SELECTED CASE STUDIES IN NLP

Current Trends in Artificial Intelligence

VUB, May 10, 2019

Chris Develder et al.

GHENT UNIVERSITY

IDLab
INTERNET & DATA LAB
imec

Self-Introduction – Chris Develder

- Professor at UGent since Oct. 2007
  - Research Interests:
    - **Natural language processing (NLP)** for information extraction (IE)
    - Data analytics and machine learning for **smart grids**
    - Past: track record in dimensioning and optimizing **optical networks**
  - Visiting researcher at UC Davis, CA, USA, Jul-Oct. 2007 (optical networks)
  - Visiting researcher at Columbia Univ., NY, USA, 2013-15 (IE & information retrieval)
- Industry Experience: Network planning/design tools
  - OPNET Technologies, 2004-05
- PhD on optical packet switching, UGent, 2003

See http://users.ugent.be/~cdvelder/ and https://ugentt2k.github.io/

Self-introduction – T2K team @ IDLab, UGent


Chris Develder

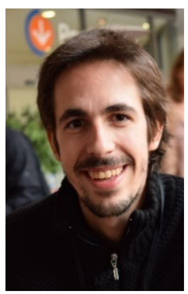

Thomas Demeester

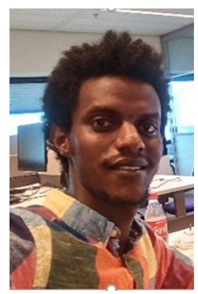

Johannes Deleu


Lucas Sterckx

UGENT
T2K


Klim Zaporojets


Giannis Bekoulis


Semere Kiros Bitew


Amir Hadifar

What is Natural Language Processing?

Human language

Non-trivial useful output

"NLP takes as input text in human language and processes it in a way that suggests an intelligent process was involved"

- Yoav Goldberg, Introduction to NLP

What is Natural Language Processing?

Can focus on content (e.g., semantics, intent), or on form (e.g., sentiment, psychological state …)

Text in human language

"Meaning"
"Insights"

"NLP takes as input text in human language and processes it in a way that suggests an intelligent process was involved"

- Yoav Goldberg, Introduction to NLP

What is Natural Language Processing?

Text in human language

Text in <u>another</u> language

“NLP takes as input text in human language and processes it in a way that suggests an intelligent process was involved”

- Yoav Goldberg, Introduction to NLP

What is Natural Language Processing?

Data in structured form

Text in human language

"NLP takes as input text in human language and processes it in a way that suggests an intelligent process was involved"

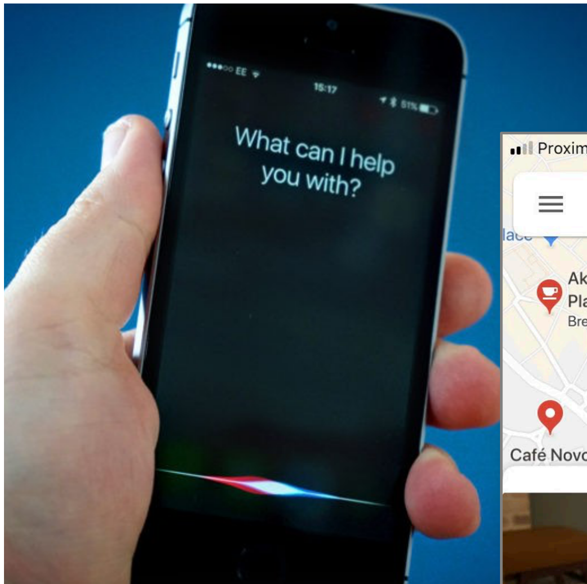
- Yoav Goldberg, Introduction to NLP

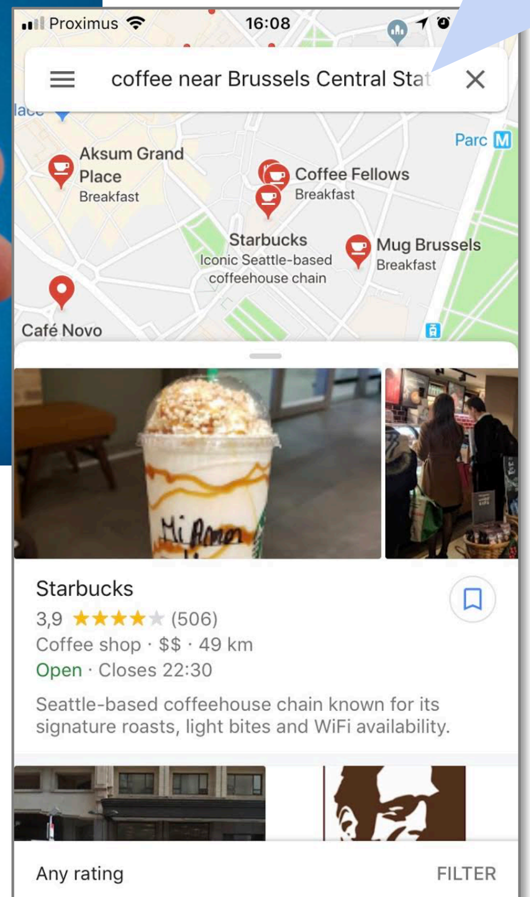
Evolution of NLP techniques

- **1950 – 1990s** – Write many rules
- **1990s – 2000s** – Corpus-based statistics
- **2000s - ~2014** – Supervised machine learning
- **2014 – today** – "Deep learning"

GHENT UNIVERSITY

NLP today … speech interfaces



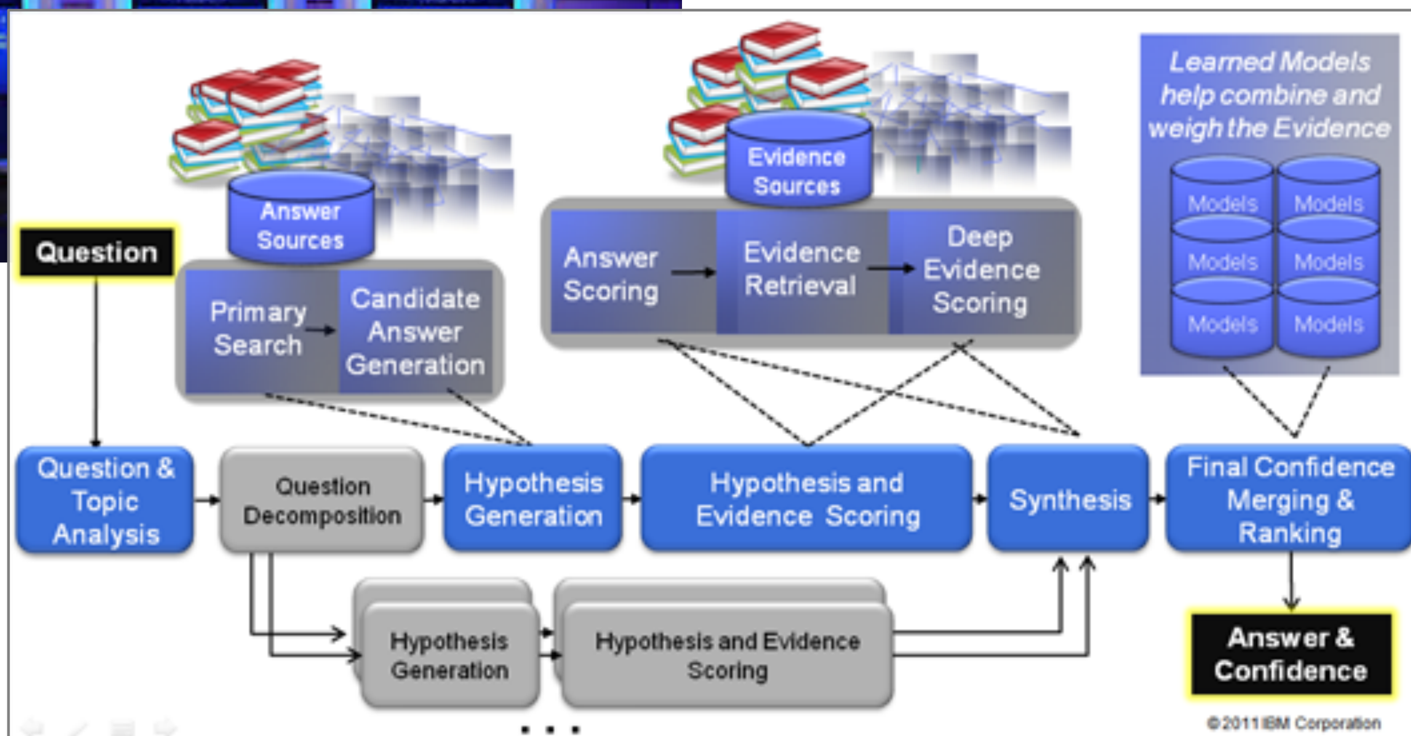


"Coffee near Brussels Central Station"

GHENT UNIVERSITY

NLP today … question answering

IBM Watson: 25 engineers; 4 years; 200 subsystems; 2,880 cores; 15 TB storage …



NLP today … machine translation

OUTLINE

- **INTRO**: Why NLP? Why neural networks for NLP?
- **PART I:** Joint entity recognition and relation extraction
- **PART II:** Automated lyrics annotation
- **PART III:** Explaining character-aware NNs for word-level prediction
- **PART IV:** Predefined sparseness in recurrent sequence models

PART I:
Joint entity recognition & relation extraction

G. Bekoulis, J. Deleu, T. Demeester and C. Develder, **"Joint entity recognition and relation extraction as a multi-head selection problem"**, Expert Syst. Appl., Vol. 114, Dec. 2018, pp. 34-45.

G. Bekoulis, J. Deleu, T. Demeester and C. Develder, **"Adversarial training for multi-context joint entity and relation extraction"**, in Proc. Conf. Empirical Methods in Natural Lang. Processing (EMNLP 2018), Brussels, Belgium, 31 Oct. - 4 Nov. 2018.

G. Bekoulis, J. Deleu, T. Demeester and C. Develder, **"An attentive neural architecture for joint segmentation and parsing and its application to real estate ads"**, Expert Syst. Appl., Vol. 102, 15 Jul. 2018, pp. 100-112.

Problem: Real estate information extraction

INPUT: Advertisement

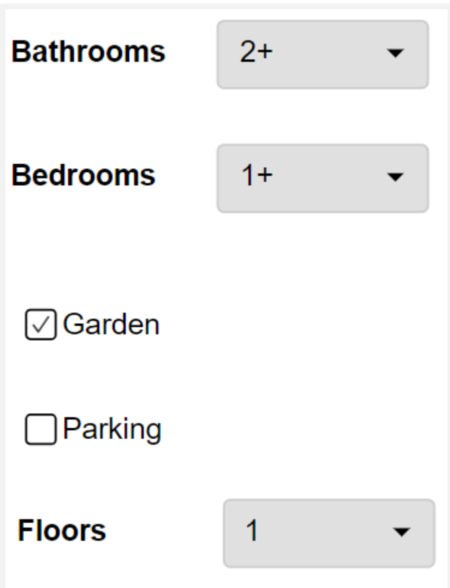
The property includes the apartment house with a garage. The house has living room, kitchen and bathroom with shower.

OUTPUT: Property structure

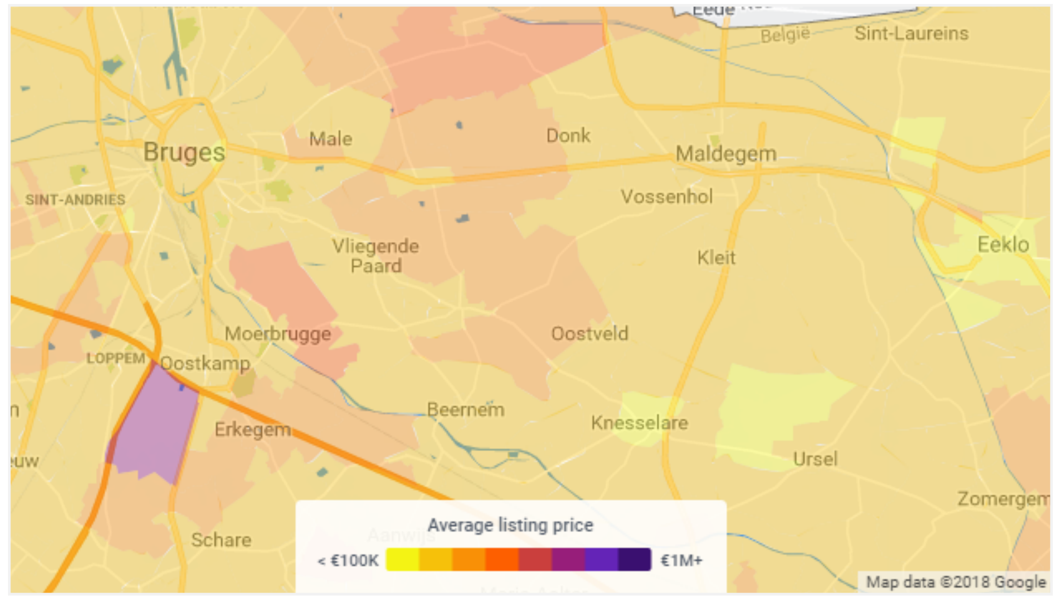
property
- house | mention = 'apartment house'
  - living room | mention = 'living room'
  - kitchen | mention = 'kitchen'
  - bathroom | mention = 'bathroom'
    - shower | mention = 'shower'
- garage | mention = 'garage'

Why is this useful?

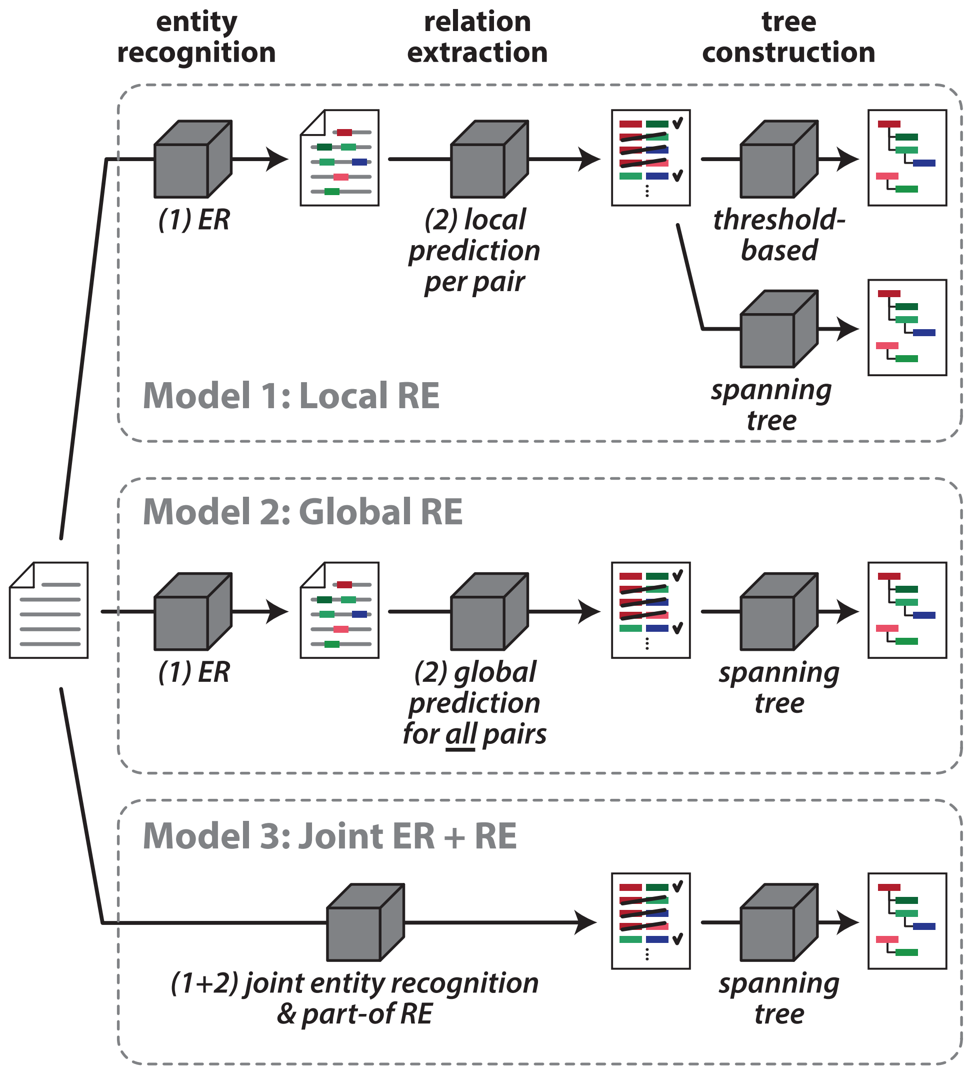
Specialized filtering



Automatic price prediction



Our solutions



GHENT UNIVERSITY

TWO-STEP MODEL

(1) Entity recognition

(2) Construct property tree

Entity recognition = Sequence labeling

- Classical NLP task = NER, named entity recognition

contentSkip to site indexPoliticsSubscribeLog InSubscribeLog InToday's PaperAdvertisementSupported **ORG** byF.B.I. Agent Peter Strzok **PERSON** , Who Criticized Trump **PERSON** in Texts, Is FiredImagePeter Strzok, a top F.B.I. **GPE** counterintelligence agent who was taken off the special counsel investigation after his disparaging texts about President Trump **PERSON** were uncovered, was fired. CreditT.J. Kirkpatrick **PERSON** for The New York TimesBy Adam Goldman **ORG** and Michael S. SchmidtAug **PERSON** . 13 **CARDINAL** , 2018WASHINGTON **CARDINAL** — Peter Strzok **PERSON** , the F.B.I. **GPE** senior counterintelligence agent who disparaged President Trump **PERSON** in inflammatory text messages and helped oversee the Hillary Clinton **PERSON** email and Russia **GPE** investigations, has been fired for violating bureau policies, Mr. Strzok **PERSON** 's lawyer said Monday **DATE** .Mr. Trump and his allies seized on the texts — exchanged during the 2016 **DATE** campaign with a former F.B.I. **GPE** lawyer, Lisa Page — in **PERSON** assailing the Russia **GPE** investigation as an illegitimate "witch hunt." Mr. Strzok **PERSON** , who rose over 20 years **DATE** at the F.B.I. **GPE** to become one of its most experienced counterintelligence agents, was a key figure in the early months **DATE** of the inquiry.Along with writing the texts, Mr. Strzok **PERSON** was accused of sending a highly sensitive search warrant to his personal email account.The F.B.I. **GPE** had been under immense political pressure by Mr. Trump **PERSON** to dismiss Mr. Strzok **PERSON** , who was removed last summer **DATE** from the staff of the special counsel, Robert S. Mueller III **PERSON** . The president has repeatedly denounced Mr. Strzok **PERSON** in posts on

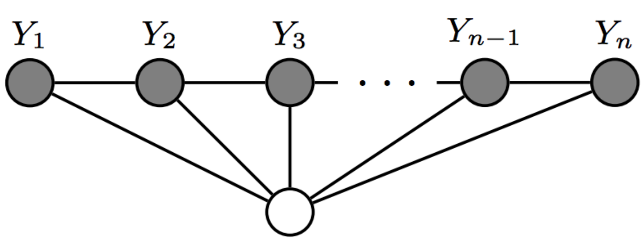
Entity recognition = Sequence labeling

- Classical NLP task = NER, named entity recognition
  - Types of "entities":
    - geo = geographical entity
    - org = Organization
    - per = Person
    - gpe = Geopolitical Entity
    - tim = Time indicator
    - art = Artifact
    - eve = Event
    - nat = Natural Phenomenon
  - Encoding of "labels": BIO
    - B = beginning
    - I = inside
    - O = outside

Entity recognition = Sequence labeling

- Classical NLP task = NER, named entity recognition

- Solution: Conditional Random Fields (CRF)
  - undirected graphical mode or Markov random field
  - globally conditioned on random variable representing observation sequences



$\boldsymbol{X} = X_1, \ldots, X_{n-1}, X_n$

$$p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\lambda}) = \frac{1}{Z(\boldsymbol{x})} \prod_i \Psi_i(\boldsymbol{y},\boldsymbol{x})$$

learnable parameters

$$\Psi_i(\boldsymbol{y},\boldsymbol{x}) = exp\left(\sum_j \lambda_j \underbrace{t_j(y_{i-1}, y_i, \boldsymbol{x}, i)}_{\text{transition feature function}} + \sum_k \mu_k \underbrace{s_k(y_i, \boldsymbol{x}, i)}_{\text{state feature function}}\right)$$

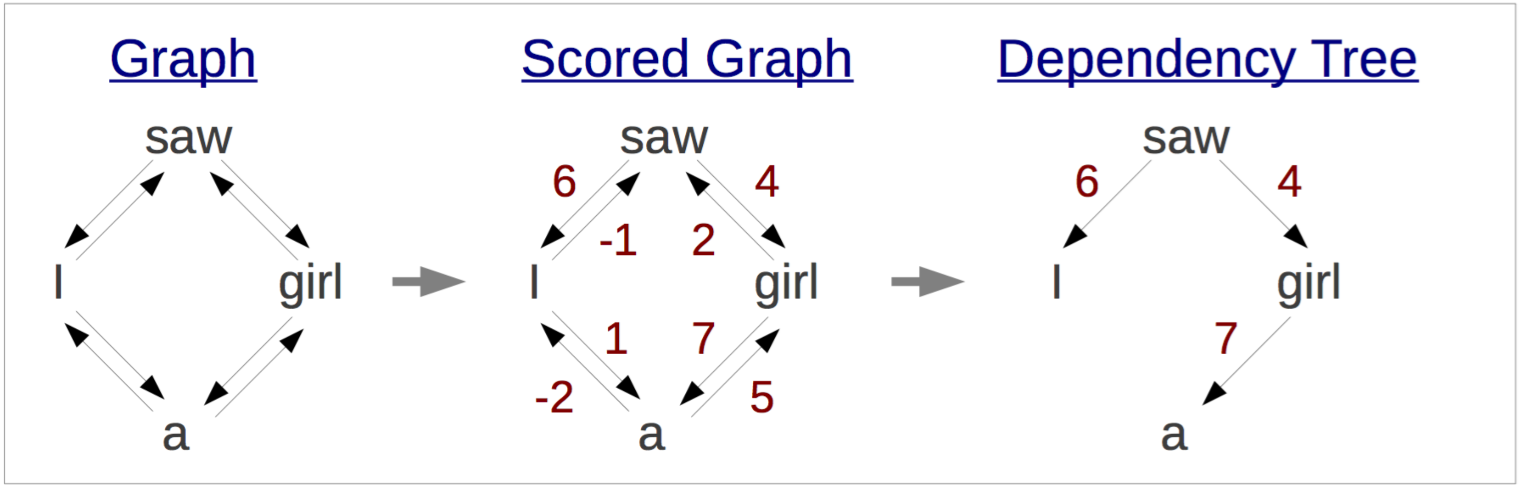
Structure prediction = Dependency parsing

- Classical NLP task = dependency parsing

Structure prediction = Dependency parsing

- Classical NLP task = dependency parsing

- Solutions:
  - Graph-based model = find the maximum spanning tree
    - Edge represents potential dependency
    - Assign score to each edge (with machine learning)
    - Keep the tree with the highest score

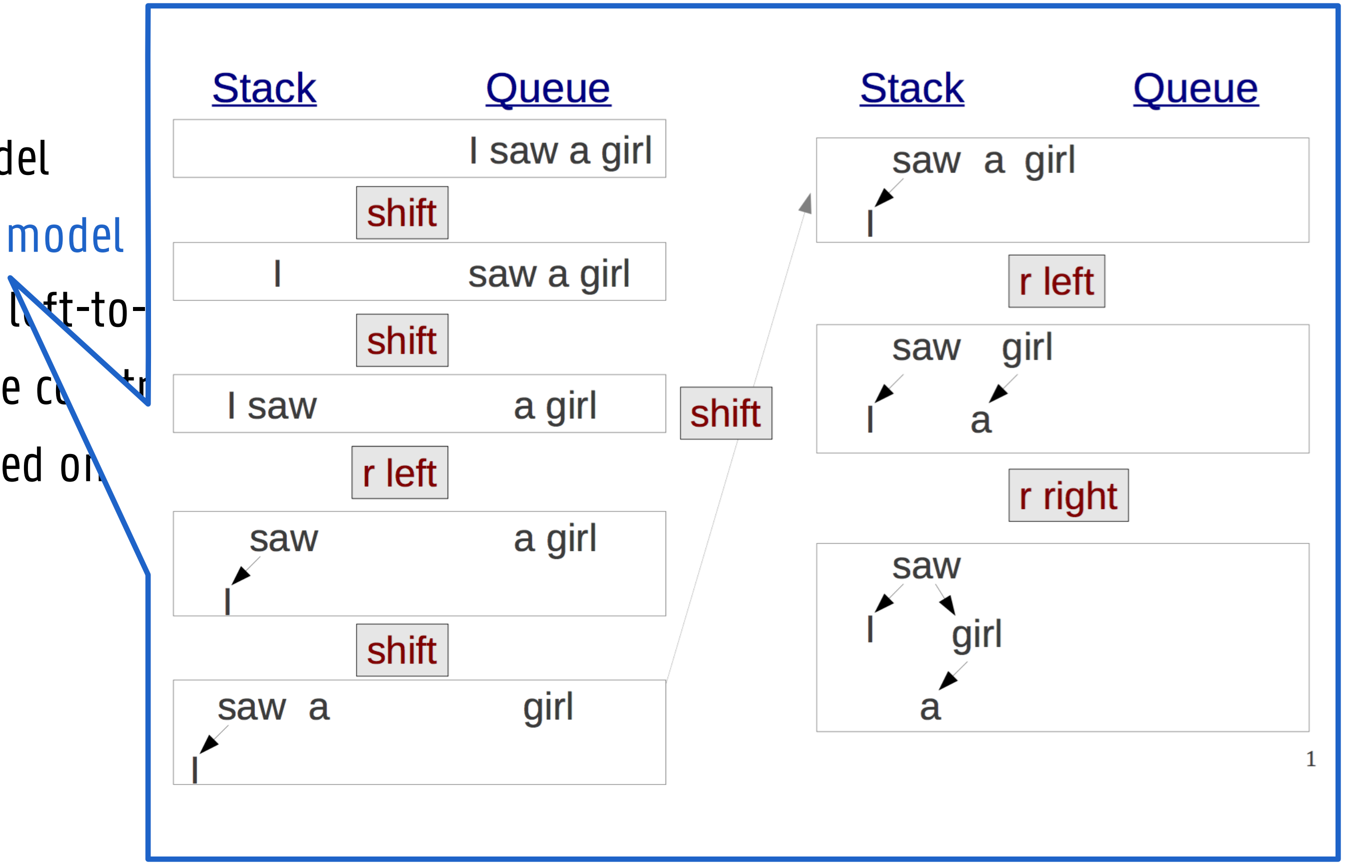


Structure prediction = Dependency parsing

- Classical NLP task = dependency parsing
- Solutions:
  - Graph-based model
  - Transition-based model
    - Process text left-to-right
    - Stepwise tree construction
    - Decision based on feature representation of stack & queue

Structure prediction = Dependency parsing

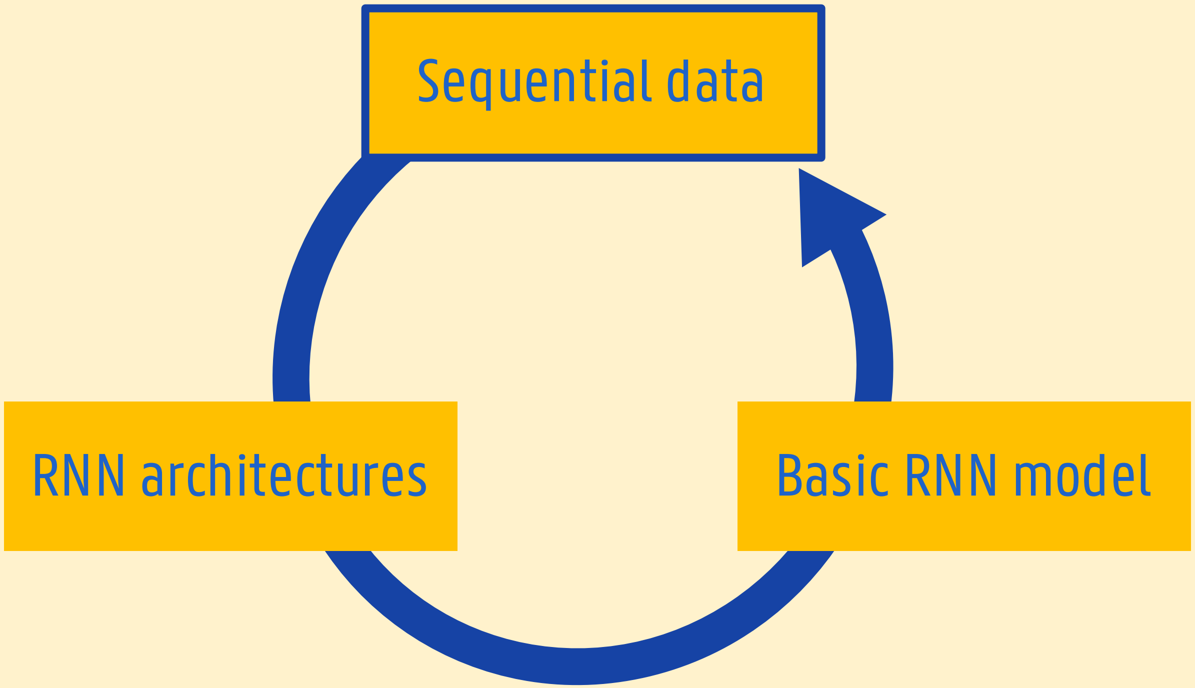
- Classical NLP task = dependency parsing
- Solutions:
  - Graph-based model
  - Transition-based model
    - Process text l…ft-to-…
    - Stepwise tree c…
    - Decision based o…



INTERMEZZO 1
Introduction to RNNs for NLP

Goal of this intermezzo …

- Recurrent neural network basics
- Conceptual overview of RNN architectures



Tasks with sequential data

- Named entity recognition

Comedian Zelensky wins Ukraine's elections. → Comedian **Zelensky** wins **Ukraine**'s elections.

- Text categorization

Parkinson's implant 'transforms lives'
A treatment that has restored the movement of patients with …
→
economy
conflict
health
gossip

- Sentiment classification

Predictable sequel with crass, suggestive humor → ★☆☆☆☆

- Machine translation

Je suis ravi de vous rencontrer. → I'm pleased to meet you.

- Speech-to-text

→ Winter is coming.

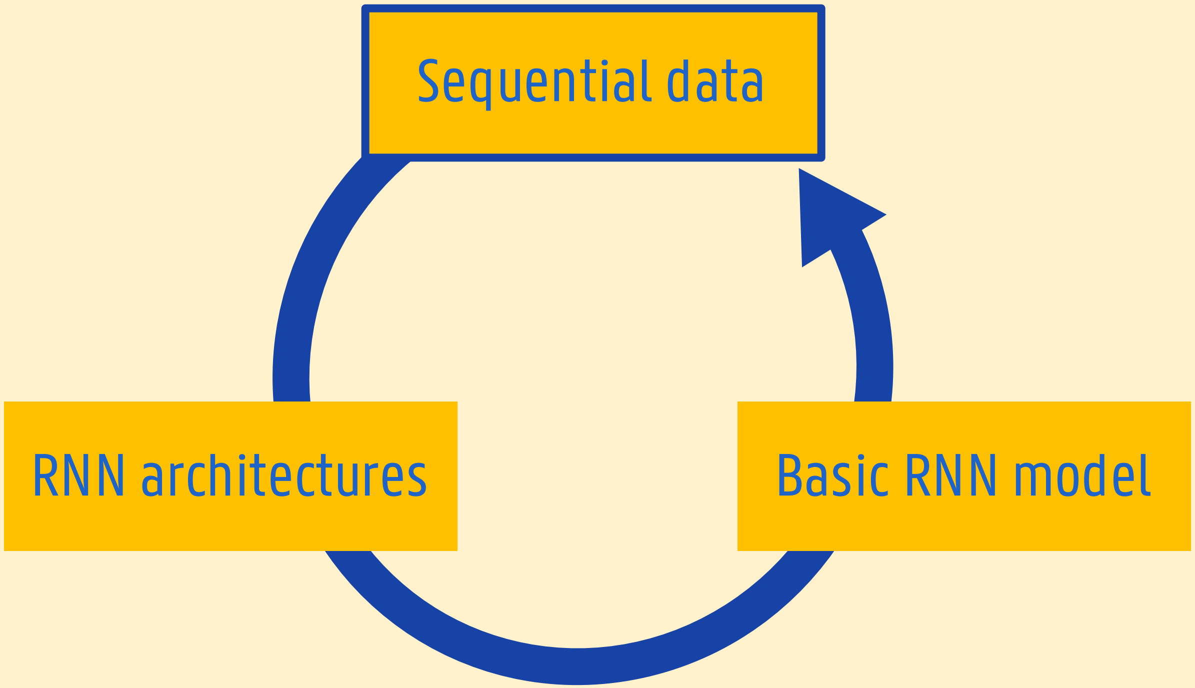
- Caption generation

→ A man in black armor with a sword

GHENT UNIVERSITY

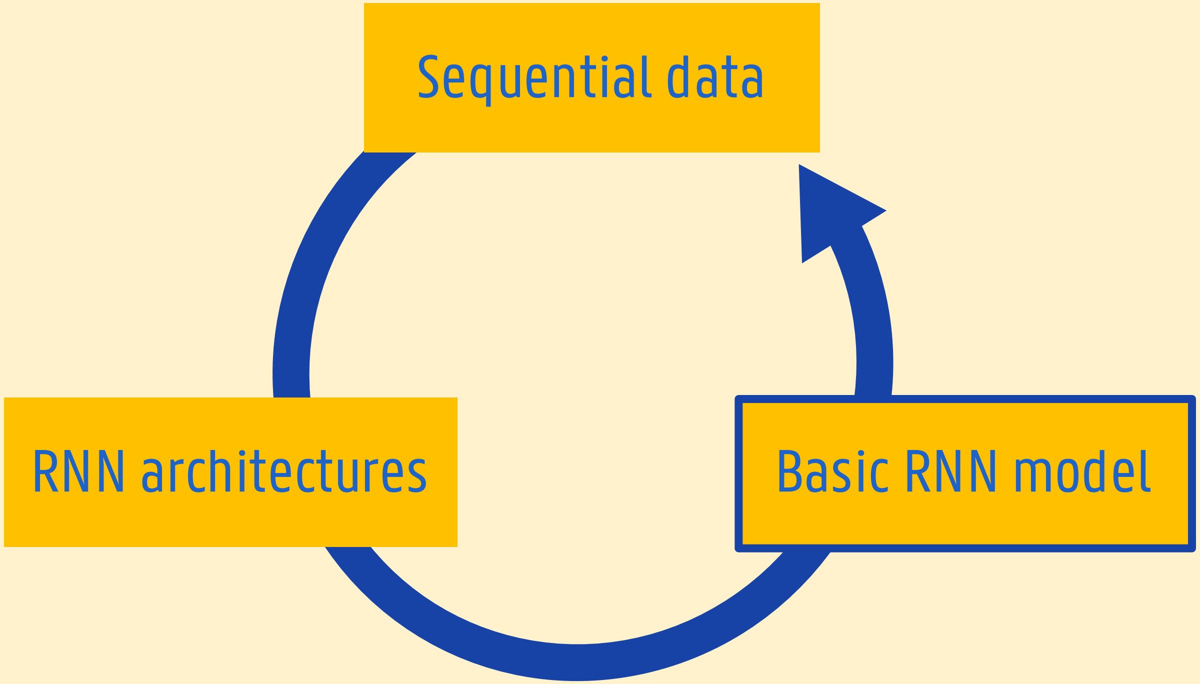
Goal of this intermezzo …

- Recurrent neural network basics
- Conceptual overview of RNN architectures



Goal of this intermezzo …

- Recurrent neural network basics
- Conceptual overview of RNN architectures



Notations

- Input sequence:

$$x^{<1>} \qquad x^{<2>} \qquad \ldots \qquad x^{<t>} \qquad \ldots \qquad x^{<T_x>}$$

input item at position t
($t \in [1, \ldots, T_x]$)

T_x = length of input sequence

- Output sequence:

$$\hat{y}^{<1>} \qquad \hat{y}^{<2>} \qquad \ldots \qquad \hat{y}^{<t>} \qquad \ldots \qquad \hat{y}^{<T_y>}$$

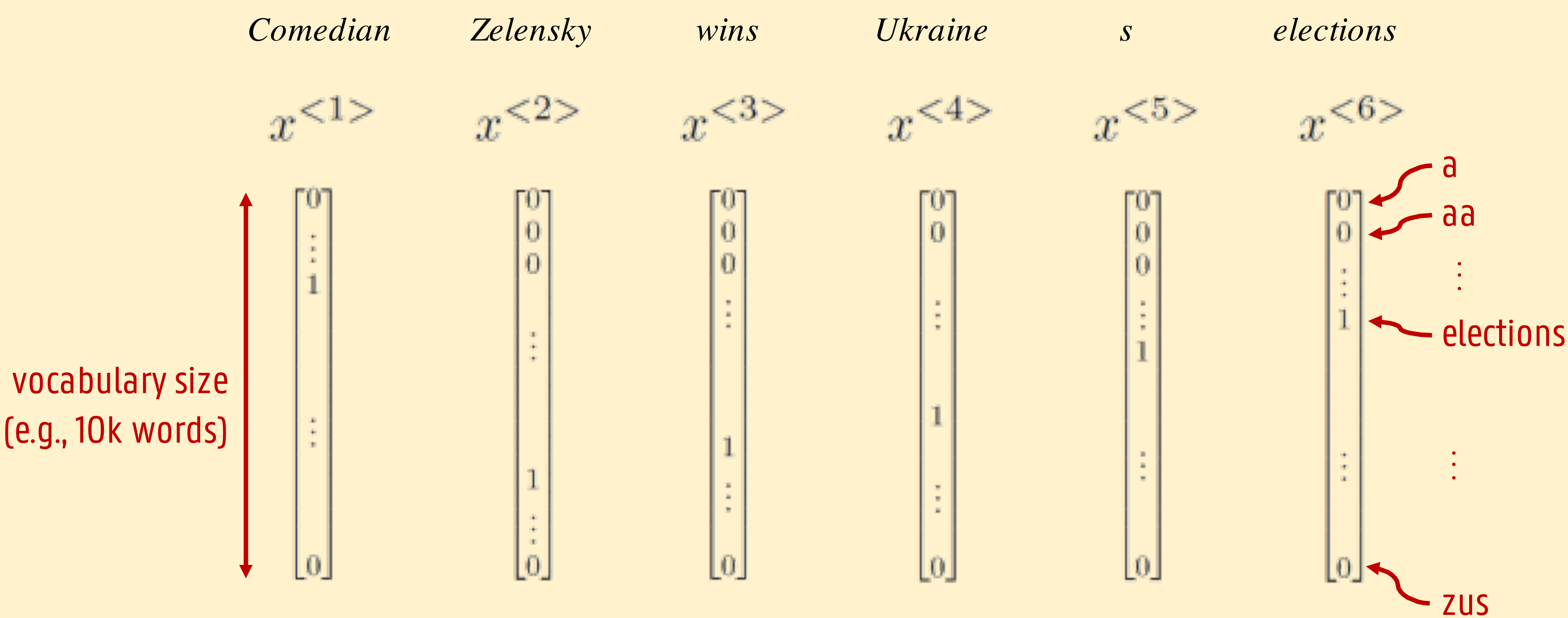
distinction between **predicted** model output $\hat{y}^{<t>}$
vs. **actual** output item $y^{<t>}$ at position t

T_x = length of input sequence
(not always $T_x = T_y$)

GHENT UNIVERSITY

Example sequence representation

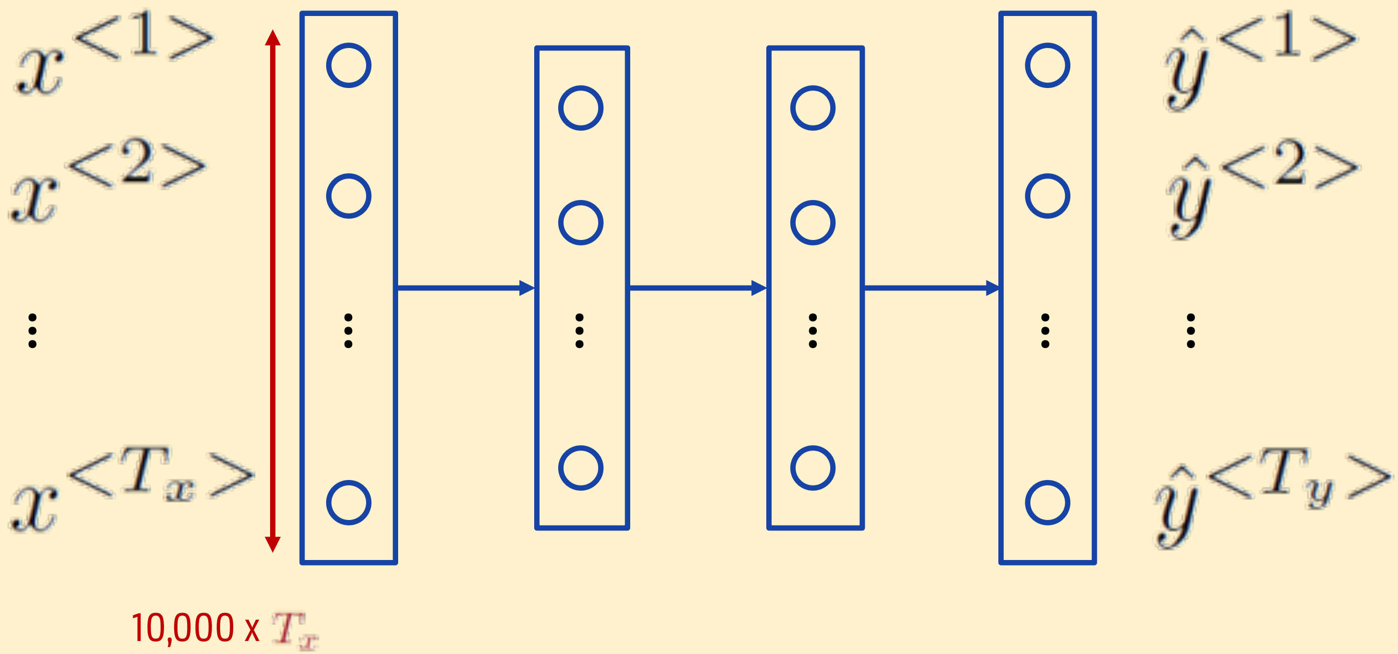
- Input sequence:



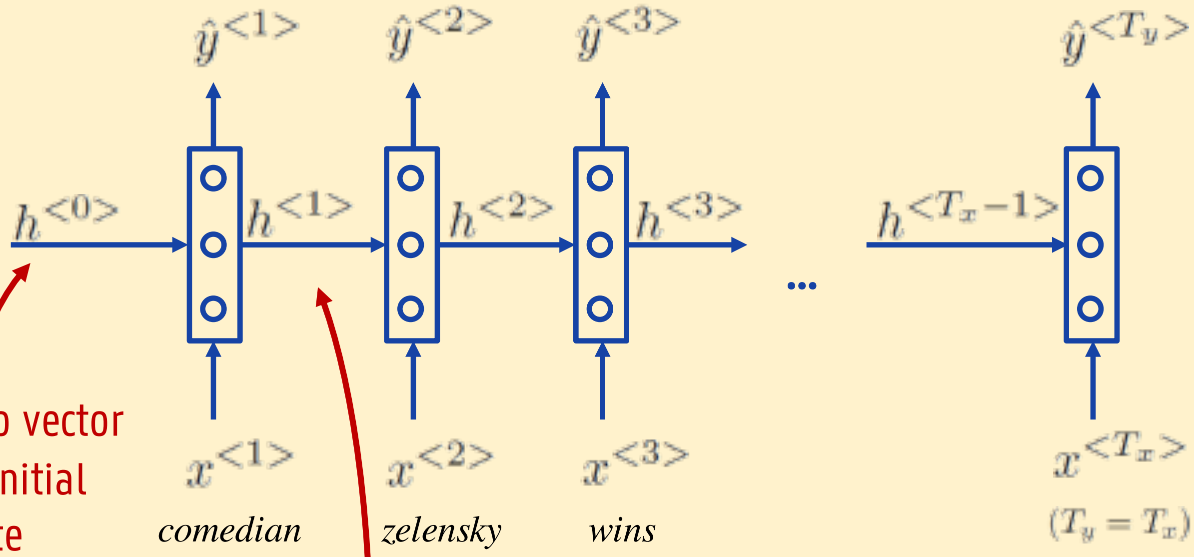
This example: "one-hot" word representations
(smarter choices are possible, cf. "embeddings")

Why not simple "feed-forward" neural nets?

- Problems:
  - How to deal with **variable length** sequences?
  - How to **share features** among words at different positions?



Recurrent neural network

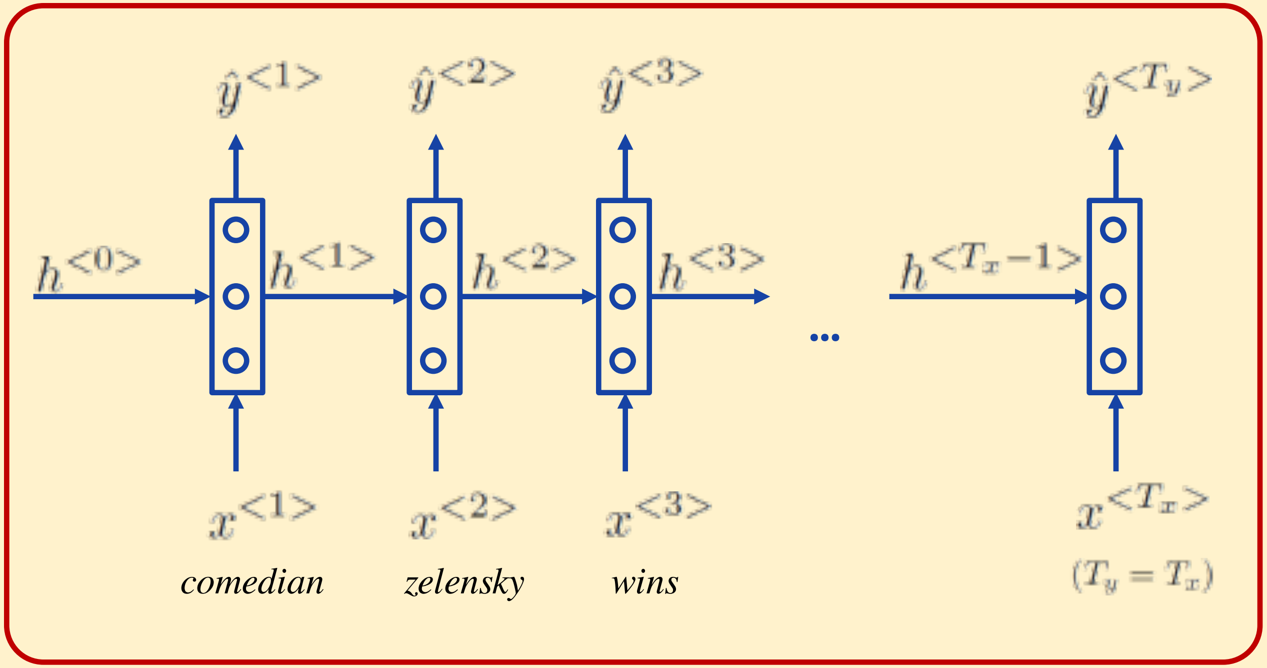


zero vector
as initial
state

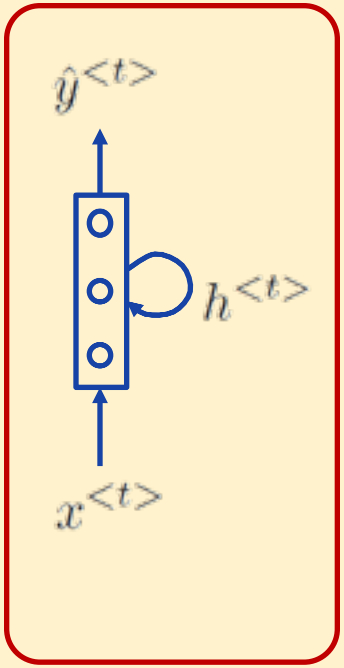
activation
of the neural network
= RNN “state”

GHENT UNIVERSITY

Recurrent neural network



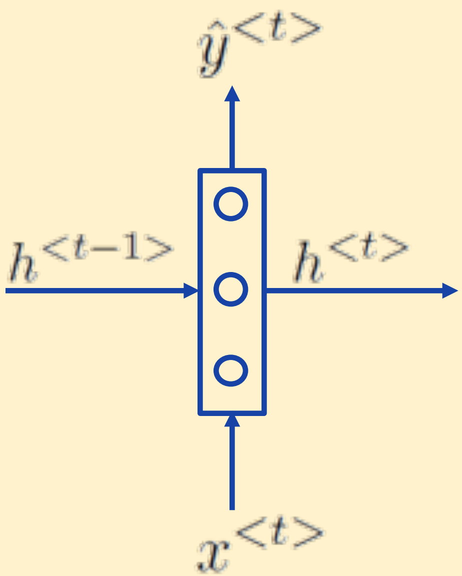
"unrolled" RNN representation



"compact" RNN representation

Elman RNN model

- Output at step *t* = based on current input + previous state
- All past sequence items are compressed into the previous state



activation function $f = \tanh(.)$ or $\mathrm{ReLU}(.)$

weight matrix

bias

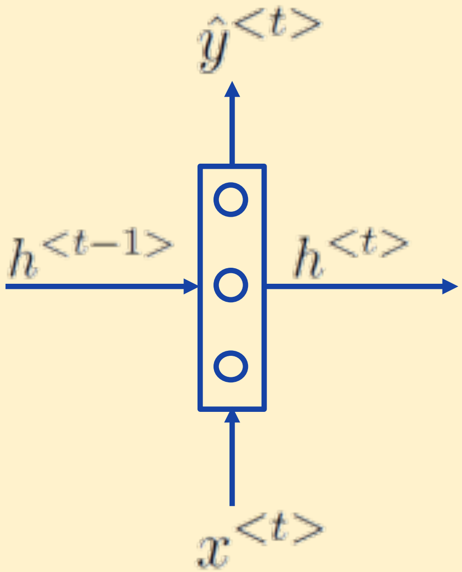
$$h^{<t>} = f(W_h \, x^{<t>} + U_h \, h^{<t-1>} + b_h)$$

$$\hat{y}^{<t>} = g(W_y \, h^{<t>} + b_y)$$

activation function $g = \sigma(.)$

Elman RNN model

- Output at step *t* = based on current input + previous state
- All past sequence items are compressed into the previous state



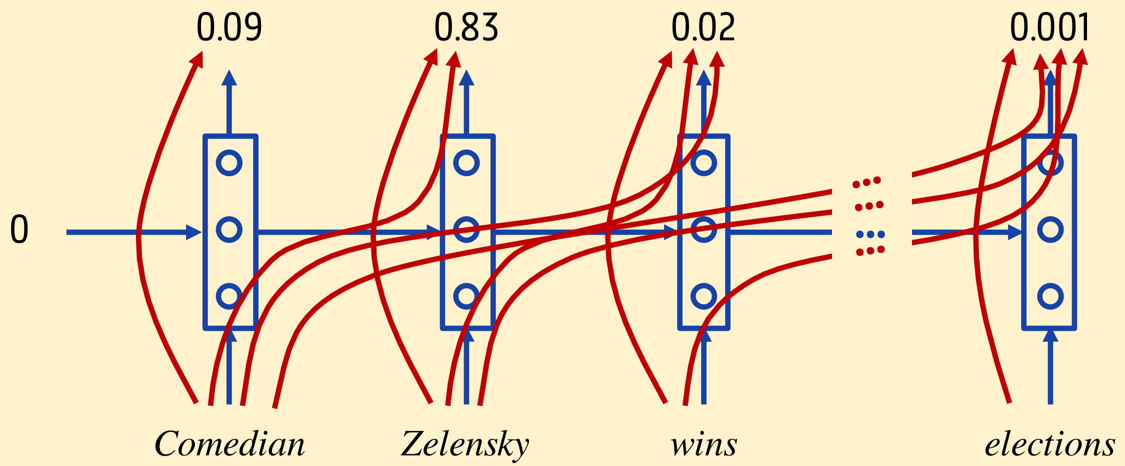
"recurrent" term

$$h^{<t>} = f\left(W_h\, x^{<t>} + \boxed{U_h\, h^{<t-1>}} + b_h\right)$$

$$\hat{y}^{<t>} = g\left(W_y\, h^{<t>} + b_y\right)$$

- "Vanilla" RNN model
- Known issues during training, suffers from short-range memory …

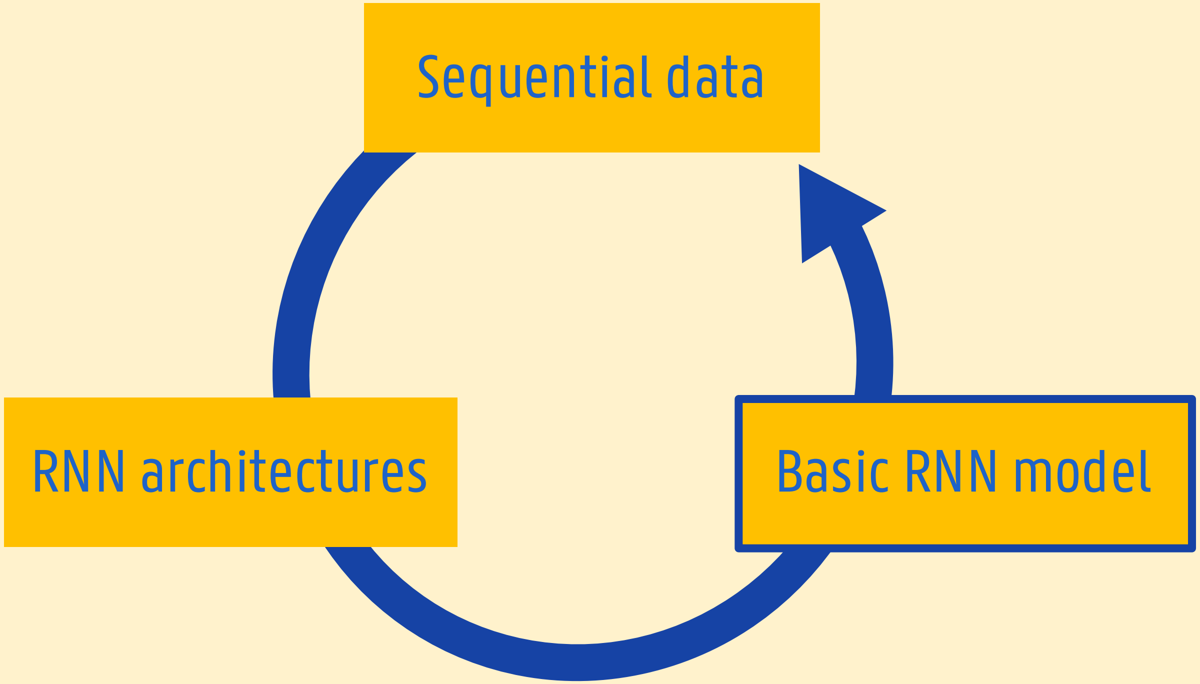
Forward propagation



these dependencies are important for training the RNN with the backpropagation-through-time algorithm

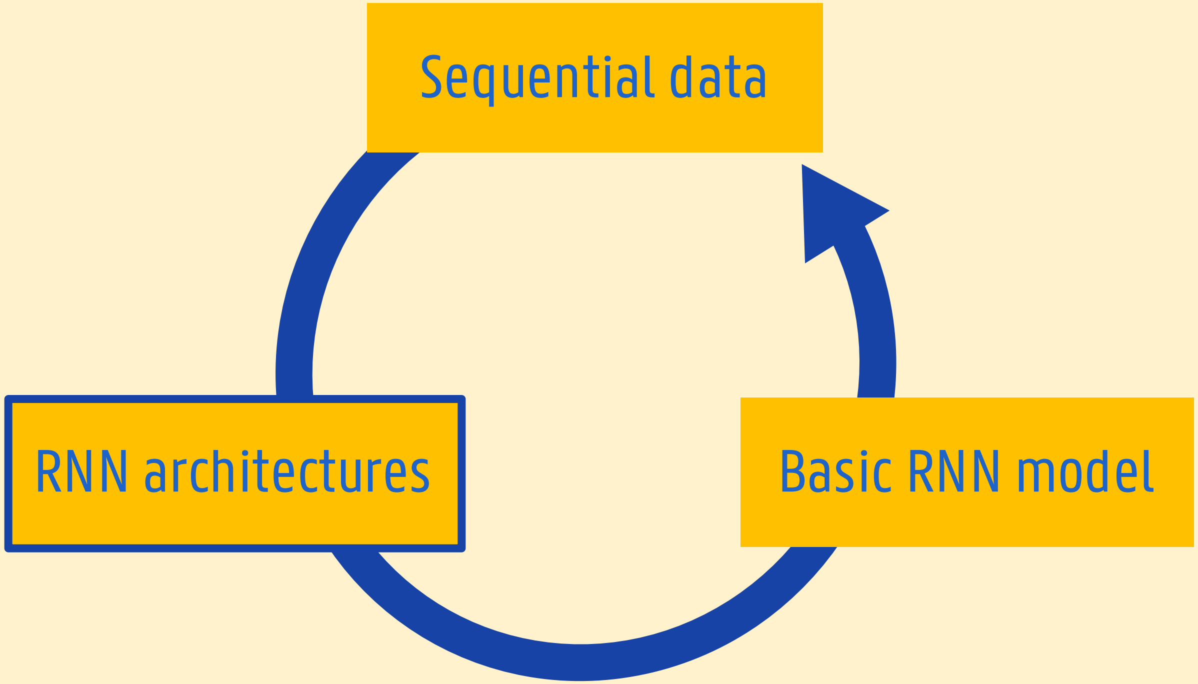
Goal of this intermezzo ...

- Recurrent neural network basics
- Conceptual overview of RNN architectures



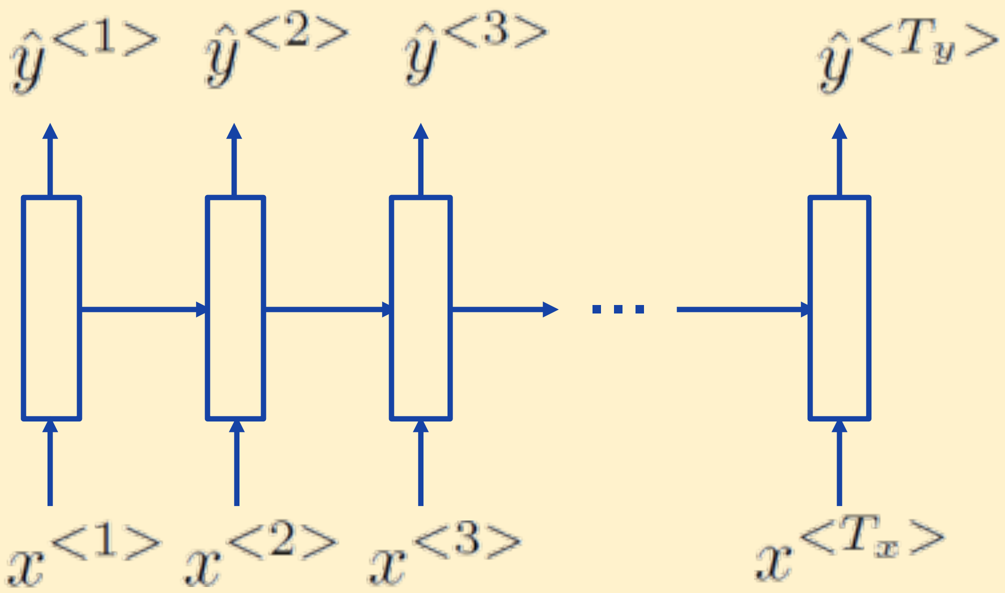
Goal of this intermezzo …

- Recurrent neural network basics
- Conceptual overview of RNN architectures



Many-to-many architecture

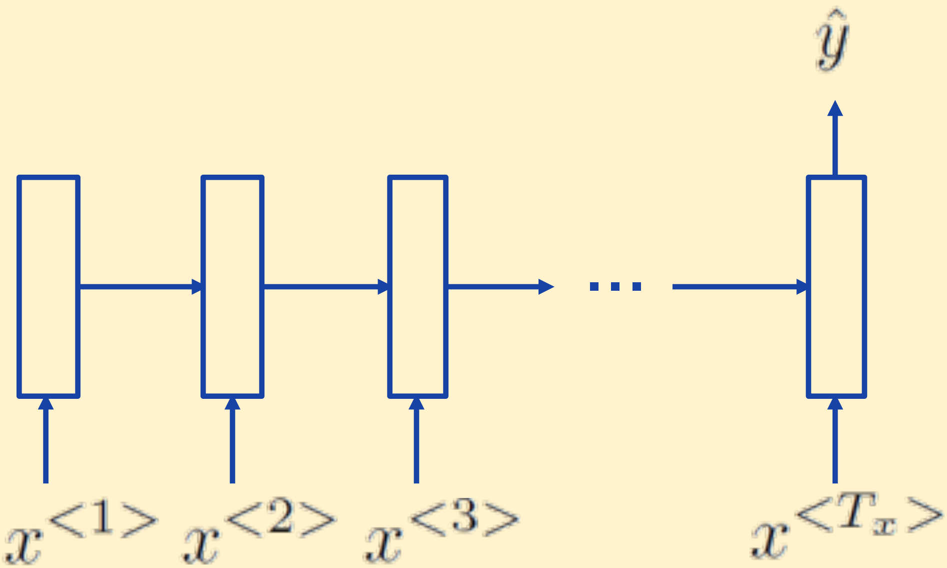
output sequence of equal length



input sequence

Many-to-one architecture

single output

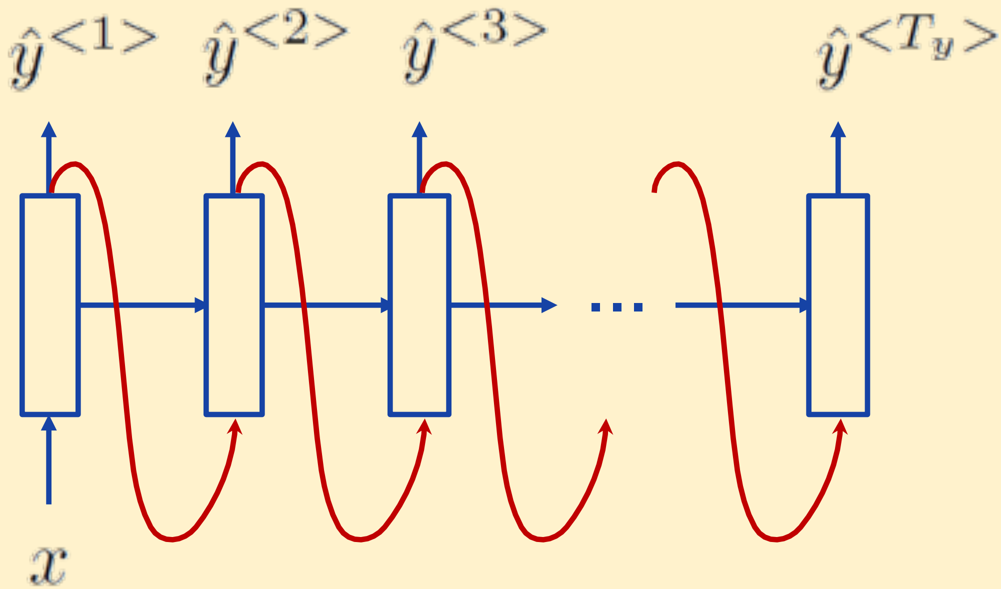


last state $h^{<T_x>}$ can be seen as a representation (or 'summary') of the entire input sequence

input sequence

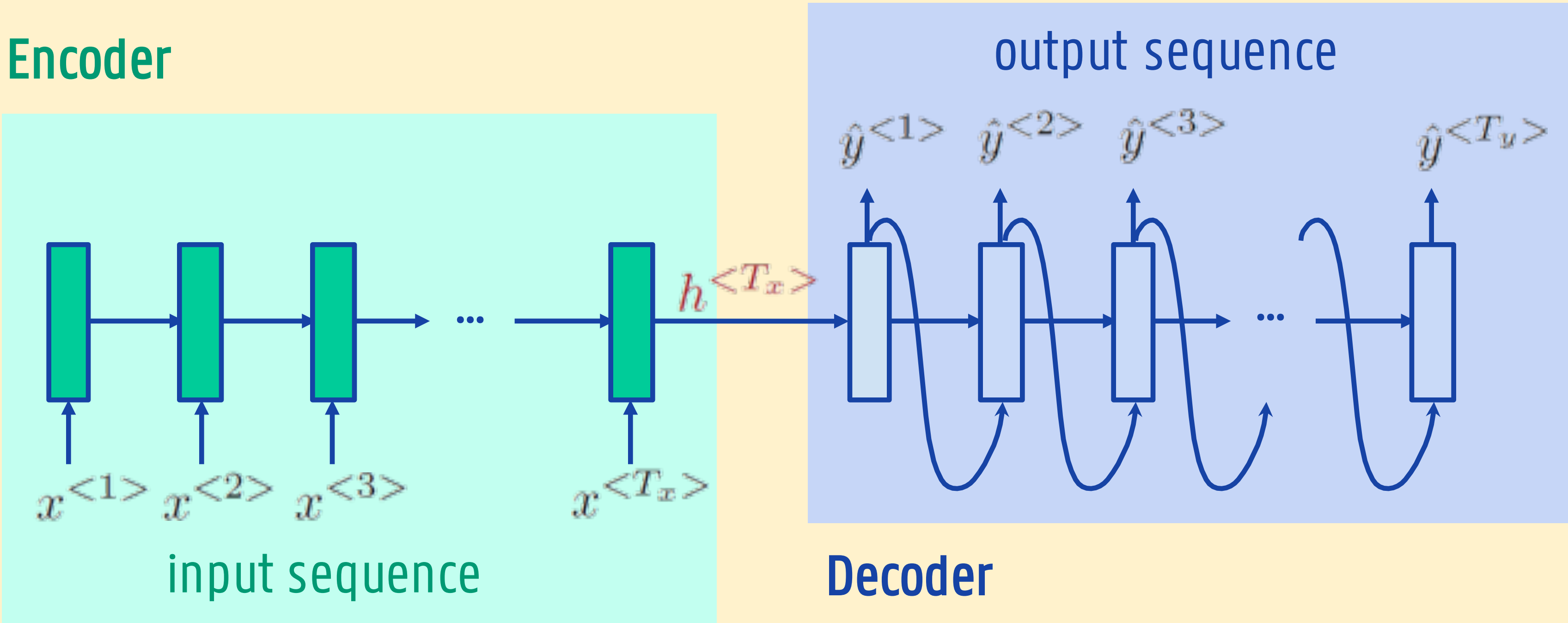
One-to-many architecture

output sequence



single input item

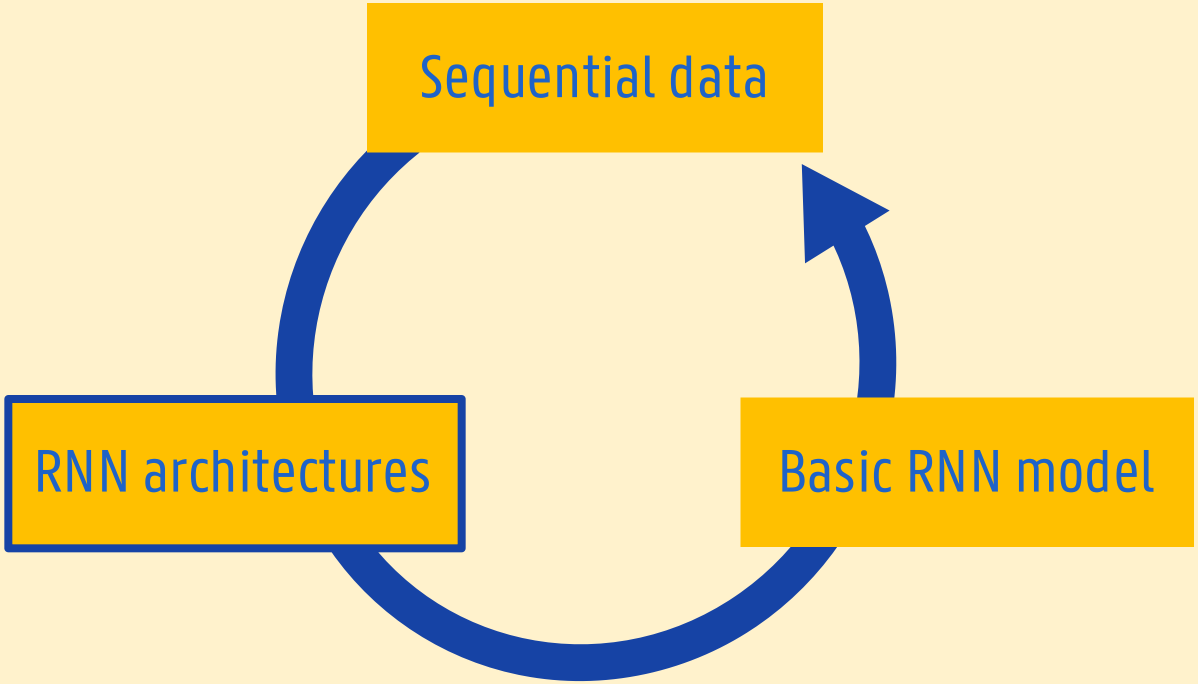
Encoder/decoder architecture



GHENT UNIVERSITY

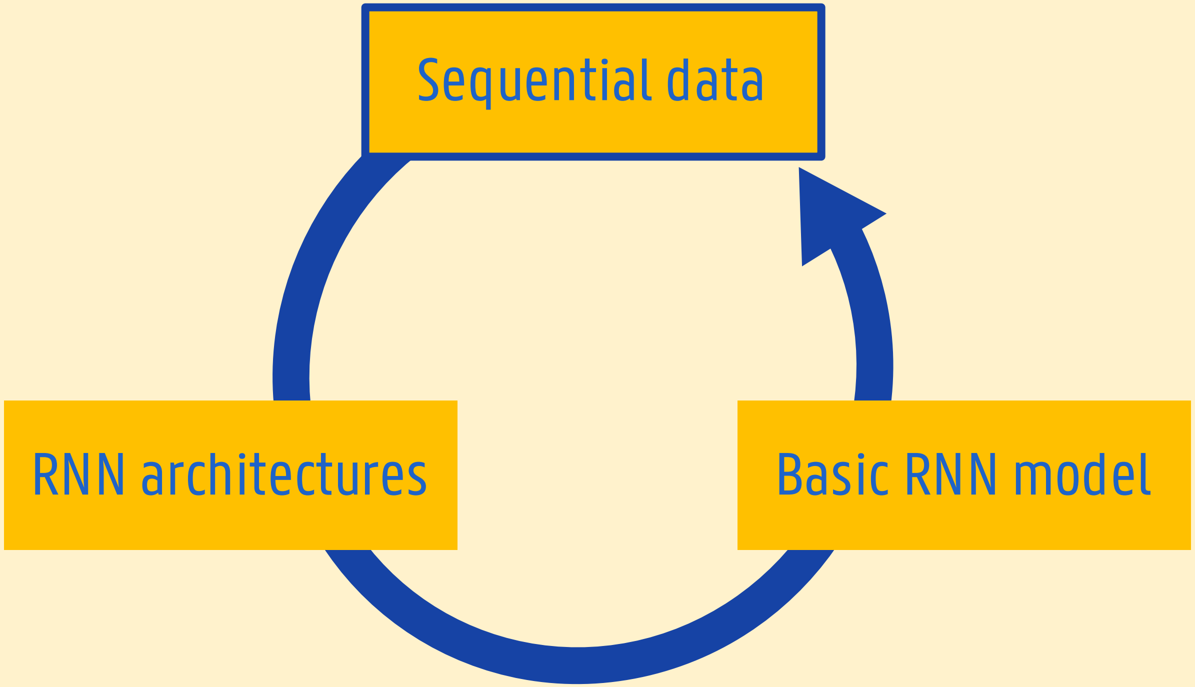
Goal of this intermezzo …

- Recurrent neural network basics
- Conceptual overview of RNN architectures



Goal of this intermezzo …

- Recurrent neural network basics
- Conceptual overview of RNN architectures



Tasks with sequential data

- Named entity recognition
  → Many-to-many

  Comedian Zelensky wins Ukraine's elections. ➡ Comedian **Zelensky** wins **Ukraine**'s elections.

- Text categorization
  → Many-to-one

  Parkinson's implant 'transforms lives'
  A treatment that has restored the movement of patients with …
  ➡ economy / conflict / **health** / gossip

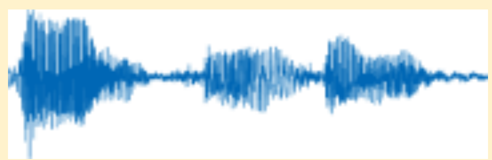
- Sentiment classification
  → Many-to-one

  Predictable sequel with crass, suggestive humor ➡ ★☆☆☆☆

- Machine translation
  → Encoder/decoder

  Je suis ravi de vous rencontrer. ➡ I'm pleased to meet you.

- Speech-to-text
  → Encoder/decoder

  
  ➡ Winter is coming.

- Caption generation
  → One-to-many

  
  ➡ A man in black armor with a sword

INTERMEZZO 2
Notions of “embeddings”

GHENT UNIVERSITY

Why dense word vectors?

- **What?**
  - Vector representation = short (50-1000) + dense (mostly non-zero)

- **Why?**
  - Easier to use as features (less parameters)
  - May generalize better
  - May better capture synonymy
  - …

  → They work better in practice!

Examples

- **Word2vec -** https://code.google.com/archive/p/word2vec/
- **Glove -** http://nlp.stanford.edu/projects/glove
- **Fasttext -** http://www.fasttext.cc/

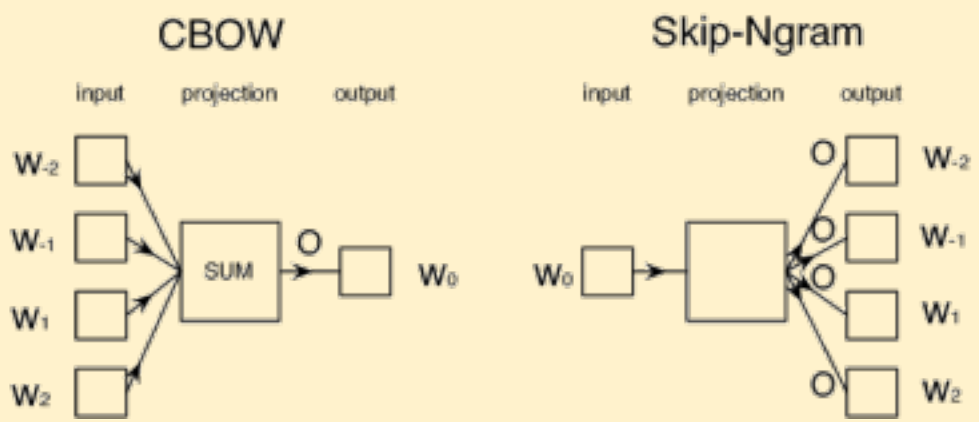
Recent approaches use contextualized representations,

i.e., dependent on surrounding words:

- **ELMO -** https://allennlp.org/elmo
- **Bert -** https://github.com/google-research/bert
- ...

Word2vec

- Idea:
  - Look at words in context
  - Rather than counting how often *w* occurs near another, say “apricot”, train a classifier on a binary <u>prediction</u> task: is *w* likely to occur near “apricot”?
  - Use classifier weights as the embeddings
- Two classification tasks:
  - CBOW = continuous bag-of-words
  - Skip-gram



Word2vec – Skip-gram training

Training sentence:

... lemon, a tablespoon of apricot jam a pinch ...

c1 c2 t c3 c4

positive examples +

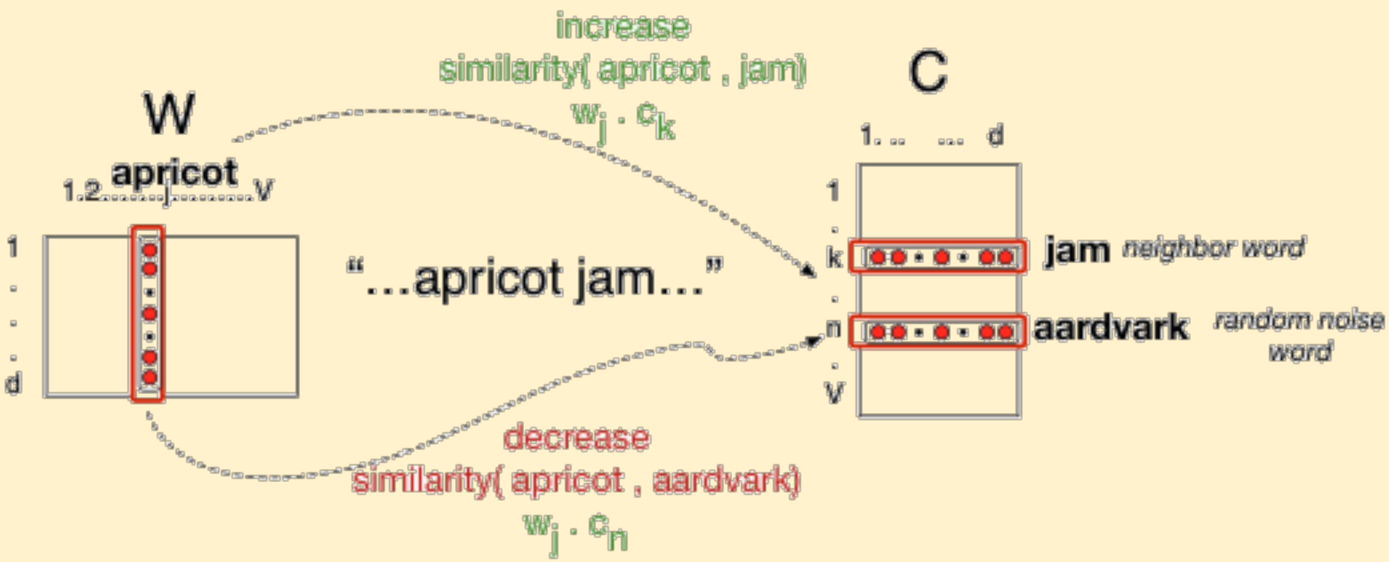
t	c
apricot	tablespoon
apricot	of
apricot	preserves
apricot	or

negative examples -

t	c	t	c
apricot	aardvark	apricot	twelve
apricot	puddle	apricot	hello
apricot	where	apricot	dear
apricot	coaxial	apricot	forever

Word2vec - Training

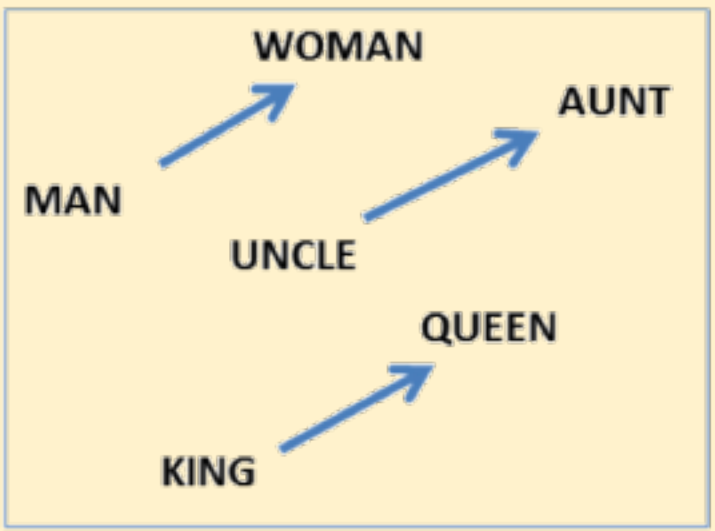
- Words V as vectors of fixed length (say 300)
- Initialize randomly, i.e., 300 x V random parameters
- Adjust word vectors over training set, to
  - Maximize similarity target word, context word pairs (t, c)
  - Minimize similarity of (t,c) pairs from negative data

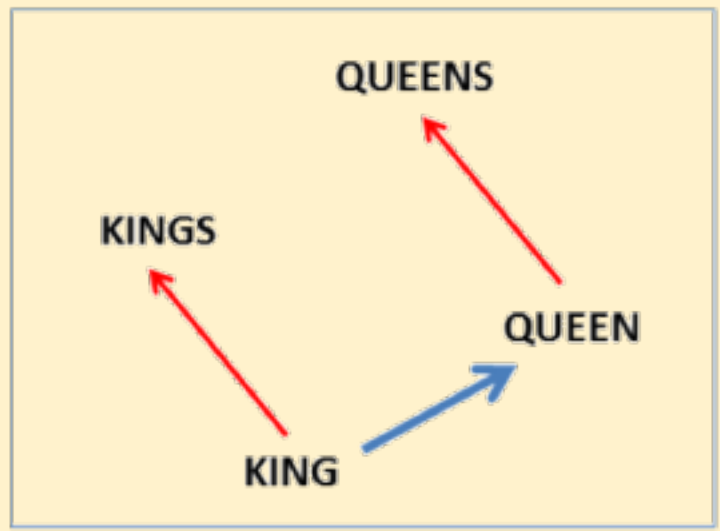


Embeddings capture relational meaning

vector(*'king'*) - vector(*'man'*) + vector(*'woman'*) ≈ vector('queen')

vector(*'Paris'*) - vector(*'France'*) + vector(*'Italy'*) ≈ vector('Rome')



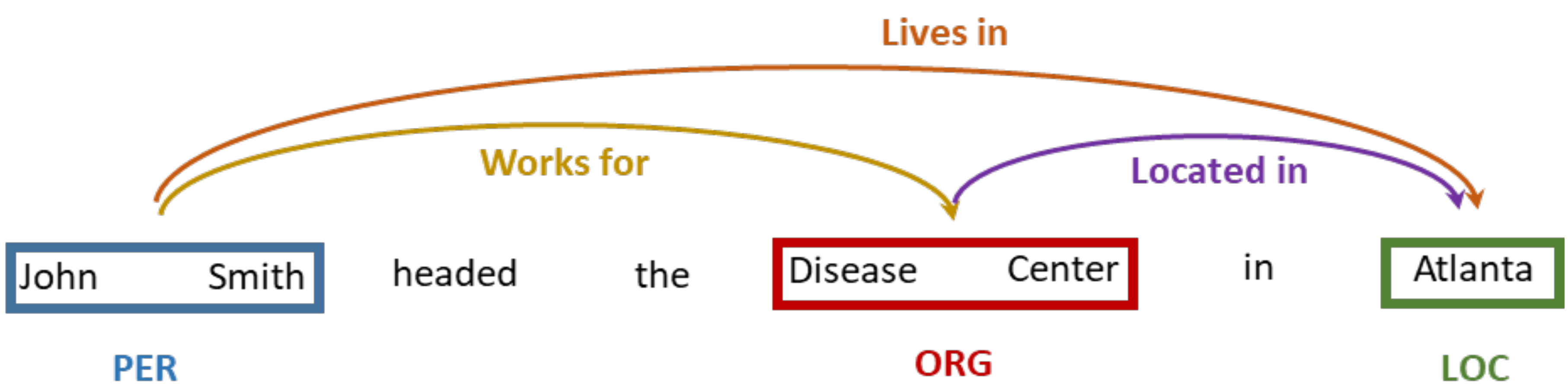


JOINT MODEL

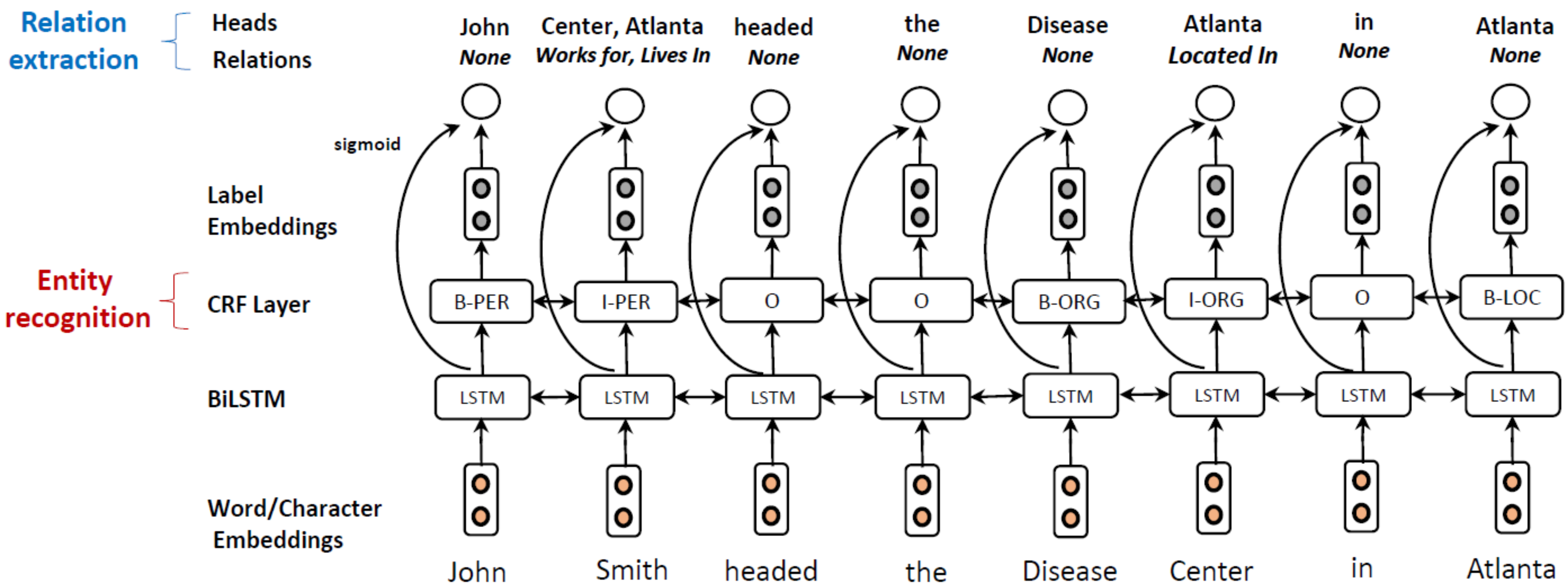
Extract entities + construct property tree at once

Goal: Joint entity recognition and relation extraction

- Solving two tasks at once:
  1. Entity recognition
  2. Relation extraction
- Use adversarial training



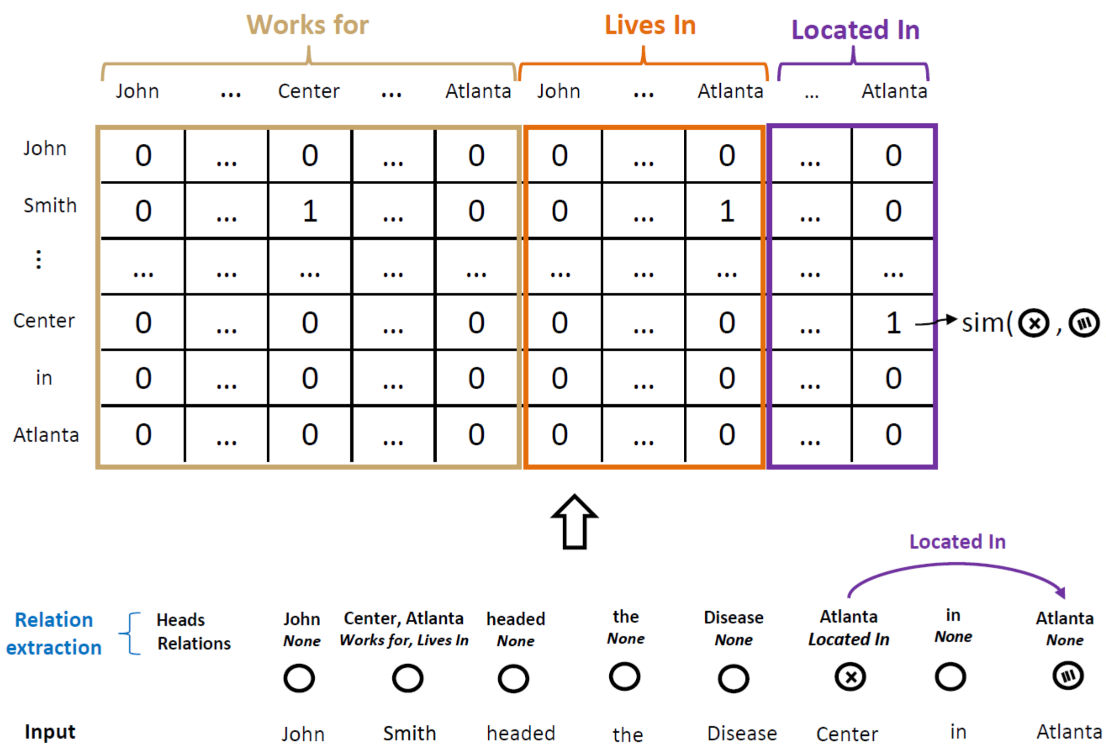
Overall model architecture



GHENT UNIVERSITY

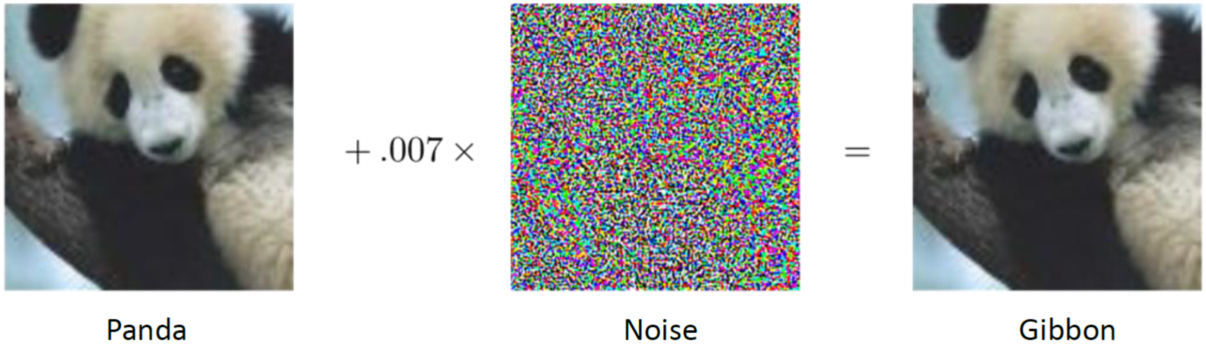
Relation extraction: Multihead selection

- Scoring matrix for each potential relation
- Score for (A,B) indicates probability that A is head of B



Adversarial training

- **Idea:**
  Regularization method to improve the robustness of neural network methods by adding small perturbations in the training data

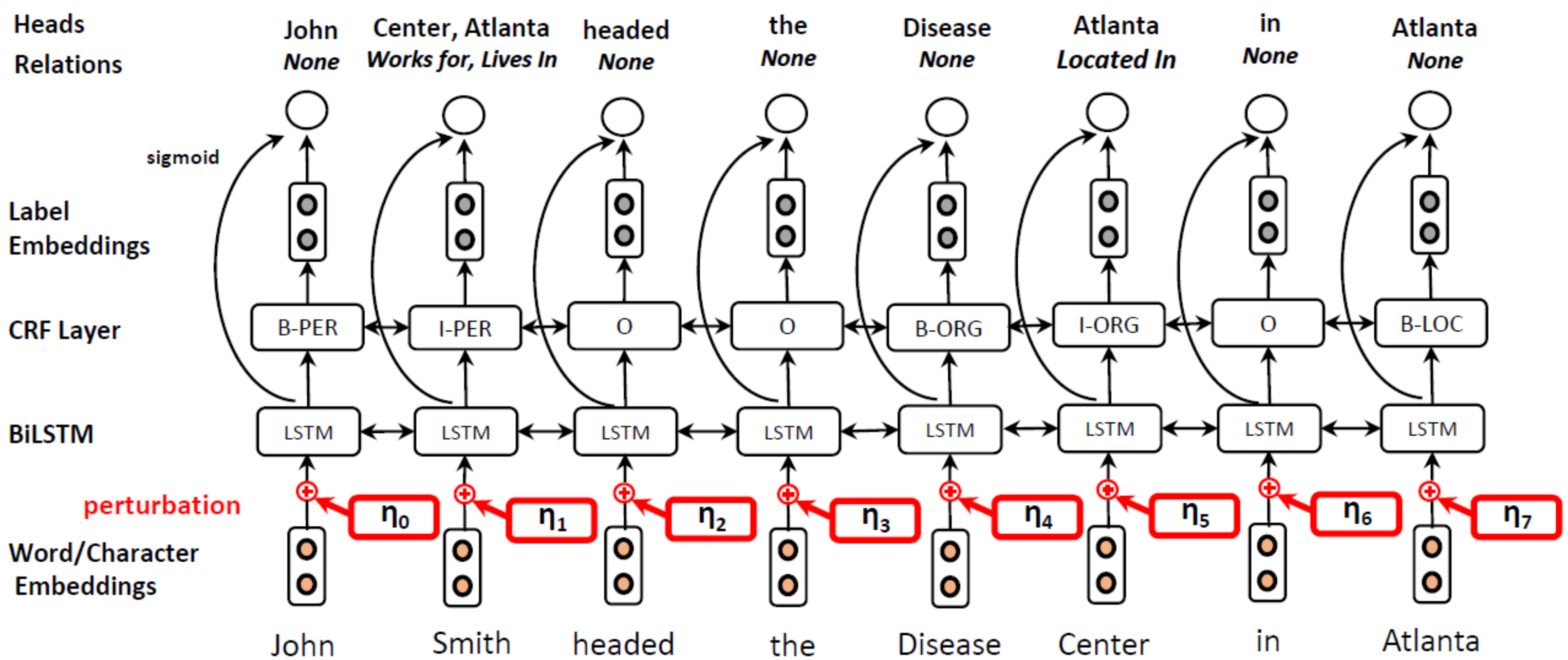


Source: Goodfellow et al. (2015).

- **Application in NLP:**
  - Text classication (Miyato et al., 2017)
  - Relation extraction (Wu et al., 2017)
  - POS tagging (Yasunaga et al., 2018)

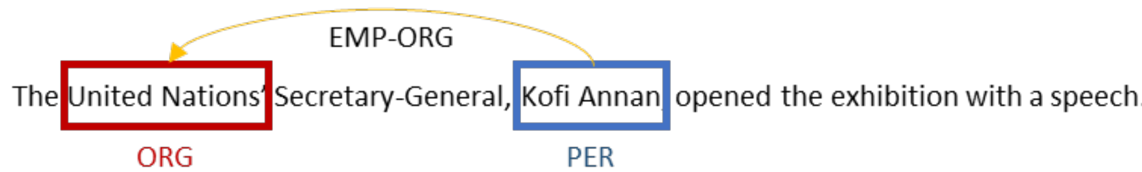
Overall model architecture + Adversarial training

- Idea: Adding worst case noise from the perspective of the loss

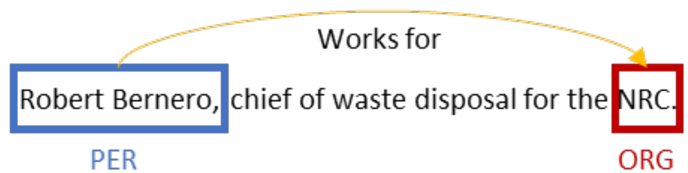


Experimental evaluation: Datasets

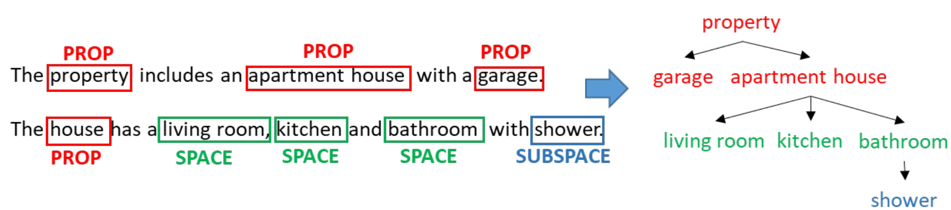
- **ACE04**
  (NER + relation extraction)



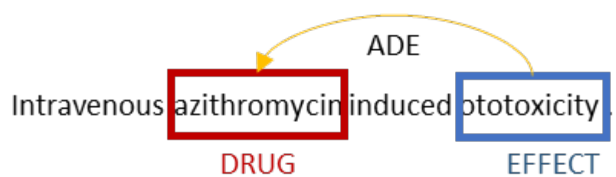
- **CoNLL04**
  (NER + relation extraction)



- **DREC**
  (real estate)



- **ADE**
  (adverse drug effects)



Experimental results

Performance close or better compared to feature based models

	Settings	Features	Entity	Relation	Overall F_1	
ACE 04	Miwa and Bansal (2016)	✓	81.80	48.40	65.10	
	Katiyar and Cardie (2017)	✗	79.60	45.70	62.65	$\simeq$
	baseline	✗	81.16	47.14	64.15	
	baseline + AT	✗	**81.64**	**47.45**	**64.54**	
CoNLL 04	Gupta et al. (2016)	✓	92.40	69.90	81.15	
	Gupta et al. (2016)	✗	88.80	58.30	73.60	
	Adel and Schütze (2017)	✗	82.10	62.50	72.30	$\simeq$
	baseline EC	✗	**93.26**	67.01	80.14	
	baseline EC + AT	✗	93.04	**67.99**	**80.51**	
	Miwa and Sasaki (2014)	✓	80.70	61.00	70.85	
	baseline	✗	83.04	61.04	72.04	$>$
	baseline + AT	✗	**83.61**	**61.95**	**72.78**	
DREC	Bekoulis et al. (2018)	✗	79.11	49.70	64.41	
	baseline	✗	82.30	52.81	67.56	
	baseline + AT	✗	**82.96**	**53.87**	**68.42**	
	baseline	✗	81.39	52.26	66.83	
	baseline + AT	✗	**82.04**	**53.12**	**67.58**	
ADE	Li et al. (2016)	✓	79.50	63.40	71.45	
	Li et al. (2017)	✓	84.60	71.40	78.00	$\gg$
	baseline	✗	86.40	74.58	80.49	
	baseline + AT	✗	**86.73**	**75.52**	**81.13**	

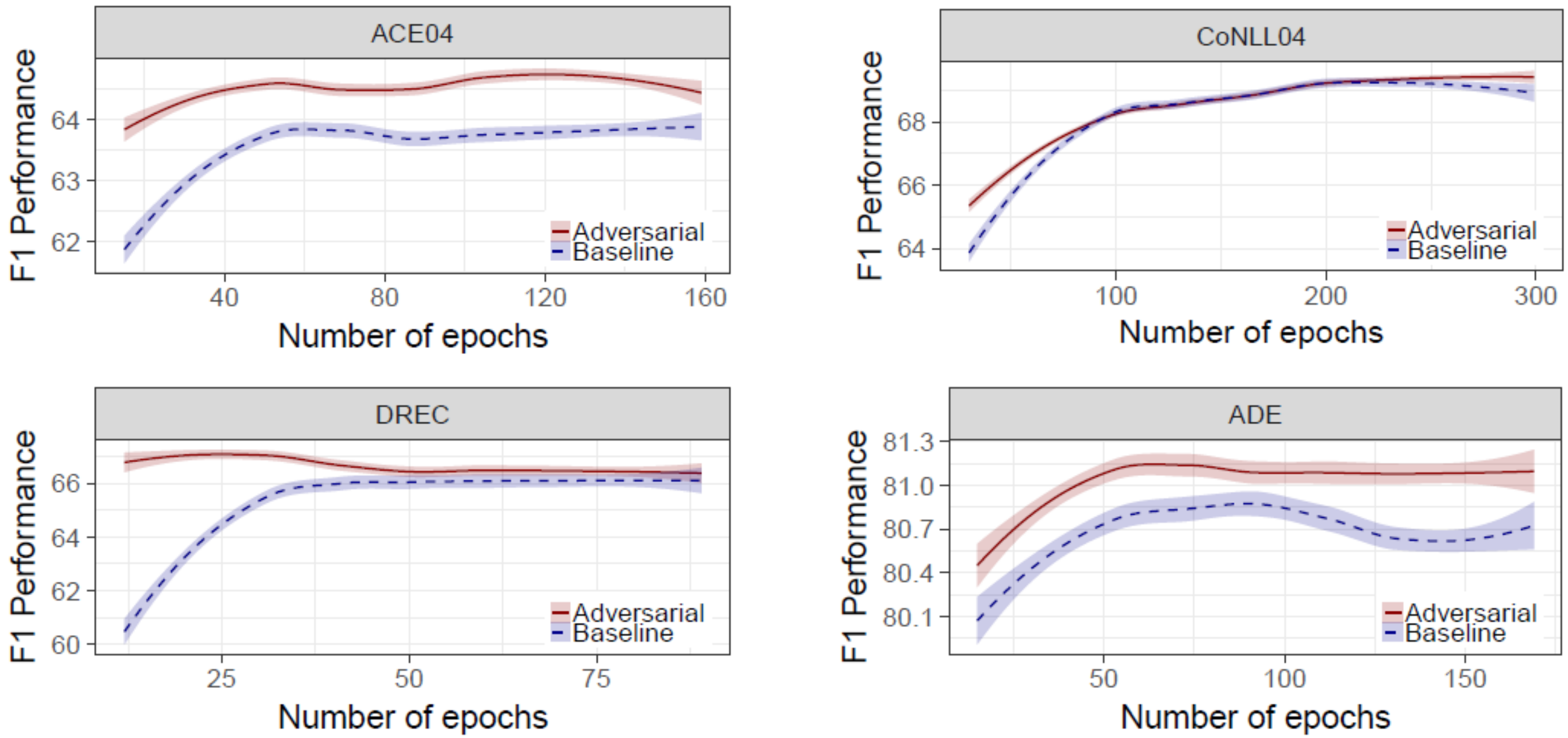
Experimental results

Improvement for both entities and relations

	Settings	Features	Entity ⇓	Relation ⇓	Overall F_1
ACE 04	Miwa and Bansal (2016)	✓	81.80	48.40	65.10
	Katiyar and Cardie (2017)	✗	79.60	45.70	62.65
	baseline	✗	81.16	47.14	64.15
	baseline + AT	✗	**81.64**	**47.45**	**64.54**
CoNLL 04	Gupta et al. (2016)	✓	92.40	69.90	81.15
	Gupta et al. (2016)	✗	88.80	58.30	73.60
	Adel and Schütze (2017)	✗	82.10	62.50	72.30
	baseline EC	✗	**93.26**	67.01	80.14
	baseline EC + AT	✗	93.04	**67.99**	**80.51**
	Miwa and Sasaki (2014)	✓	80.70	61.00	70.85
	baseline	✗	83.04	61.04	72.04
	baseline + AT	✗	**83.61**	**61.95**	**72.78**
DREC	Bekoulis et al. (2018)	✗	79.11	49.70	64.41
	baseline	✗	82.30	52.81	67.56
	baseline + AT	✗	**82.96**	**53.87**	**68.42**
	baseline	✗	81.39	52.26	66.83
	baseline + AT	✗	**82.04**	**53.12**	**67.58**
ADE	Li et al. (2016)	✓	79.50	63.40	71.45
	Li et al. (2017)	✓	84.60	71.40	78.00
	baseline	✗	86.40	74.58	80.49
	baseline + AT	✗	**86.73**	**75.52**	**81.13**

Experimental results

- AT outperforms the neural baseline model consistently across multiple and diverse datasets
- Improvement of AT depends on the dataset



GHENT UNIVERSITY

Conclusions on joint entity + relation extraction

- Proposed a **new joint model** that outperforms all previous methods that do not rely on external features or NLP tools
- Studied effectiveness of **adversarial training** as a regularization method over a multi-context baseline joint model
- Large scale experimental evaluation
- Improvement for each task (i.e., entity and relation extraction) separately, as well as the overall performance of the baseline joint model

PART II:
Automated lyrics annotation

L. Sterckx, J. Naradowsky, B. Byrne, T. Demeester and C. Develder, **"Break it down for me: A study in automated lyric annotation"**, in Proc. Conf. Empirical Methods in Natural Lang. Processing (EMNLP 2017), Copenhagen, Denmark, 7-11 Sep. 2017, pp. 2064-70.

L. Sterckx, J. Deleu, C. Develder and T. Demeester, **"Prior Attention for Style-aware Sequence-to-Sequence Models"**, arXiv preprint, Jun. 2018. https://arxiv.org/abs/1806.09439

Automated lyric annotation

Real G's Move in Silence Like Lasagna

When you pronounce lasagna, the G is silent.
Real gangsters (G's) move in silence too:
they keep their activities out of the spotlight.

Influence on language

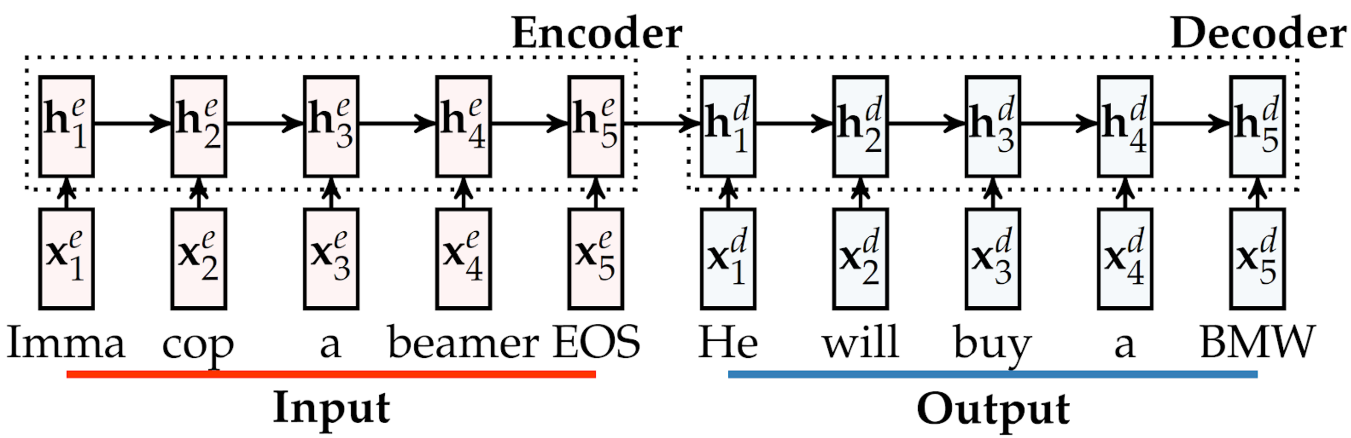
Popular

1	Codeine Dreaming (feat. Lil Wayne)	EXPLICIT	128,086,050
2	Sucker For Pain (with Wiz Khalifa, Imagine Dragons, Logic &...	EXPLICIT	511,919,272
3	Love U Better (feat. Lil Wayne & The-Dream)	EXPLICIT	86,190,486
4	The Way I Are (Dance With Somebody) [feat. Lil Wayne]		112,010,793
5	6 Foot 7 Foot	EXPLICIT	155,839,469
6	A Milli	EXPLICIT	149,009,460
7	Love Me	EXPLICIT	164,835,871
8	Running Back (feat. Lil Wayne)	EXPLICIT	72,760,580
9	Forever	EXPLICIT	208,147,497
10	Lollipop	EXPLICIT	120,216,454

Neural Machine Translation

- Problem
  - NLP models are typically trained on “clean” data
  - How to deal with slang, street language?

- Solution
  - Normalize slang to formal English using Machine Translation



GHENT UNIVERSITY

Data

- **Problem**

  Machine translation systems require thousands of examples

- **Solution**

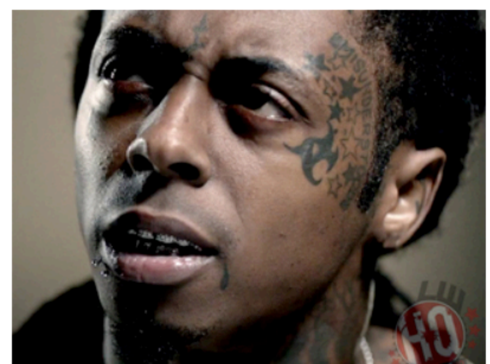
  genius.com - Platform for social annotation

GENIUS

[Hook]
Six-foot, seven-foot, eight-foot bunch
Six-foot, seven-foot, eight-foot bunch

[Verse 1: Lil Wayne]
Excuse my charisma, vodka with a spritzer
Swagger down pat; call my shit Patricia
Young Money militia and I am the commissioner
You no wan' start Weezy 'cause the 'F' is for finisher
So misunderstood, but what's a world without enigma?
Two bitches at the same time; synchronized swimmers
Got the girl twisted 'cause she open when you twist her

Wayne's out of this world persona is confusing to some, but he embraces the role of hip-hop's confusing enigma – it keeps things interesting. He even got "misunderstood" tattooed on his face!



GHENT UNIVERSITY

Machine Translation Baselines

Lyric:	`and when I slept, I dream g's, son I need some`
Human:	he dreams about large amounts of money because he is in need of it .
SMT:	and when he slept, and he's dream, g's son, he needs some
Seq2Seq:	when he sleeps , he dreams of getting money.
Retrieval:	g's refer to the ghetto gangsters in london that the kid has become involved with
Lyric:	`Mossberg pump I'm ridin' shotgun literally`
Human:	riding shotgun means to ride up front in the passenger seat
SMT:	Mossberg pump - action shotgun, he's saying he's ridin' shotgun literally.
Seq2Seq:	Mossberg is a firearms manufacturer. He is riding shotgun with his shotgun.
Retrieval:	he fully respects the people mentioned earlier and is riding for them.
Lyric:	`and my eyes red cause of all that haze`
Human:	his eyes turn red due to lack of hydration from the weed one smokes (in this case , smoking purple haze)
SMT:	and his eyes are red because of all that haze.
Seq2Seq:	his eyes are red because of all the weed he smokes.
Retrieval:	his eyes are red as a sign of the fatigue of travel and , possibly , from drugs and alcohol .

Evaluation

- Data:

# Lyric Annotation pairs	803,720
# Lyric tokens	13,677,332
# Annotation tokens	38,255,671
⊘ Tokens per Lyric	15
⊘ Tokens per Annotation	43
$\lvert V_{\text{lyrics}} \rvert$	124,022
$\lvert V_{\text{annot}} \rvert$	260,427

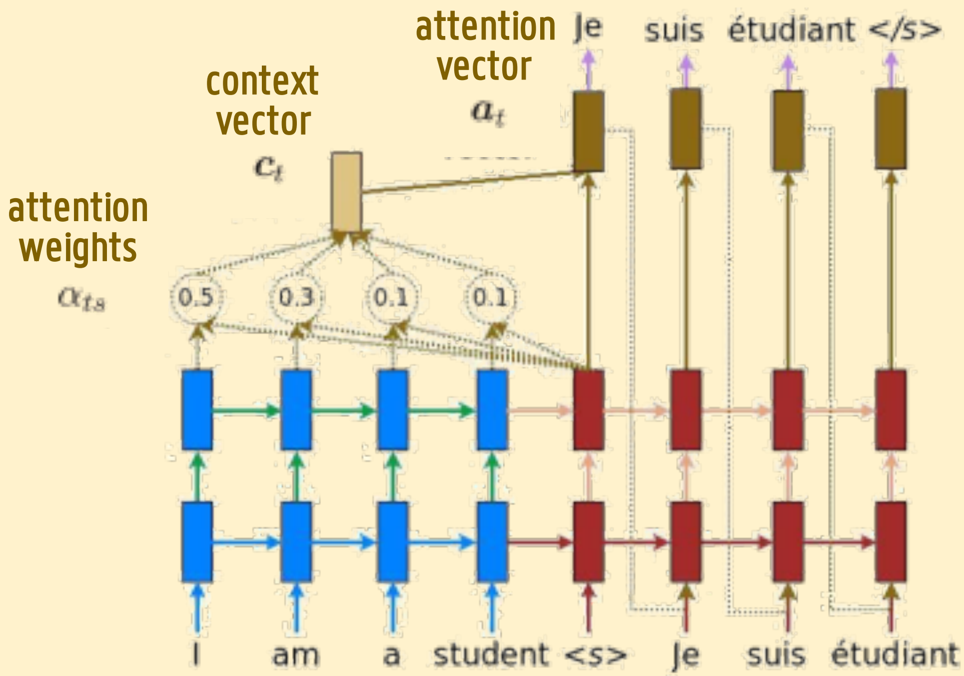
Components:
- **Precision**: fraction of translation words correct?
- **Recall**: how many target words matched?
- **Penalty**: mismatch in word order, length …

Higher score = better

- Results:

	Properties		Automated Evaluation				Human Evaluation	
	Length Ratio	Profanity/Tok.	BLEU	iBLEU	METEOR	SARI	Fluency	Information
Human	1.19	0.0027	-	-	-	-	3.93	3.53
SMT (Sent.)	1.23	0.0068	6.22	1.44	12.20	38.42	3.82	3.31
Seq2Seq (Sent.)	1.05	0.0023	5.33	3.64	9.28	36.52	3.76	3.25
Seq2Seq	1.32	0.0022	5.15	3.46	10.56	36.86	3.83	3.34
Retrieval	1.18	0.0038	2.82	2.27	5.10	32.76	3.93	2.98

INTERMEZZO: Attention



$$\alpha_{ts} = \frac{\exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)}$$

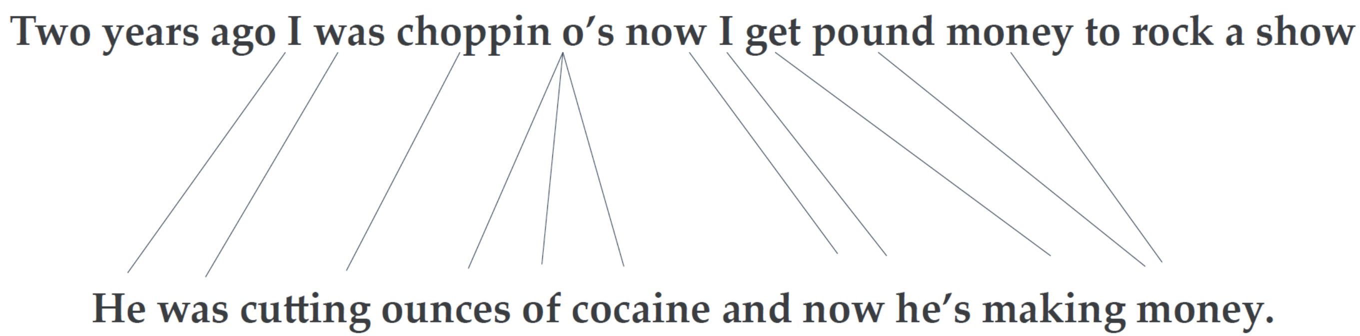
$$\boldsymbol{c}_t = \sum_s \alpha_{ts} \bar{\boldsymbol{h}}_s$$

$$\boldsymbol{a}_t = f(\boldsymbol{c}_t, \boldsymbol{h}_t) = \tanh(\boldsymbol{W}_c[\boldsymbol{c}_t; \boldsymbol{h}_t])$$

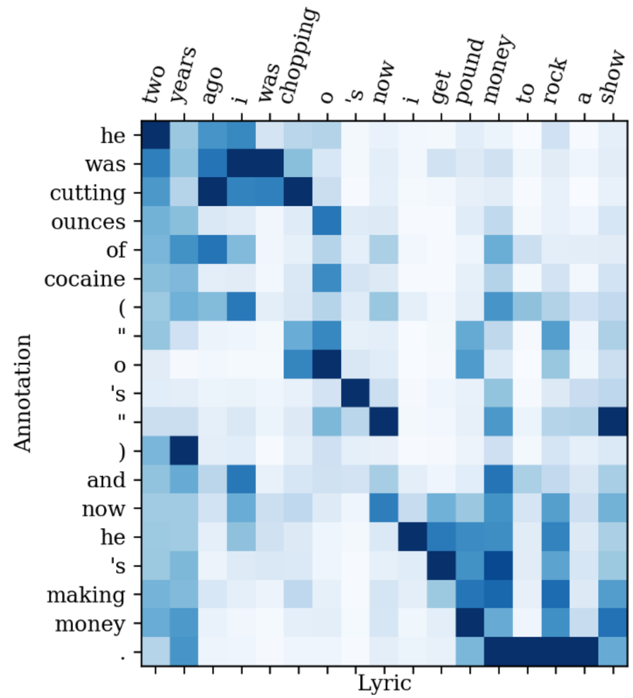
$$\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \boldsymbol{W} \bar{\boldsymbol{h}}_s \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W}_1 \boldsymbol{h}_t + \boldsymbol{W}_2 \bar{\boldsymbol{h}}_s\right) \end{cases}$$

Source: https://www.tensorflow.org/alpha/tutorials/text/nmt_with_attention

Alignment



Two years ago I was choppin o's now I get pound money to rock a show
He was cutting ounces of cocaine and now he's making money.



two
years
ago
i
was
chopping
o
's
now
i
get
pound
money
to
rock
a
show
he
was
cutting
ounces
of
cocaine
(
"
o
's
"
)
and
now
he
's
making
money
.
Annotation
Lyric

PART III:
Explaining character-aware neural networks for word-level prediction

F. Godin, K. Demuynck, J. Dambre, W. De Neve and T. Demeester, **"Explaining character-aware neural networks for word-level prediction: Do they discover linguistic rules?",** in Proc. Conf. Empirical Methods in Natural Lang. Processing (EMNLP 2018), Brussels, Belgium, 31 Oct. - 4 Nov. 2018.

GHENT UNIVERSITY

Word-level prediction tasks?

- **Morphological tagging**: predict morphological labels for a word (gender, tense, singular/plural, ...)

económicas → lemma = económico
económicas → gender = feminine
económicas → number = plural

- Manual annotations available for subset of words

económicas → lemma = económico
económicas → gender = feminine
económicas → number = plural

Interpretability

E.g., Brill's transformation-based error-driven tagger*:

Template	➡	**Rule**
Change the most-likely tag **X** to **Y** if the last (1,2,3,4) characters of the word are **x**		Change the tag **common noun** to **plural common noun** if the word has suffix **-s**

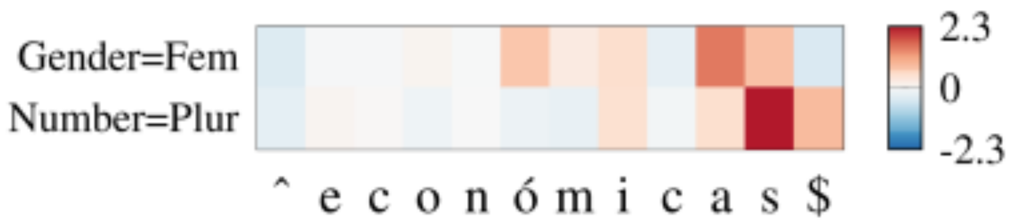
- Rule-based / tree-based systems
  - ⟹ Transparent: follow the trace!
- Shallow statistical models (e.g., logistic regression, CRFs…)
  - ⟹ Essentially: __**weights**__ x features
- Neural network models
  - ⟹ .

*: E. Brill, "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging", *Computational linguistics*, *21*(4), 543-565.

Proposed method

- We present contextual decomposition (CD) for CNNs
  - Extends CD for LSTMs (Murdoch et al. 2018)
  - White box approach to interpretability

- We trace back morphological tagging decisions to the character-level
  Research questions:
  - Which characters are important?
  - Same patterns as linguistically known?
  - Difference between CNN and BiLSTM?



Up next

- Contextual decomposition for CNNs
  - Concept of CD
  - CD applied to CNNs = convolution + pooling + classification layer

- Experiments
  - Datasets
  - Architectures: CNN vs BiLSTM
  - **Q1**: Visualization of character contributions?
  - **Q2**: Agreement with manual (expert) segmentation?
  - **Q3**: Which patterns found? Compositions of patterns?

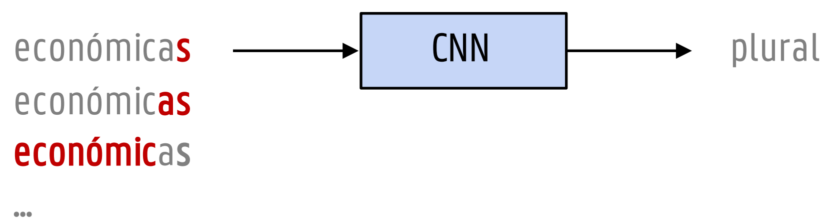
Contextual decomposition for CNNs

Contextual Decomposition (CD)

Idea: Every output value can be "decomposed" in

- **Relevant** contributions originating from the input we are interested in (e.g., some specific characters)
- Irrelevant contributions originating from all the other inputs (e.g., all other characters in a word)

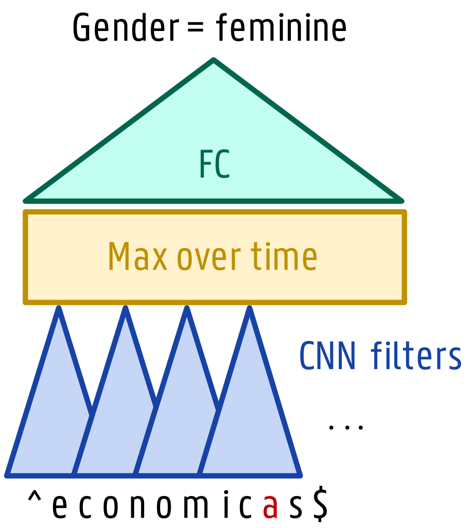
$$z = \underbrace{\beta}_{\text{relevant}} + \underbrace{\gamma}_{\text{irrelevant}}$$



Contextual Decomposition for CNNs

- Three main components of CNN
  - Convolution
  - Activation function
  - Max-over-time pooling

- Classification layer



Contextual Decomposition for CNNs: Convolution

Output of single convolutional filter at timestep t :

$$z_t = \sum_{i=0}^{n-1} W_i \cdot \boldsymbol{x}_{t+i} + b$$

$$\beta_t = \sum_{i=0}^{n-1} W_i \cdot \boldsymbol{x}_{t+i} \qquad (t+i) \in \boldsymbol{S}$$

Relevant 9

$+$

$$\gamma_t = \sum_{i=0}^{n-1} W_i \cdot \boldsymbol{x}_{t+i} \qquad (t+i) \notin \boldsymbol{S}$$

Irrelevant 8, 10, 11

$+ \quad b$

n = filter size
S = Indexes of of relevant inputs
W_i = i^{th} column of filter W

Contextual Decomposition for CNNs: Activation function

- **Goal:** linearize activation function to split output

$$
\begin{aligned}
c_t =& f_{ReLU}(z_t) \\
=& f_{ReLU}(\beta_{z,t} + \gamma_{z,t} + b) \\
=& L_{ReLU}(\beta_{z,t}) \\
& + [L_{ReLU}(\gamma_{z,t}) + L_{ReLU}(b)] \\
=& \beta_{c,t} + \gamma_{c,t}
\end{aligned}
$$

- **Linearization formula**:

$$
L_f(y_k) = \frac{1}{M_N} \sum_{i=1}^{M_N} [f(\sum_{l=1}^{\pi_i^{-1}(k)} y_{\pi_i(l)}) - f(\sum_{l=1}^{\pi_i^{-1}(k)-1} y_{\pi_i(l)})]
$$

Average over all possible component orderings

Function of the first terms up to and <u>including</u> y_k

Function of the first terms up to (but <u>excluding</u>) y_k

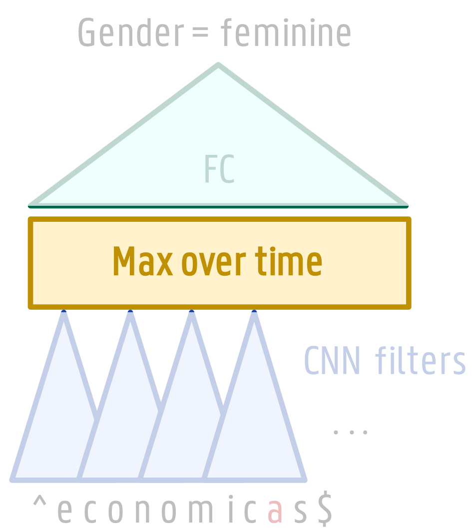
Contextual Decomposition for CNNs: Max pooling

- Max-over-time pooling:

$$c = \max_t (c_t)$$

- Determine t and split that particular instance

$$\beta_c + \gamma_c = \max_t (\beta_{c,t} + \gamma_{c,t})$$



Contextual Decomposition for CNNs: Classification layer

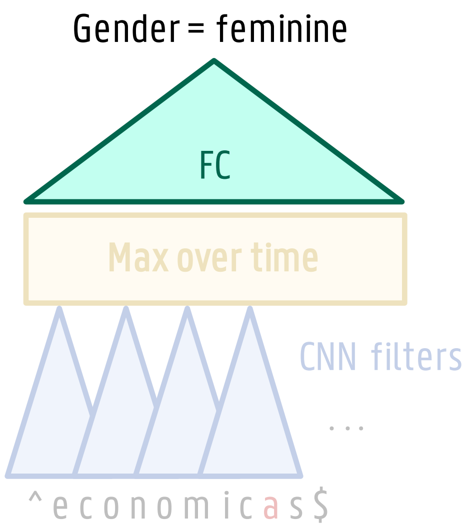
- Probability of certain class from softmax-layer:

$$p_j = \frac{e^{W_j \cdot \boldsymbol{x} + b_j}}{\sum_{i=1}^{C} e^{W_i \cdot \boldsymbol{x} + b_i}}$$

- Simplify to linear part, i.e., weight matrix:

$$W_j \cdot \boldsymbol{x} + b_j = \boxed{W_j \cdot \boldsymbol{\beta}} + W_j \cdot \boldsymbol{\gamma} + b_j$$

Relevant contribution to class j



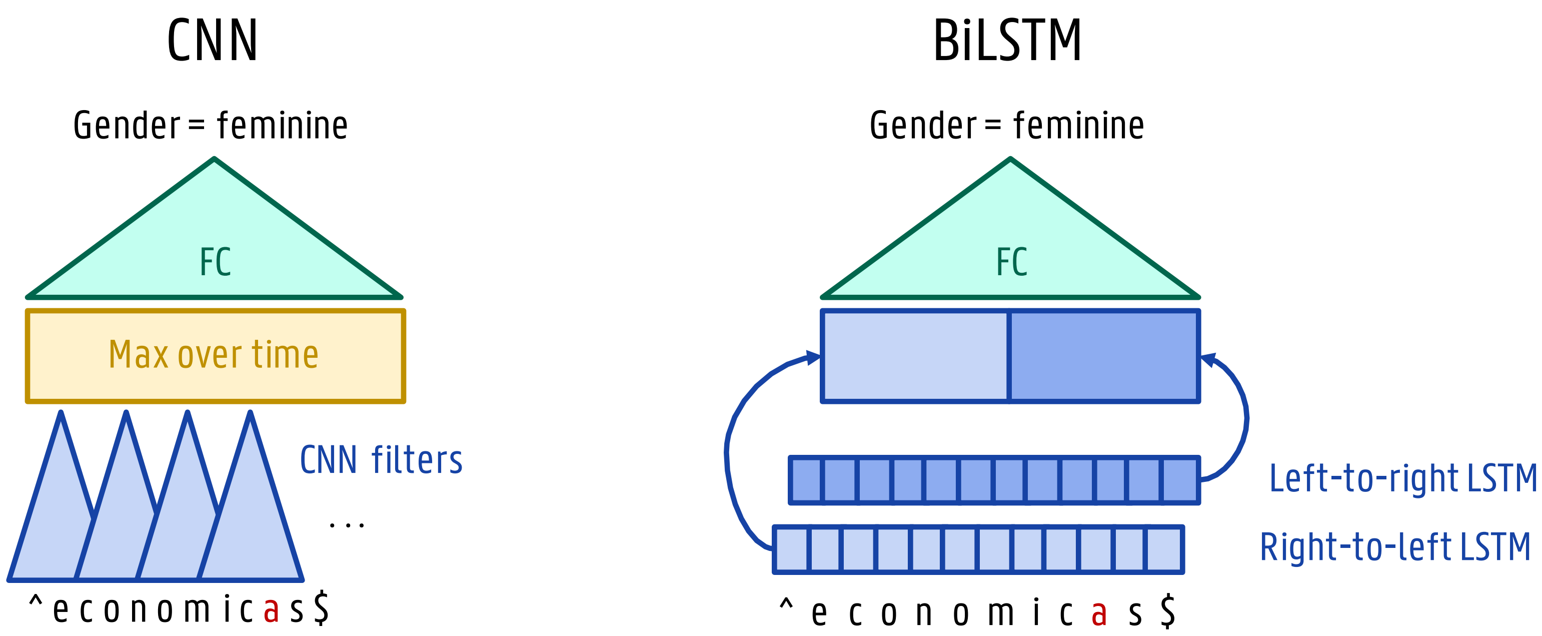
Experiments

Datasets

- Universal dependencies 1.4:
  - Finnish, Spanish and Swedish
  - Select all unique words and their morphological labels

- Manual annotations and segmentations of 300 test set words

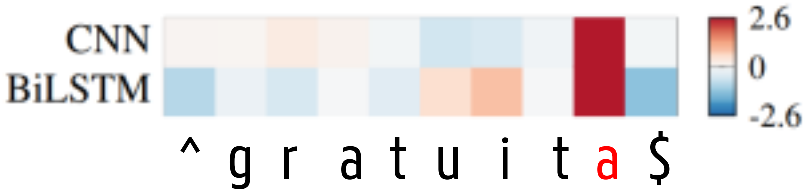
Architectures: CNN vs BiLSTM



GHENT UNIVERSITY

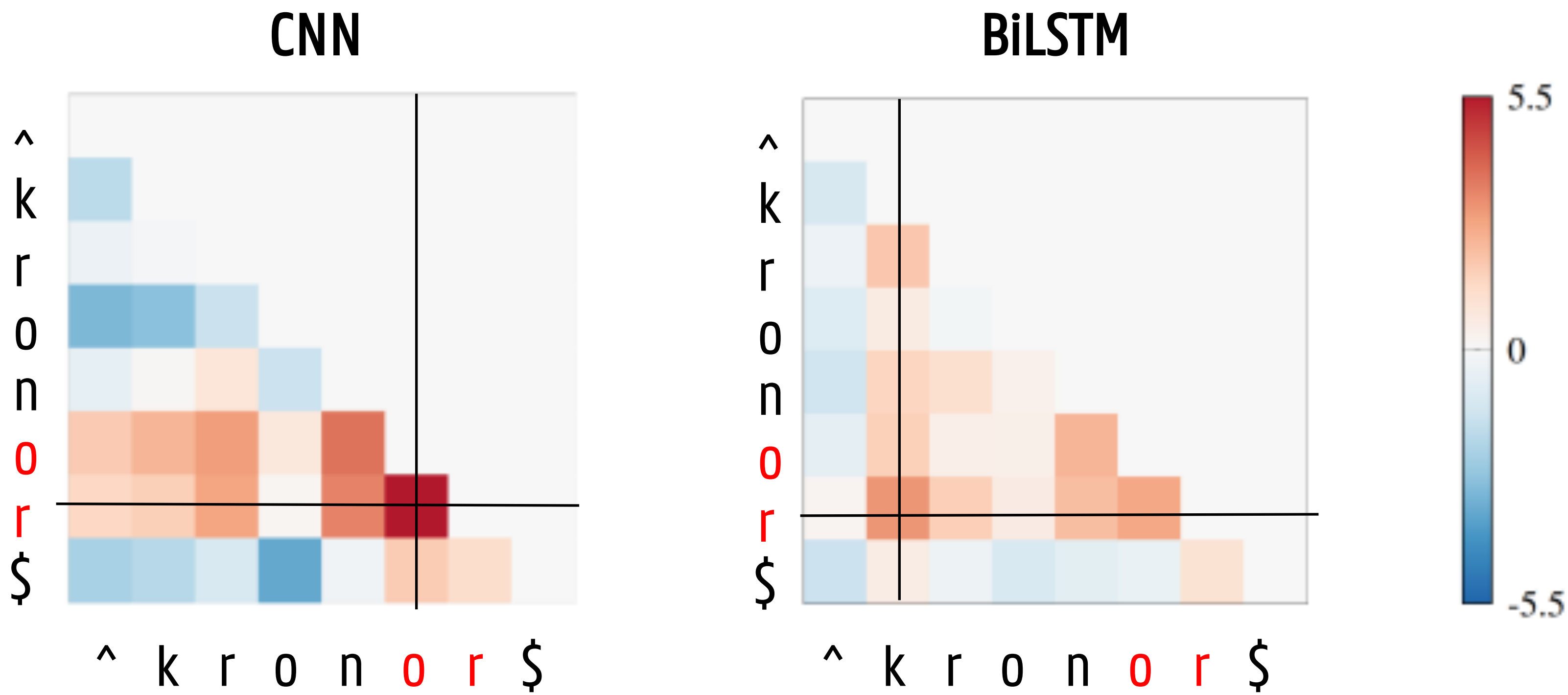
Q1: Visualization of character contributions?

Spanish



Label: Gender = feminine

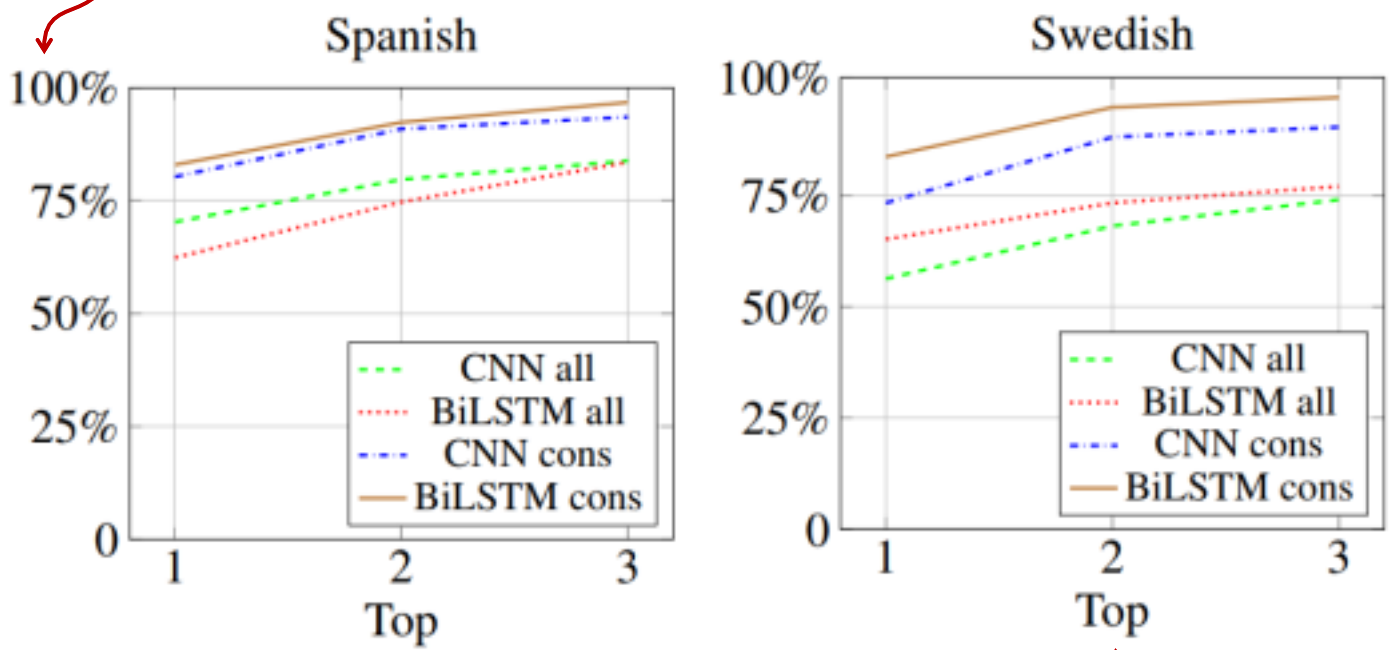
Q1: Visualization of character contributions?



Label: number = plural

Q2: Agreement with manual (expert) segmentation?

% test words for which gold label is
among top-k sequences



Top-k character
sequences considered

all = every possible combination of characters
cons = all consecutive character n-grams

Q3: Which patterns found? Compositions of patterns?

Spanish:

- Linguistic rules for feminine gender:
  - Feminine adjectives often end with “a”
  - Nouns ending with “dad” or “ión” are often feminine
- Patterns found:
  - “a” is a very important pattern
  - “dad” and “sió” are import trigrams

		One character	Two characters	Three characters	Examples
Spanish Gend=Fem	BiL.	a (69%), i (16%), d (6%), e (4%)	as (23%), a$ (13%), ad (7%), ia (5%)	ia$ (4%), ad$ (3%), da$ (3%), ca$ (2%)	toleranc**ia**, ciud**ad**
	CNN	a (77%), ó (14%), n (4%), d (3%)	a$ (34%), as (20%), da (8%), ió (7%)	dad (5%), da$ (4%), a_ió (4%), sió (2%)	firm**as**, precis**ión**

GHENT UNIVERSITY

Q3: Which patterns found? Compositions of patterns?

Swedish:

- Linguistic rules for feminine gender:
  - 5 suffixes: or, ar, (e)r, n, and no ending

- Patterns found:
  - “or” and “ar”
  - But also “na” and “rn” → “na” is definite article in plural forms

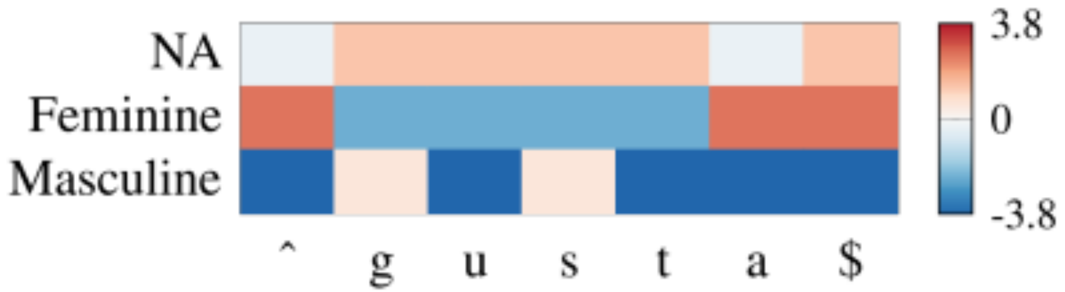
			One character	Two characters	Three characters	Examples
Swedish	Numb=Plur	BiL.	n (25%), r (19%), a (14%), g (7%)	na (13%), a__r (4%), or (3%), n__r (3%)	iga (5%), rna (3%), ner (1%), der (1%)	kron**or**, perio**der**
		CNN	n (21%), a (18%), r (15%), d (5%)	rn (8%), na (5%), or (4%), er (3%)	rna (7%), arn (3%), iga (2%), n_ar (2%)	kraftе**rna**, sak**er**

GHENT UNIVERSITY

Q3: Which patterns found? Compositions of patterns?

- How do positive and negative patterns interact?
  Consider the Spanish verb “gusta”
  - Gender = Not Applicable (NA)
  - We know that suffix “a” is indicator for gender=feminine

- Consider most positive/negative set of characters per class:



⟹ Stem provides counter-evidence for “gender = feminine”

Wrap-up

- We introduced a white box approach to understanding CNNs

- We showed that:
  - BiLSTMs and CNNs sometimes choose different patterns
  - The learned patterns coincide with our linguistic knowledge
  - Sometimes other plausible patterns are used

PART IV:
Predefined sparseness in recurrent sequence models

T. Demeester, J. Deleu, F. Godin and C. Develder, **"Predefined sparseness in recurrent sequence models"**, in Proc. SIGNLL Conf. Comput. Natural Language Learning (CoNLL 2018), Brussels, Belgium, 31 Oct. - 1 Nov. 2018.

Getting more out of big data …

"Big fat neural networks trained on huge amounts of data can solve everything"

https://imgs.xkcd.com/comics/machine_learning.png

Getting more out of big data …

"Big fat neural networks trained on huge amounts of data can solve everything"

… or should we rather

- Do more with <u>less</u> data
- Do the same with <u>smaller</u> models

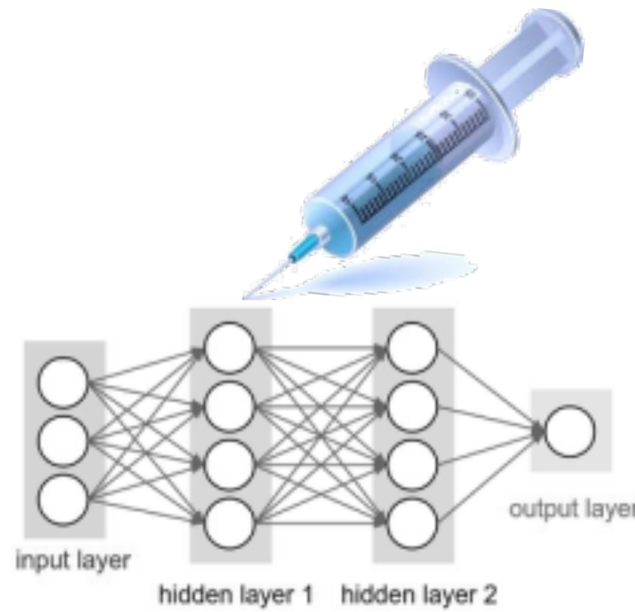
"We choose to tackle problems that no one else can"
– Sander Dieleman, Deepmind

- Dozens of GPU cores in parallel ← OK
- Training takes over a week ← OK
- Lots of hyperparameter tuning ← **NOT OK**

Getting more out of big data …

Recent trends towards **"injecting" extra knowledge**

- **Implicitly**: pretraining on large datasets
  - Howard, Ruder. "**ULMFiT**: Universal language model fine-tuning for text classification". ACL 2018.
  - Clark, Lee, Zettlemoyer. **ELMO**: "Deep contextualized word representations". NAACL 2018.
  - Devlin, Chang, Lee, Toutanova. **BERT**: "Pre-training of deep bidirectional transformers for language understanding".



- **Explicitly**: e.g., logical rules, reasoning tools
  - Demeester, Rocktaschel, Riedel. "Lifted rule injection for relation embeddings". EMNLP 2016
  - Minervini, Demeester, Rocktäschel, Riedel. "Adversarial sets for regularising neural link predictors". UAI 2017.
  - Manhaeve, Dumančić, Kimmig, Demeester, De Raedt. **DeepProbLog**: "Neural probabilistic logic programming". NeurIPS 2018

Getting more out of big data …

Two complementary approaches:

1. Do more with less

2. Insert “knowledge”

Getting more out of big data …

Two complementary approaches:

1. Do more with less
2. Insert “knowledge”

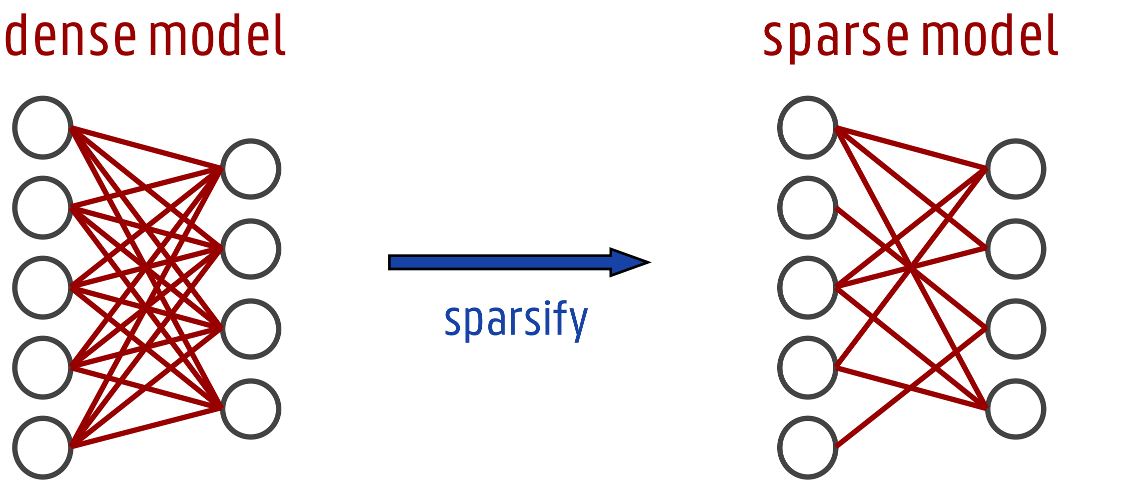
T. Demeester, J. Deleu, F. Godin and C. Develder, **“Predefined sparseness in recurrent sequence models”**, in Proc. SIGNLL Conf. Comput. Natural Language Learning (CoNLL 2018), Brussels, Belgium, 31 Oct. – 1 Nov. 2018.

Smaller models

For specific NLP applications

GHENT UNIVERSITY

Sparse neural networks



'smaller' model
(lower memory footprint)

Sparsifying by weight pruning

- ✔ Highly sparse with accuracy close to dense models [1]
- ✔ Large sparse networks can be better than small dense models [2]
- ✘ **<u>But then</u>**: large dense network needed during training!

⇒ **Goal:** Models that are sparse from the start, i.e., "predefined sparseness"

[1] Narang et al. "Exploring sparsity in RNNs" (ICLR 2017)
[2] Kalchbrenner et al. "Efficient neural audio synthesis" (ICML 2018)

Inspiration from literature

"Application of sparse coding in language processing is far from extensive, when compared to speech processing" [3]

$\Longrightarrow$ Need for sparse models in NLP!

"Natural language is high-rank" [4]

$\Longrightarrow$ How to train large sparse representations despite memory constraints?

[3] Wang et al. "Deep and sparse learning in speech and language processing: An overview" BICS 2016
[4] Yang et al. "Breaking the softmax bottleneck: A high-rank RNN language model." ICLR 2018

Predefined sparseness in NLP

Two experiments (many others are possible)

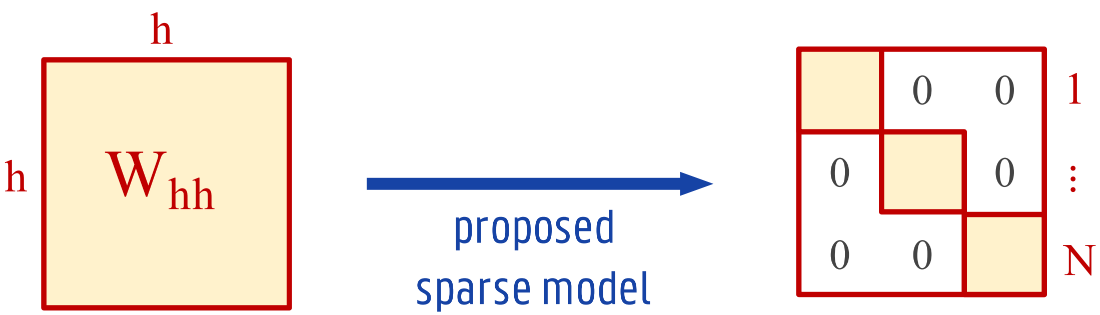
- Predefined sparseness in **recurrent sequence encoders**
- Predefined sparseness on the **word embedding** level

Predefined sparseness for RNNs

Predefined sparseness for RNNs

Any recurrent cell (RNN, LSTM, GRU, ...) has 2 types of matrices

- **Hidden-to-hidden:**

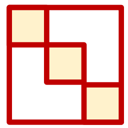
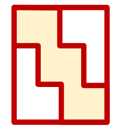


block-diagonal
(density 1/N)

- **Input-to-hidden:**

(density γ)

Predefined sparseness for RNNs

With sparse W_{hh}  and W_{hi} 

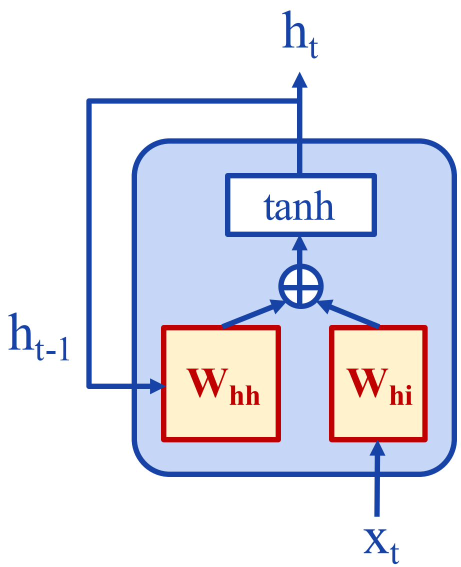
- the number of hidden-to-hidden interactions is strongly reduced (cf. weight dropping in W_{hh} [5])
- not all hidden dimensions have access to each input dimension

⇒ **Why** this particular choice?

[5] Merity et al. "Regularizing and optimizing LSTM language models." ICLR 2018.

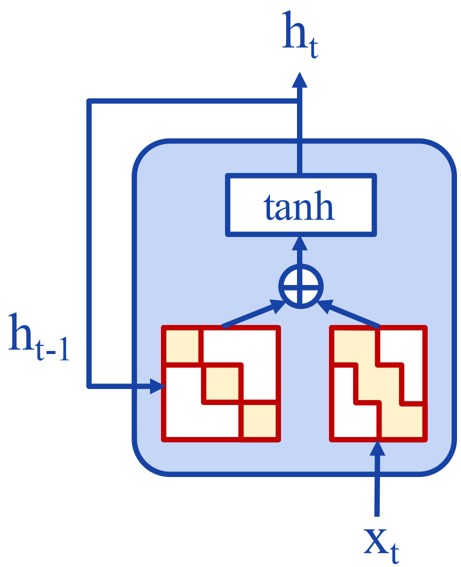
Motivation for chosen sparse matrices

Vanilla RNN:



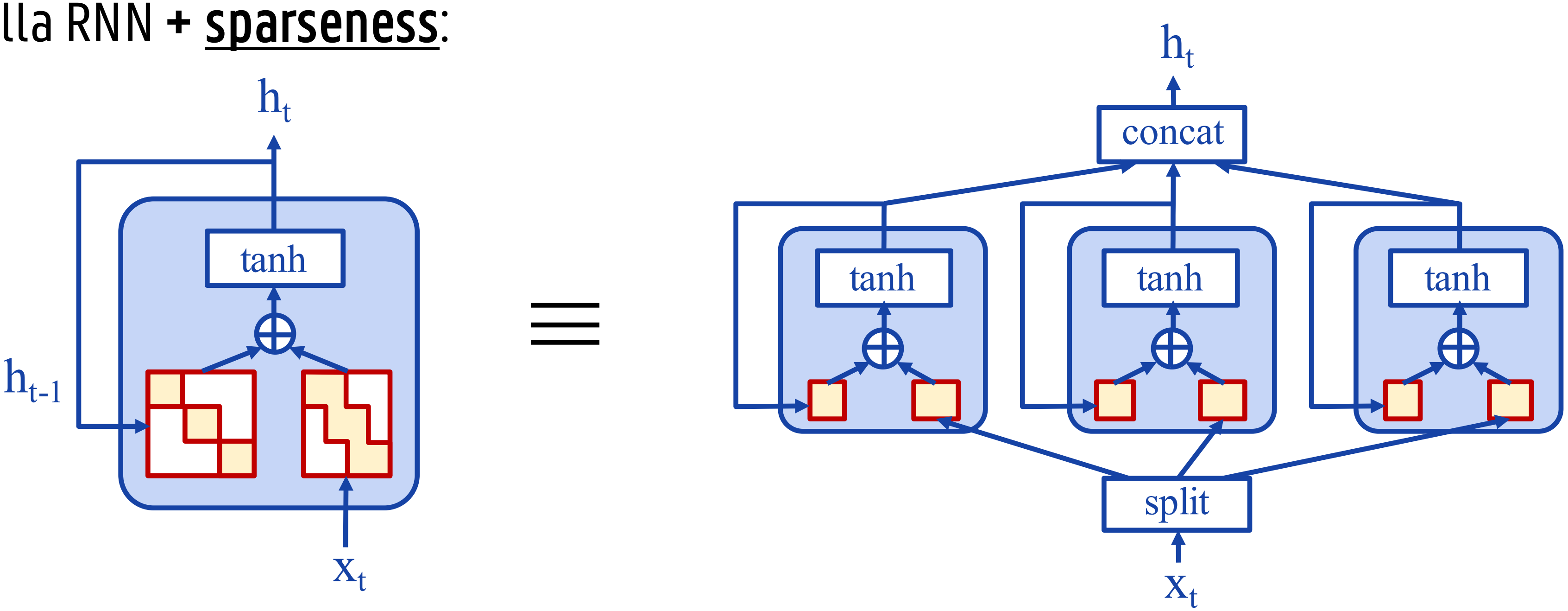
Motivation for chosen sparse matrices

Vanilla RNN + **sparseness**:



Motivation for chosen sparse matrices

Vanilla RNN + **sparseness**:



$\Rightarrow$ Resulting RNN is equivalent to N smaller dense RNNs in parallel

- Only possible with output divided into disjoint segments
- But input can be (partly) shared between components
- Holds for vanilla RNN, LSTM, GRU,...
- Allows using standard tools (CuDNN)

Language modeling with sparse LSTM

- **Baseline**: AWD-LSTM model [5] with 3-layer stacked LSTM

- **Sparse counterpart**:
  - Middle LSTM hidden size x 1.5 (from 1150 to 1725)
  - Sparse; same number of parameters
  - Same regularization settings

[5] Merity et al. "Regularizing and optimizing LSTM language models." ICLR 2018.

Language modeling with sparse LSTM

- First train run (500 epochs):

Model	Penn Treebank test perplexity
reported [5]	58.8
baseline	58.8 ± 0.3
sparse LSTM	57.9 ± 0.3

- Further training ("finetune"): Sparse model overfits

[5] Merity et al. "Regularizing and optimizing LSTM language models." ICLR 2018.

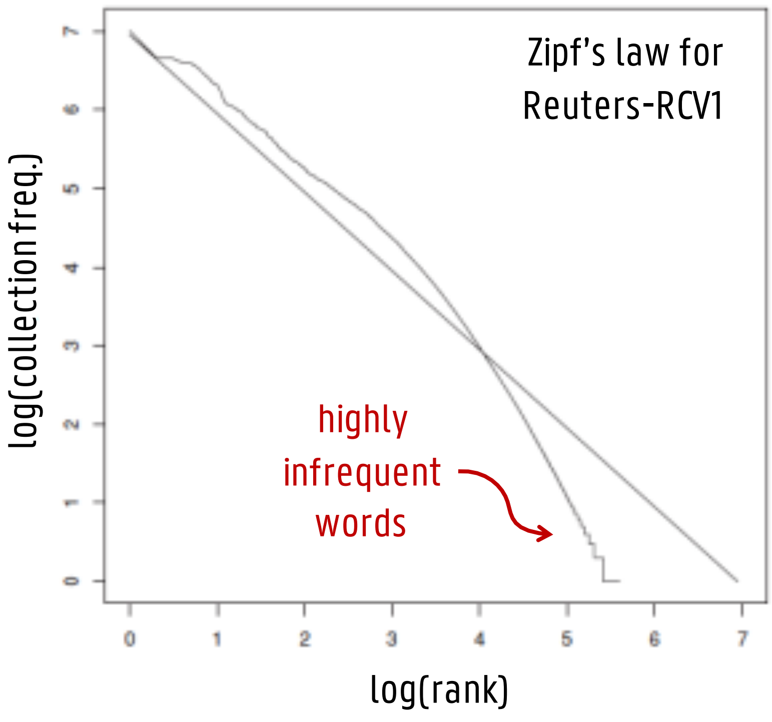
Predefined sparseness in word embeddings

Predefined sparseness in word embeddings

- **Goal:**
  decide upfront which entries in embedding matrix $E \in \mathbb{R}^{V \times k}$ are 0

- **Observation:**
  word occurrence frequencies follow Zipf's law

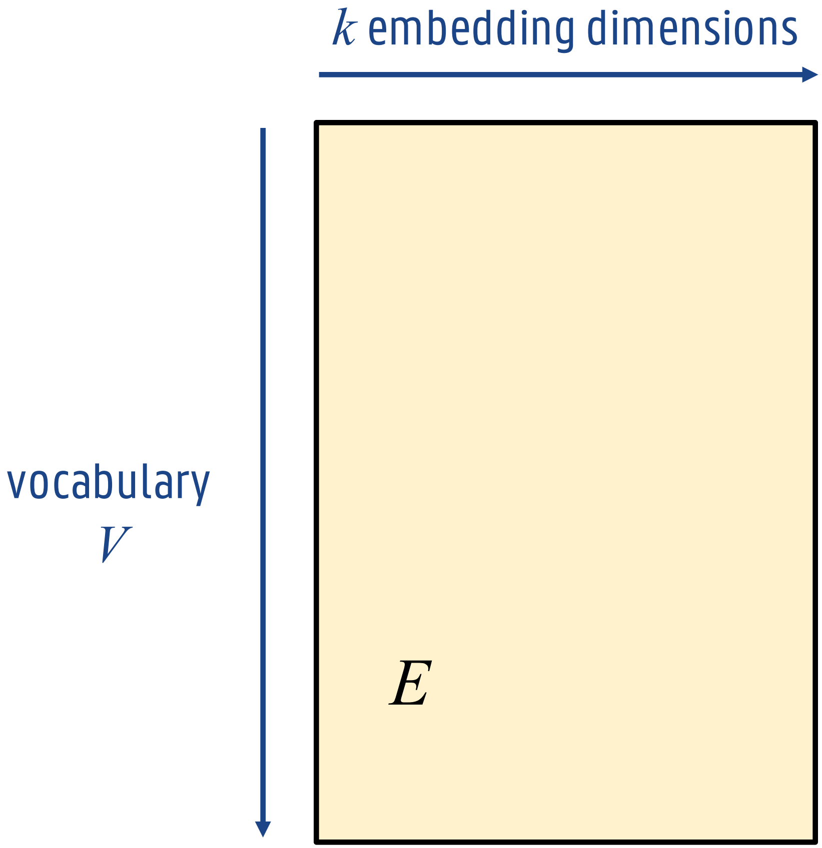
$\Rightarrow$ Representing long tail of **rare terms** with **short embeddings** would greatly reduce memory requirements (to store non-zero elements)



source: Manning, Schütze, Raghavan, "Introduction to Information Retrieval", Cambridge UP, 2009

Predefined sparseness in word embeddings

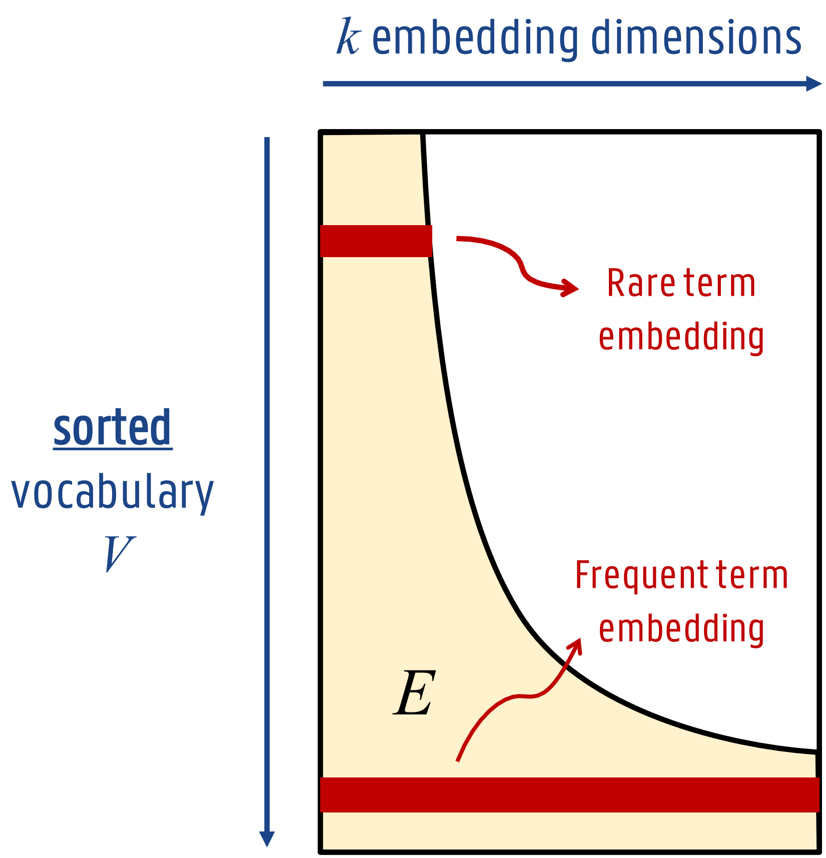
- Predefined sparse embedding matrix E ?



trainable parameters = $k\ V$

Predefined sparseness in word embeddings

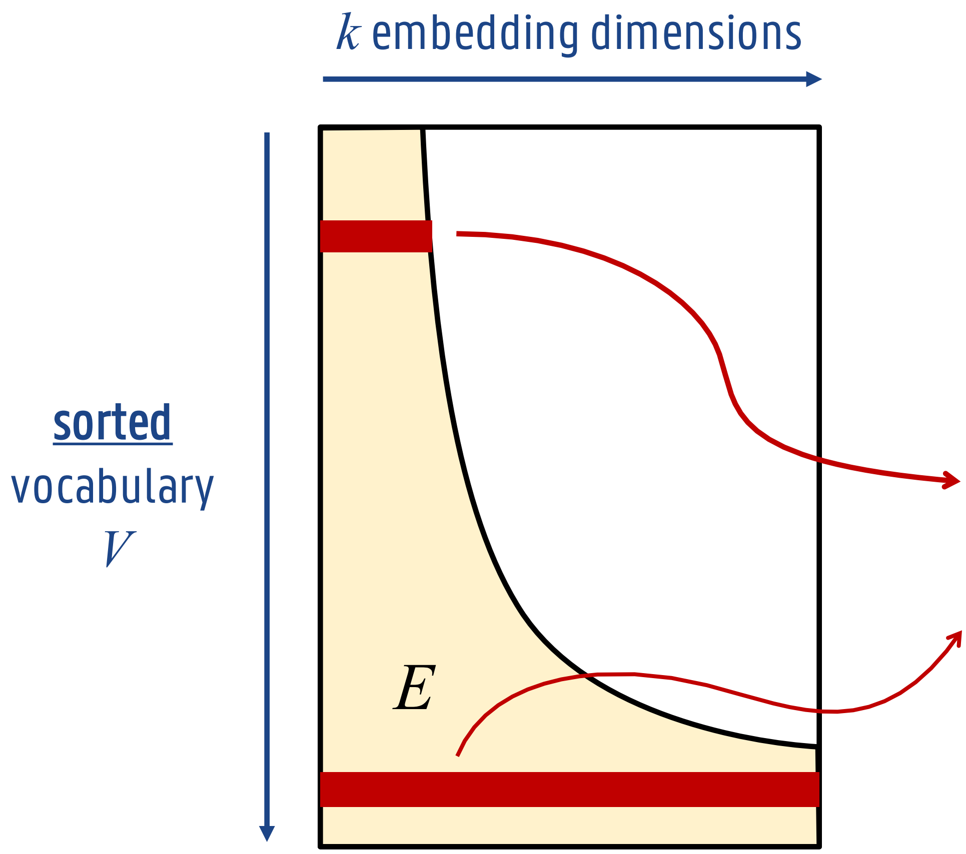
- Predefined sparse embedding matrix E ?



\# trainable parameters = $k\ V\ \delta_E$

Predefined sparseness in word embeddings

- Predefined sparse embedding matrix E ?



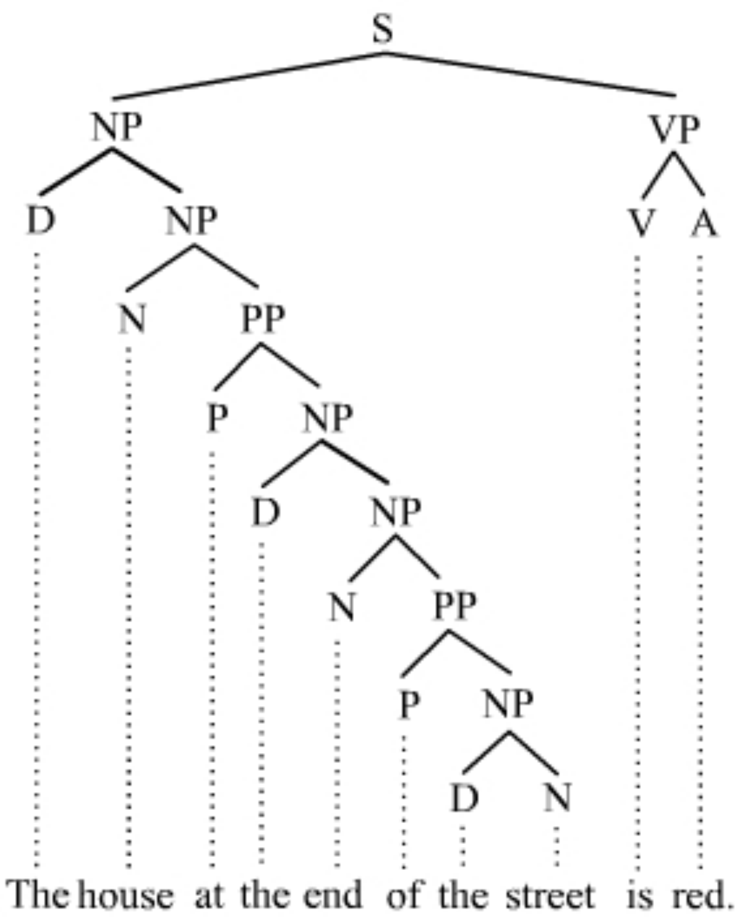
trainable parameters = $k\ V\ \delta_E$

Sparse embedding space

- 'first' dimensions model many rare terms
- remaining dimensions model few frequent terms

Predefined sparseness in word embeddings

- Experimental setup:
  - POS tagging on Penn Treebank
  - Very small model (else too easy!)
  - 20-D word embeddings (876k params)
  - BiLSTM state size 10+10 (3k params)

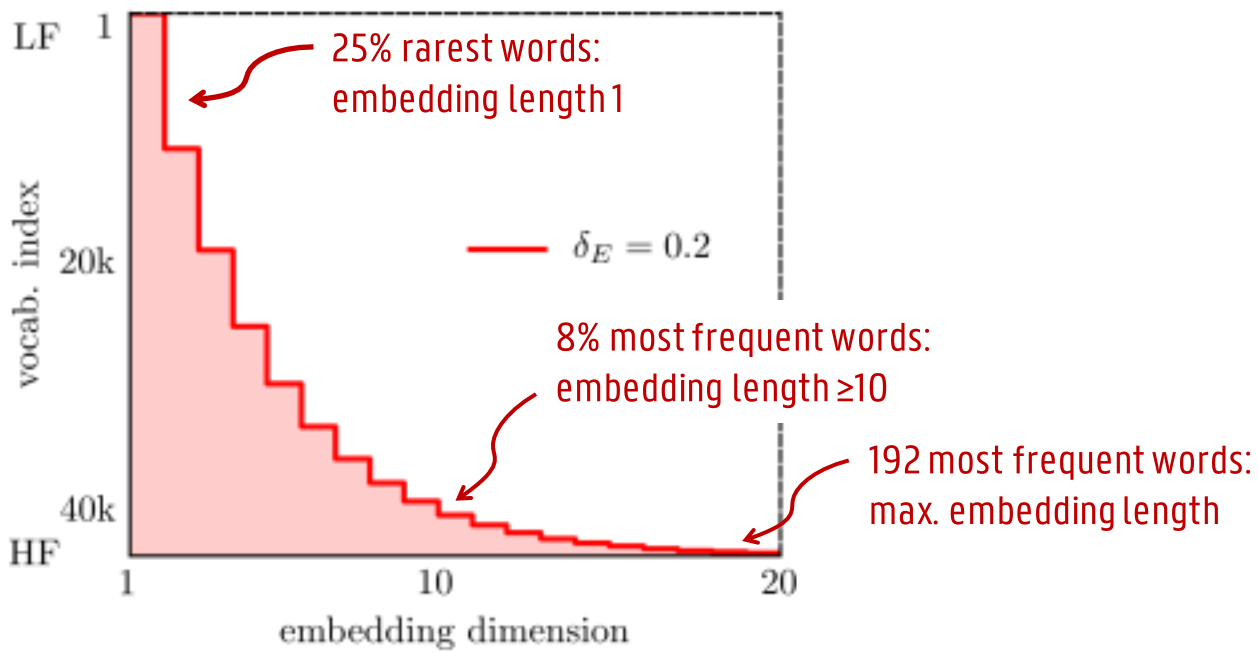


NP = noun phrase	VP = verb phrase	PP = prepositional phrase
N = noun	V = verb	P = preposition
D = determiner	A = adjective	

GHENT UNIVERSITY

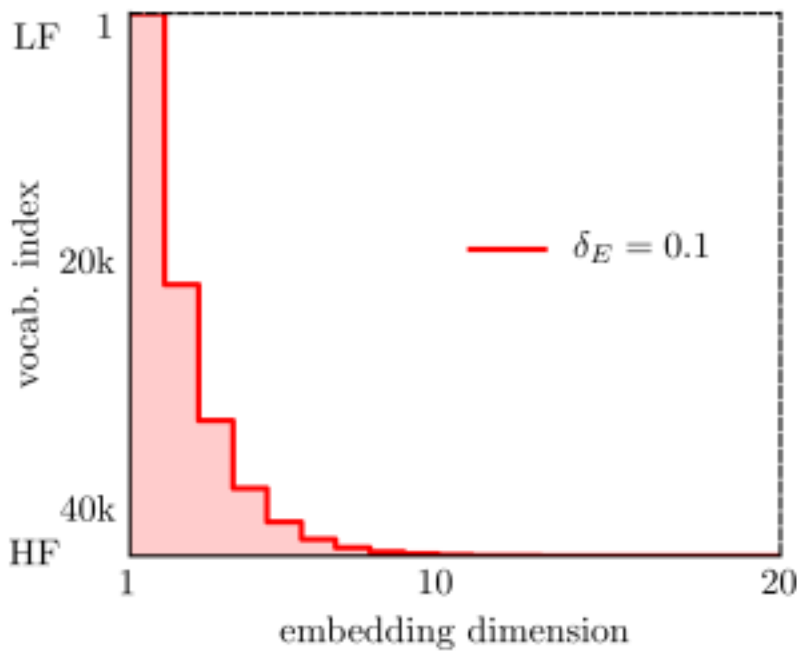
Predefined sparseness in word embeddings

- Experimental setup:
  - POS tagging on Penn Treebank
  - Very small model (else too easy!)
  - 20-D word embeddings (876k params)
  - BiLSTM state size 10+10 (3k params)
- Embedding matrix:



Predefined sparseness in word embeddings

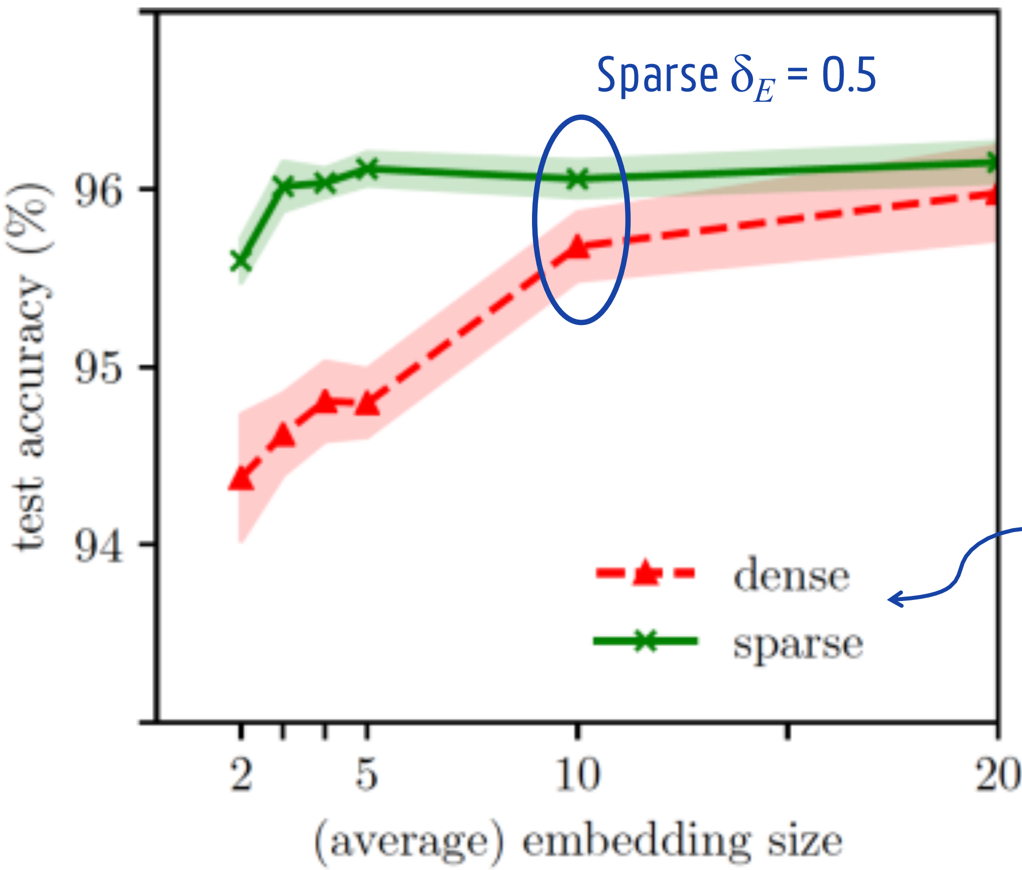
- Experimental setup:
  - POS tagging on Penn Treebank
  - Very small model (else too easy!)
  - 20-D word embeddings (876k params)
  - BiLSTM state size 10+10 (3k params)
- Embedding matrix:



GHENT UNIVERSITY

Predefined sparseness in word embeddings

Resulting POS tag accuracy:

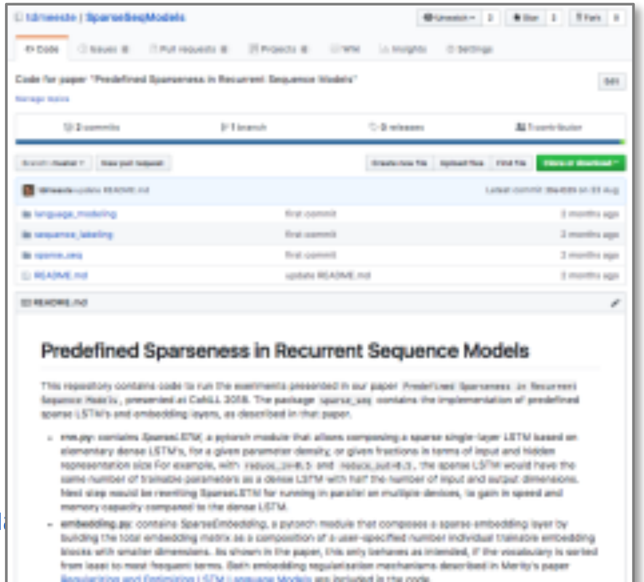


E has same number of trainable parameters for given x-value

Conclusions

- Simple ideas for **predefined sparseness in RNNs** and **embedding layers**
- Predefined Sparseness has **potential in NLP**
- But … further investigation needed
  (for very large representation sizes for large vocabularies, etc.)

Note: “predefined sparseness” code available
https://github.com/tdmeeste/SparseSeqModels



GHENT
UNIVERSITY

Wrap-up

Take-away messages

- Neural models for natural language:
  - (Sub)words can be represented in dense **embeddings**
  - Sentence = sequence of words, Word = sequence of letters
  - Process sequences using either **RNNs** or **CNNs**

- Sample applications:
  - Joint entity recognition and relation extraction → end-to-end model based on BiLSTMs
  - Automated lyrics annotation → example of seq2seq (or encoder/decoder) application
  - Explaining character-aware NNs for word-level prediction → provides explaination
  - Predefined sparseness in recurrent sequence models → decide on embedding structure

Thank you.
Any questions?

chris.develder@ugent.be

http://users.ugent.be/~cdvelder

https://ugentt2k.github.io