Deep Learning for NLP

Lecture 7 – Word Embeddings 3 (Sentence Embeddings)

Dr. Steffen Eger
Wei Zhao
Niraj Pandey



Natural Language Learning Group (NLLG)
Technische Universität Darmstadt

This lecture

1. **Embeddings of sentences (or even documents)**

2. (Problems with) Evaluation of Sentence Embeddings

Embedding of sentences

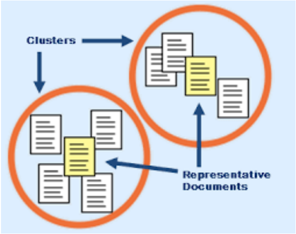
- Our methods so far only embedded *words* in a low dimensional dense space

- How about larger objects such as phrases, sentences, or even whole documents?

  - Would be cool if we could represent the meaning of a sentence in a low-dimensional space
    - Why?

Why sentence/document embeddings?

- For clustering



- For retrieval
  - Given question, give me an answer
  - Given sentence, give me a similar sentence
  - Given sentence, give me a translated sentence

- As an alternative to sentence representations learned from word-level models (e.g. CNN)
  - Particularly, when task-specific training data is small

Sentence Embeddings: Naive approaches

- **Naïve approach number 1:**
  - Treat sentence as long word, predict surrounding sentences like in CBOW or SKIP-GRAM
    - The cat sat on the mat → The_cat_sat_on_the_mat
  - Problems with this approach? Extreme data sparsity

- **Naïve approach number 2:**
  - Take some sort of mean (e.g. arithmetic mean of words in the sentences = centroid)
    - Embedding of “cat sat on the mat” is the average embedding of all of words in the sentence

Sentence Embeddings: Naive approaches

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Naïve approach number 2:**
  - Take some sort of mean (e.g. arithmetic mean of words in the sentences = centroid)
    - Embedding of “cat sat on the mat” is the average embedding of all of words in the sentence
    - Problems with this:
      - Half of all words in a sentence are function words (“noise”) which shouldn’t contribute a lot
        - Have to discard high frequency words (e.g. use stop word list or determine them by counting)
        - Or even better: weight them down via, e.g., inverse document frequency
      - Word order is ignored. Embedding for “cat sat on the mat” and “mat cat sat on the” are the same
    - However, the mean (weighted) word vector is often a reasonable baseline

Sentence Embeddings

- To outline more sophisticated approaches, we briefly need to peek ahead

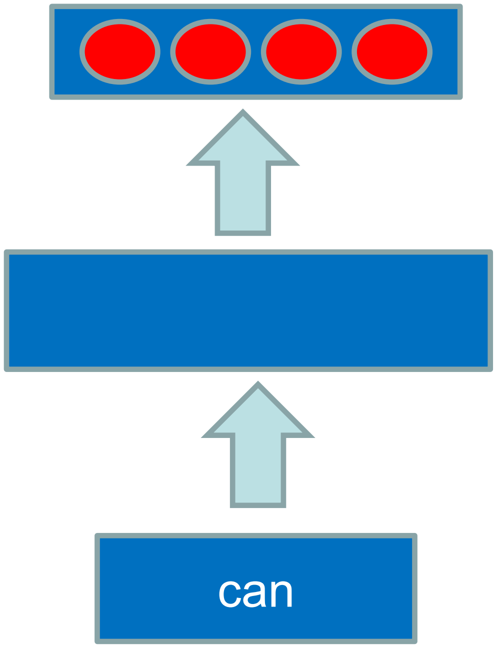
- And introduce so-called **encoder-decoder models**, discussed in more detail in Lecture 9
  - To understand these, we first need to understand **recurrent neural nets** (Lecture 8)

Excursion 1: RNNs

Motivation

We want to do POS tagging

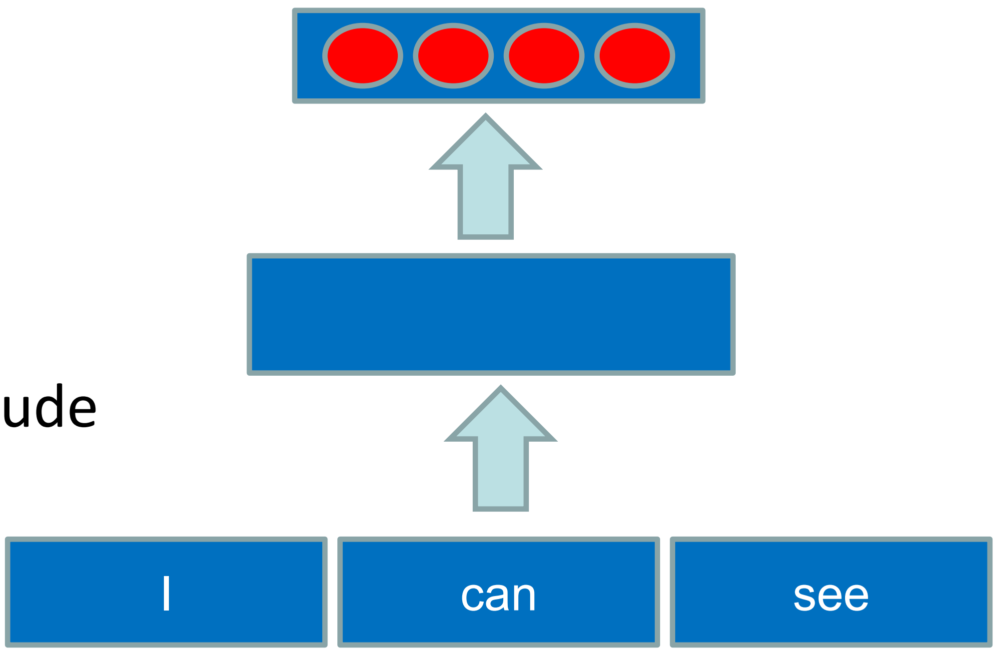
- That is, label each token in a sentence with its part-of-speech (= word class)
- I can see the cat



Motivation

We want to do POS tagging

- That is, label each token in a sentence with its part-of-speech (= word class)
- I can see the cat

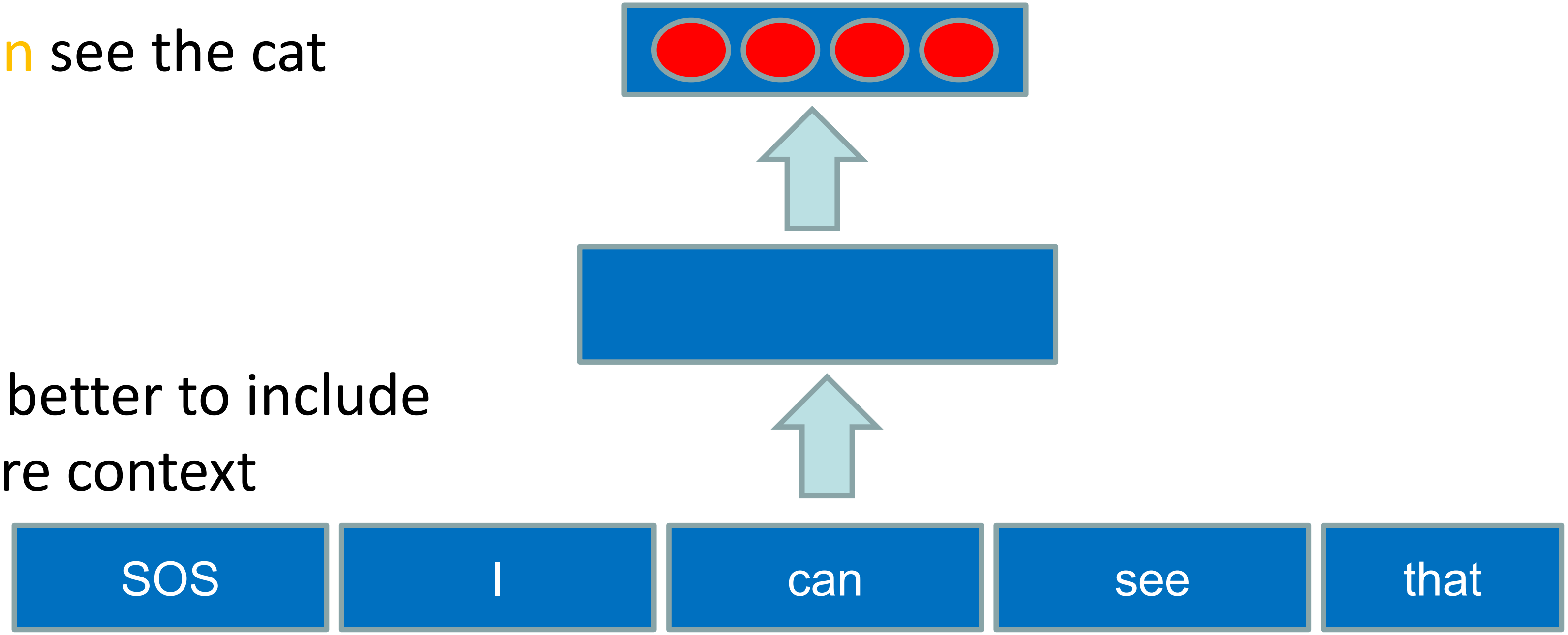


- It's better to include context

Motivation

We want to do POS tagging

- That is, label each token in a sentence with its part-of-speech (= word class)
- I can see the cat
- It's better to include more context



Motivation

We want to do POS tagging

- That is, label each token in a sentence with its part-of-speech (= word class)
- I can see the cat

- Problem 1: How much context?
- Problem 2: We cannot simply add more and more context because
  - We have many more parameters then

    → Overfitting

    → Speed

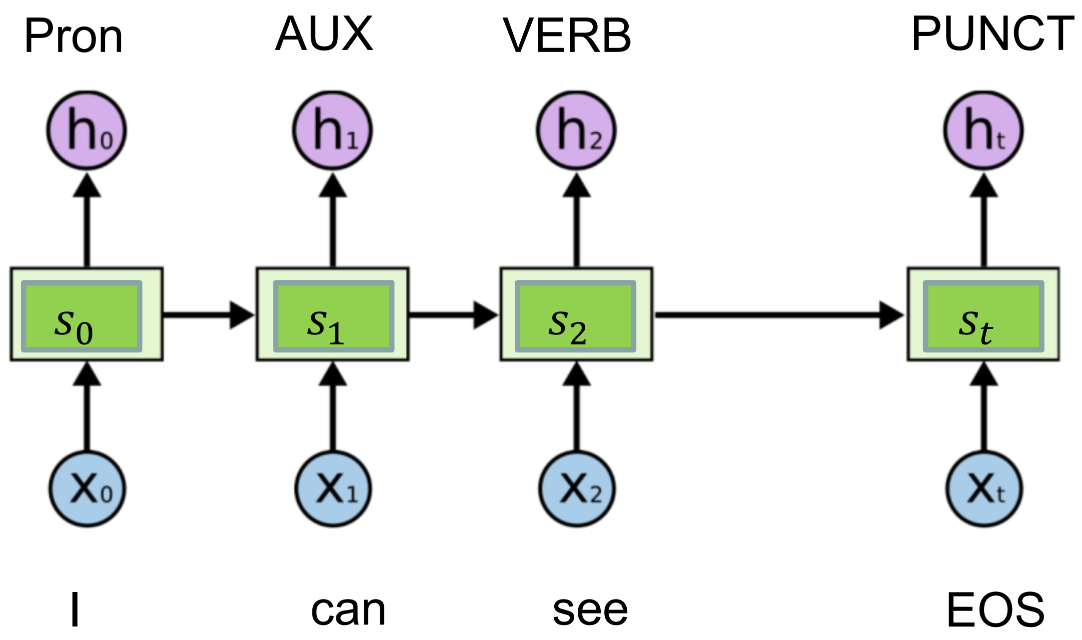
Motivation

We want to do POS tagging

- That is, label each token in a sentence with its part-of-speech (= word class)
- I can see the cat

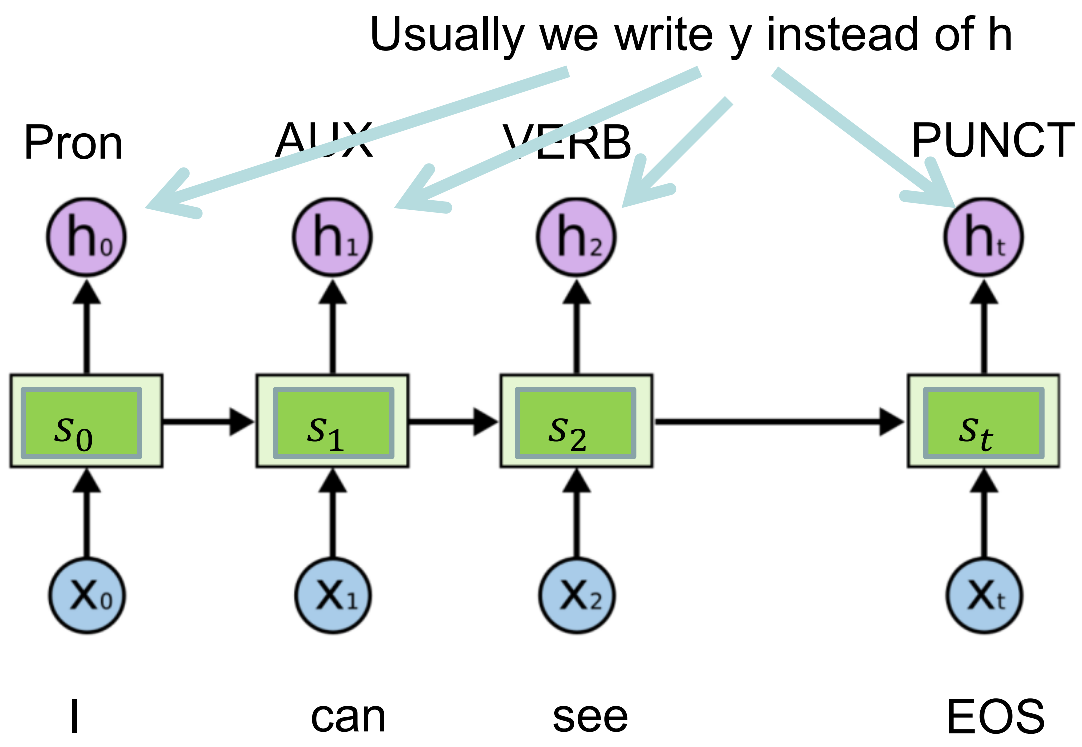
- We want a different architecture
  - with infinite context size
  - that has few parameters
  - makes a prediction at each time step

RNN model

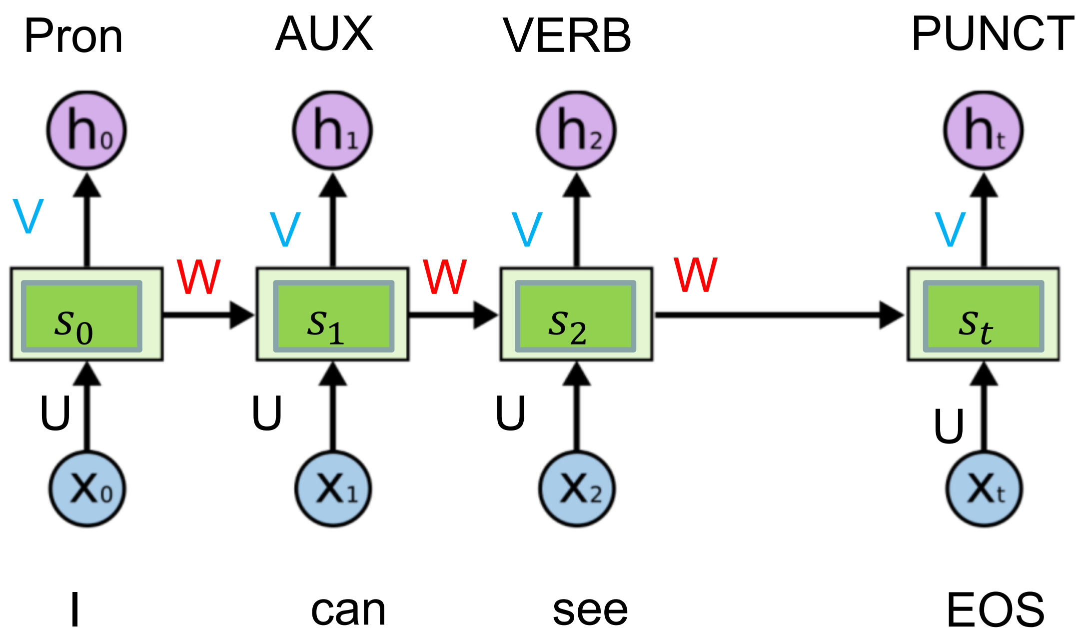


Pron
AUX
VERB
PUNCT
h_0
h_1
h_2
h_t
s_0
s_1
s_2
s_t
x_0
x_1
x_2
x_t
I
can
see
EOS

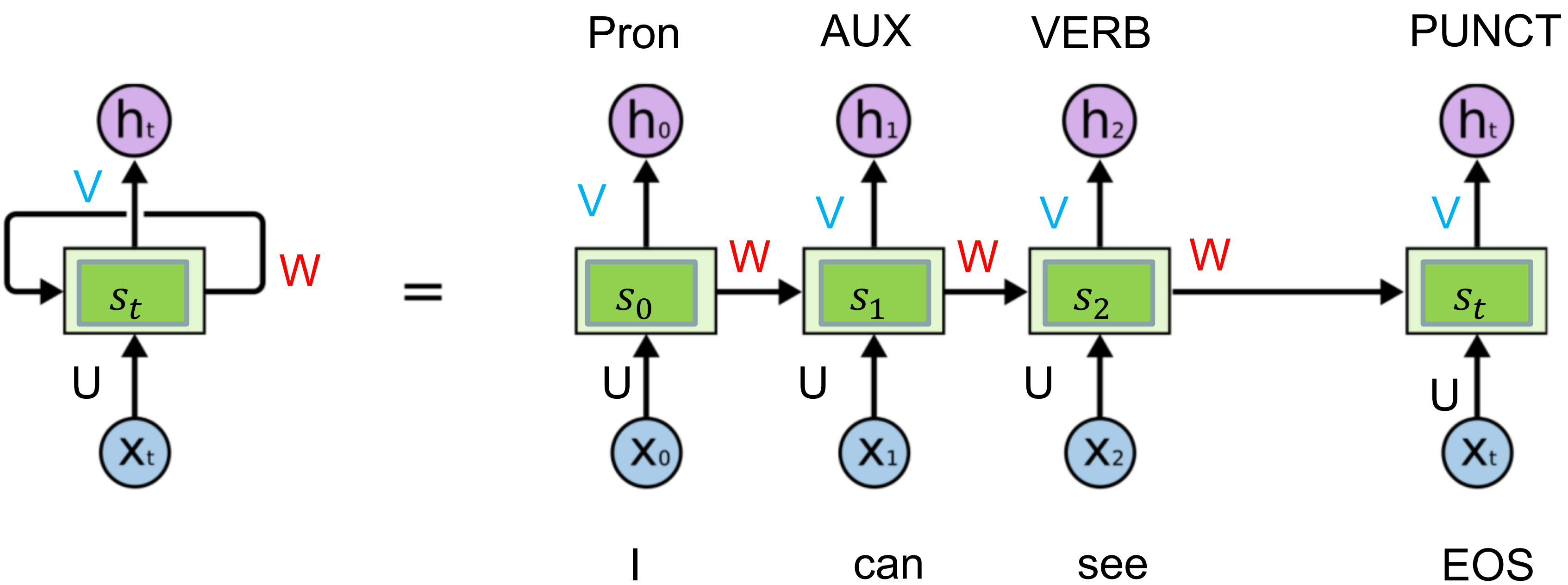
RNN model


Usually we write y instead of h
Pron
AUX
VERB
PUNCT
h_0
h_1
h_2
h_t
s_0
s_1
s_2
s_t
x_0
x_1
x_2
x_t
I
can
see
EOS

RNN model


Pron
AUX
VERB
PUNCT
h0
h1
h2
ht
V
V
V
V
W
W
W
s_0
s_1
s_2
s_t
U
U
U
U
X0
X1
X2
Xt
I
can
see
EOS

RNN model

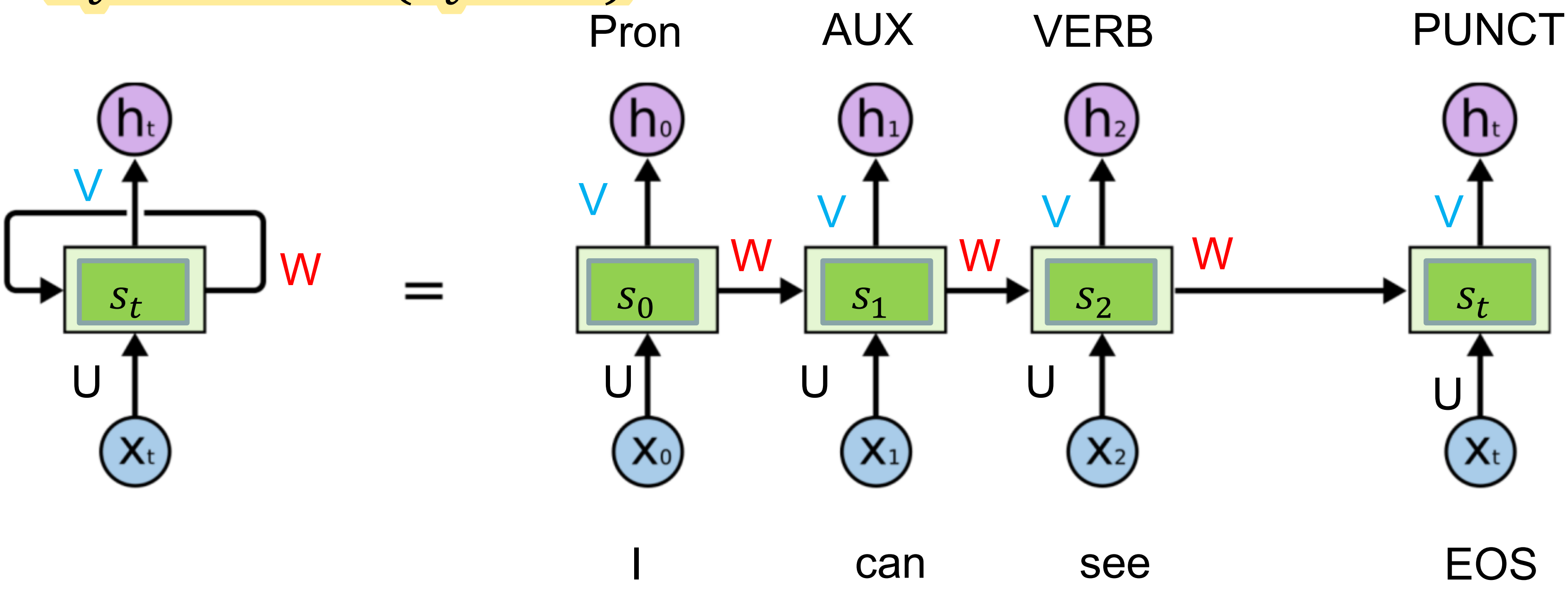


Pron
AUX
VERB
PUNCT
h_t
h_0
h_1
h_2
h_t
V
W
U
s_t
s_0
s_1
s_2
s_t
=
x_t
x_0
x_1
x_2
x_t
I
can
see
EOS

Math

Math: $\boldsymbol{s}_t = f(\boldsymbol{x}_t \boldsymbol{U} + \boldsymbol{s}_{t-1} \boldsymbol{W} + \boldsymbol{b})$

$\boldsymbol{h}_t = \text{softmax}(\boldsymbol{s}_t \boldsymbol{V} + \boldsymbol{c})$



Benefits

- Infinite influence from the past – in theory
  - If you make this bidirectional, you also have infinite influence from the future
- Fewer parameters, via parameter sharing and “small” matrices U, V, W

RNN – Example

- Input: “A rusty can”
- Embeddings: $\boldsymbol{x}_1 = (1,0,0), \boldsymbol{x}_2 = (1,1,2), \boldsymbol{x}_3 = (1,-1,1)$
- Truth: DET,ADJ,NOUN, encoded as 1-hot vectors (in a 4-d label space)
- Activations: ReLU for hidden layer, Softmax for output layer

RNN – Example

Initialization:

- $\boldsymbol{U} = \begin{pmatrix} 1 & 1 \\ 2 & 0 \\ 0.5 & 1 \end{pmatrix}$

- $\boldsymbol{W} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

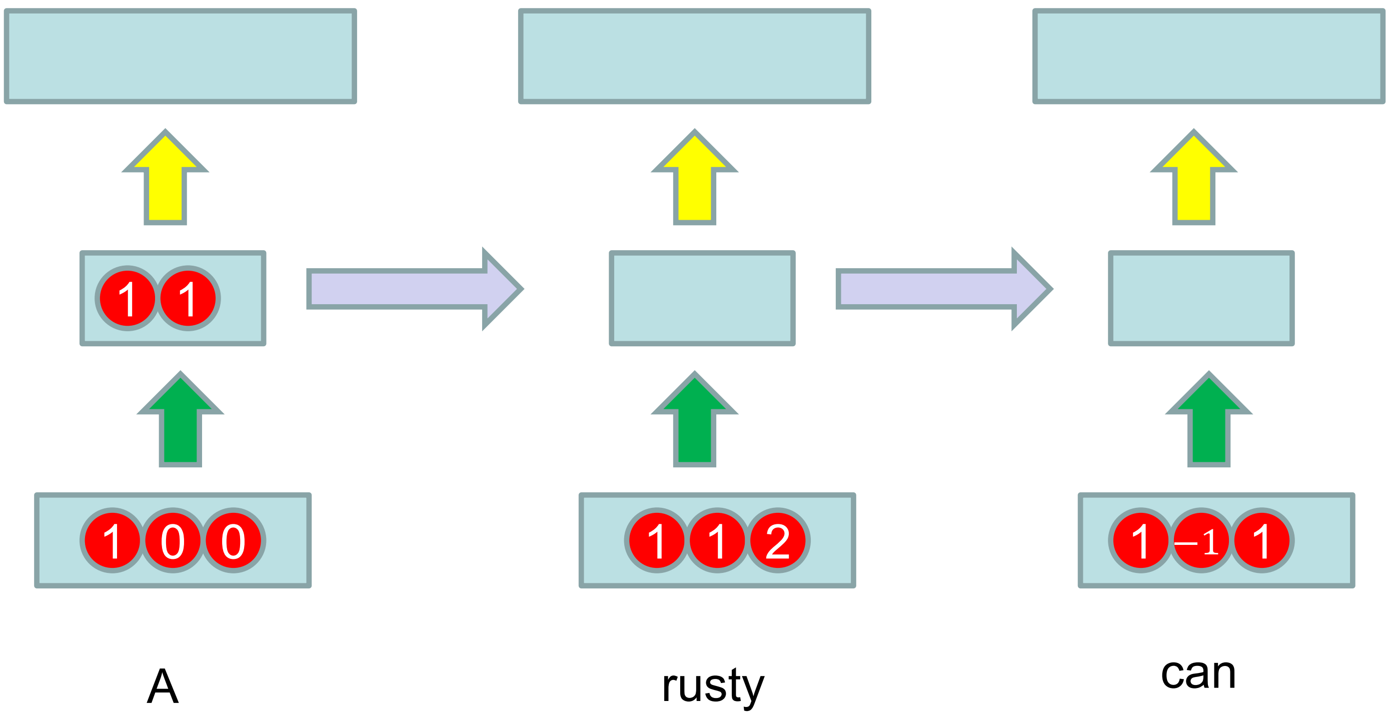
- $\boldsymbol{V} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & \frac{1}{3} & -1 \end{pmatrix}$

- $\boldsymbol{b} = \boldsymbol{c} =$ zero-vectors of appropriate size
- $\boldsymbol{h}_0 = (0,0)$

RNN – Example

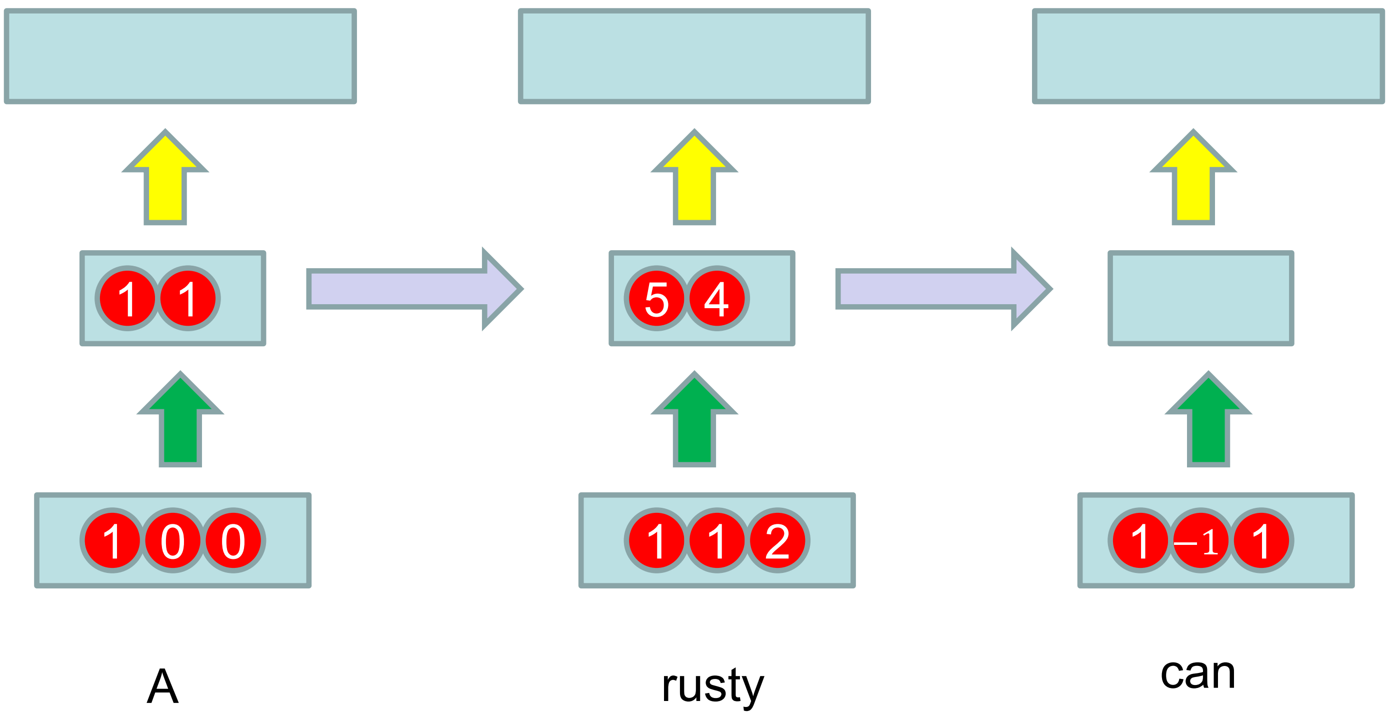
$$h_1 = \sigma_H(x_1 U + h_0 W + b)$$

$$h_1 = (1,1)$$



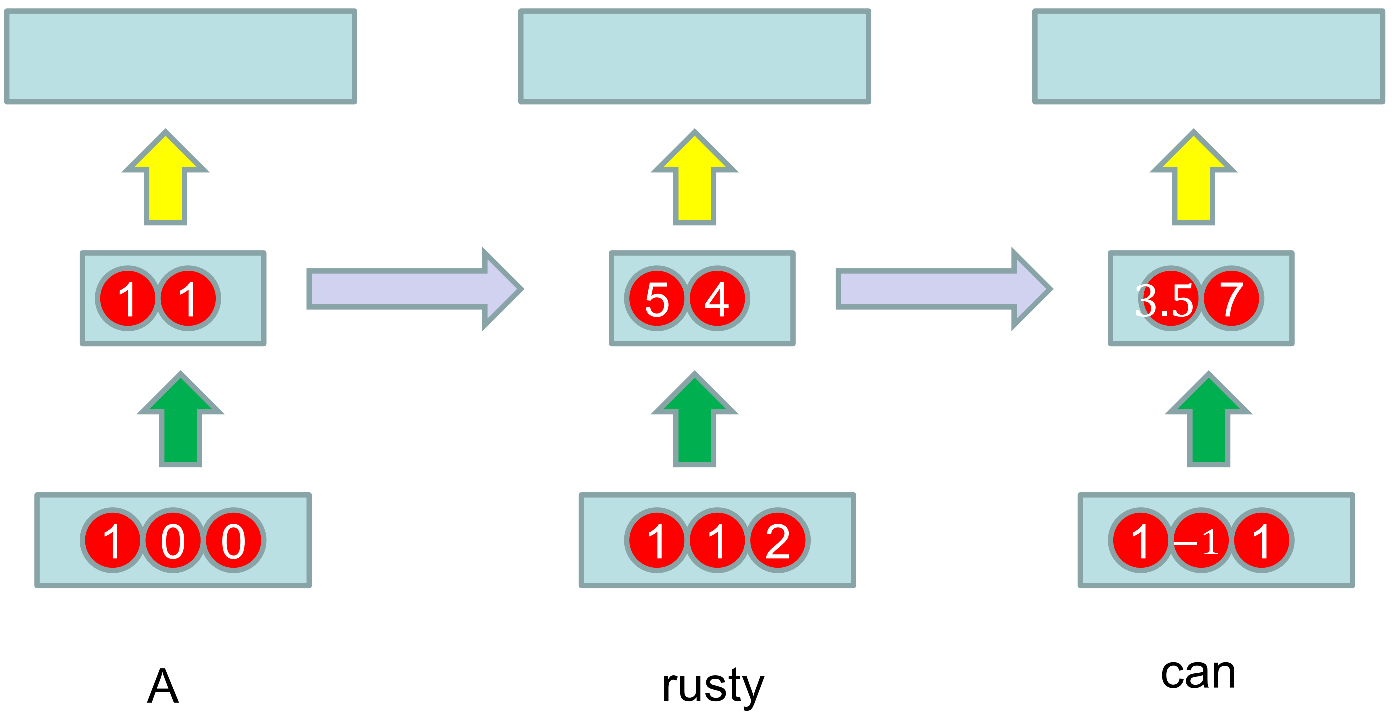
RNN – Example

$$\boldsymbol{h}_2 = \sigma_H(\boldsymbol{x}_2\boldsymbol{U} + \boldsymbol{h}_1\boldsymbol{W} + \boldsymbol{b})$$
$$\boldsymbol{h}_2 = (5,4)$$



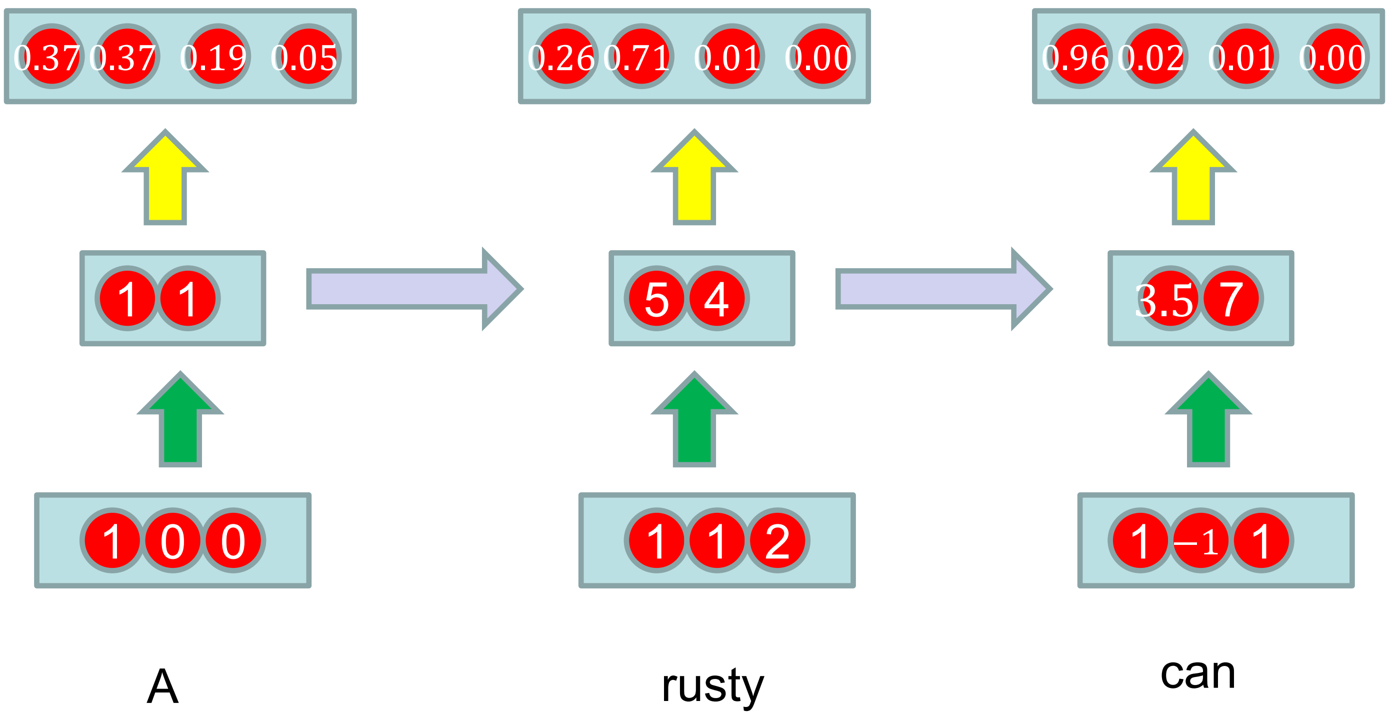
RNN – Example

$$\boldsymbol{h}_3 = \sigma_H(\boldsymbol{x}_3\boldsymbol{U} + \boldsymbol{h}_2\boldsymbol{W} + \boldsymbol{b})$$
$$\boldsymbol{h}_3 = (3.5,7)$$

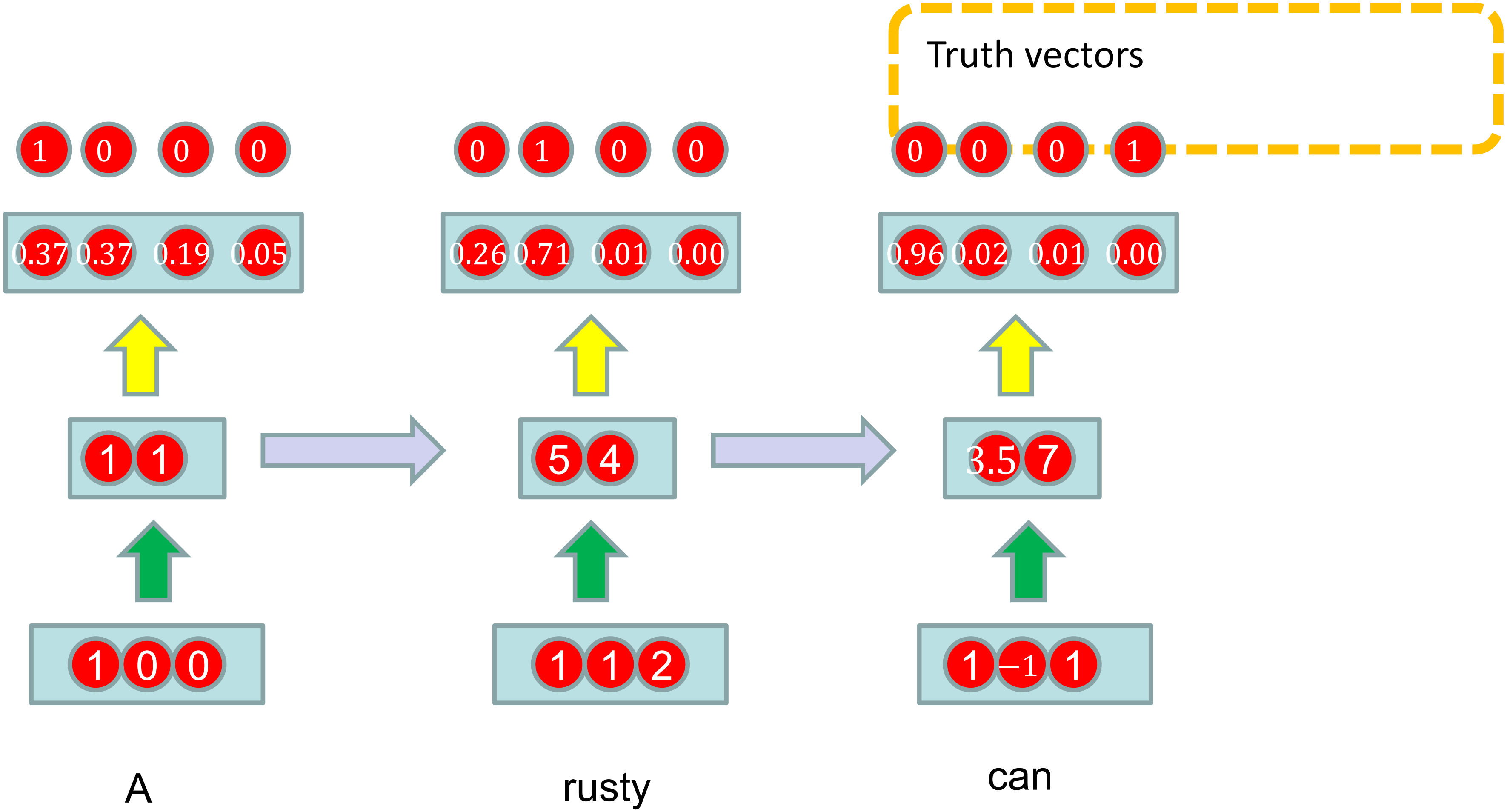


RNN – Example

$$y_t = \sigma_Y(h_t V + c)$$



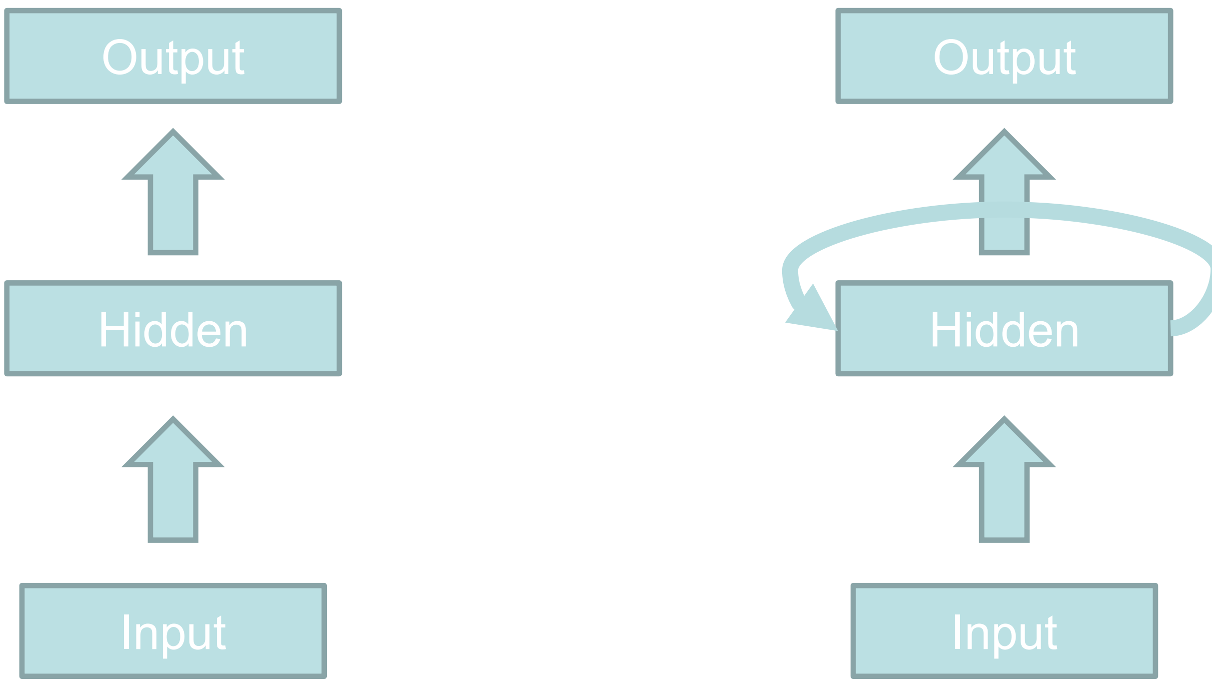
RNN – Example


Truth vectors
1 0 0 0
0 1 0 0
0 0 0 1
0.37 0.37 0.19 0.05
0.26 0.71 0.01 0.00
0.96 0.02 0.01 0.00
1 1
5 4
3.5 7
1 0 0
1 1 2
1 −1 1
A
rusty
can

Excursion 2: Encoder-Decoder Models

RNNs

A recurrent neural net (RNN) is a MLP with additional feedback loop

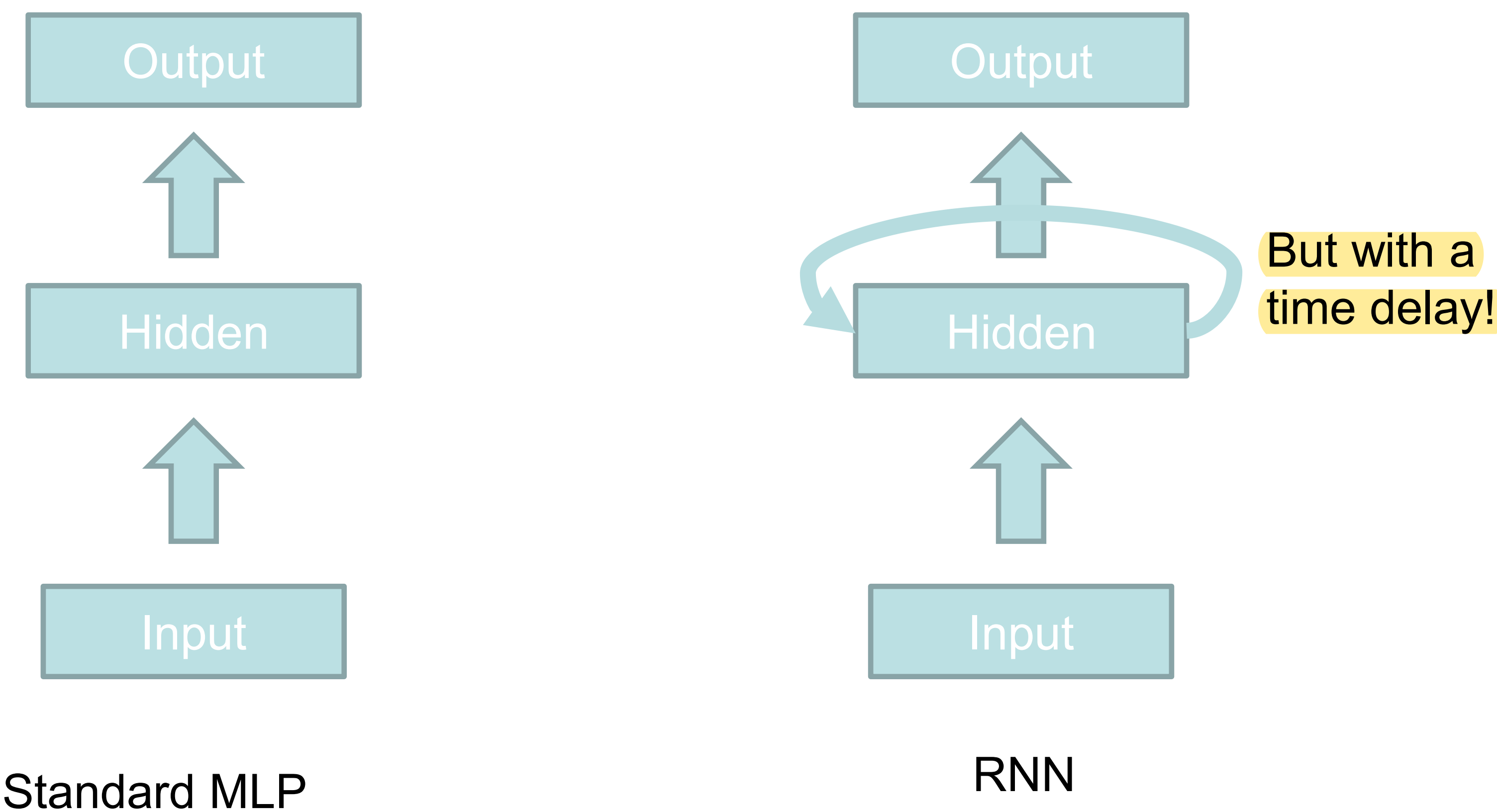


Standard MLP

RNN

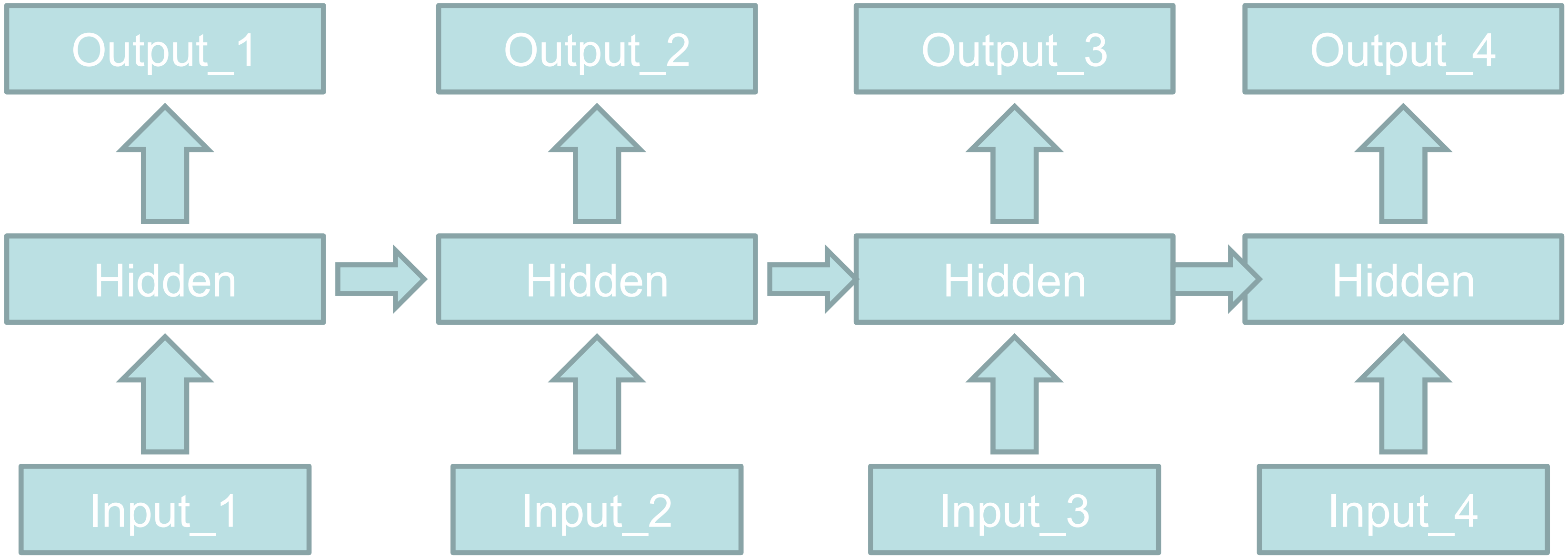
RNNs

A recurrent neural net (RNN) is a MLP with additional feedback loop



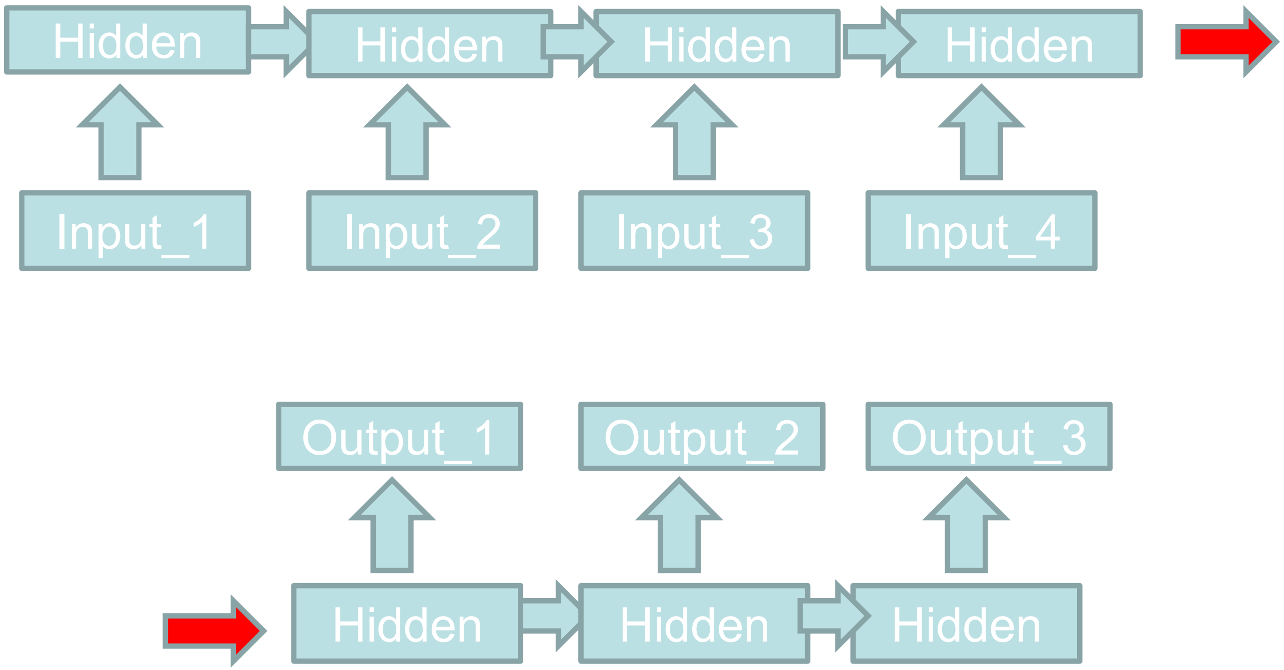
RNNs

- The feedback loop can capture information from “previous time steps”
- Thus, RNNs are really sequential models – they model *variable-length* sequences
- An RNN “unrolled in time”:



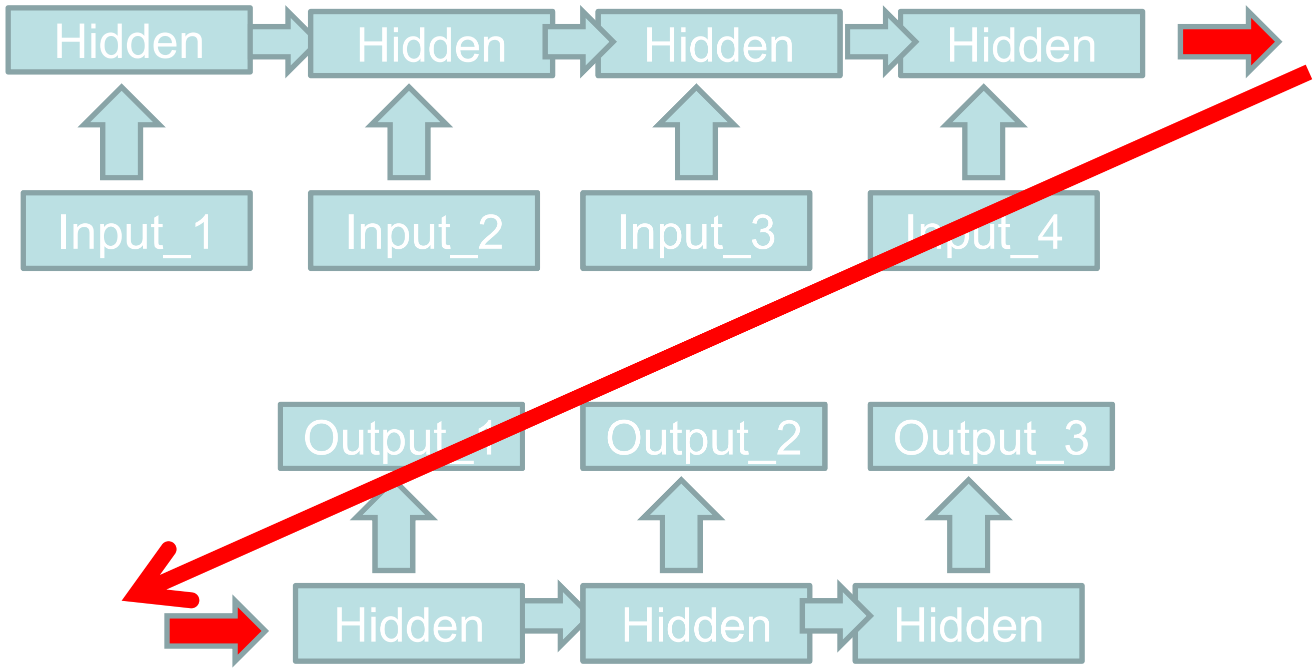
From RNNs to Encoder-Decoder Models

- Encoder-Decoder Models: We stack two RNNs together
- And make some further design changes



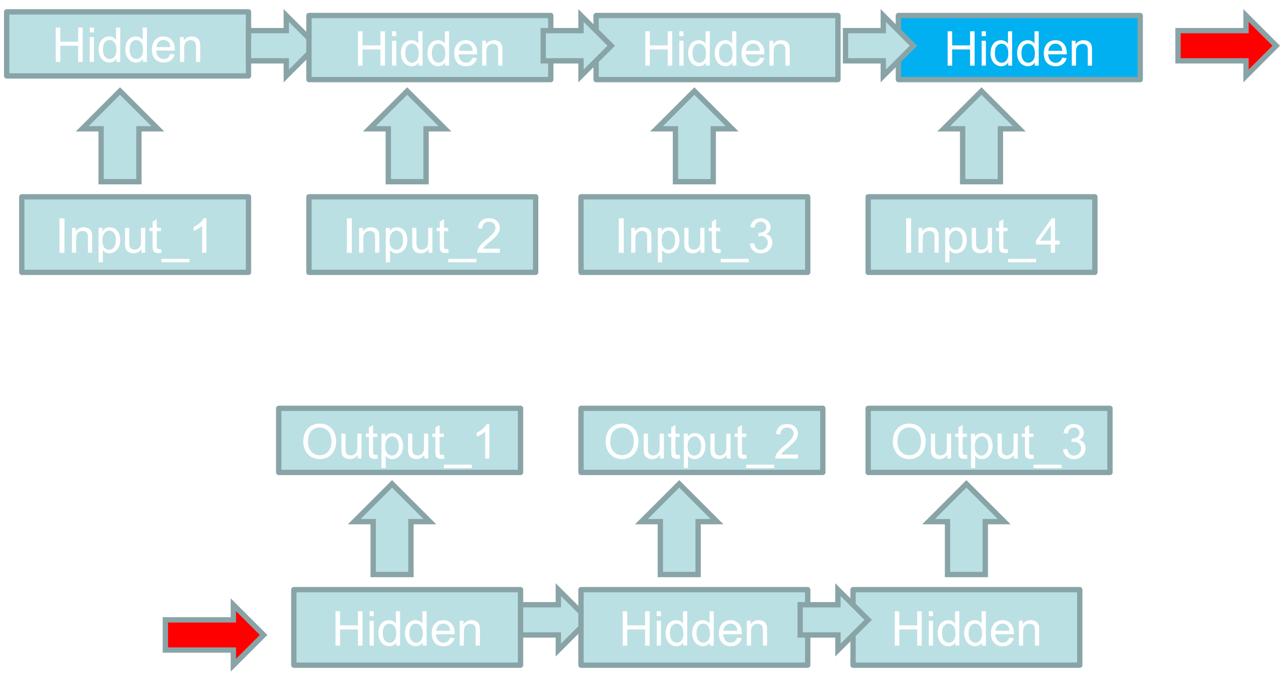
Encoder-Decoder models

- Encoder-Decoder Models: We stack two RNNs together
- And make some further design changes



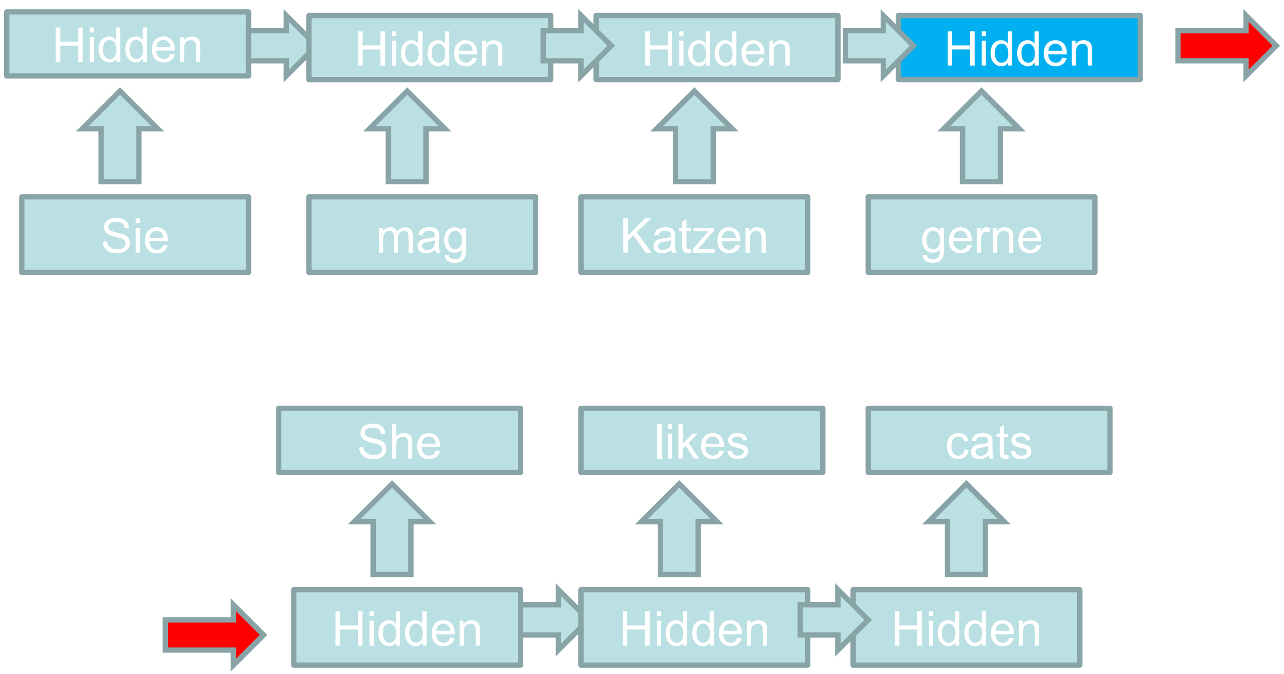
Encoder-Decoder models

- Encoder-Decoder Models: We stack two RNNs together
- The last hidden layer in the input is taken as representation of the input



Application of Encoder-Decoder models

- Encoder-Decoder Models are typically emloyed in Machine Translation
- E.g. Translate a German sentence into an English sentence



Back to SE: Complex/costly methods

Idea

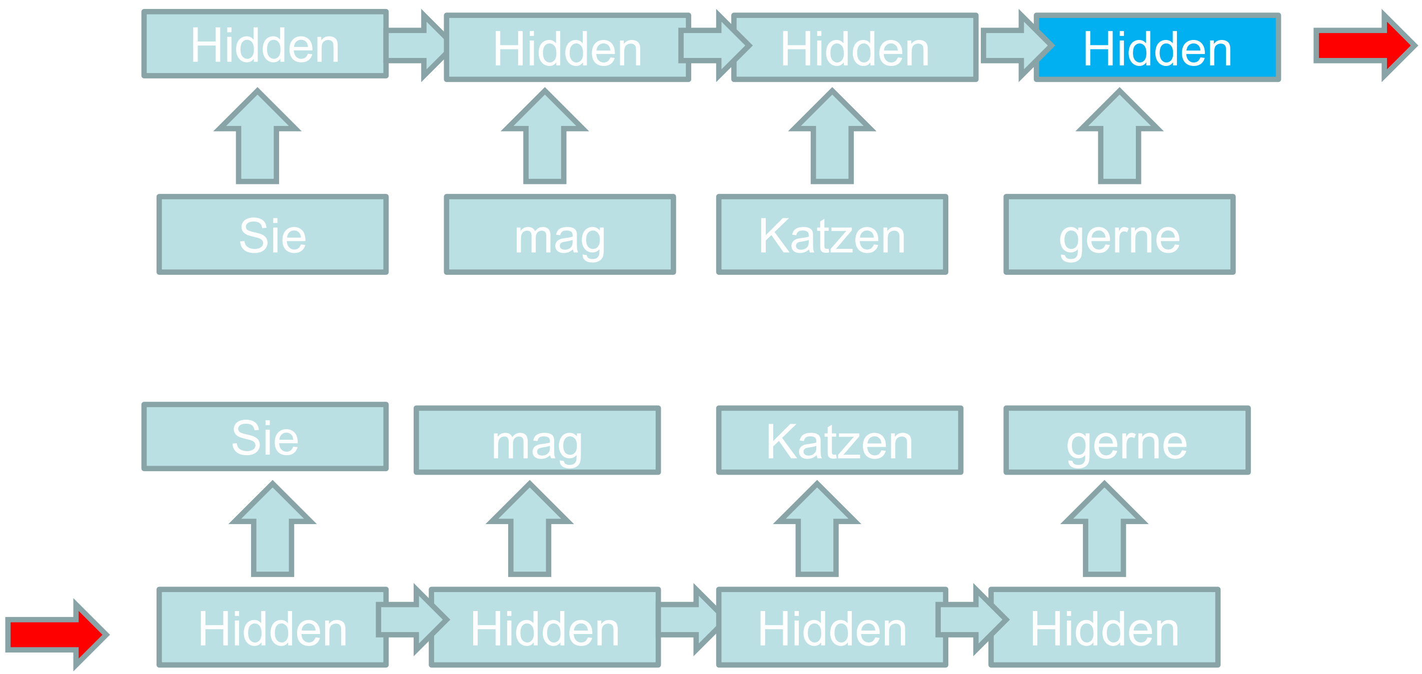
- For sentence embeddings, we can exactly take such encoder-decoder models

- E.g. take an encoder-decoder model, let the input sequence equal the output sequence, and take the final hidden vector on the input side to be the **sentence representation**

  - Such an approach is sometimes called an *auto-encoder*

Sequential Denoising Autoencoders

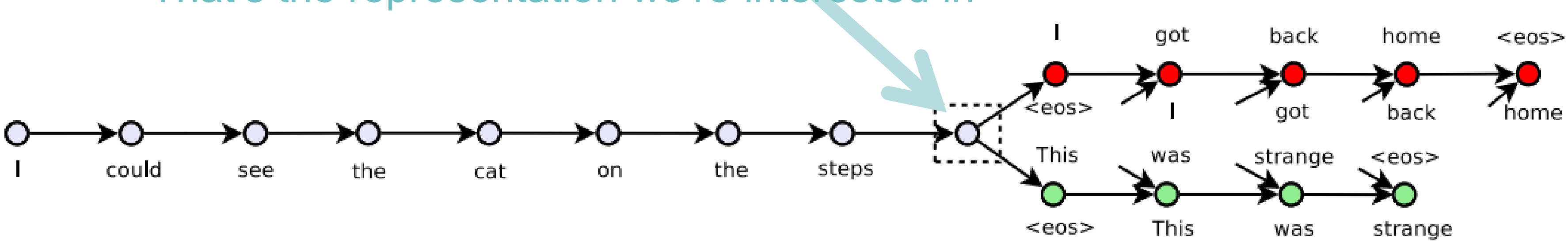
- This is the idea of Hill et al. (2015): **SDAE**
- They in addition do something called *denoising* – they corrupt the input a little



Skip-thought vectors

- Another possibility is to predict the *context sentences*, similarly as in Skip-Gram
- This is the idea of Kiros et al. (2015): **Skip-Thought Vectors**

That's the representation we're interested in



- One can of course easily extend these ideas
  - E.g. predict the current, previous and next sentence, etc.
- What is the difference to our naïve idea number 1?

Comparison: Skip-thoughts vs. SDAE

- Skip-thoughts requires text in context – e.g. a novel where preceding and following sentences are coherent

- SDAE only requires individual sentences without context
  - Could be applied easier to, e.g., Twitter etc.
  - Can make use of more data

InferSent

- It is **supervised** rather than unsupervised as the two methods before

- It trains on high-quality data (Stanford Natural Language Inference Data - SNLI)

- Paper: Supervised Learning of Universal Sentence Representations from Natural Language Inference Data

InferSent – SNLI Training data

SNLI Corpus

- Stanford Natural Language Inference corpus

  Premise: Girl in a red coat, blue head wrap and jeans is making a snow angel.

  Hypothesis: A girl outside plays in the snow.

  Label: entailment

- 570k premise/hypothesis/label triplets
- Labels: "entailment", "contradiction", "neutral"
- http://nlp.stanford.edu/projects/snli/

64

InferSent – SNLI Training data

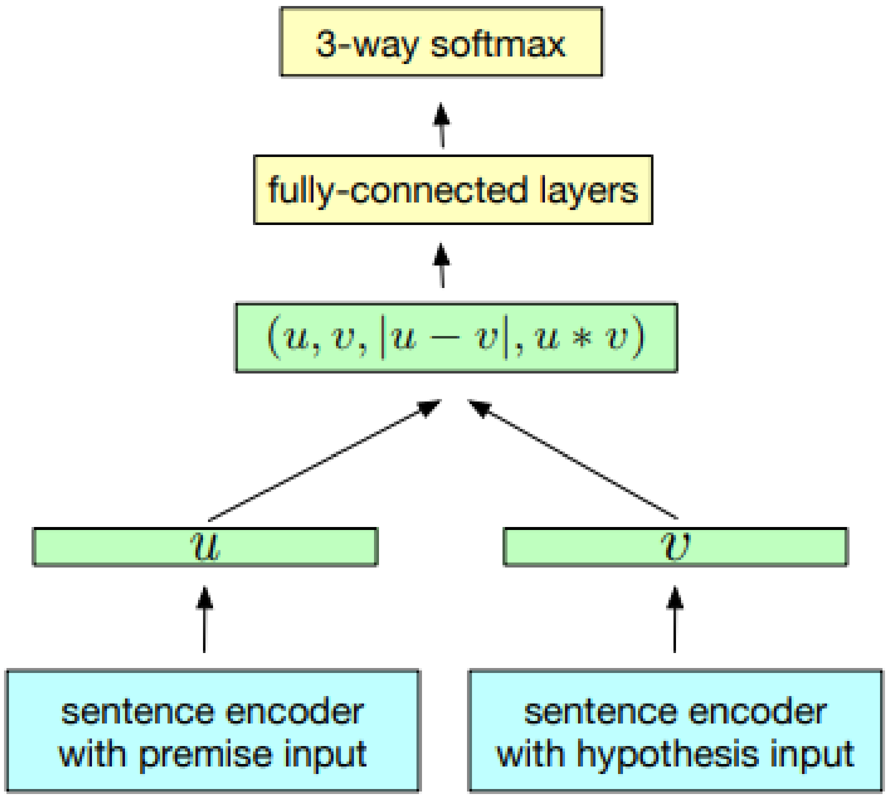


Figure 1: **Generic NLI training scheme.**

InferSent – General Outline

Sentence Embedding

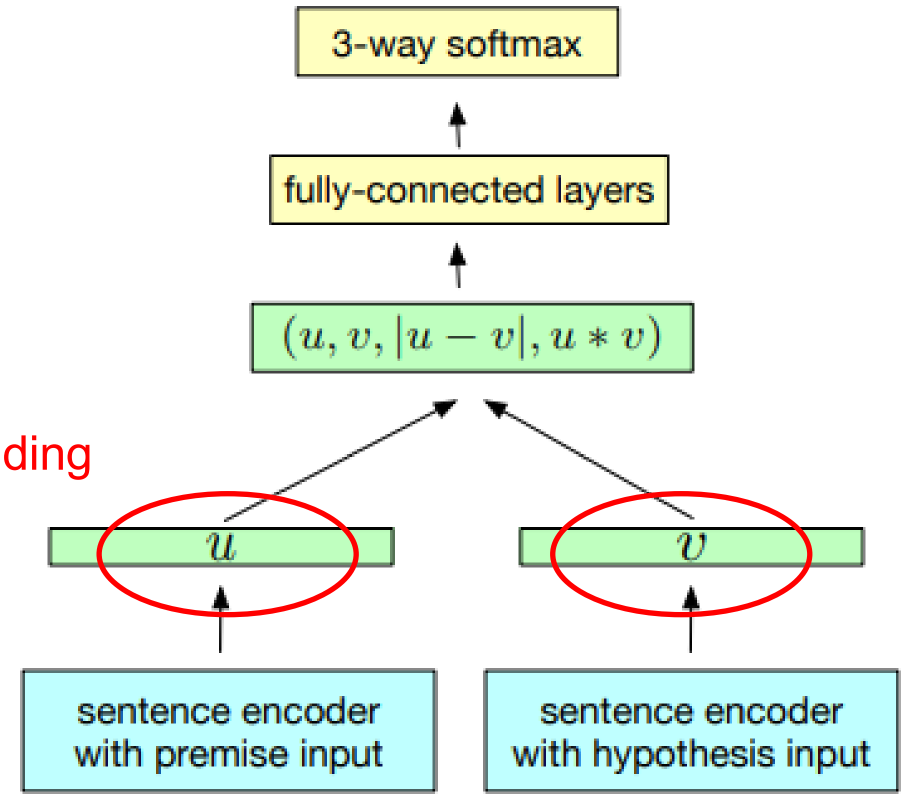


Figure 1: **Generic NLI training scheme.**

InferSent – Computing the sentence embedding

- They use an LSTM, an RNN variant (see Lecture 8)
- Their LSTM is bidirectional

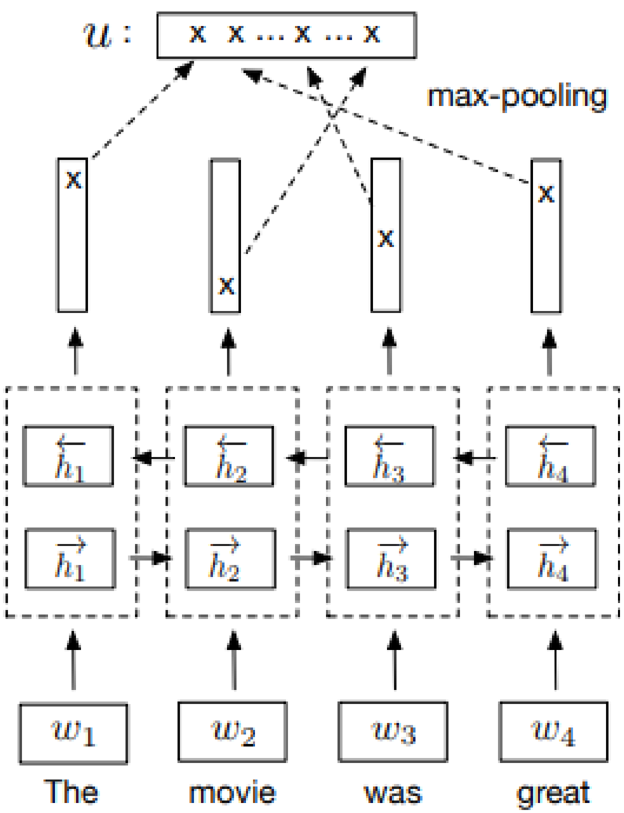


Figure 2: **Bi-LSTM max-pooling network.**

Back to SE: Simple/cheap methods

Why simple?

- The previous models were costly, because **at test time** one would have to run a new sentence through an RNN to embed
  - There are many matrix-vector multiplications involved
  - May be slow and memory intensive

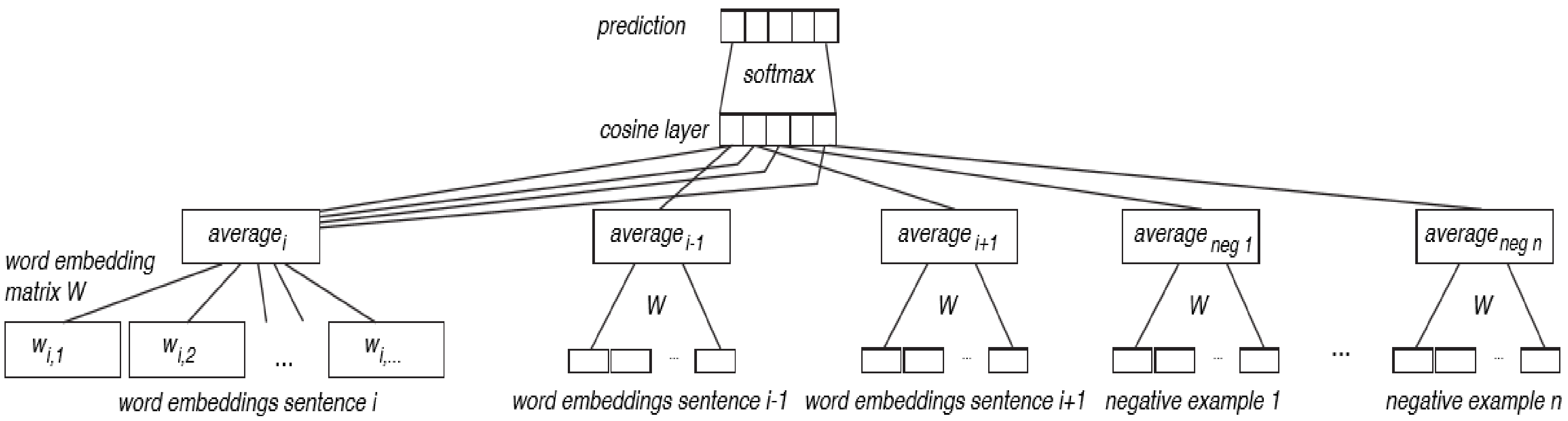
- Now we discuss simpler techniques, especially at test time

Siamese CBOW: Idea

- Proposed by Kenter et al. (2016)
- The goal of this model is to assign each word a word embedding such that the averaged word embeddings of “similar” sentences are close
- They also include a negative sampling strategy

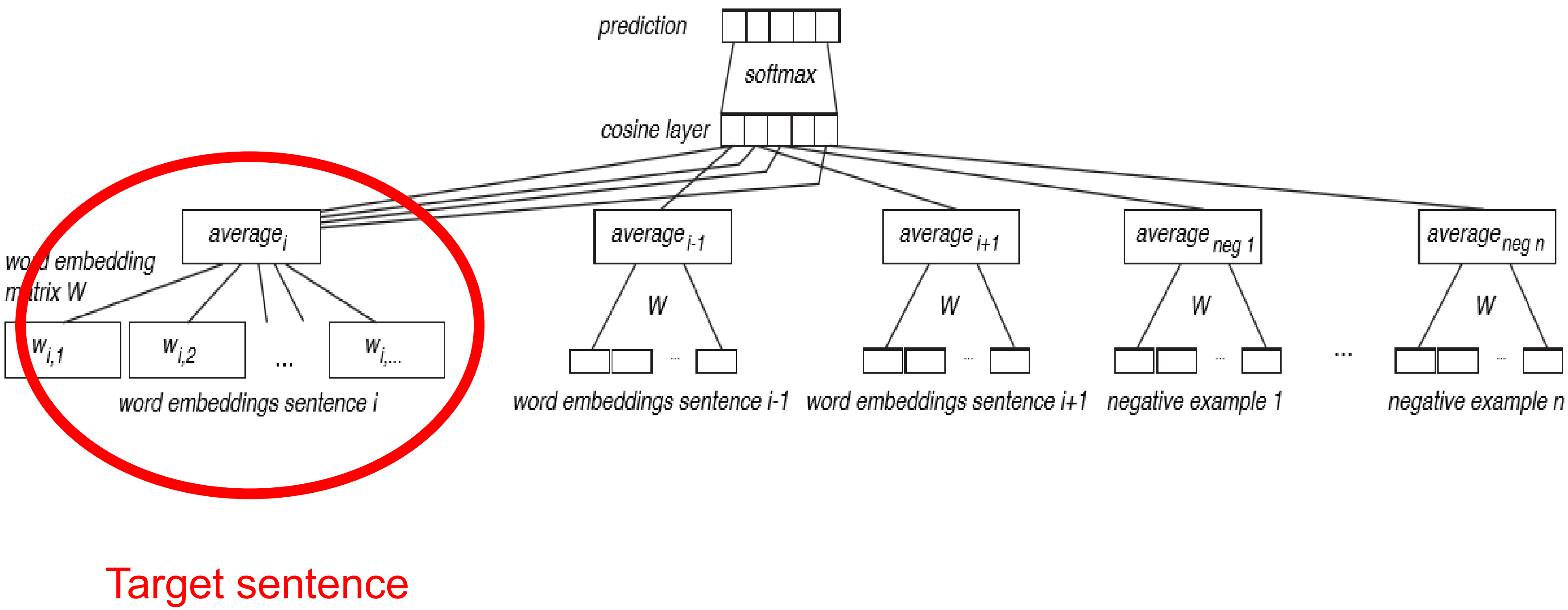
Siamese CBOW: Model

- Their full model visualized:



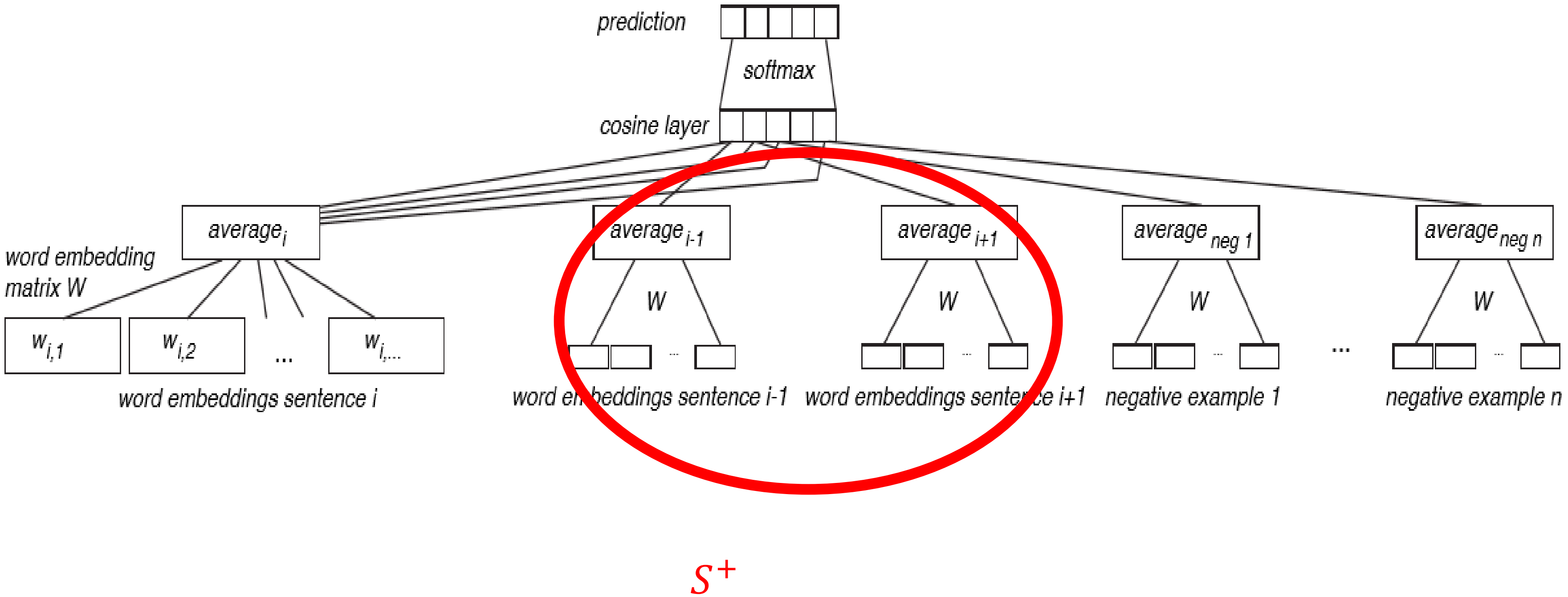
Siamese CBOW: Model

- Their full model visualized:



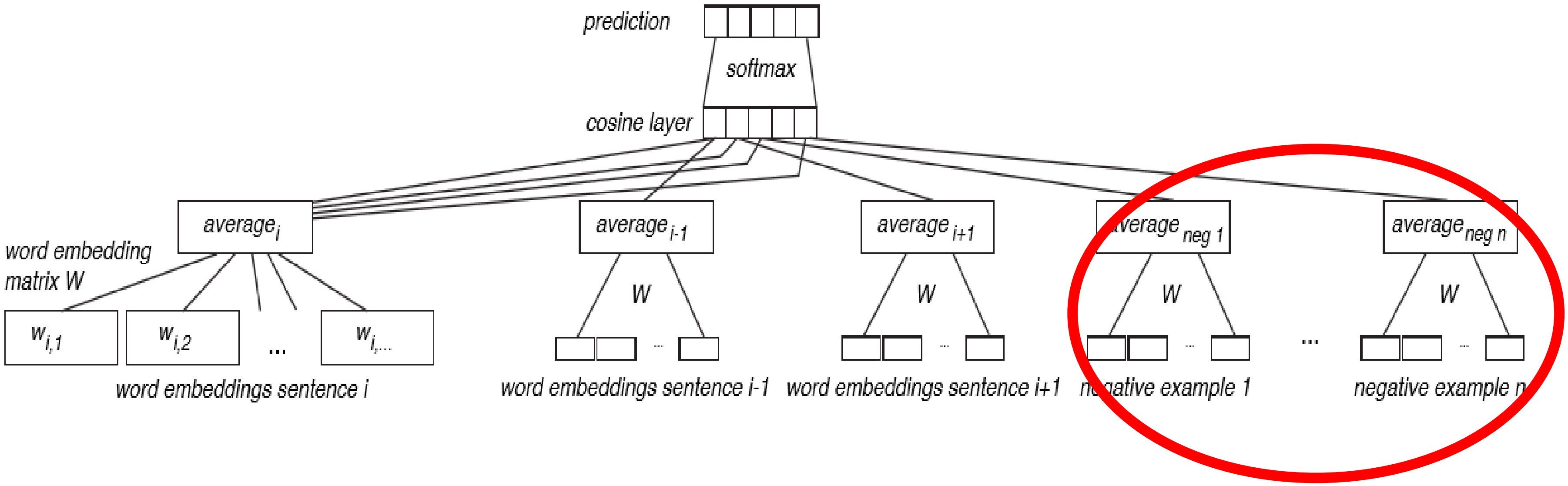
Siamese CBOW: Model

- Their full model visualized:



Siamese CBOW: Model

- Their full model visualized:



S^{-}

Siamese CBOW: Introspection

Some qualitative insights:

- "As Siamese CBOW directly averages word embeddings for sentences, we expect it to learn that words with little semantic impact have a low vector norm"
- "Indeed, we find that the 10 words with lowest vector norm are *to, of, and, the, a, in, that, with, on, as*"

Concatenated Power Mean Embeddings
(https://github.com/UKPLab/arxiv2018-xling-sentence-embeddings)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Proposed by Rückle et al. (2018)
- 1st Idea is to generalize the average to the so-called *power mean*
  - Power mean of numbers $x_1, \dots, x_n$
  - $M_p(x_1, \dots, x_n) = \left(\frac{1}{n}\sum_i x_i^p\right)^{1/p}$
    - $p = -\infty$: $M_p = \min(x_1, \dots, x_n)$
    - $p = +\infty$: $M_p = \max(x_1, \dots, x_n)$
    - $p = 1$: ?
    - $p = 2$: quadratic mean
    - $p = 0$: geometric mean
    - .....

Concatenated Power Mean Embeddings

- 1st Idea is to generalize the average to the so-called *power mean*
  - Now instead of taking a per-dimension standard average
  - One takes a per-dimension power mean average
    - Concatenate different power mean representations
  - Why?
- 2nd Idea is to concatenate diverse averaged word embeddings
  - Such as Glove,Word2Vec,….
  - Why?

Example

	This	is	awesome
e.g. Glove	$\begin{bmatrix}1\\2\\3\end{bmatrix}$	$\begin{bmatrix}1\\2\\2\end{bmatrix}$	$\begin{bmatrix}-2\\-1\\1\end{bmatrix}$
Attract Repel	$\begin{bmatrix}1\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\1\\1\end{bmatrix}$	$\begin{bmatrix}2\\-1\\-4\end{bmatrix}$

$$\xrightarrow{\text{a. mean}} \begin{bmatrix}0\\1\\2\\1\\0\\-1\end{bmatrix} \qquad \xrightarrow{\text{max}} \begin{bmatrix}1\\2\\3\\2\\1\\1\end{bmatrix}$$

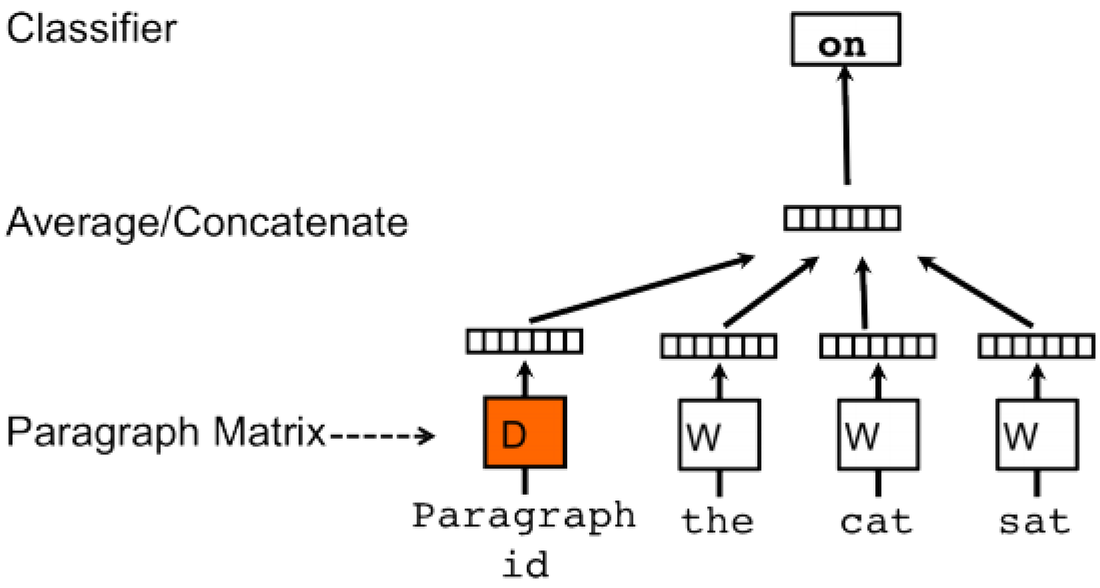
Sentence Embedding

Paragraph Vectors

TECHNISCHE UNIVERSITÄT DARMSTADT

- Described in Le and Mikolov (2014)
- The idea is to assign to a paragraph (one sentence or several) a vector such that we can predict *words* in a text
- This model learns word vectors and paragraph vectors at the same time
- Very similar to CBOW and Skip-Gram model, but with an id for each sentence/paragraph

Paragraph Vectors



Paragraph Vectors

TECHNISCHE UNIVERSITÄT DARMSTADT

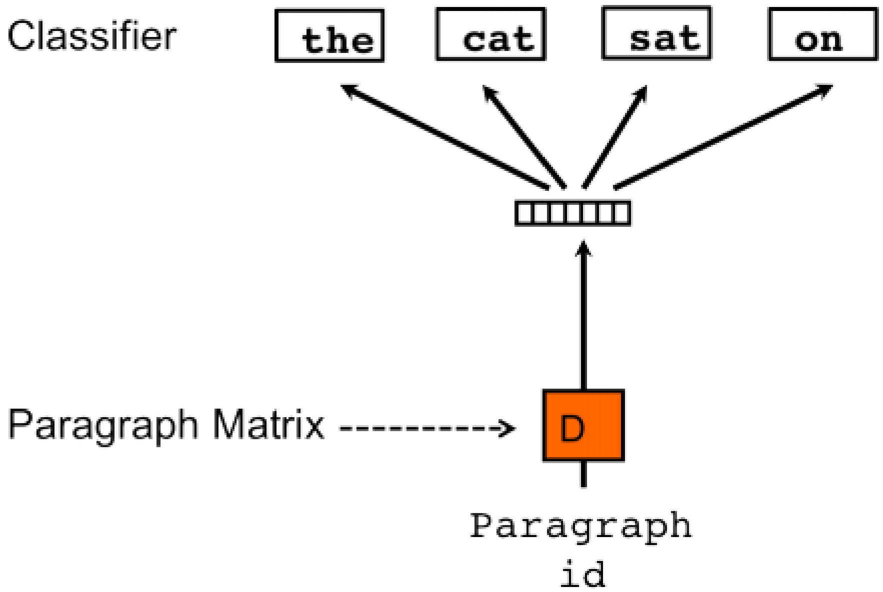


Figure 3. Distributed Bag of Words version of paragraph vectors. In this version, the paragraph vector is trained to predict the words in a small window.

NLLG

Paragraph Vectors

- "The paragraph vector is shared across all contexts generated from the same paragraph but not across paragraphs. The word vector matrix […], however, is shared across paragraphs"

This lecture

1. Embeddings of sentences (or even documents)

2. **(Problems with) Evaluation of Sentence Embeddings**

Evaluation of Sentence Embeddings

- As for word embeddings
  - Extrinsic
    - Feed in to some task
    - Usually apply simple classifier on top of embeddings
      - E.g. logistic regression
  - Intrinsic
    - Direct introspection of embeddings

Extrinsic evaluation - Scheme

A) Take your sentence embedding model

B) Embed sentences in an extrinsic task

C) Train classifier on embedded sentences

D) Repeat with different sentence embedding model and compare performances

Extrinsic tasks

name	N	task	C	examples
MR	11k	sentiment (movies)	2	"Too slow for a younger crowd , too shallow for an older one." (neg)
CR	4k	product reviews	2	"We tried it out christmas night and it worked great ." (pos)
SUBJ	10k	subjectivity/objectivity	2	"A movie that doesn't aim too high , but doesn't need to." (subj)
MPQA	11k	opinion polarity	2	"don't want"; "would like to tell"; (neg, pos)
TREC	6k	question-type	6	"What are the twin cities ?" (LOC:city)
SST	70k	sentiment (movies)	2	"Audrey Tautou has a knack for picking roles that magnify her [..]" (pos)

Table 1: **Classification tasks**. C is the number of class and N is the number of samples.

Intrinsic evaluation - Scheme

A) Take your sentence embedding model

B) Embed sentence pairs in an intrinsic task

C) Use cosine to measure distance between pairs

D) Correlate with human judgments

Intrinsic tasks

name	task	N	premise	hypothesis	label
SICK-R	STS	10k	"A man is singing a song and playing the guitar"	"A man is opening a package that contains headphones"	1.6
STS14	STS	4.5k	"Liquid ammonia leak kills 15 in Shanghai"	"Liquid ammonia leak kills at least 15 in Shanghai"	4.6

Table 2: **Natural Language Inference and Semantic Textual Similarity tasks**. NLI labels are contradiction, neutral and entailment. STS labels are scores between 0 and 5.

Problems with Evaluation of Sentence (and Word!) Embeddings

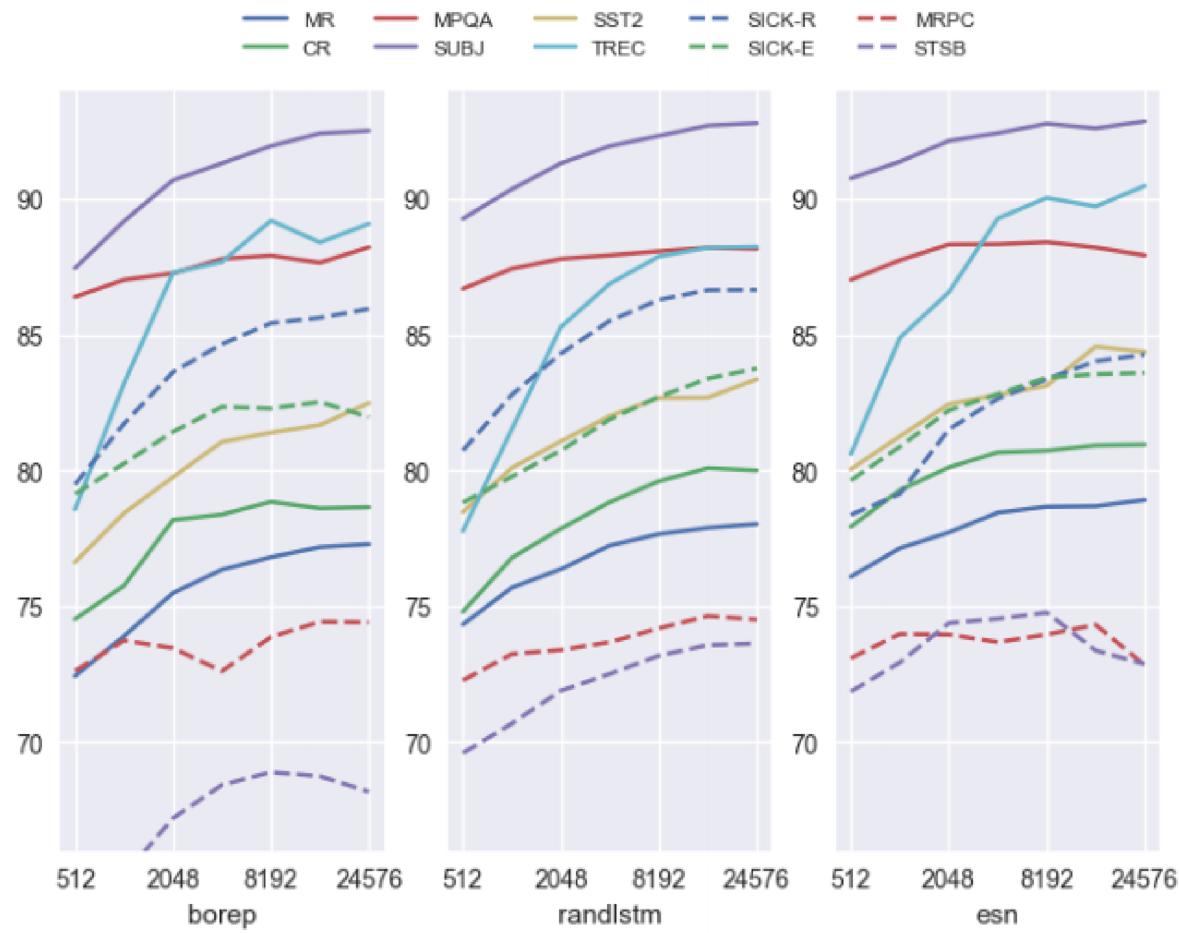
(1) Researchers come up with models of vastly different sizes
- 300d, 600d, 700d, 3600d, 4096d, 4800d
- Comparison is unfair

(2) Different models trained on different datasets (Wikipedia, common crawl, Toronto Book corpus, …)

(3) Which classifier to use on top of embeddings in extrinsic tasks?

Sizes

Wieting and Kiela (2019), ICLR



Sizes

Eger et al. (2019), Problems with Eval of Sentence Emb., Repl4NLP

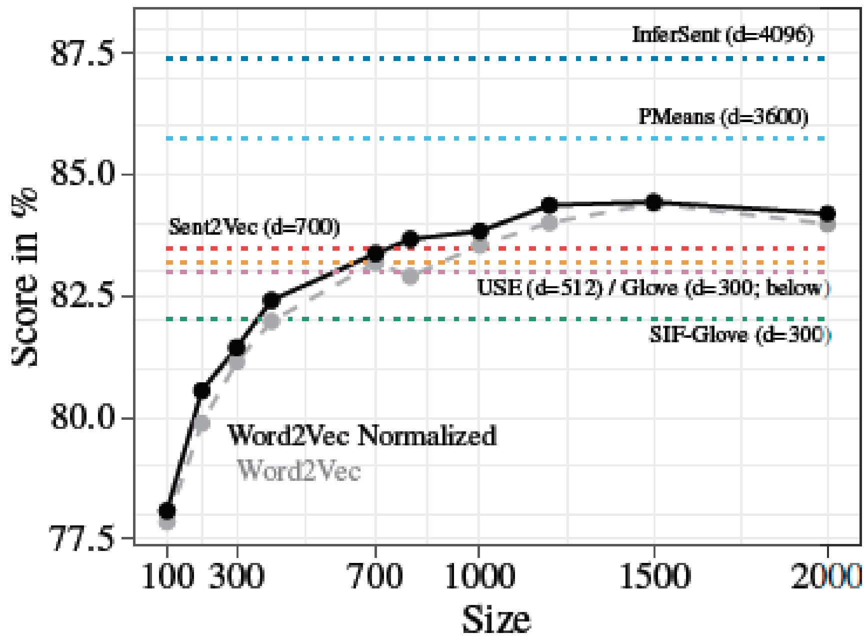


Figure 1: Avg. score across 6 transfer tasks for different sizes of Word2Vec embeddings vs. scores of other encoders (with constant embedding sizes as given in Table 1). 'Word2Vec Normalized' is discussed below.

Introspection of Sentence Embeddings

Linguistic Probing Tasks

TECHNISCHE UNIVERSITÄT DARMSTADT

- What linguistic information is captured in embeddings?
  - Sentence length
  - Word order
  - Whether a certain word is in the sentence
  - Agreement between subject and verb (*she likes cats* vs. *she like cats*)

- Extrinsic and intrinsic evaluation give limited insights
  - Because they are complex tasks and may require several knowledge nuggets

- Probing tasks introspect embeddings → help to **interpret** them

Linguistic Probing Tasks

Table 4: Linguistic probing tasks description and samples.

Task	Description	Example	Output
Bigram Shift (BShift)	Whether two words (tokens) in a sentence have been inverted	This is my Eve Christmas .	Inverted
Coordination Inversion (CoordInv)	Sentences comprised of two coordinate clauses. Detect whether clauses are inverted	I returned to my work , and Lisa headed for her office .	Inverted
Object Number (ObjNum)	Number of the direct object in the main clause (singular and plural)	He received the 200 points .	NNS (Plural)
Sentence Length (SentLen)	Predict the sentence length among 6 classes, which are length intervals	I can 't wait to show you and Mr. Taylor .	9 – 12 words
Semantic Odd Man Out (SOMO)	Random noun or verb replaced in the sentence by another noun or verb. Detect whether the sentence has been modified	Tomas surmised as well .	Changed

Linguistic Probing Tasks

Subject Number (SubjNum)	Number of the subject in the main clause (singular and plural)	If there was ever a time to let loose , this vacation would have to be it .	Singular
Past Present (Tense)	Whether the main verb in the sentence is in the past or present tense	She smiled at him , her eyes alight with love .	Present
Top-Constituent (TopConst)	Classification task, where the classes are given by the 19 most common top-constituent sequences in the corpus	Did he buy anything from Troy ?	VBD_NP_VP_
Depth of Syntactic Tree (TreeDepth)	Predict the maximum depth of the syntactic tree of the sentence	The leaves were in various of stages of life .	10
Word Content (WC)	Predict which of the target words (among 1000) appear in the sentence	She eyed him skeptically .	eyed

Mandatory reading for next week

Wieting and Kiela (2019), No Training Required: Exploring Random Encoders for Sentence Classification

References

- Hill, F., Cho, K. & Korhonen, A.: Learning Distributed Representations of Sentences from Unlabelled Data, in *Proceedings of NAACL-HLT*, 2016
- Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R.S., Torralba, A., Urtasun, R. & Fidler, S.: Skip-Thought Vectors, in *Advances in Neural Information Processing Systems 28*, 2015
- Le, Q. & Mikolov, T.: Distributed representations of sentences and documents, in *Proceedings of the 31st International Conference on Machine Learning*, 2014
- Kenter, T., Borisov, A. & Rijke, M.: Siamese CBOW: Optimizing Word Embeddings for Sentence Representations, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016
- Newman, M.E.J.: Power laws, Pareto distributions and Zipf's law, in Contemporary Physics, 2005
- Xu, Y. & Kemp, C.: A Computational Evaluation of Two Laws of Semantic Change, in *CogSci*, 2015
- Hamilton, W.L., Leskovec, J. & Jurafsky, D.: Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics,* 2016
- Eger, S. & Mehler, A.: On the Linearity of Semantic Change: Investigating Meaning Variation via Dynamic Graph Models, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016
- Bamman, D., Dyer, C. & Smith, N.A.: Distributed Representations of Geographically Situated Language, in Proceedings of d An nual Meeting of the Association for Computational Linguistics, 2015
- Asgari, E. & Mofrad, M.R.K.: Comparing Fifty Natural Languages and Twelve Genetic Languages Using Word Embedding Language Divergence (WELD) as a Quantitative Measure of Language Distance, in *Proceedings of the Workshop on Multilingual and Cross-lingual Methods in NLP*, 2016
- Eger, S., Hoenen, A. & Mehler, A.: Language classification from bilingual word embedding graphs, in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016b
- Hamilton et al.: Cultural Shift or Linguistic Drift? Comparing Two Computational Measures of Semantic Change, EMNLP, 2016b

References

- Pagliardini et al., Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features, 2018
- Arora et al., A SIMPLE BUT TOUGH-TO-BEAT BASELINE FOR SENTENCE EMBEDDINGS, 2017