CSCI 5922 - NEURAL NETWORKS AND DEEP LEARNING

RECENT ADVANCES IN DEEP LEARNING FOR NATURAL LANGUAGE PROCESSING

OVERVIEW

- Latent Semantic Analysis, Deerwester et al, 1988 [link] [wikipedia]
- A Neural Probabilistic Language Model, Bengio et al, 2003 [link]
- Recurrent Neural Network-Based Language Model, Mikolov et al, 2010 [link]
- Linguistic Regularities in Continuous Space Word Representations, Mikolov et al, 2013 [link]
- Distributed Representations of Words and Phrases and their Compositionality, Mikolov et al, 2013 [link]
- **Murad Chowdhury presents**: Attention is All you Need, Vaswani et al 2017 [link]
- GLUE Benchmark [link]
- Deep Contextualized Word Representations, Peters et al, 2018 [link]
- Improving Language Understanding by Generative Pre-Training, Redford et al, 2018 [link] (GPT)
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Devlin et al, 2018 [link]

TERM-DOCUMENT MATRIX

Consider a corpus consisting of three sentences. Each sentence is considered a document.

**THE DOCTOR PRESCRIBED MEDICINE TO THE PATIENT.
THE PHYSICIAN PRESCRIBED ANTIBIOTICS.
THE DOCTOR WAS PATIENT WITH HER CHILD.**

This can be represented as a term-document matrix with terms as rows and documents as columns. Let's call it X.

$$\mathbf{X} \in \mathbb{R}^{|V| \times |D|}$$

where

$|V|$: The number of words in the vocabulary

$|D|$: The number of documents in the corpus

Term	Doc1	Doc2	Doc3
antibiotics	0	1	0
child	0	0	1
doctor	1	0	1
her	0	0	1
medicine	1	0	0
patient	1	0	1
physician	0	1	0
prescribed	1	1	0
the	2	1	1
to	1	0	0
was	0	0	1
with	0	0	1

SYNONYMY

Synonyms have the same referent but different forms. In e.g. English, the forms would be spelled differently, whereas in e.g. Chinese the forms would be different characters.

SIMILAR WORDS, DIFFERENT REPRESENTATIONS.

In a term-document matrix, the dot product of the row vectors of synonymous terms is

$$\mathbf{x}_{\text{doctor}} \; \mathbf{x}_{\text{physician}} = 0$$

Term	Doc1	Doc2	Doc3
antibiotics	0	1	0
child	0	0	1
doctor	1	0	1
her	0	0	1
medicine	1	0	0
patient	1	0	1
physician	0	1	0
prescribed	1	1	0
the	2	1	1
to	1	0	0
was	0	0	1
with	0	0	1

POLYSEMY

A polysemous word has multiple meanings, depending on context.

DIFFERENT WORDS, SAME REPRESENTATION.

Here, the dot product of a polysemous word with itself is

$$\mathbf{x}_{\text{patient}} \, \mathbf{x}_{\text{patient}} = 1$$

and the masked dot product of two documents containing different meanings of a polysemous word

$$\mathbf{x}_{\text{doc}_i} \, \mathbf{x}_{\text{doc}_j} = 1$$

Term	Doc1	Doc2	Doc3
antibiotics	0	1	0
child	0	0	1
doctor	1	0	1
her	0	0	1
medicine	1	0	0
patient	1	0	1
physician	0	1	0
prescribed	1	1	0
the	2	1	1
to	1	0	0
was	0	0	1
with	0	0	1

HANDLING SYNONYMY WITH SINGULAR VALUE DECOMPOSITION

Reduced rank SVD

$$\mathbf{X} \in \mathbb{R}^{|V|\times|D|}$$
$$\mathbf{U} \in \mathbb{R}^{|V|\times K}$$
$$\mathbf{\Sigma} \in \mathbb{R}^{K\times K}$$
$$\mathbf{V^T} \in \mathbb{R}^{K\times|D|}$$

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

Orthogonal matrix with left singular vectors (each row corresponds to a term).

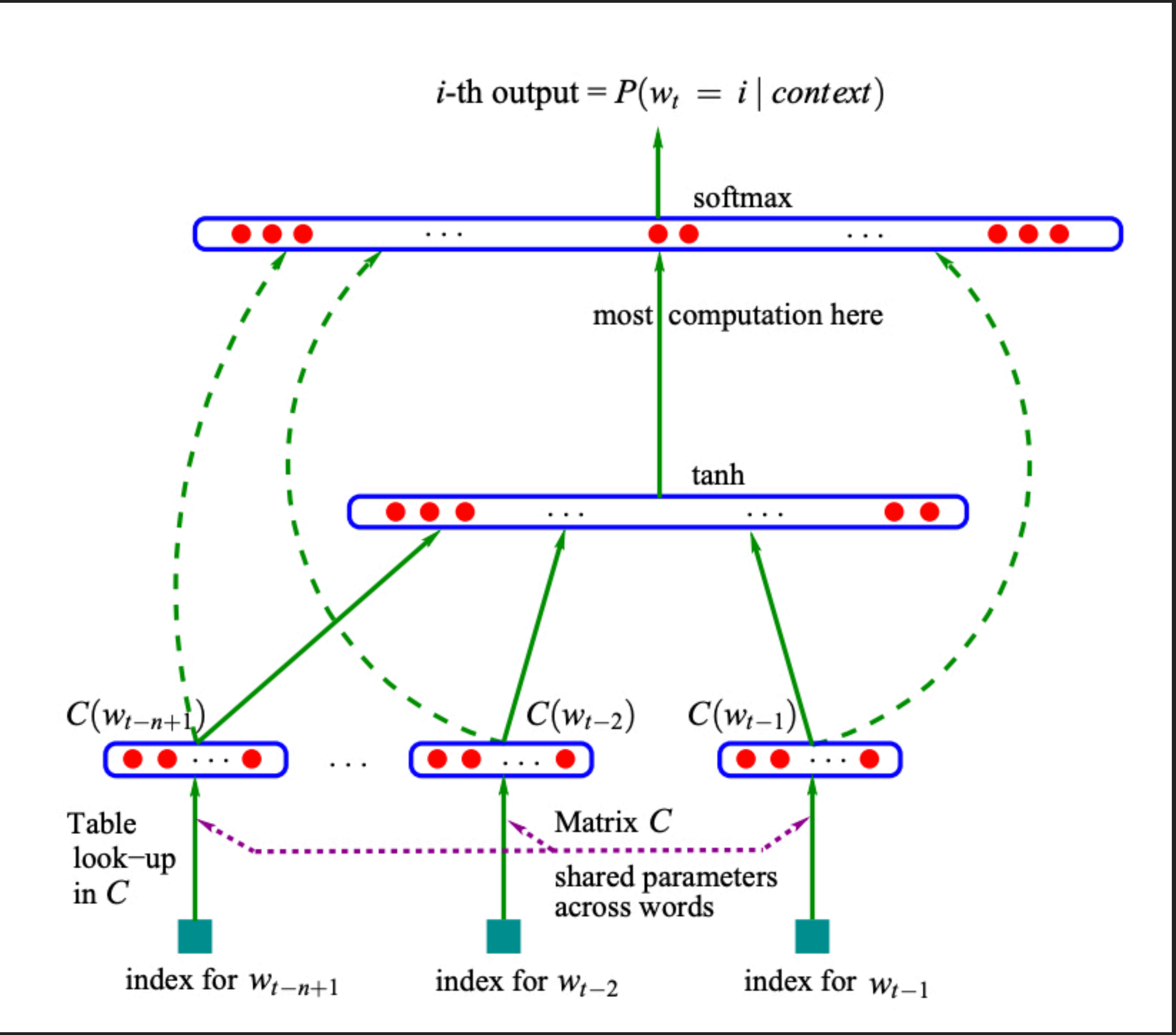
Diagonal matrix with singular values.

Orthogonal matrix richt singular vectors (each column corresponds to a document).

THIS DOES NOT HELP WITH POLYSEMY.

BENGIO ET AL, 2003 [LINK]

- Associate a distributed representation (a vector) with each word in the vocabulary.
- To predict the next word, express the joint probability of word sequences in terms of the feature vectors of these words in the sequence.
- Simultaneously learn the distributed representations and the parameters of the classifier.



BENGIO ET AL, 2003 [LINK]

- Results on the Brown corpus.
- *Direct* indicates whether there are direct connections from input to output.
- *Mix* indicates whether network output probability and trigram-model probability are averaged.

	n	c	h	m	direct	mix	train.	valid.	test.
MLP1	5		50	60	yes	no	182	284	268
MLP2	5		50	60	yes	yes		275	257
MLP3	5		0	60	yes	no	201	327	310
MLP4	5		0	60	yes	yes		286	272
MLP5	5		50	30	yes	no	209	296	279
MLP6	5		50	30	yes	yes		273	259
MLP7	3		50	30	yes	no	210	309	293
MLP8	3		50	30	yes	yes		284	270
MLP9	5		100	30	no	no	175	280	276
MLP10	5		100	30	no	yes		265	**252**
Del. Int.	3						31	352	336
Kneser-Ney back-off	3							334	323
Kneser-Ney back-off	4							332	321
Kneser-Ney back-off	5							332	321
class-based back-off	3	150						348	334
class-based back-off	3	200						354	340
class-based back-off	3	500						326	**312**
class-based back-off	3	1000						335	319
class-based back-off	3	2000						343	326
class-based back-off	4	500						327	312
class-based back-off	5	500						327	312

MIKOLOV ET AL, 2010 [LINK]

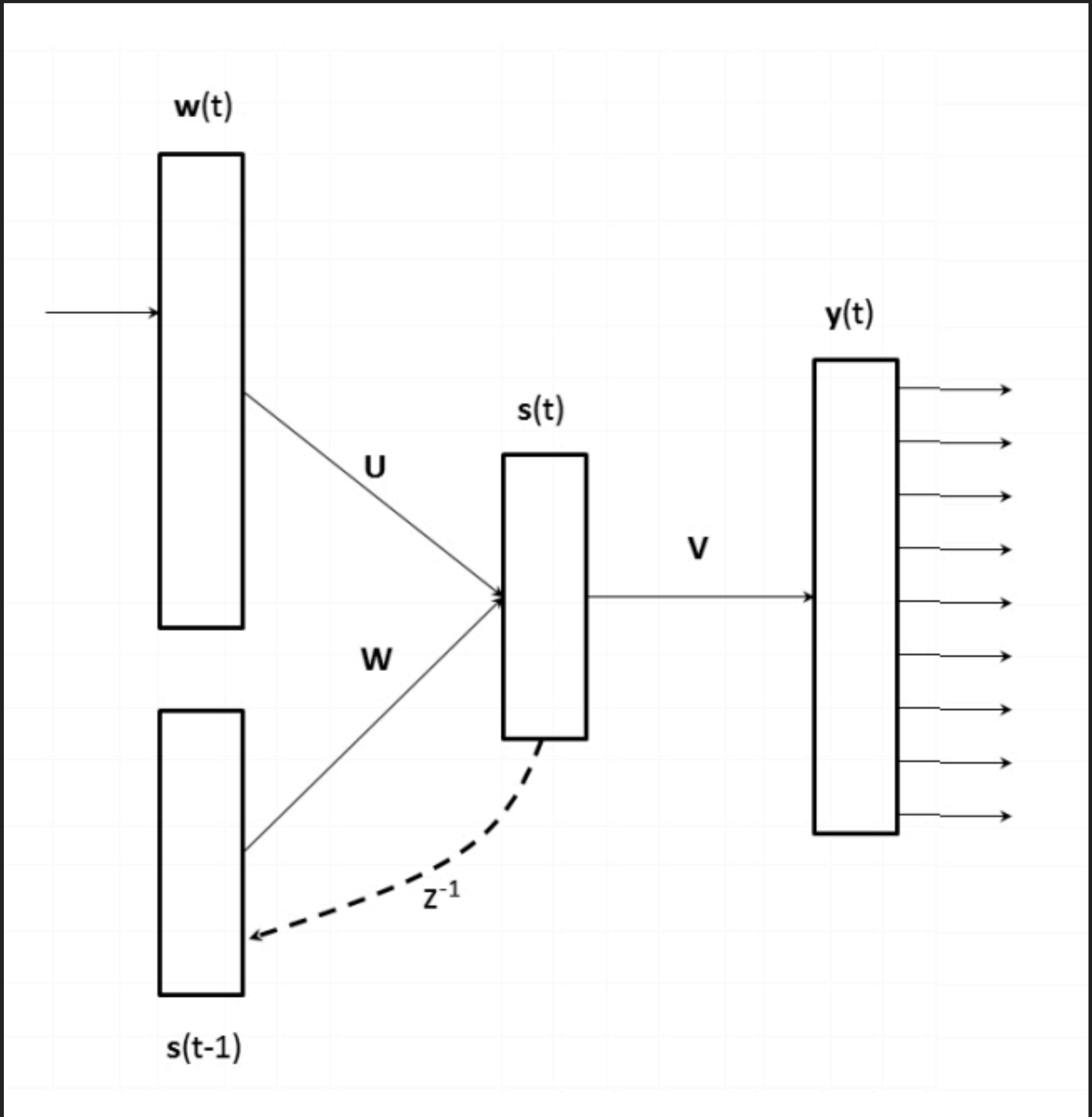
$$x(t) = w(t) + s(t-1)$$

$$s_j(t) = f\left(\sum_i x_i(t) u_{ji}\right)$$

$$y_k(t) = g\left(\sum_j s_j(t) v_{kj}\right)$$

$$f(z) = \frac{1}{1+e^{-z}}$$

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$



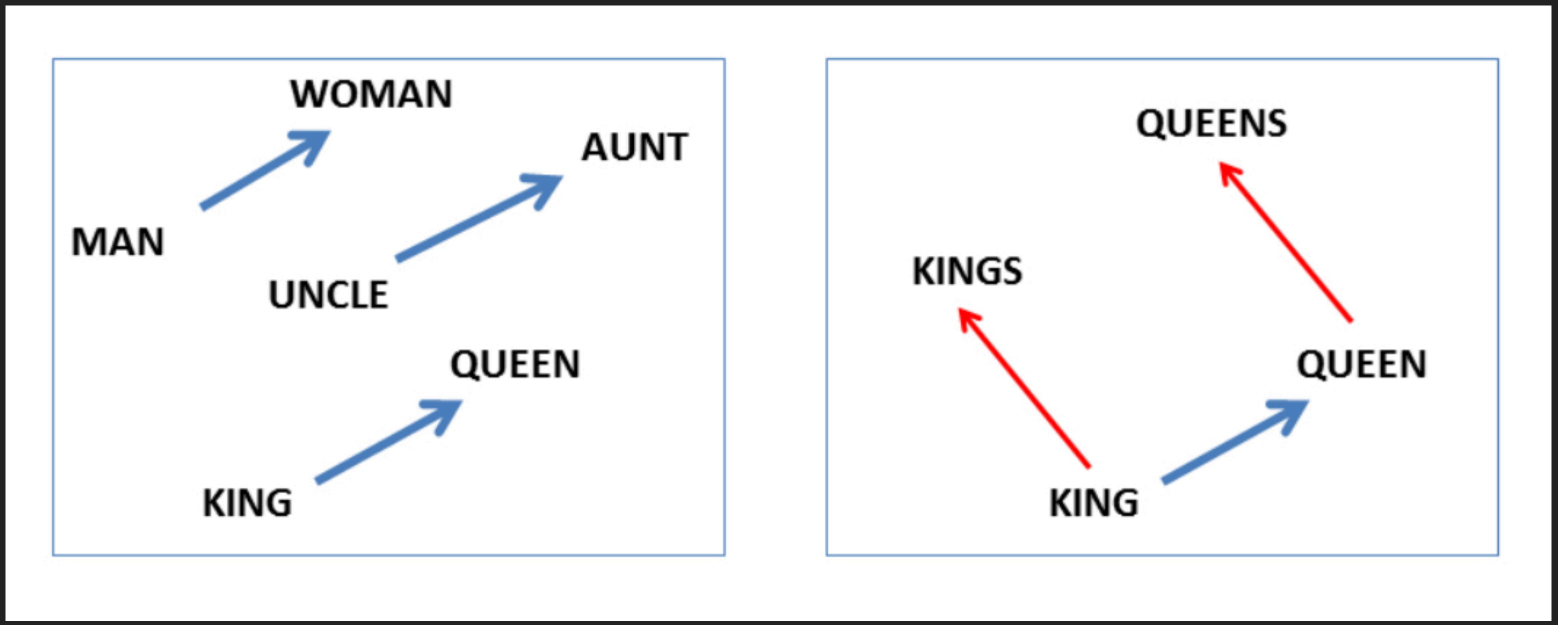
MIKOLOV ET AL, 2013 [LINK]

Given unit-normalized distributed word representations learned by a recurrent network, and given an analogy question a:b c:d, take the word embeddings x_i for words a, b, c, d and compute:

$$y = x_b - x_a + x_c$$

Then find the nearest word w* to y.

$$w^* = \text{argmax}_w \frac{x_w y}{||x_w||\,||y||}$$



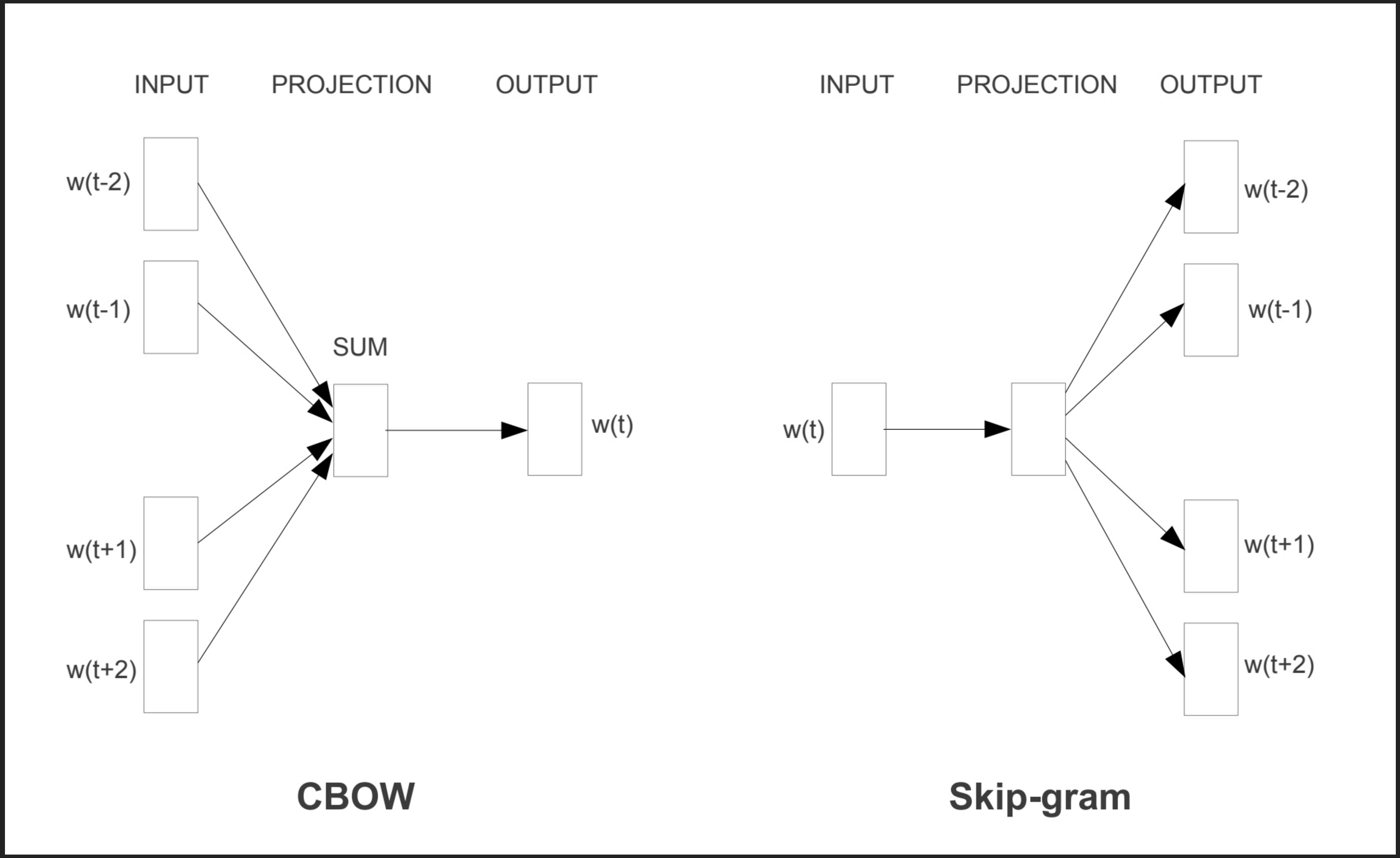
Then plug w* into the analogy to complete it. If w* is c, the model completes the analogy correctly.

LINGUISTIC REGULARITIES IN CONTINUOUS SPACE WORD REPRESENTATIONS

Method	Adjectives	Nouns	Verbs	All
LSA-80	9.2	11.1	17.4	12.8
LSA-320	11.3	18.1	20.7	16.5
LSA-640	9.6	10.1	13.8	11.3
RNN-80	9.3	5.2	30.4	16.2
RNN-320	18.2	19.0	45.0	28.5
RNN-640	21.0	25.2	54.8	34.7
RNN-1600	**23.9**	**29.2**	**62.2**	**39.6**

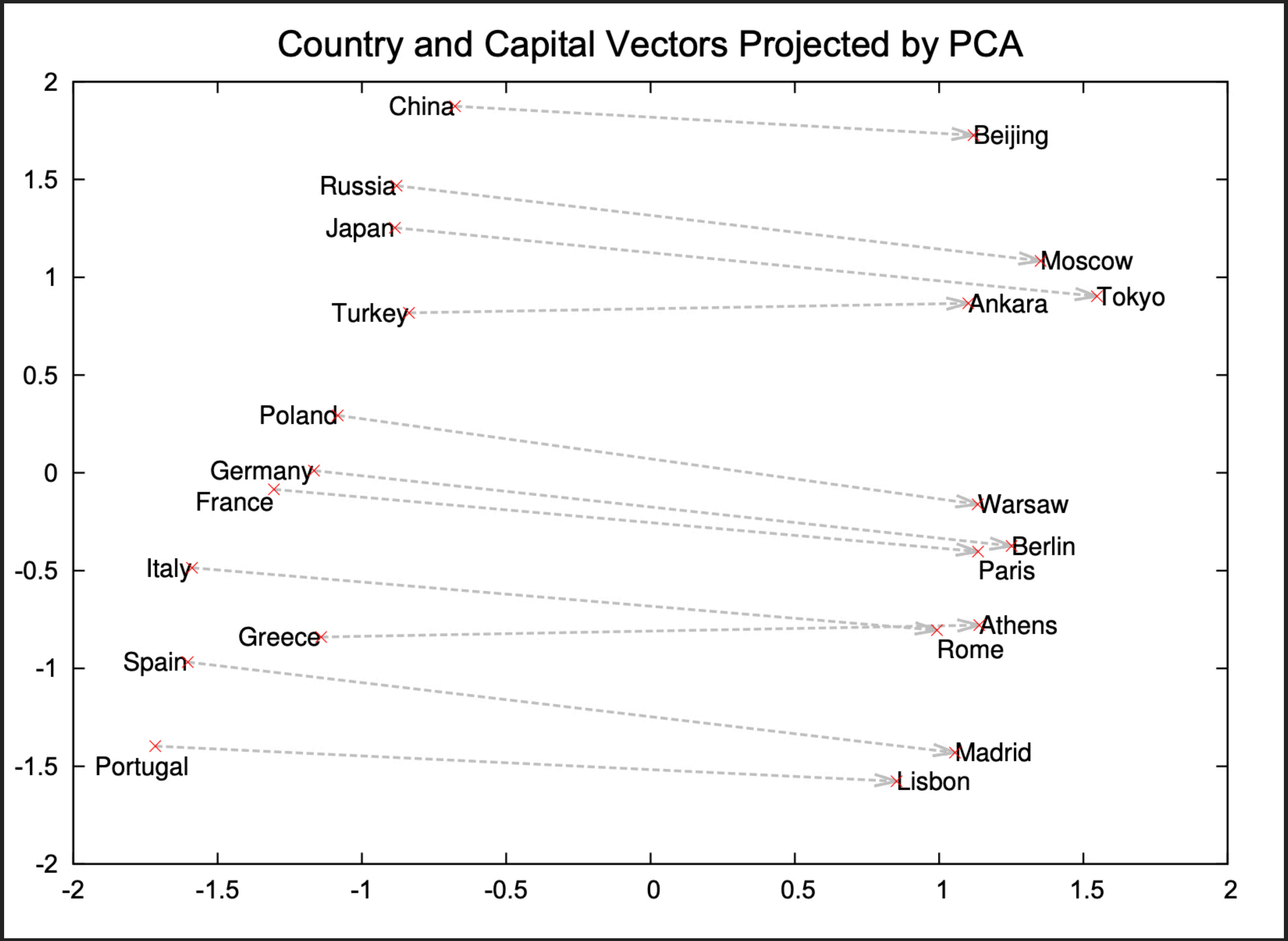
Method	Adjectives	Nouns	Verbs	All
RNN-80	**10.1**	8.1	**30.4**	**19.0**
CW-50	1.1	2.4	8.1	4.5
CW-100	1.3	4.1	8.6	5.0
HLBL-50	4.4	5.4	23.1	13.0
HLBL-100	7.6	**13.2**	30.2	18.7

MIKOLOV ET AL, 2013 [LINK]



A more efficient way of computing word2vec – called negative sampling – was introduced later in 2013 [link]. What's inefficient about the above?

GLUE BENCHMARK



What might explain the slope of Turkey-Ankara?

PETERS ET AL, 2018 [LINK]

- Basic idea: use bidirectional language model to obtain contextual word representations.
- Transfer features from all layers of network to supervised tasks.
- Obtain SOTA performance!

For a given supervised task, the non-contextual embeddings and contextual hidden states are reweighted using a softmax s^task to create a task-specific ELMo vector.

$$
\begin{aligned}
R_k &= \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1, \ldots, L\} \\
&= \{\mathbf{h}_{k,j}^{LM} \mid j = 0, \ldots, L\},
\end{aligned}
$$

$$
\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}.
$$

Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

MURAD CHOWDHURY PRESENTS

WANG ET AL, 2019 [LINK]

The GLUE Benchmark has two parts. The first is a set of datasets for different tasks (see below). These datasets vary in size and all have pre-allocated test sets. The training sets can be used in a multitask learning setting.

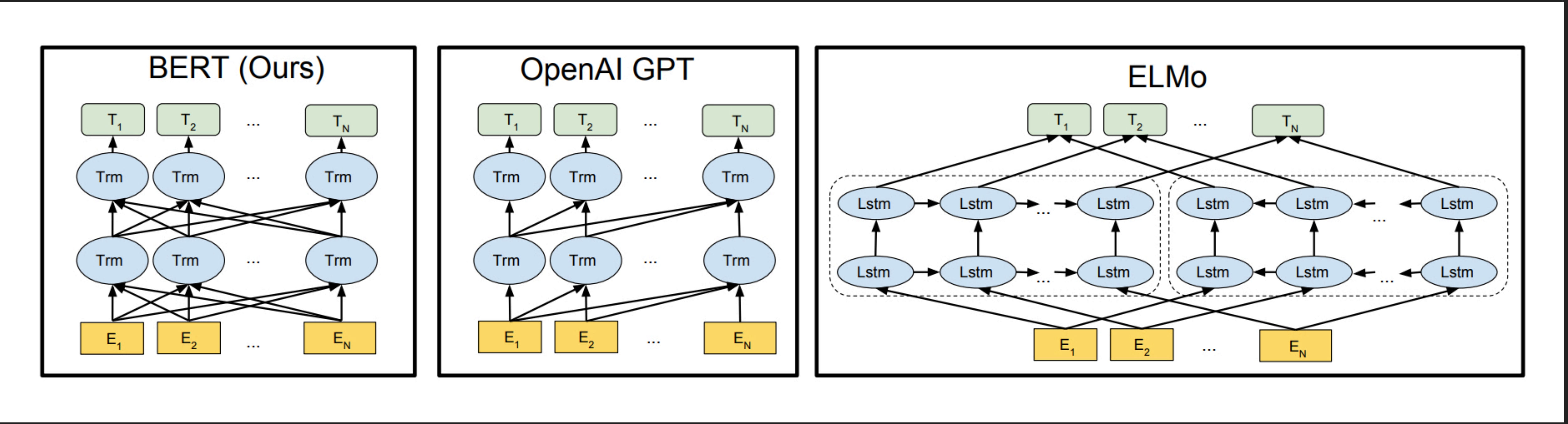
Corpus	|Train|	|Test|	Task	Metrics	Domain
			Single-Sentence Tasks		
CoLA	8.5k	**1k**	acceptability	Matthews corr.	misc.
SST-2	67k	1.8k	sentiment	acc.	movie reviews
			Similarity and Paraphrase Tasks		
MRPC	3.7k	1.7k	paraphrase	acc./F1	news
STS-B	7k	1.4k	sentence similarity	Pearson/Spearman corr.	misc.
QQP	364k	**391k**	paraphrase	acc./F1	social QA questions
			Inference Tasks		
MNLI	393k	**20k**	NLI	matched acc./mismatched acc.	misc.
QNLI	105k	5.4k	QA/NLI	acc.	Wikipedia
RTE	2.5k	3k	NLI	acc.	news, Wikipedia
WNLI	634	**146**	coreference/NLI	acc.	fiction books

WANG ET AL, 2019 [LINK]

The second part is a set of diagnostic tests. These have no training set. They are intended for evaluation purposes only, and they test features of language that are considered essential to natural language understanding.

Coarse-Grained Categories	**Fine-Grained Categories**
Lexical Semantics	Lexical Entailment, Morphological Negation, Factivity, Symmetry/Collectivity, Redundancy, Named Entities, Quantifiers
Predicate-Argument Structure	Core Arguments, Prepositional Phrases, Ellipsis/Implicits, Anaphora/Coreference Active/Passive, Nominalization, Genitives/Partitives, Datives, Relative Clauses, Coordination Scope, Intersectivity, Restrictivity
Logic	Negation, Double Negation, Intervals/Numbers, Conjunction, Disjunction, Conditionals, Universal, Existential, Temporal, Upward Monotone, Downward Monotone, Non-Monotone
Knowledge	Common Sense, World Knowledge

DEVLIN ET AL, 2018 [LINK]



BERT is fully bi-directional. ELMO is two independent LSTMs consuming a sentence and its reverse. OpenAI GPT is a forward-only transformer.

BERT is trained like a Cloze test.

The other day I was on a _____ in the park and I saw a squirrel.

Except that multiple words are deleted. Unlike a de-noising autoencoder, which is trained to reconstruct the input completely, BERT is trained to predict only the missing words.

DEVLIN ET AL, 2018 [LINK]

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

GLUE BENCHMARK AS OF APRIL 2019

	Rank	Name	Model	URL	Score
	1	GLUE Human Baselines	GLUE Human Baselines		87.1
+	2	Microsoft D365 AI & MSR AI	MT-DNN++ (BigBird)		83.8
+	3	王玮	ALICE large (Alibaba DAMO NLP)		83.3
	4	Stanford Hazy Research	Snorkel MeTaL		83.2
	5	Anonymous Anonymous	BERT + BAM		82.3
	6	张倬胜	SemBERT		82.0
+	7	Jason Phang	BERT on STILTs		82.0
+	8	Jacob Devlin	BERT: 24-layers, 16-heads, 1024-hidde		80.5
	9	Neil Houlsby	BERT + Single-task Adapters		80.2
	10	Alec Radford	Singletask Pretrain Transformer		72.8
	11	GLUE Baselines	BiLSTM+ELMo+Attn		70.0