
- Merge these heads and perform further operations.
- Main advantage is that using more than one attention heads, reduces variance and also allows for better learning over time.
- Its possible that the different heads learn to attend to different things, but final output is same.

Scaled Dot-Product Attention

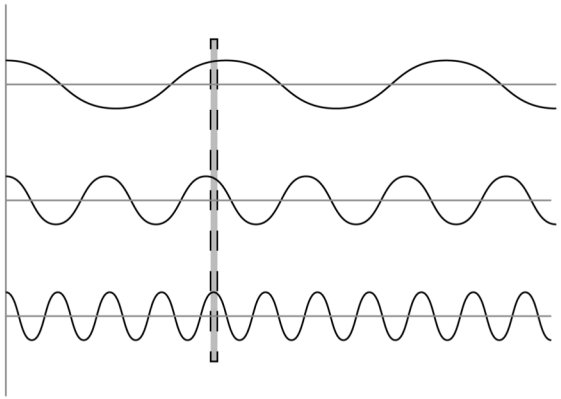


Multi-Head Attention



Transformer: Usage of Positional Encoding

- Model processes both the input and output sentences simultaneously, so it does not have an understanding of positions and places of words in sequence.
- Use a cosine based encoding method: values in between -1 and +1.
- Each positional vector then behaves like a giant array of switches which are either off or on or somewhere in the middle.



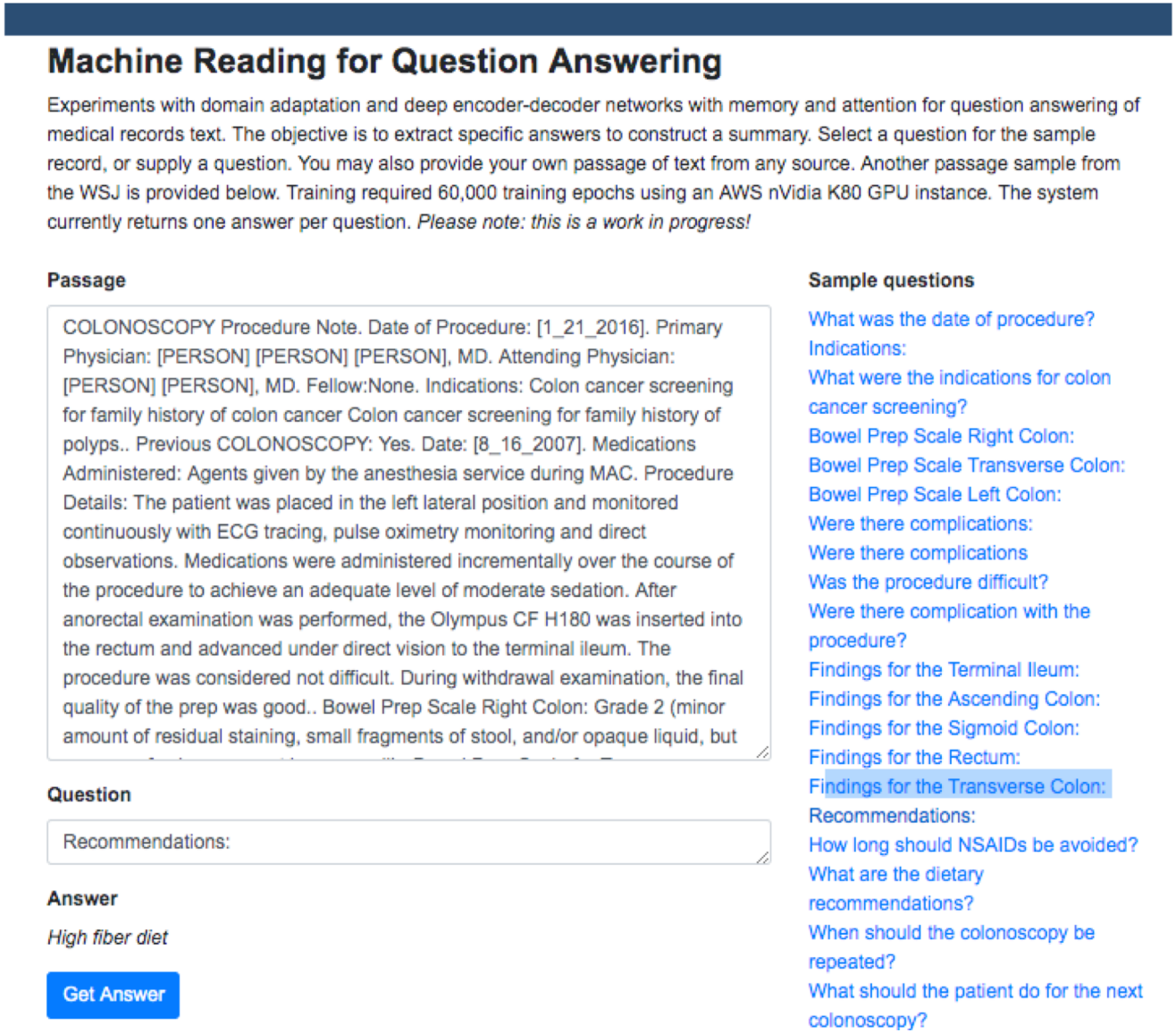
If we select any column *(a vector for single position)*, values change like a large array of switches which are either on or off or somewhere in the middle.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*. https://nlp.stanford.edu/pubs/rajpurkar2016squad.pdf

Jay Urbain, Bradley Crotty. Medical Record Reading via Deep Learning via Deep Attention Networks. nVidia Global Technology Conference, March 2019.

Jay Urbain, Bradley Crotty, Bradley Taylor. Medical Record Reading via Deep Learning Question and Answering. AMIA 2019 Informatics Summit, March 2019.

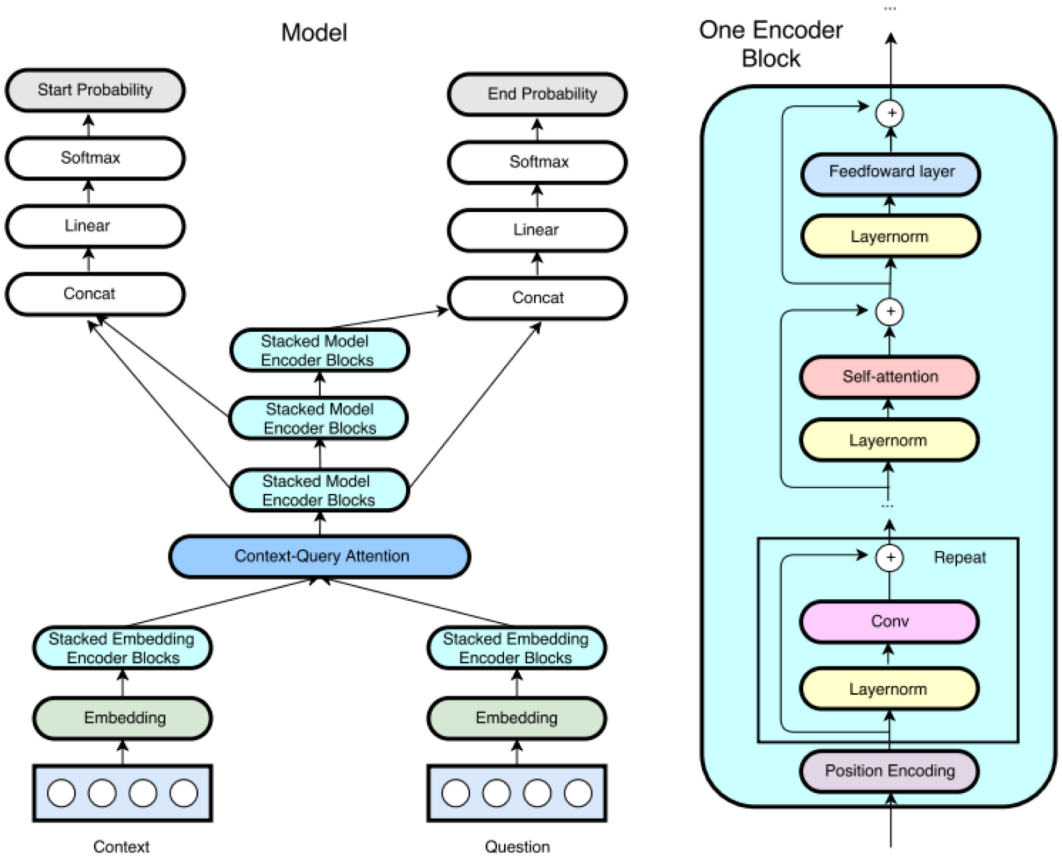
http://ec2-54-163-221-189.compute-1.amazonaws.com:8080/



Model 3: QANet Transformer + Convultional Embedding Layer

Combining CNN with Attention

Yu, A. W., Dohan, D., Luong, M. T., Zhao, R., Chen, K., Norouzi, M., & Le, Q. V. (2018). QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. arXiv preprint arXiv:1804.09541.



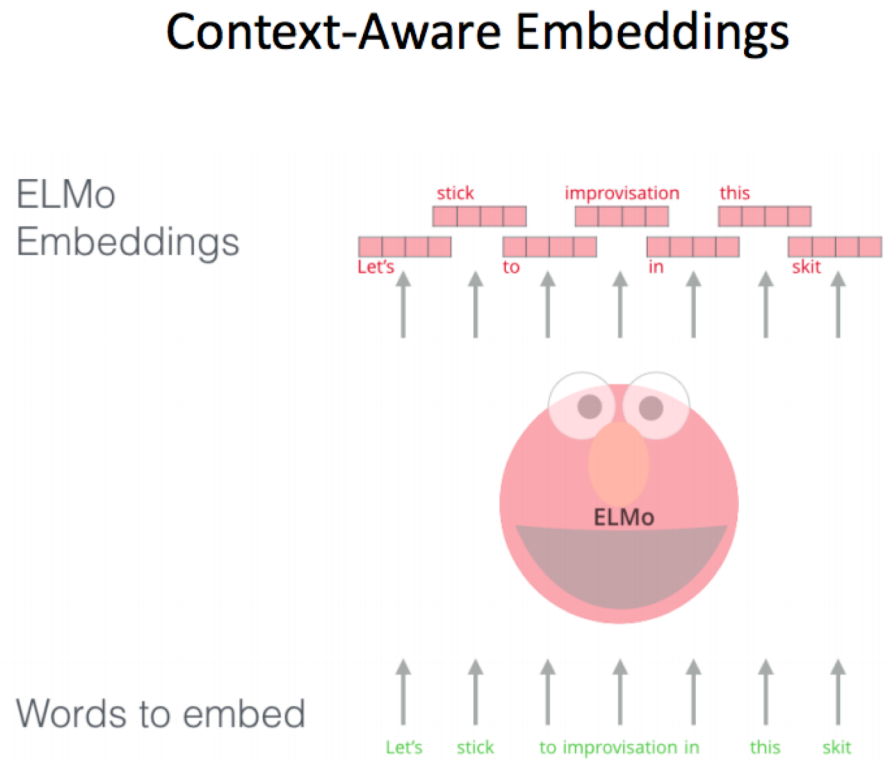
Future: Pretrained Language Models with Fine-Tuning

Breakthrough NLP Developments in 2017 & 2018

- Attention, Self-Attention
- Transformer – Universal Encoder
- Idea: Learn a language model (general model of language) using unsupervised pretraining and domain/application specific fine tuning /transfer learning.
  - ULMFit
  - ElMO
  - OpenAI Transformer - GPT
  - BERT and Natural Language Processing
  - GPT-2
  - Microsoft's New MT-DNN

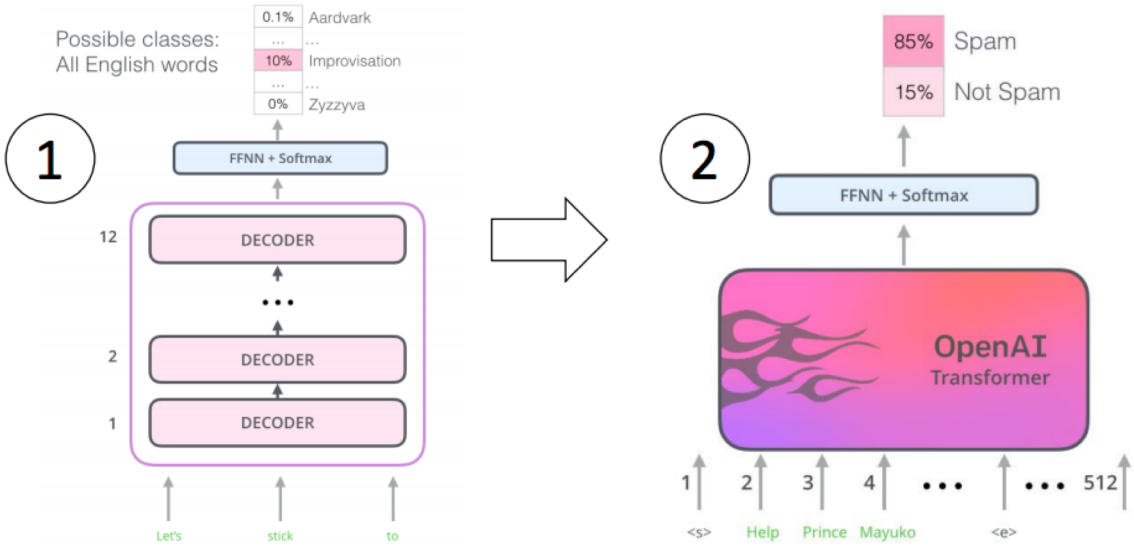
ELMo

- ELMo - deep contextualized word representation
  - Language model using transformer
  - Models complex characteristics of word use
  - How these uses vary across linguistic contexts.
- Word vectors are learned functions of the internal states of a deep bidirectional language model (biLM), which is pre-trained on a large text corpus.
- Can be added to existing models.
- Significantly improves the state of the art across a broad range of challenging NLP problems: including question answering, textual entailment and sentiment analysis



Add reference - AllenAI

OpenAI Transformer



Works in two stages:

1) Train a transformer model on a very large amount of data in an unsupervised manner — using language modeling as a training signal .

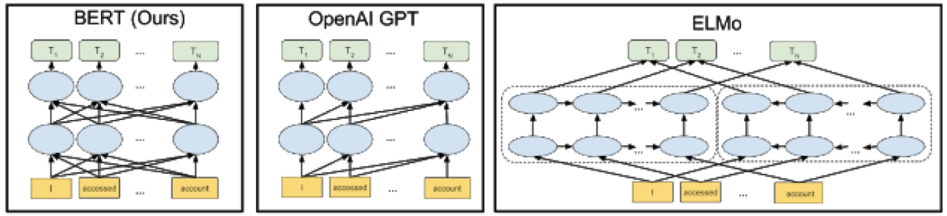
2) Fine-tune this model on much smaller supervised datasets to help it solve specific tasks.

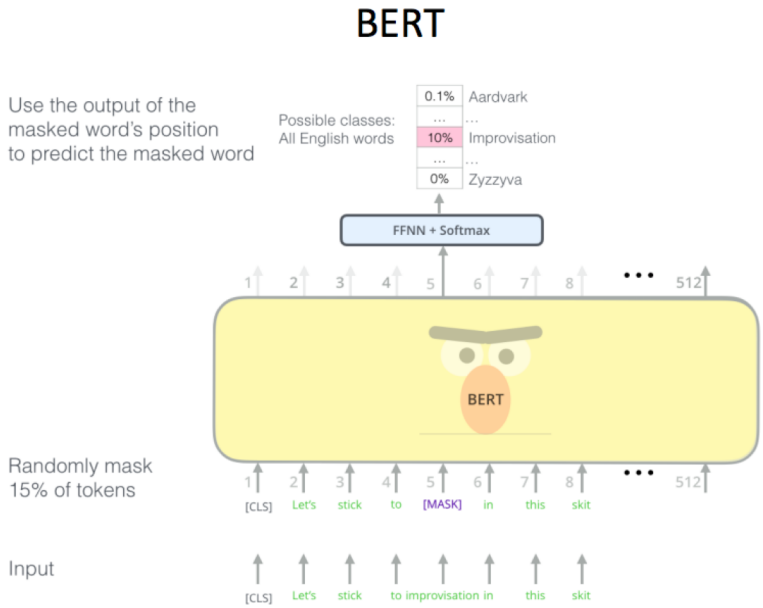
BERT - Bidirectional Encoder Representations from Transformers

BERT builds upon recent work in pre-training contextual representations — including Semi-supervised Sequence Learning.

Unlike ELMo, and ULMFit BERT is the first deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus (Wikipedia).

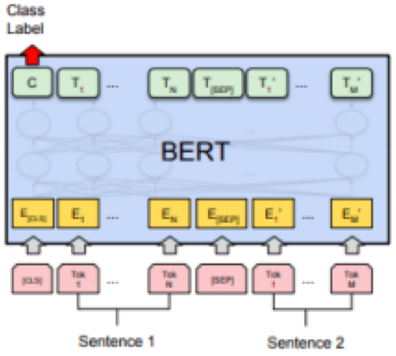
.



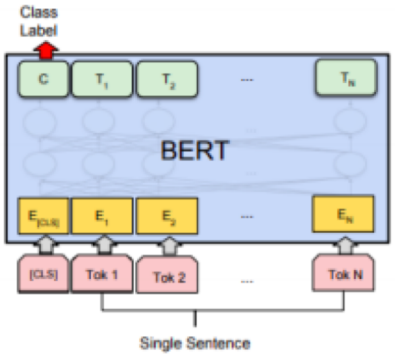


Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." (2018).

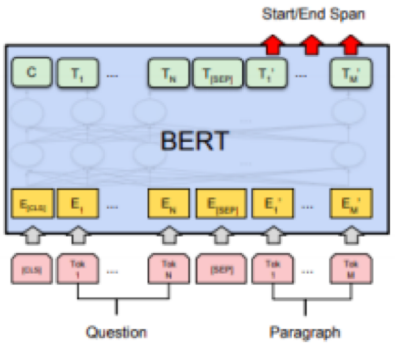
BERT Applications



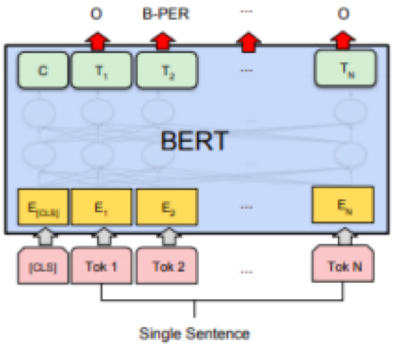
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA



(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Now you can use BERT:

- Create contextualized word embeddings (like ELMo)
- Sentence classification
- Sentence pair classification
- Sentence pair similarity
- Multiple choice
- Sentence tagging
- Question answering

Additional References

- Neural Responding Machine for Short-Text Conversation (2015-03)
- A Neural Conversational Model (2015-06)
- A Neural Network Approach to Context-Sensitive Generation of Conversational Responses (2015-06)
- The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems (2015-06)
- Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models (2015-07)
- A Diversity-Promoting Objective Function for Neural Conversation Models (2015-10)
- Attention with Intention for a Neural Network Conversation Model (2015-10)
- Improved Deep Learning Baselines for Ubuntu Corpus Dialogs (2015-10)
- A Survey of Available Corpora for Building Data-Driven Dialogue Systems (2015-12)
- Incorporating Copying Mechanism in Sequence-to-Sequence Learning (2016-03)
- A Persona-Based Neural Conversation Model (2016-03)
- How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation (2016-03)

Learning resources

Stanford class on deep learning for NLP.http://cs224d.stanford.edu/syllabus.html

Hinton's Coursera Course. Get it right from the horse's mouth. He explains things well.
https://www.coursera.org/course/neuralnets

Online textbook in preparation for deep learning from Yoshua Bengio and friends. Clear and understandable.
http://www.iro.umontreal.ca/~bengioy/dlbook/

TensorFlow tutorials.
https://www.tensorflow.org/versions/r0.8/tutorials/index.html

Additional Sources

- https://cis.ctsi.mcw.edu/deid/
- https://cis.ctsi.mcw.edu/nlp/
- https://github.com/jayurbain/machine-learning
- https://github.com/jayurbain/ctsi-mcw-deid-service
- https://github.com/jayurbain/ctsi-mcw-deid-service