Neural Networks for Natural Language Processing

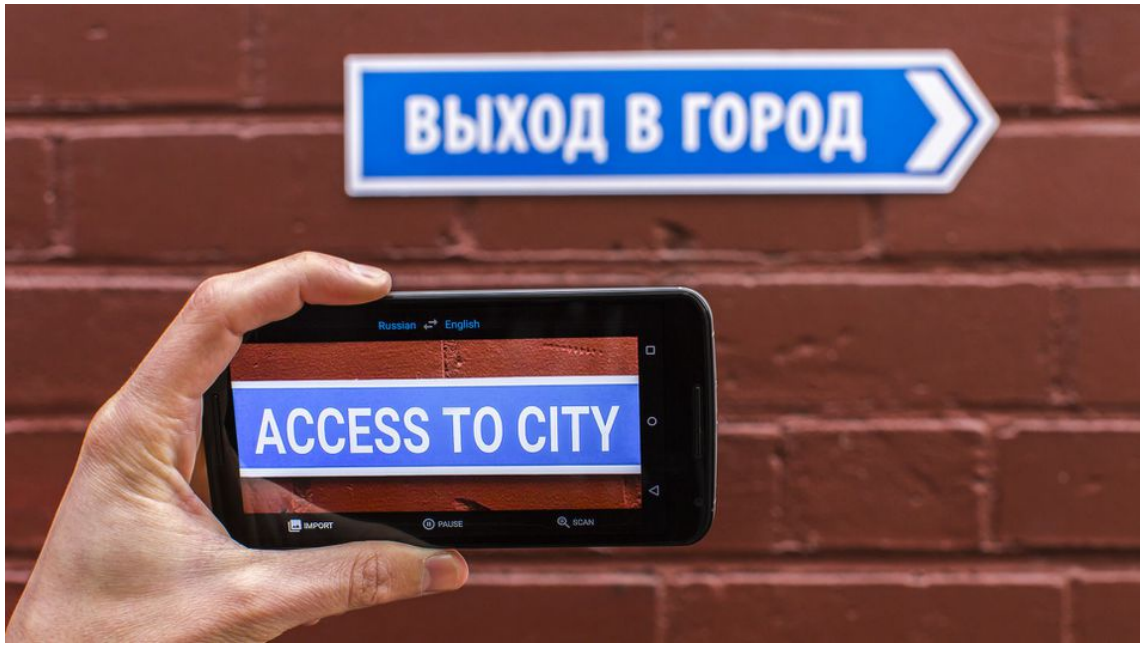
Alexandre Allauzen

Université Paris-Sud / LIMSI-CNRS

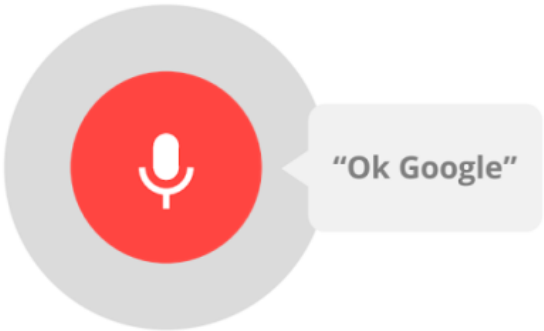
19/01/2017

Outline

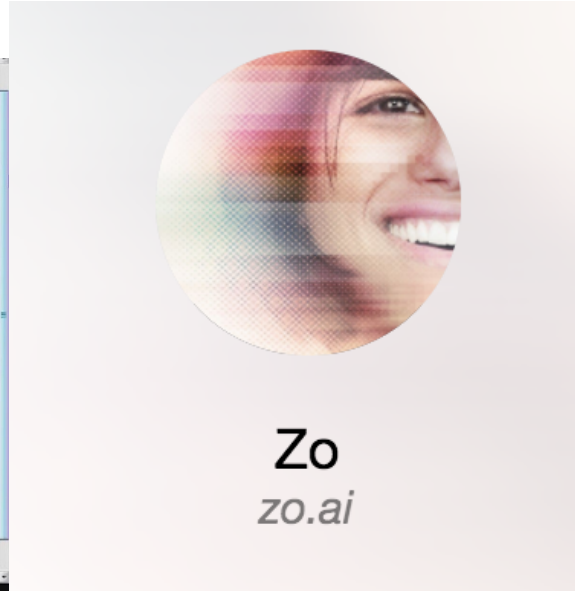
“Successful” applications of Natural Language Processing (NLP)



“Successful” applications of Natural Language Processing (NLP)



“Successful” applications of Natural Language Processing (NLP)



Some NLP tasks

Machine Translation

第13届上海国际电影节开幕...

The 13th Shanghai Film festival ...

Word Sense Disambiguation

I need new batteries for my **mouse**

Paraphrase

XYZ acquired ABC yesterday

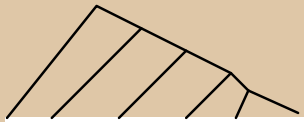
ABC has been taken over by XYZ

Spam detection

Let's go to Agra!

Buy V1AGRA ...

Syntactic Parsing



I see him with a telescope

Summarization

The Dow Jones is up

The S&P 500 jumped

Housing price rose

Economy is good

Part of Speech (POS) tagging

ADJ	ADJ	NOUN	VERB	ADV
Colorless	green	ideas	sleep	furiously

Coreference resolution

Carter told Mubarak **he** shouldn't run again

Dialog / Question Answering

Where is a Bug's life playing ?

Sept Parnassien at 7:30

Ambiguous, noisy and with great variability

Why NLP is so hard ?

Named entities and Idioms

- Where is A Bug's Life playing (...)
- Let It Be was recorded (...)
- Push the daisies
- lose face

Non-canonical language

Great job `@justinbieber`! Were SOO PROUD of what youve done!

U taught us 2 #neversaynever & you yourself should never give up either

Neologism

- unfriend, retweet,
- bromance, +1, ...

World knowledge

- Mary and Sue are sisters
- Mary and Sue are mothers

Ambiguous, noisy and with great variability

Why NLP is so hard ?

Named entities and Idioms

- Where is A Bug's Life playing (...)
- Let It Be was recorded (...)
- Push the daisies
- lose face

Non-canonical language

Great job `@justinbieber`! Were SOO PROUD of what youve done!

U taught us 2 #neversaynever & you yourself should never give up either

Neologism

- unfriend, retweet,
- bromance, +1, ...

World knowledge

- Mary and Sue are sisters
- Mary and Sue are mothers

Hospitals are Sued by 7 Foot Doctors
Kids Make Nutritious Snacks
Iraqi Head Seeks Arms

Statistical NLP

A very successful approach, indeed

From Peter Norvig (`http://norvig.com/chomsky.html`)

- **Search engines**: 100% of major players are trained and probabilistic.
- **Speech recognition**: 100% of major systems ...
- **Machine translation**: 100% of top competitors ...
- **Question answering**: the IBM Watson system ...

Statistical NLP

A very successful approach, indeed

From Peter Norvig (`http://norvig.com/chomsky.html`)

- **Search engines**: 100% of major players are trained and probabilistic.
- **Speech recognition**: 100% of major systems ...
- **Machine translation**: 100% of top competitors ...
- **Question answering**: the IBM Watson system ...

Today, add **neural network/deep-learning**

Statistical NLP

A very successful approach, indeed

From Peter Norvig (`http://norvig.com/chomsky.html`)

- **Search engines**: 100% of major players are trained and probabilistic.
- **Speech recognition**: 100% of major systems ...
- **Machine translation**: 100% of top competitors ...
- **Question answering**: the IBM Watson system ...

Today, add **neural network/deep-learning**

Statistical NLP ?

Using statistical techniques to **infer structures** from text based on **statistical language modeling**.

Statistical NLP - a (very) brief history

1970 -1983: Early success in speech reocgnition

- Hidden Markov models for acoustic modeling
- The first notion of language modeling as a Makov Chain

1983 - : Dominance of empiricism and statistical methods

- Incorporate probabilities for most language processing
- Use large corpora for training and evaluation

2003 - : Neural networks

- As a component at the beginning ...
- to end-to-end systems today

NLP: Statistical issues

Data sparsity in high dimension

For most of NLP tasks:

- model structured data
- with very peculiar and sparse distributions
- with a large set of possible outcomes

Ambiguity and variability

- The *context* is essential.
- Language is difficult to "interpret", even for human

→ Learning to efficiently represent language data
→ Neural networks have renewed the research perspectives

Is it so important ?

It is decisive !

Machine translation issue :

Opinion mining and Stock prediction

A. Hathaway *vs* Berkshire Hathaway

Outline

Outline

n-gram language model

Applications

Automatic Speech Recognition, Machine Translation, OCR, ...

The goal

Estimate the (non-zero) probability of a word sequence for a given vocabulary

n-gram assumption

$$P(w_{1:L}) = \prod_{i=1}^{L} P(w_i | w_{i-n+1:i-1}), \quad \forall i, w_i \in \mathcal{V}$$

Discrete n-gram model (conventional)

A word given its context

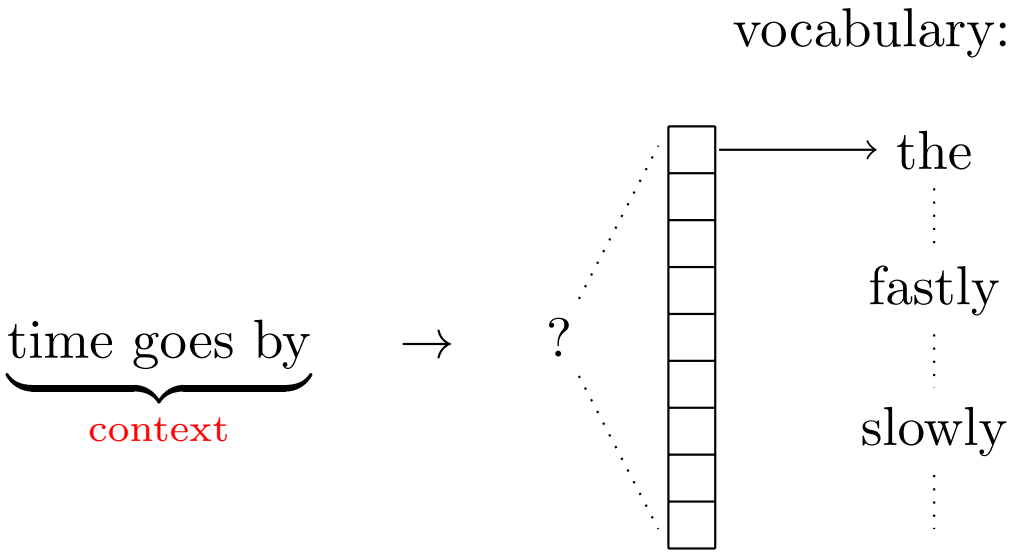
$$n = 4\text{: } P(w_i = ? | \underbrace{w_{i-3}, w_{i-2}, w_{i-1}}_{\text{context}})$$

$$\underbrace{\text{time goes by}}_{\text{context}} \quad \rightarrow \quad ?$$

Discrete n-gram model (conventional)

A word given its context

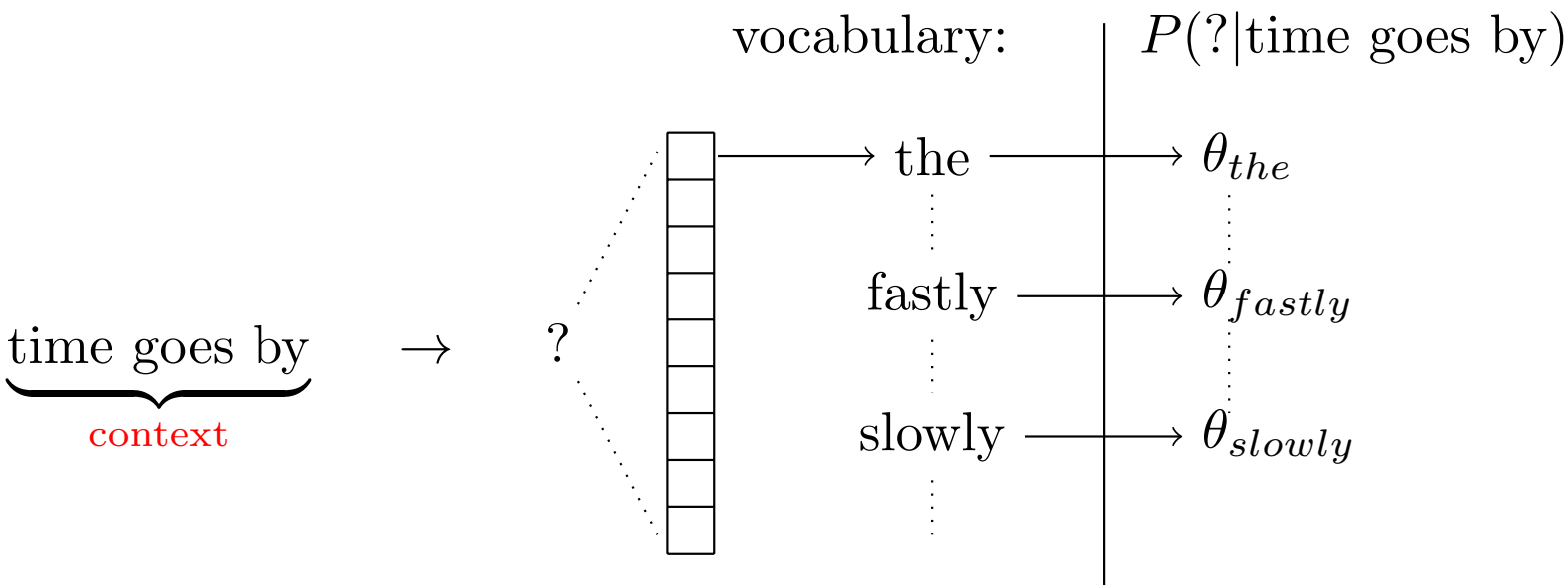
$$n = 4\text{: } P(w_i = ?|\underbrace{w_{i-3}, w_{i-2}, w_{i-1}}_{\text{context}})$$



Discrete n-gram model (conventional)

A word given its context

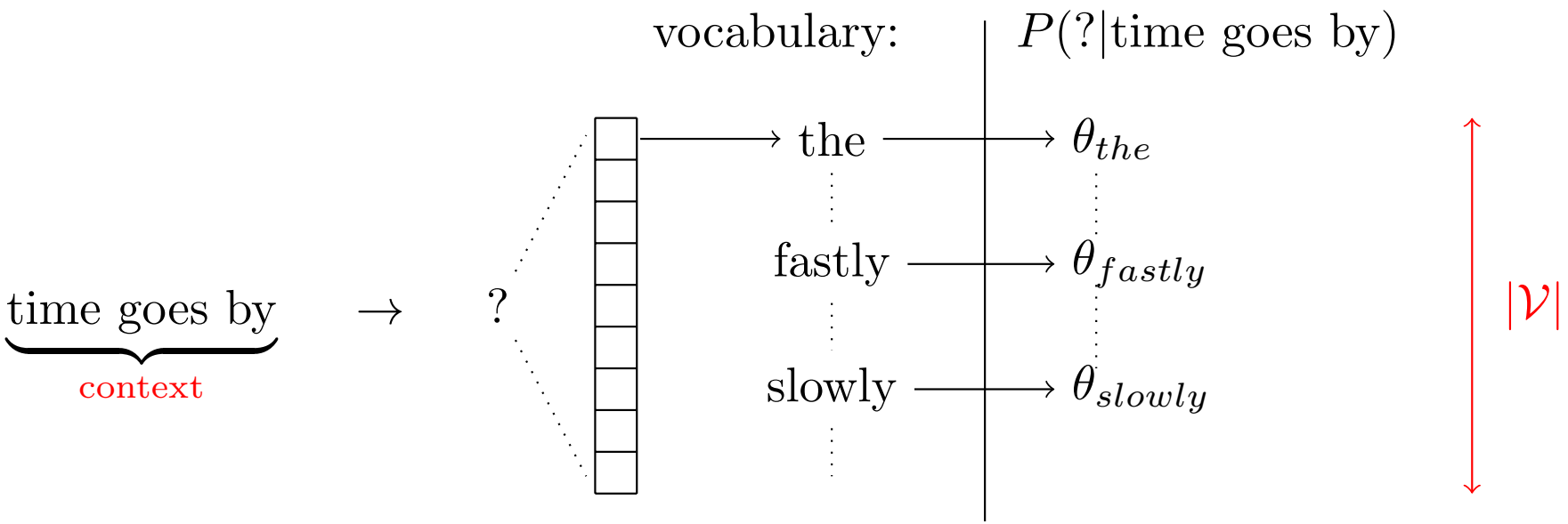
$n = 4$: $P(w_i = ? | \underbrace{w_{i-3}, w_{i-2}, w_{i-1}}_{\text{context}})$



Discrete n-gram model (conventional)

A word given its context

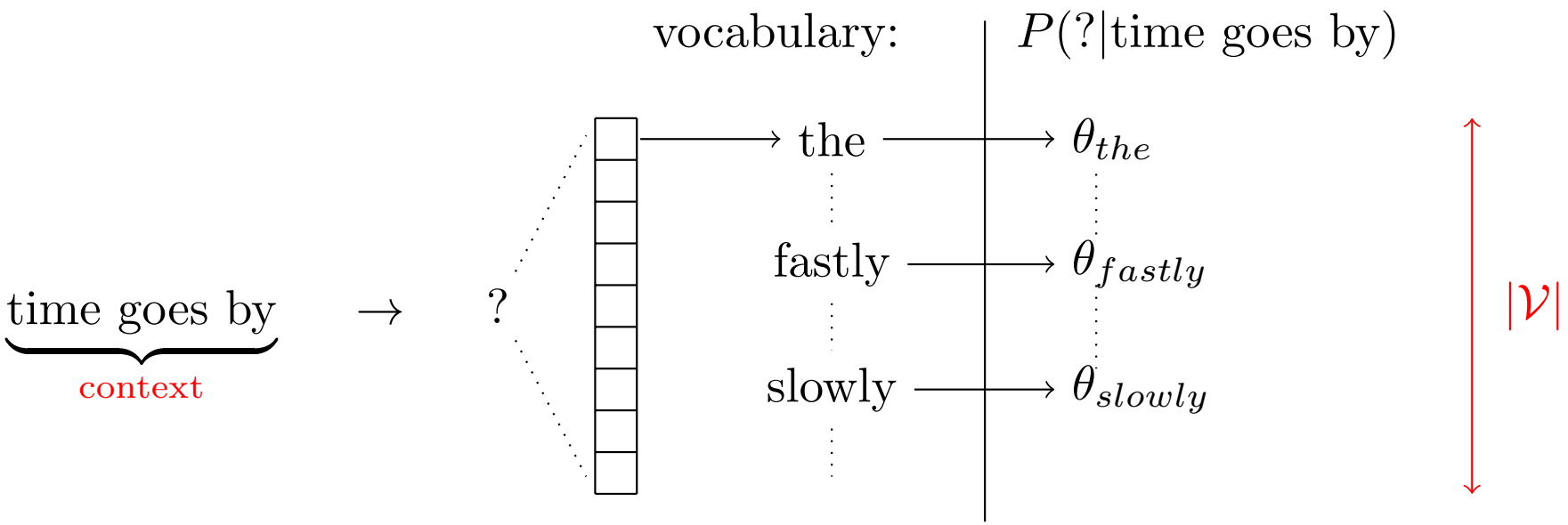
$n = 4$: $P(w_i = ? | \underbrace{w_{i-3}, w_{i-2}, w_{i-1}}_{\text{context}})$



Discrete n-gram model (conventional)

A word given its context

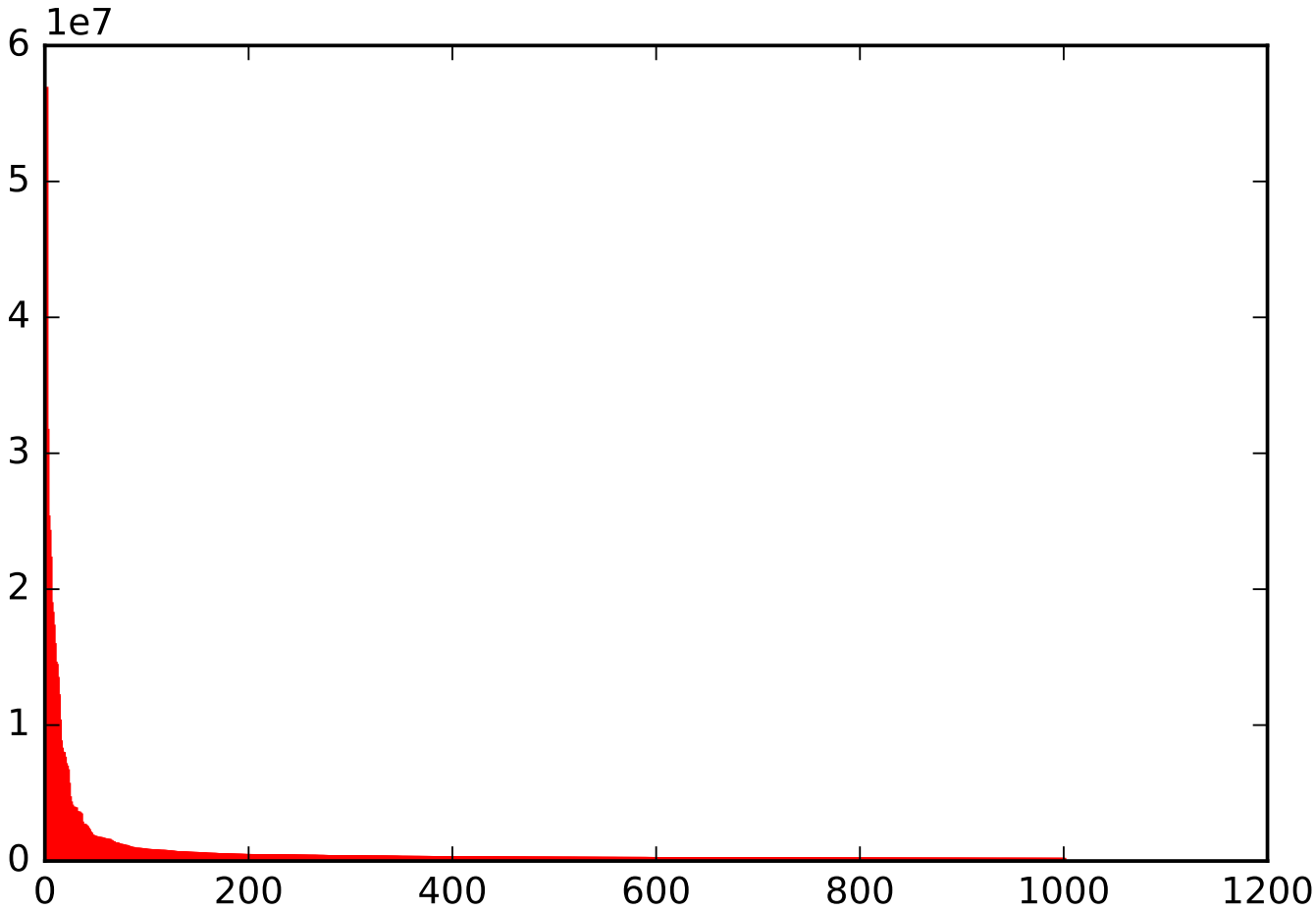
$n = 4$: $P(w_i = ? | \underbrace{w_{i-3}, w_{i-2}, w_{i-1}}_{\text{context}})$



$|\mathcal{V}|^4$ parameters, Maximum Likelihood Estimate

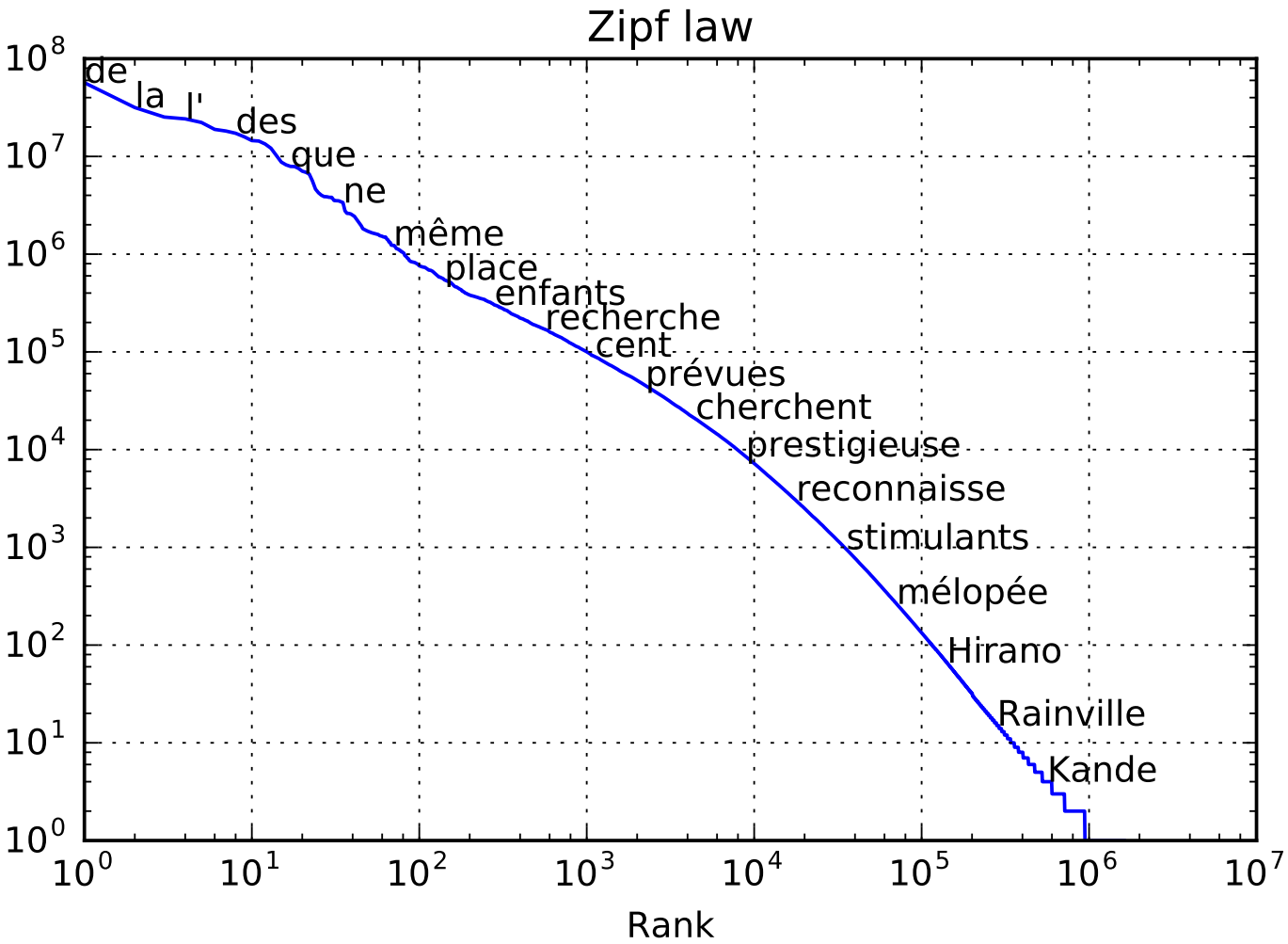
The Zipf law (for French)

$$frequency \propto 1/rank$$



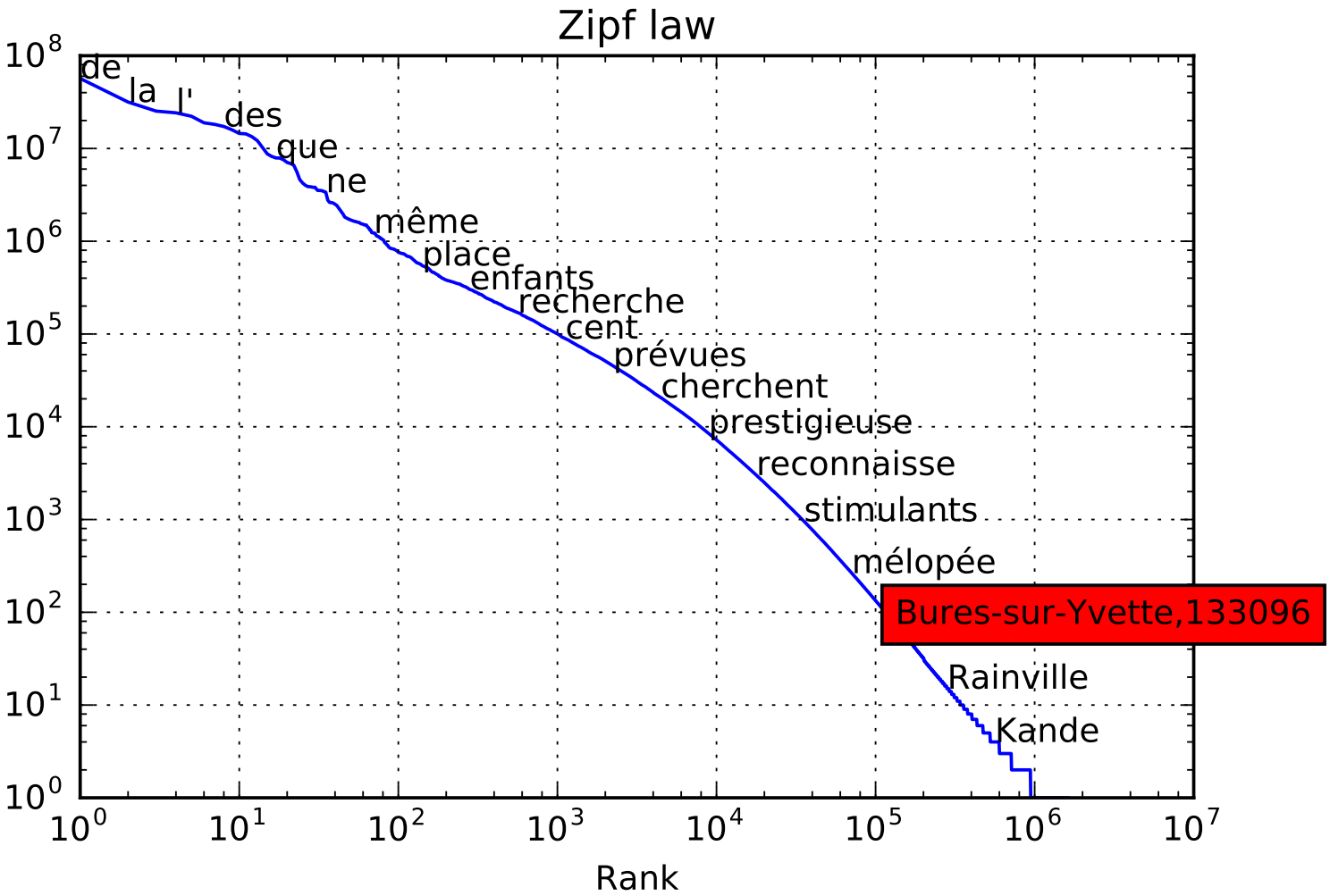
The Zipf law (for French)

$$frequency \propto 1/rank$$

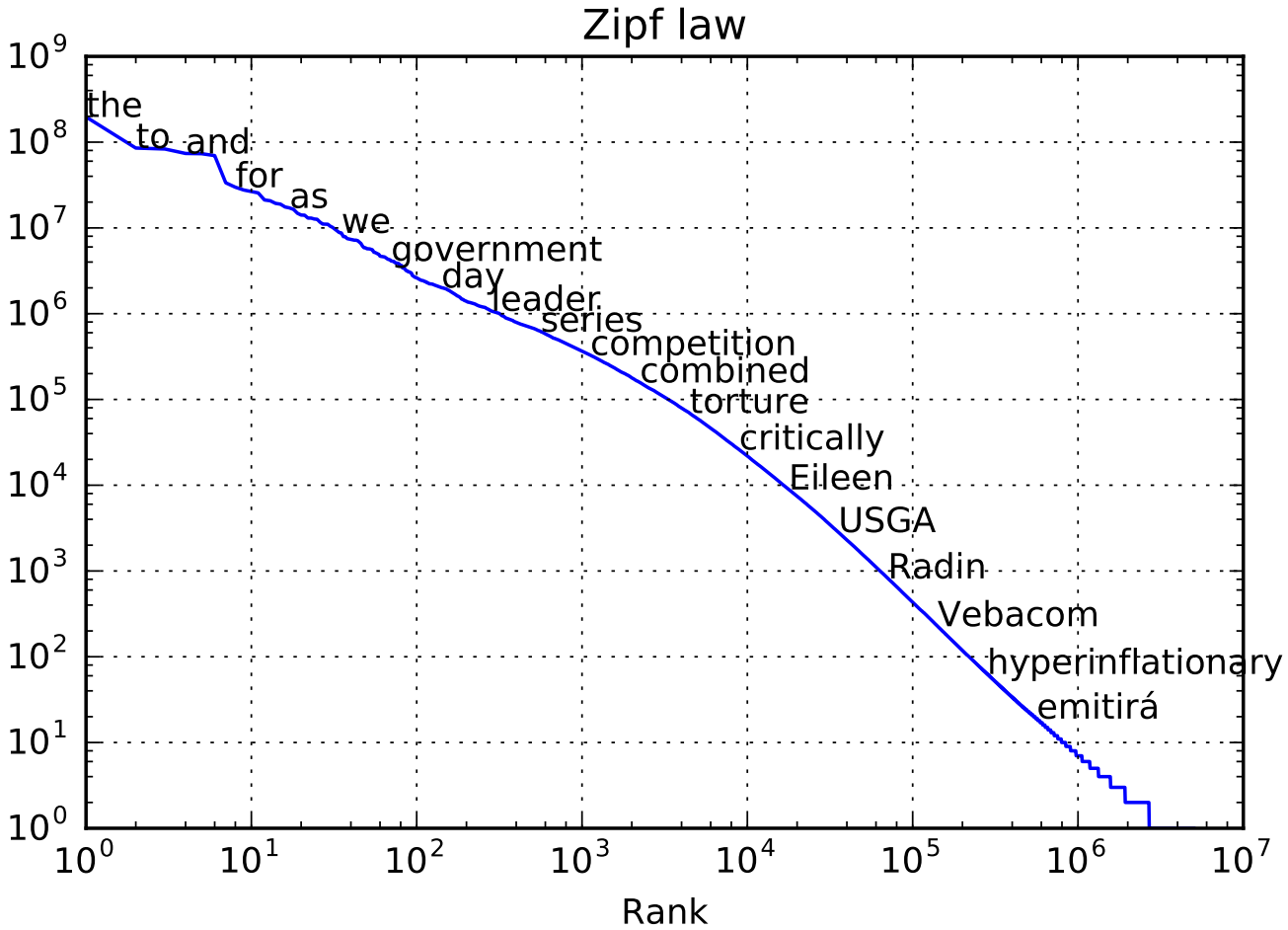


The Zipf law (for French)

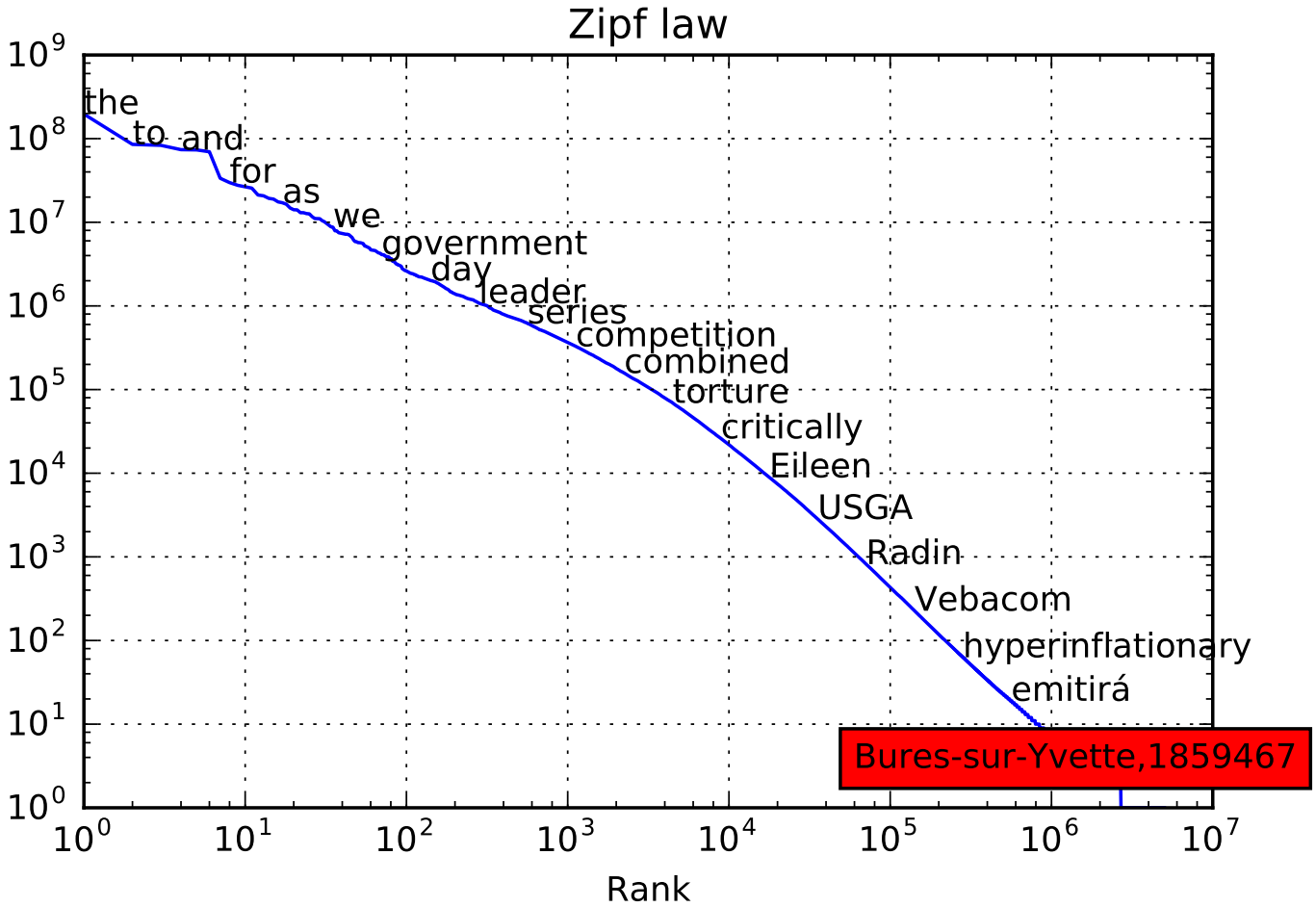
$$frequency \propto 1/rank$$



The Zipf law (for English)



The Zipf law (for English)



Consequences

Large vocabulary

- For many applications, $|\mathcal{V}| \propto 10^5$ or 10^6
- but most of the words quite never occur.

For n-gram model, it is even worse

- Most of n-grams never occur.

$\Rightarrow$ a restricted context ($n = 4, 5$ at most)

With 7 billions of running English words and $|\mathcal{V}| = 200\ 000$:

- $200\ 000^4 \approx 1.6 \times 10^{21}$ possible 4-grams,
- Hardly 1 billion observed
- Most of them just once

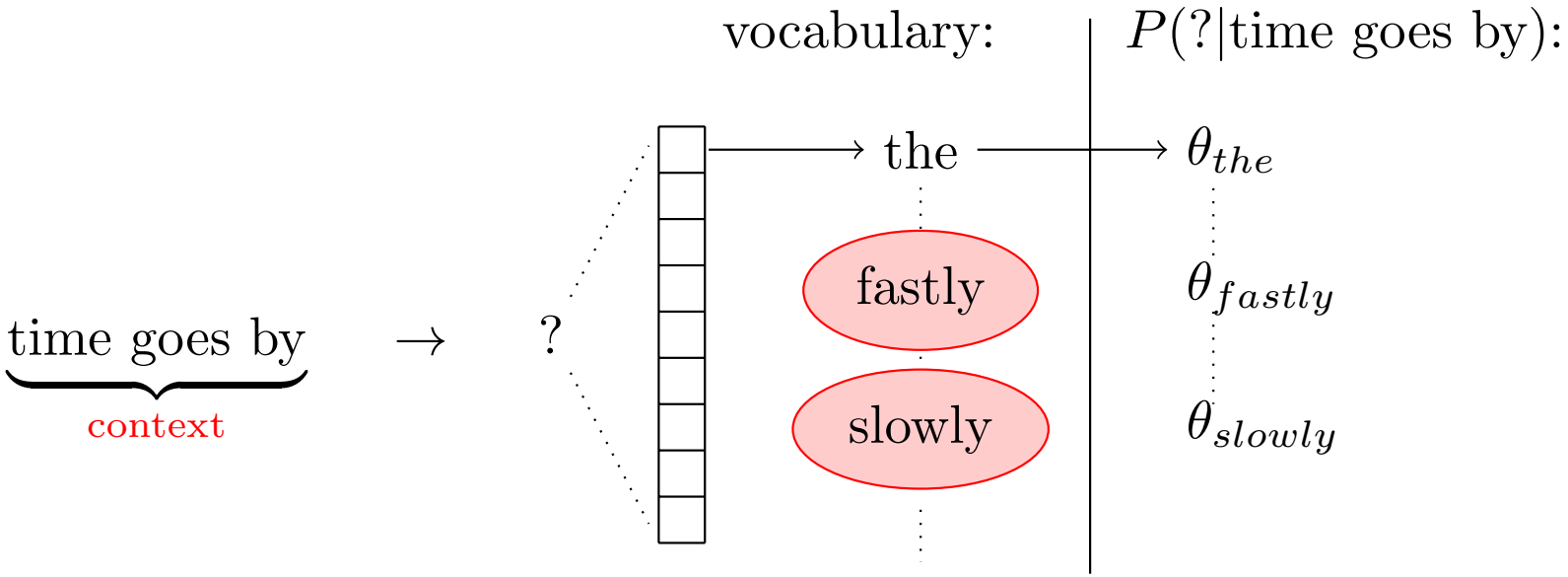
Some conventional remedies

- Increase the amount of data
- Smoothing methods (Chen and Goodman1998)

Lack of generalization - 1

A word given its context

$n = 4$:

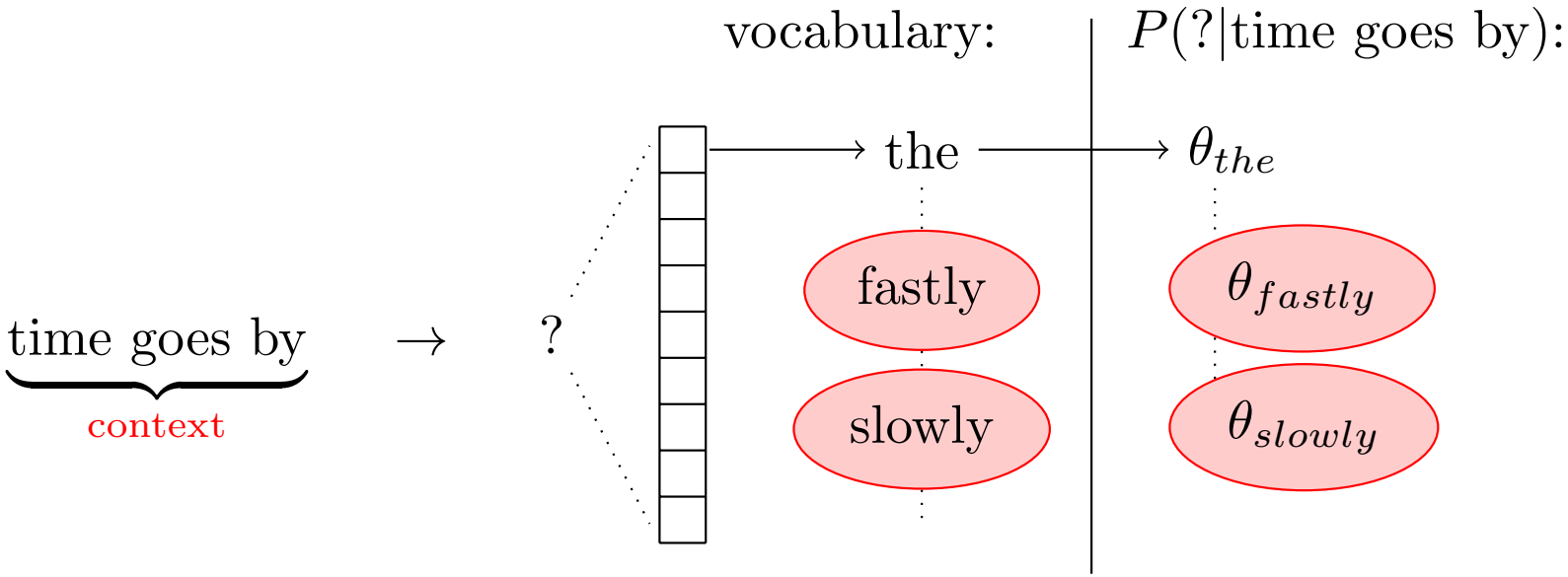


- For each context, a multinomial distribution
- A word in its context = one parameter to learn

Lack of generalization - 1

A word given its context

$n = 4$:

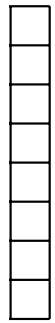


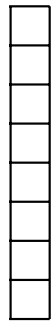
- For each context, a multinomial distribution
- A word in its context = one parameter to learn
- No parameter tying between words

$\Rightarrow$ No knowledge sharing

Lack of generalization - 2

and for different contexts

time goes by $\longrightarrow \boldsymbol{\theta}^{\text{time goes by}} =$ 

train goes by $\longrightarrow \boldsymbol{\theta}^{\text{train goes by}} =$ 

Each distribution is independant of the others

Sequence tagging

$$\mathbf{w} = w_1^L = w_1, w_2, ..., w_L$$
$$\mathbf{t} = t_1^L = t_1, t_2, ..., t_L$$

Example : Part-of-Speech (POS) tagging

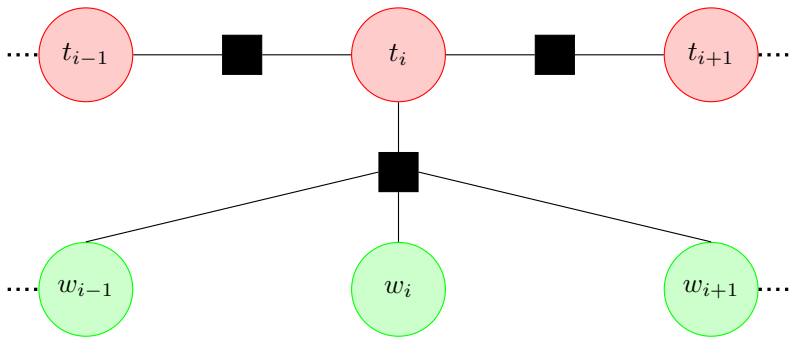
Sentence	POS-tags
Er	PPER-case=nom|@gender=masc|number=sg|person=3
fürchtet	VVFIN-mood=ind|number=sg|person=3|tense=pres
noch	ADV
Schlimmeres	NN-case=acc|gender=neut|number=sg
.	$.

Conditional Random Fields (CRF)

Linear chain

$$P(\mathbf{t}|\mathbf{w}) = \frac{1}{Z(\mathbf{w})} \prod_{i=1}^{L} \exp\Big(\langle \boldsymbol{\theta}, \boldsymbol{\phi}(t_i, t_{i-1}, \mathbf{w}) \rangle \Big)$$

$$= \frac{1}{Z(\mathbf{w})} \prod_{i=1}^{L} \exp\Big(\langle \boldsymbol{\theta}_u, \boldsymbol{\phi}_u(t_i, \mathbf{w}) \rangle + \langle \boldsymbol{\theta}_b, \boldsymbol{\phi}_b(t_i, t_{i-1}, \mathbf{w}) \rangle \Big)$$



Conditional Random Fields (CRF)

Linear chain

$$P(\mathbf{t}|\mathbf{w}) = \frac{1}{Z(\mathbf{w})} \prod_{i=1}^{L} \exp\Big(\langle \boldsymbol{\theta}, \boldsymbol{\phi}(t_i, t_{i-1}, \mathbf{w}) \rangle \Big)$$

$$= \frac{1}{Z(\mathbf{w})} \prod_{i=1}^{L} \exp\Big(\langle \boldsymbol{\theta}_u, \boldsymbol{\phi}_u(t_i, \mathbf{w}) \rangle + \langle \boldsymbol{\theta}_b, \boldsymbol{\phi}_b(t_i, t_{i-1}, \mathbf{w}) \rangle \Big)$$

The basic feature template with binary indicators:

- $\boldsymbol{\phi}_u(t_i, \mathbf{w}) = \boldsymbol{\phi}_u(t_i, w_i) = \mathbb{I}\big(w_i \wedge t_i\big)$
- $\boldsymbol{\phi}_b(t_i, t_{i-1}, \mathbf{w}) = \boldsymbol{\phi}_b(t_i, t_{i-1}) = \mathbb{I}\big(t_{i-1} \wedge t_i\big)$
- ...

A hand-crafted word representation :

$$\boldsymbol{\theta}_u \longleftrightarrow \boldsymbol{\phi}_u(t_i, w_i)$$

$$\boldsymbol{\theta}_{u,t_i} \longleftrightarrow \boldsymbol{\phi}_{u,t_i}(w_i)$$

Word representation in CRF

Word representation may include:

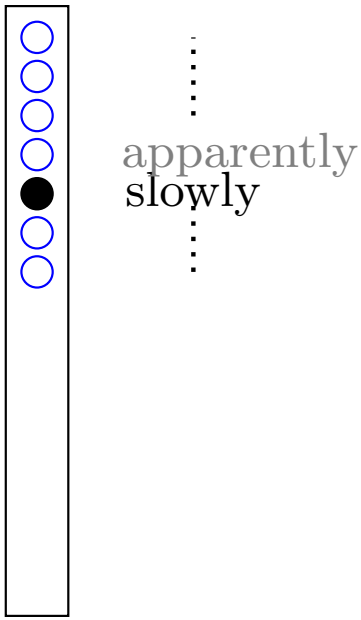
- surface form

Word representation in CRF

Word representation may include:

- surface form

$w_i \rightarrow \mathbf{x}_i =$

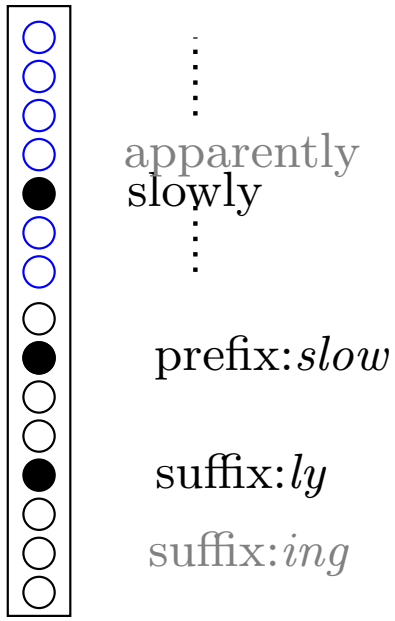


Word representation in CRF

Word representation may include:

- surface form
- prefix
- suffix
- ...

$w_i \rightarrow \mathbf{x}_i =$



Word vectors

A word is described by a feature vector: its representation or embedding

The goal

- address the sparsity issue
- a better generalization

Word vectors

A word is described by a feature vector: its representation or embedding

The goal

- address the sparsity issue
- a better generalization

Drawbacks

- "expertise" is required and dedicated to the task, the language, the domain, ...
- in practice, relies on linguistic resources

Outline

Estimate n-gram probabilities in a continuous space

Introduced in (Bengio and Ducharme2001; Bengio et al.2003) and applied to speech recognition and machine translation in (Schwenk and Gauvain2002).

In a nutshell

1. associate each word with a continuous feature vector
2. express the probability function of a word sequence in terms of the feature vectors of these words
3. learn simultaneously the feature vectors and the parameters of that probability function.

Estimate n-gram probabilities in a continuous space

Introduced in (Bengio and Ducharme2001; Bengio et al.2003) and applied to speech recognition and machine translation in (Schwenk and Gauvain2002).

In a nutshell

1. associate each word with a continuous feature vector
2. express the probability function of a word sequence in terms of the feature vectors of these words
3. learn simultaneously the feature vectors and the parameters of that probability function.

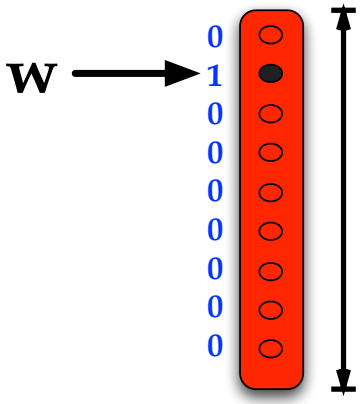
Why should it work ?

- "similar" words are expected to have a similar feature vectors
- the probability function is a smooth function of these feature values

$\Rightarrow$ a small change in the features will induce a small change in the probability

Project a words into a continuous space

- The vocabulary is a neural network layer

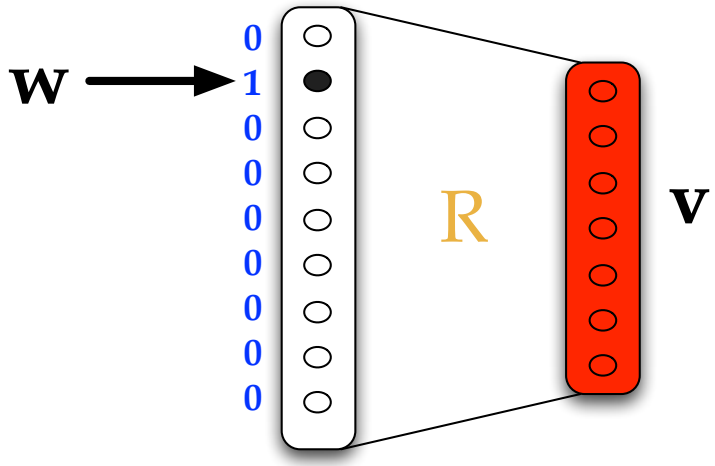


|V|: vocabulary size

- A neural network layer represents a vector of values,
- one neuron per value

Project a words into a continuous space

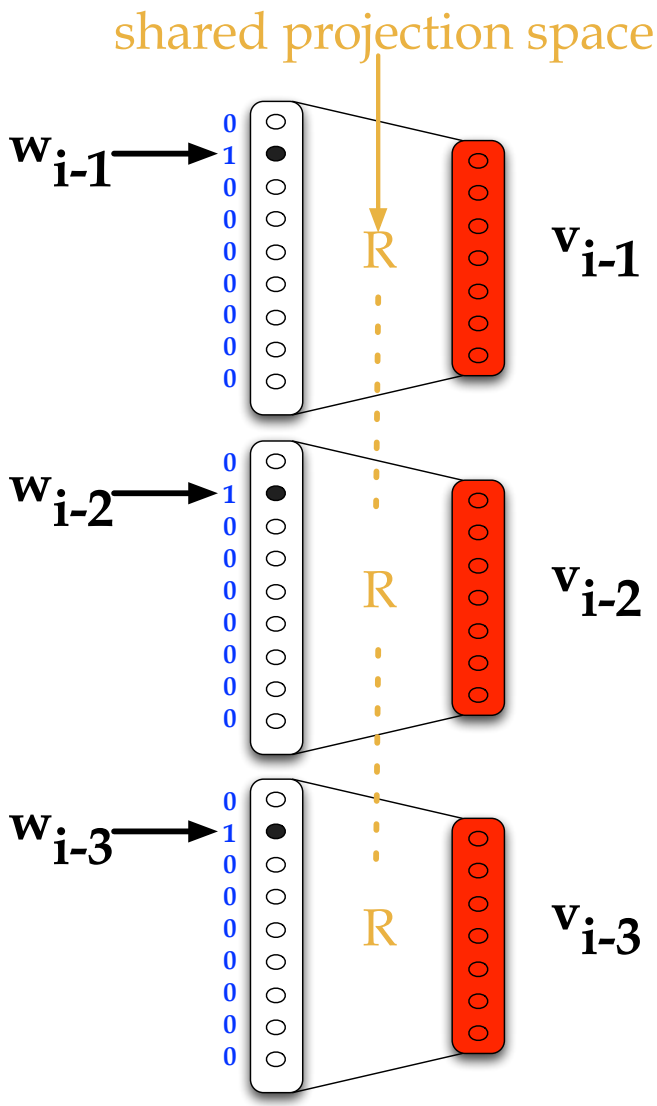
- The vocabulary is a neural network layer
- Word continuous representation:
  add a second layer fully connected



- The connection between two layers is a matrix operation
- The matrix **R** contains all the connection weights
- **v** is a continuous vector

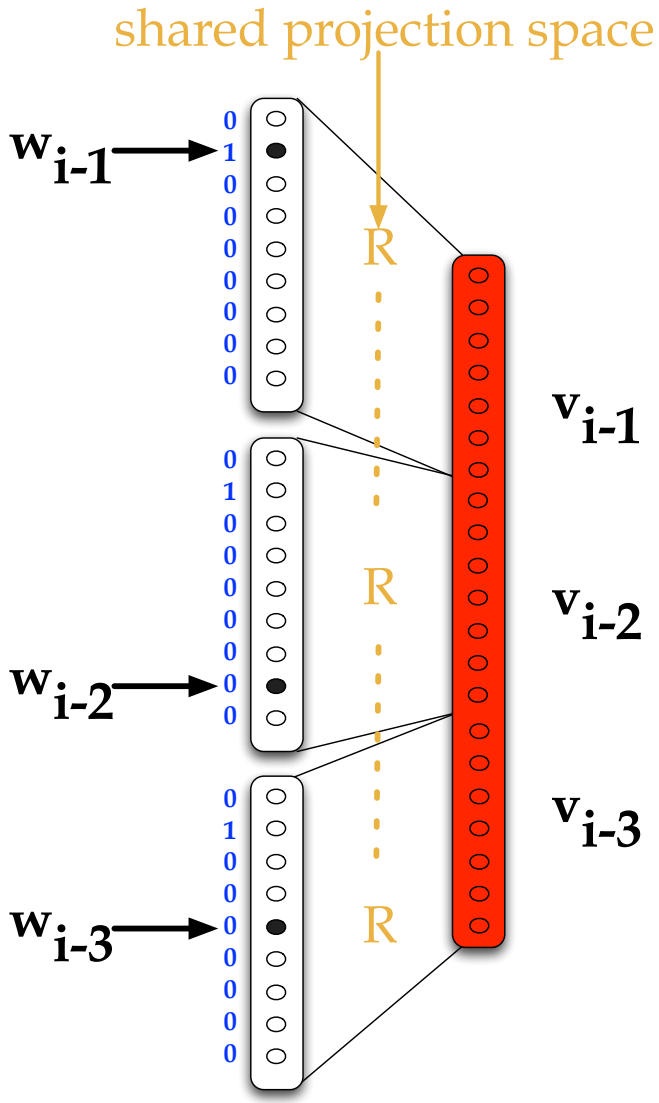
Project a words into a continuous space

- The vocabulary is a neural network layer
- Word continuous representation:
  add a second layer fully connected
- For a 4-gram, the history is a sequence of
  3 words



Project a words into a continuous space

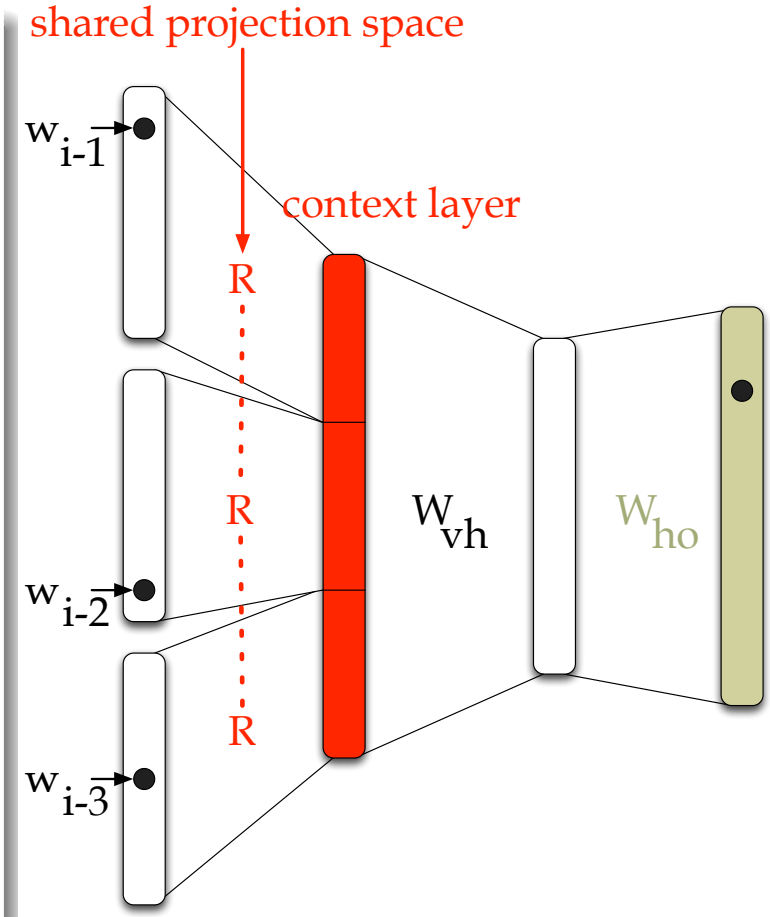
- The vocabulary is a neural network layer
- Word continuous representation: add a second layer fully connected
- For a 4-gram, the history is a sequence of 3 words
- Merge these three vectors to derive a single vector for the history



Estimate the n-gram probability

The program

- Given the history expressed as a feature vector : $\mathbf{v}$

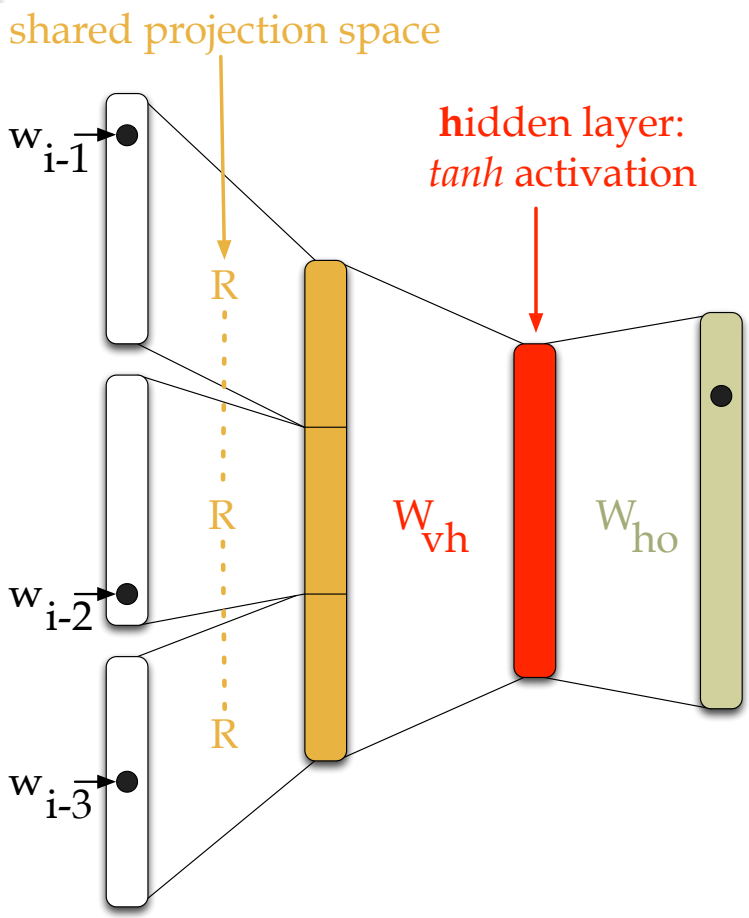


Estimate the n-gram probability

The program

- Given the history expressed as a feature vector : $\mathbf{v}$
- Create a feature vector for the word to be predicted:

$$\mathbf{h} = f(\mathbf{W}_{vh}\mathbf{v})$$



Estimate the n-gram probability

The program

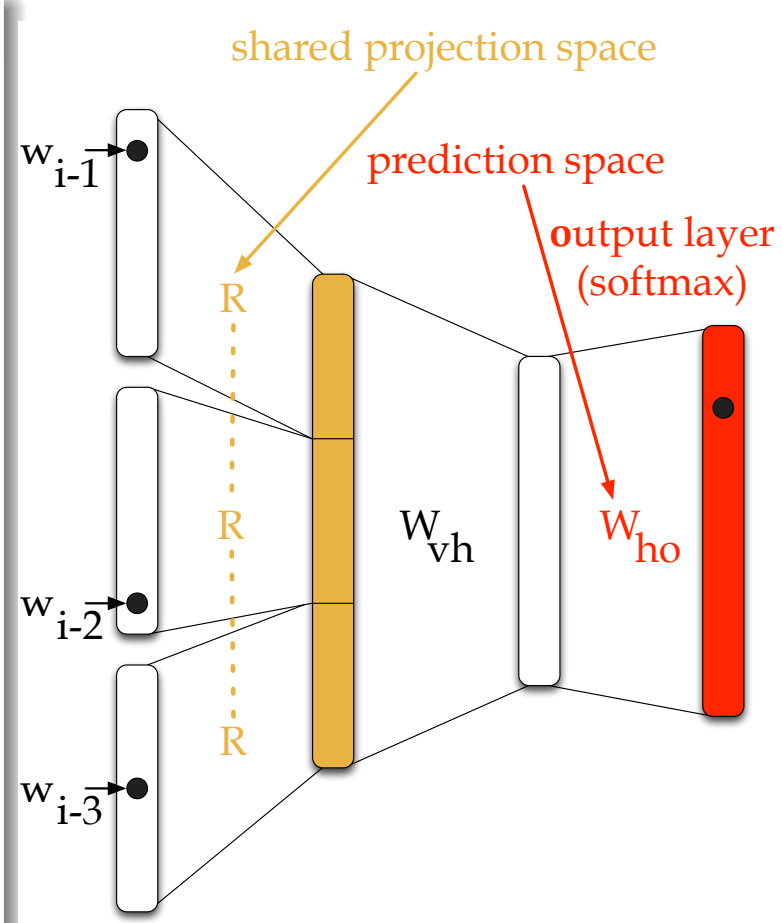
- Given the history expressed as a feature vector : $\mathbf{v}$
- Create a feature vector for the word to be predicted:

$$\mathbf{h} = f(\mathbf{W}_{vh}\mathbf{v})$$

- Estimate probabilities for all words given the history:

$$\mathbf{o} = f(\mathbf{W}_{ho}\mathbf{h})$$

$$P(w|w_{i-n+1:i-1}) = \frac{\exp(o_{w_i})}{\sum_{w \in \mathcal{V}} \exp(o_{w_i})}$$



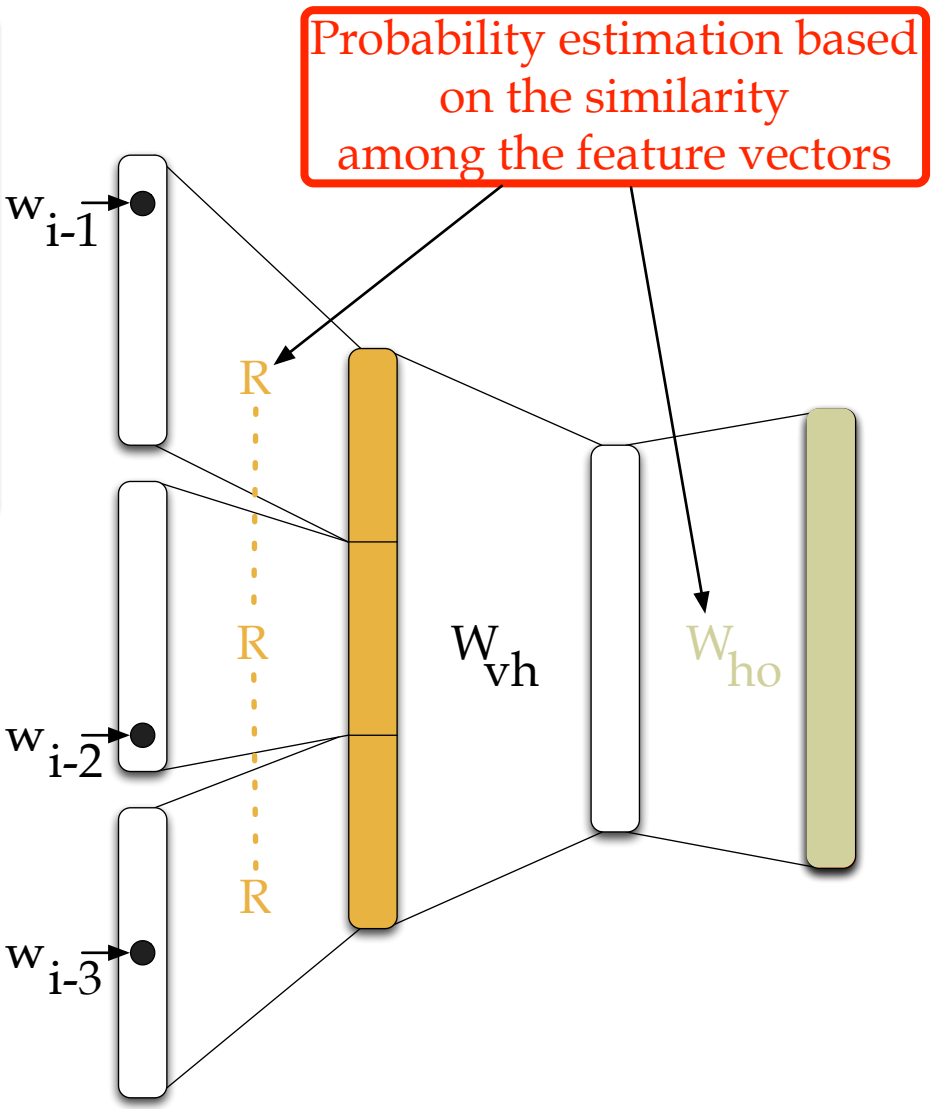
Early assessment

Key points

- The projection **in continuous spaces**

→ reduces the sparsity issues

- Learn simultaneously the projection and the prediction: $(\mathbf{R}, \mathbf{W_{vh}}, \mathbf{W_{ho}})$



Early assessment

Key points

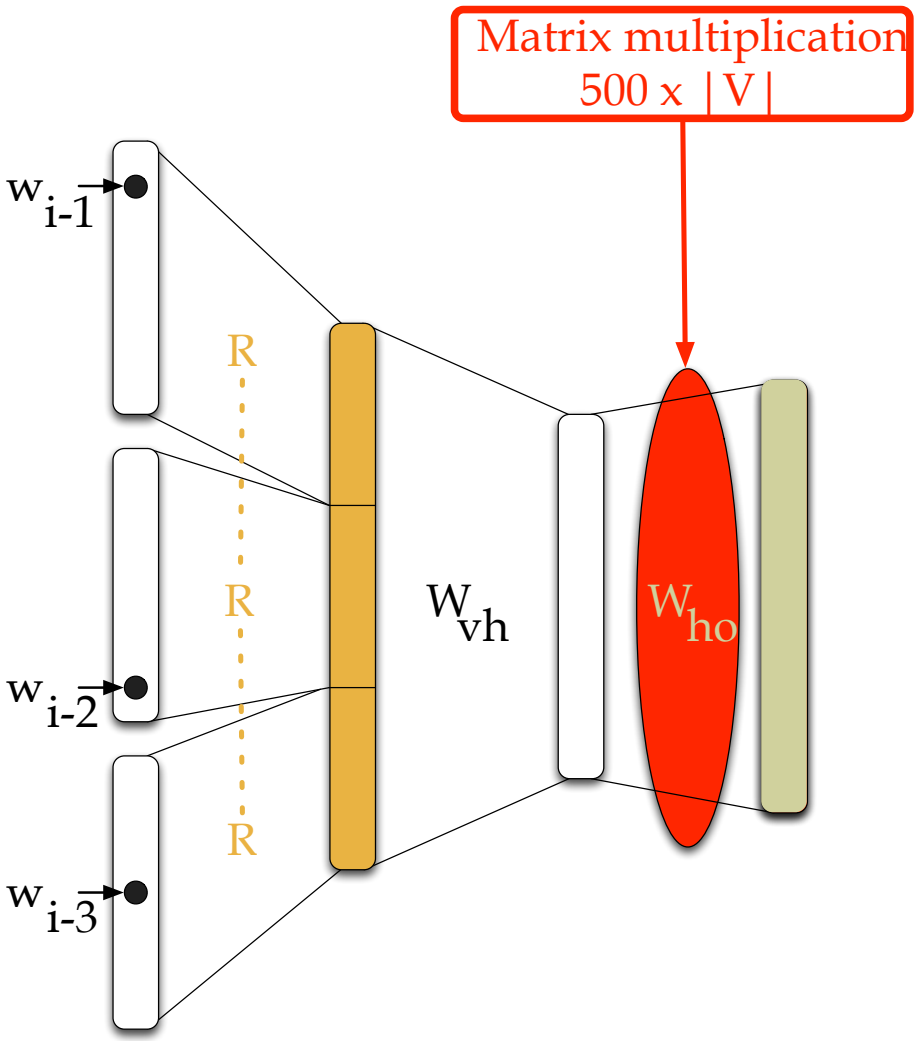
- The projection **in continuous spaces**

→ reduces the sparsity issues

- Learn simultaneously the projection and the prediction: $(\mathbf{R}, \mathbf{W_{vh}}, \mathbf{W_{ho}})$

Complexity issues

- The input vocabulary can be as large as we want.
- Increasing the order of n does not increase the complexity.
- **The problem is the output vocabulary size.**



A solution : class-based language model

Main ideas

As proposed by (Mnih and Hinton2008):

- Represent the vocabulary as a clustering tree (Brown et al.1992).
- Predict the path in this clustering tree.

A solution : class-based language model

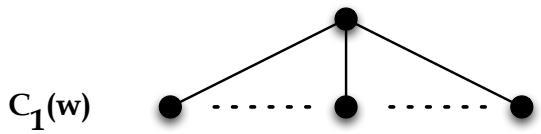
Main ideas

As proposed by (Mnih and Hinton2008):

- Represent the vocabulary as a clustering tree (Brown et al.1992).
- Predict the path in this clustering tree.

Word clustering

- Put each word w with a single root class $c_1(w)$
- Split these word classes in sub-classes $(c_2(w))$ and so on.



A solution : class-based language model

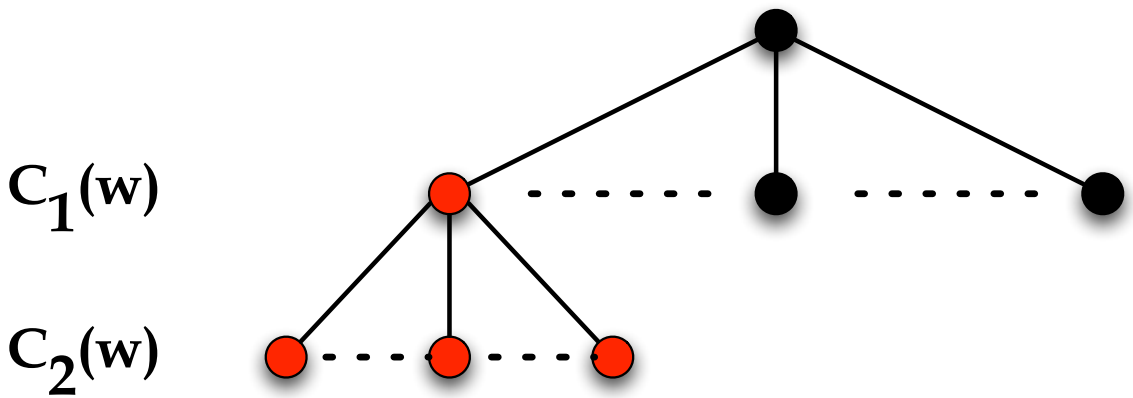
Main ideas

As proposed by (Mnih and Hinton2008):

- Represent the vocabulary as a clustering tree (Brown et al.1992).
- Predict the path in this clustering tree.

Word clustering

- Put each word w with a single root class $c_1(w)$
- Split these word classes in sub-classes ($c_2(w)$) and so on.



A solution : class-based language model

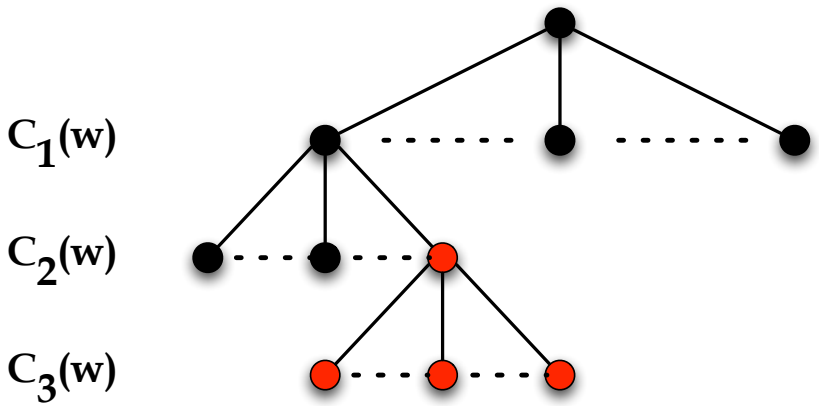
Main ideas

As proposed by (Mnih and Hinton2008):

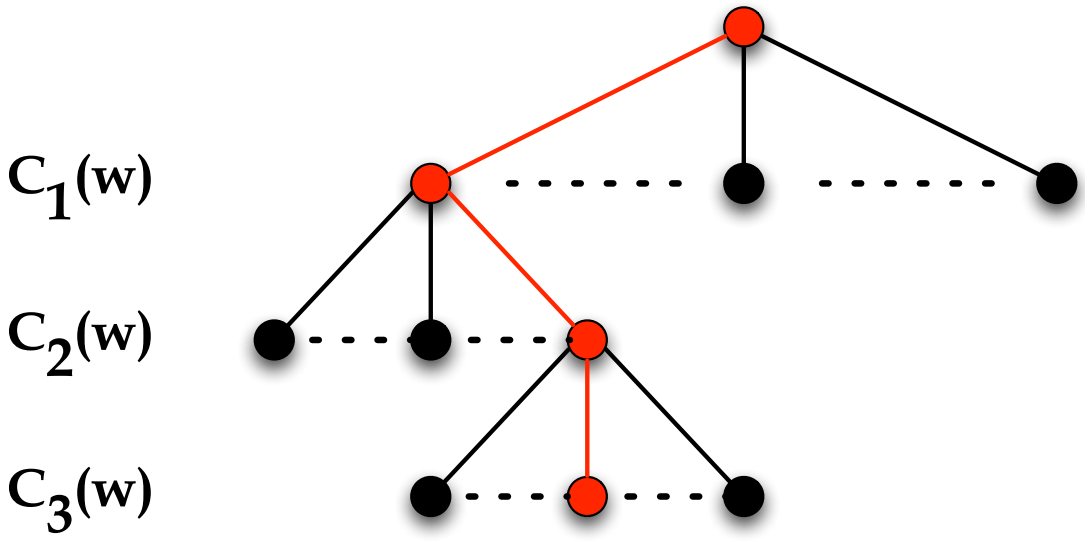
- Represent the vocabulary as a clustering tree (Brown et al.1992).
- Predict the path in this clustering tree.

Word clustering

- Put each word w with a single root class $c_1(w)$
- Split these word classes in sub-classes $(c_2(w))$ and so on.



Word probabilities

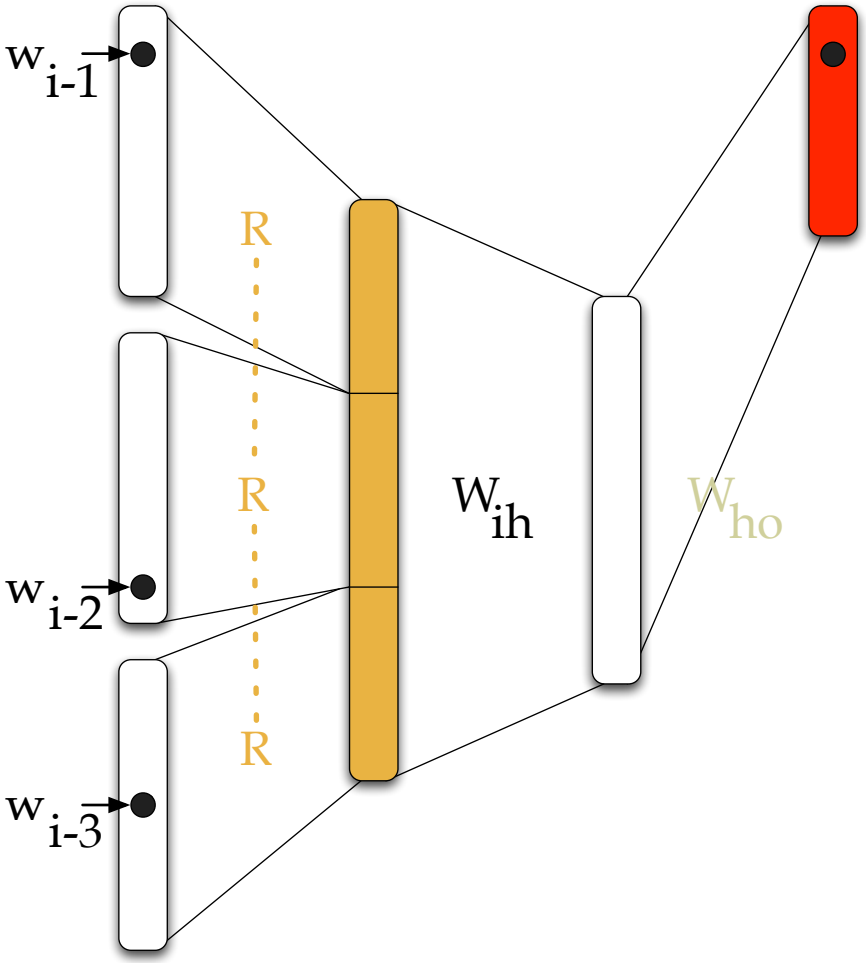


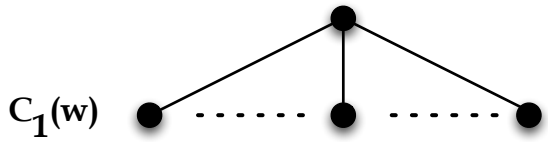
$$P(w_i|h) = P(c_1(w_i)|h) \prod_{d=2}^{D} P(c_d(w_i)|h, c_{1:d-1})$$

- $c_{1:D}(w_i) = c_1, \ldots, c_D$: path for the word w_i in the clustering tree,
- D : depth of the tree,
- $c_d(w_i)$: (sub-)class,
- $c_D(w_i)$: leaf.

The SOUL language model

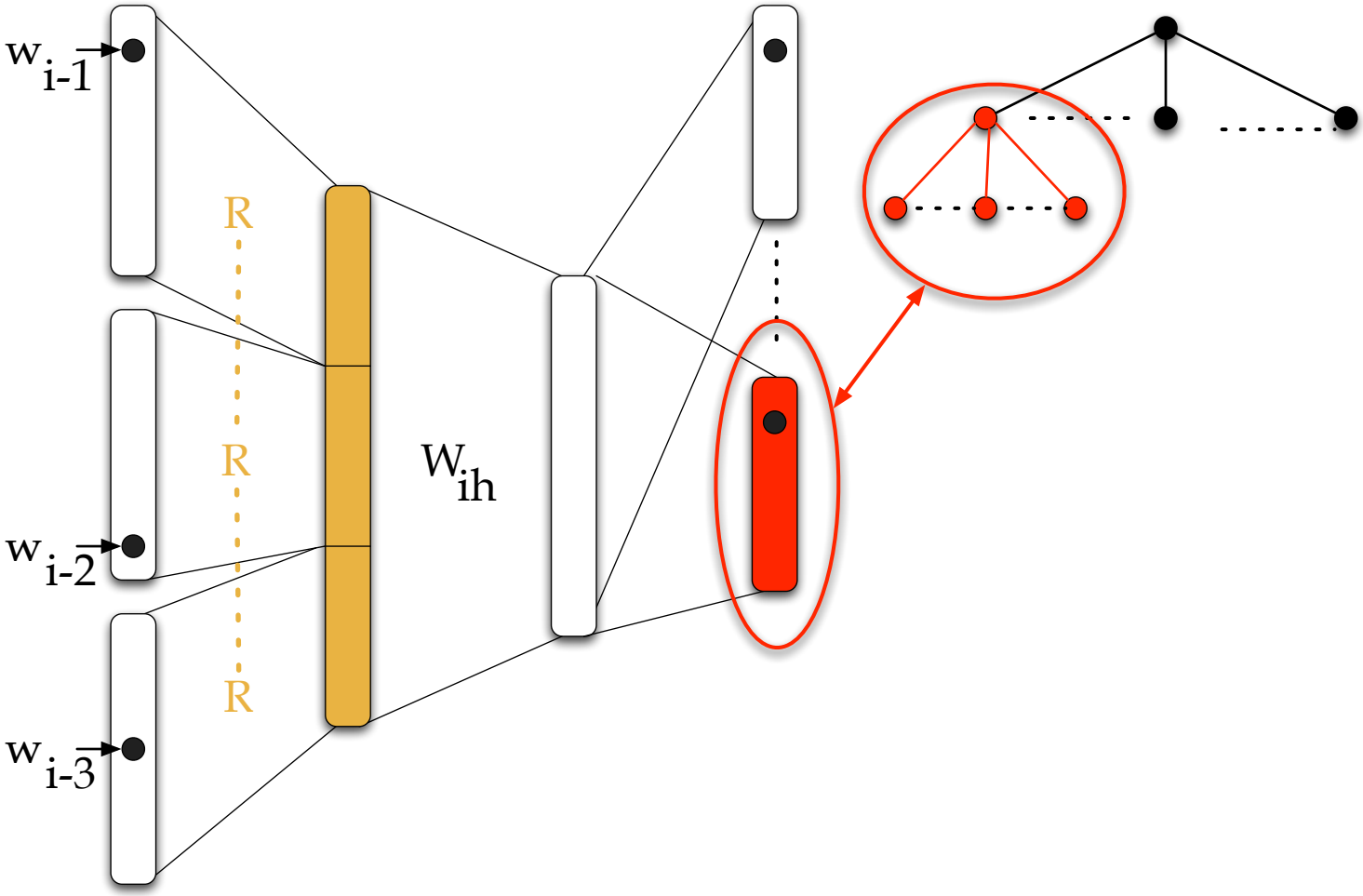
A solution for large vocabulary NLP tasks (Le et al.2011)





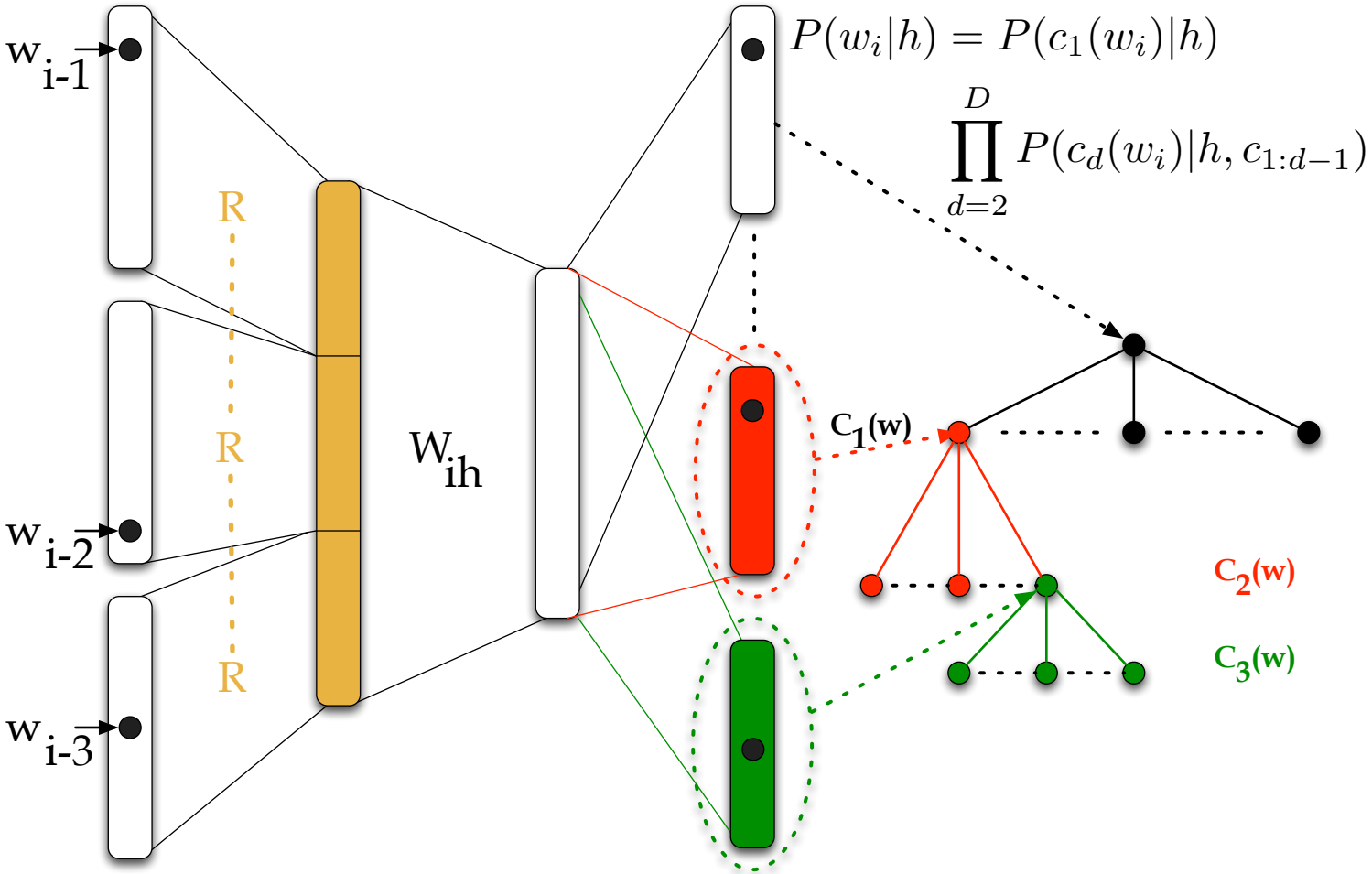
The SOUL language model

A solution for large vocabulary NLP tasks (Le et al.2011)



The SOUL language model

A solution for large vocabulary NLP tasks (Le et al.2011)



Experimental results

For automatic speech recognition

- On Mandarin Chinese and Arabic data (GALE)
- Significant improvements over state of the art systems.

For machine translation

WMT International evaluation campaign on european language pairs

best results for French-English in 2010 and 2011.

Extension to translation modeling (Le et al.2012)

Best results for French-English in 2012, 2013 and 2015.

Outline

Motivations

For morphologically-rich and non-canonical languages

- Very productive word formation processes

$\Rightarrow$ generate a proliferation of word forms.

Freundschaftsbezeigungen, görüntülenebilir $\leftrightarrow$ *MYL, AFAIK, cul8r*

Consequences

- Morphologically-rich and under-resourced language processing
- Social Media implies fast change in the language use

$\rightarrow$ An evolving vocabulary with new compound tokens, abbreviations, ...

Tokens decipherment/encoding

Ex: German POS-Tagging

The Task

The TIGER corpus defines a POS-tagging task with very rich tagset (around 600 tags)

State of the art results (Mueller et al.2013)

- A second order CRF
- with an intensive feature engineering to describe the morphology

Deep Net approach (Santos and Zadrozny2014)

- A lot of information about words can be leveraged from subword features.

$\Rightarrow$ Learn to infer a word representation from the character level

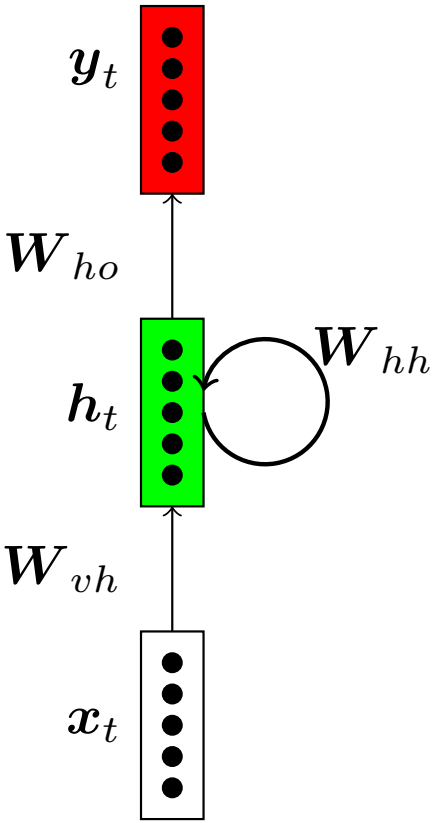
$\Rightarrow$ Make these representations aware of the context (sentence level)

A training example

Sentence	POS-tags
Er	PPER-case=nom|@gender=masc|number=sg|person=3
fürchtet	VVFIN-mood=ind|number=sg|person=3|tense=pres
noch	ADV
Schlimmeres	NN-case=acc|gender=neut|number=sg
.	$.

Recurrent network

Interlude



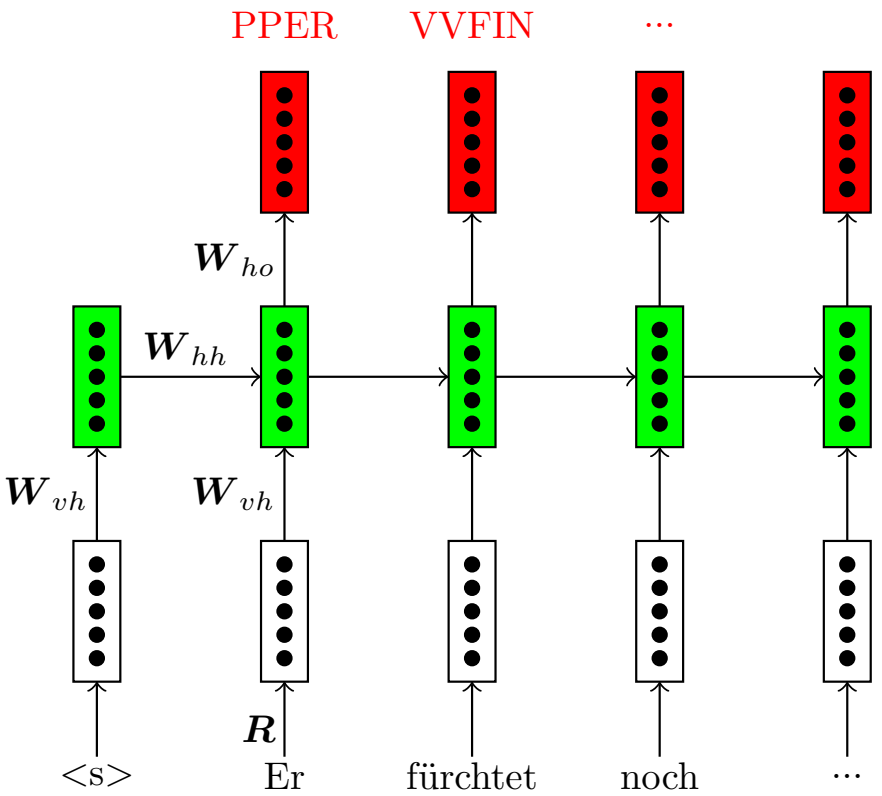
A dynamic system, at time t:

- maintains a hidden representation, the internal state: $\boldsymbol{h}_t$
- Updated with the observation of $\boldsymbol{x}_t$ and the previous state $\boldsymbol{h}_{t-1}$
- The prediction $\boldsymbol{y}_t$ depends on the internal state ($\boldsymbol{h}_t$)
- $\boldsymbol{x}_t$ comes from word embeddings

The same parameter set is shared across time steps

Recurrent network

Unfolding the structure: a deep-network



At each step t

- Read the word $w_t \rightarrow \boldsymbol{x}_t$ from $\boldsymbol{R}$
- Update the hidden state $\boldsymbol{h}_t = f(\boldsymbol{W}_{vh}\boldsymbol{x}_t + \boldsymbol{W}_{\boldsymbol{hh}}\boldsymbol{h}_{\boldsymbol{t-1}})$
- The tag at t can be predicted from $\boldsymbol{h}_t$:

$$\boldsymbol{y}_t = g(\boldsymbol{W}_{ho}\boldsymbol{h}_t)$$

- g is the softmax function over the tagset

Training recurrent model

Training algorithm

Back-Propagation through time (Rumelhart et al.1986; Mikolov et al.2011):

- for each step t
  - compute the loss gradient
  - Back-Propagation through the unfolded structure

Inference

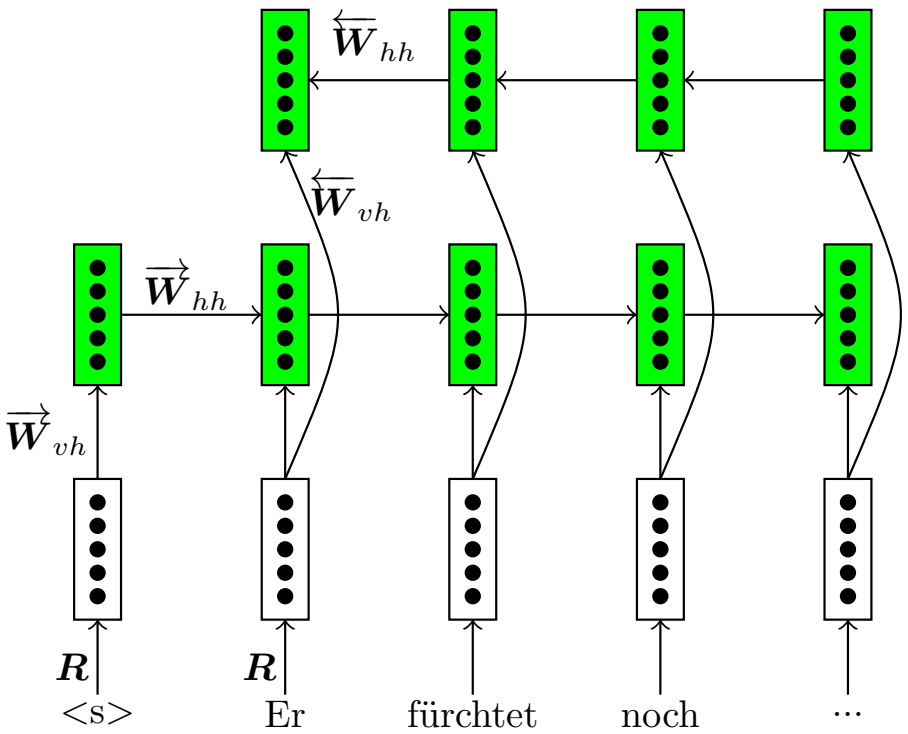
- Cannot be easily integrated to conventional approaches (ASR, SMT, ...)
- Well suited for sequence tagging
- $\boldsymbol{h}_t$ represent the word w_t in its left context
- A powerful device for end-to-end system

Known issues

- Vanishing gradient $\rightarrow$ LSTM
- Long-term memory $\rightarrow$ Bi-recurrent network

Bi-recurrent network

Unfolding the structure: a deep-network



At each step t, from left to right

- $w_t \rightarrow \boldsymbol{x}_t$
- $\overrightarrow{\boldsymbol{h}}_t = f(\overrightarrow{\boldsymbol{W}}_{vh}\boldsymbol{x}_t + \overrightarrow{\boldsymbol{W}}_{hh}\overrightarrow{\boldsymbol{h}}_{t-1})$

At each step t, from right to left

- $w_t \rightarrow \boldsymbol{x}_t$
- $\overleftarrow{\boldsymbol{h}}_t = f(\overleftarrow{\boldsymbol{W}}_{vh}\boldsymbol{x}_t + \overleftarrow{\boldsymbol{W}}_{hh}\overleftarrow{\boldsymbol{h}}_{t-1})$

For the prediction :

$$\boldsymbol{y}_t = g(\boldsymbol{W}_{ho}[\overrightarrow{\boldsymbol{h}}_t; \overleftarrow{\boldsymbol{h}}_t])$$

$[\overrightarrow{\boldsymbol{h}}_t; \overleftarrow{\boldsymbol{h}}_t]$ is a contextual representation of w_t

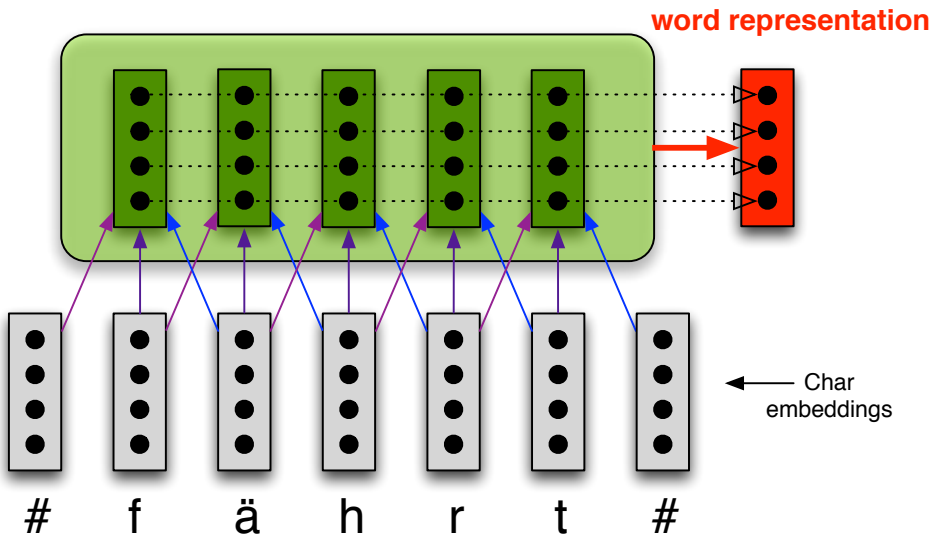
From characters to word representation

A word is a sequence of characters !

Sequence representation

- Recurrent network (Elman1990; Mikolov et al.2011)
- Convolutional network + pooling (Kalchbrenner et al.2014; Santos and Zadrozny2014)

Convolutional net + pooling



- A convolution net is applied at each position
- In 1-D, it mixes the inputs within a window (represent the local context)
- Max-pooling reduces this sequence in one vector

Putting all together

(Labeau et al.2015)

A unified model that can

- Infer word representation from the character level
- which takes sentence level information into account
- Make sequence prediction

Training

- Maximize the conditional log-likelihood of the tag sequence given the word sequence
- Optimization with AdaGrad (Duchi et al.2011)

Results

- This model achieves state of the art performance
- without any feature engineering

Outline

Summary

Neural Networks : how to efficiently represent language data

- To address the sparsity issue
- To deal with large (output) vocabulary
- To handle different kinds of contexts

Ongoing work

End-to-end neural system for complexe tasks:

- Automatic speech recognition
- Machine translation
- Summarization, ...

World Wide NLP

Everyone does not speak like the Wall Street Journal

All the other languages

- Access to resources is very uneven and patchy:

→ under-resourced languages

→ different morphological, syntactical and semantical properties

Freundschaftsbezeigungen, görüntülenebilir

Cultural heritage and digital humanities

- Allow users to communicate in their native language
- Preserve the language diversity
- Language studies

User generated content

User generated content

URaQT ;-) I<3U BFF CUL8R 4evER !!!

- Social Media implies fast change in the language use and writting style
- Spontaneous and noisy

→ An evolving vocabulary with new compound tokens, abbreviations, ...

Challenges

- Learning to decipher new texts
- NLP is not "context free"
- Linked-data processing (image, video, speech and sounds)

Thank you for your attention


Yoshua Bengio and Réjean Ducharme.
2001.
A neural probabilistic language model.
In Advances in Neural Information Processing Systems (NIPS), volume 13. Morgan Kaufmann.


Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin.
2003.
A neural probabilistic language model.
Journal of Machine Learning Research, 3:1137–1155.


Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai.
1992.
Class-based n-gram models of natural language.
Computational Linguistics, 18(4):467–479.


Stanley F. Chen and Joshua T. Goodman.
1998.
An empirical study of smoothing techniques for language modeling.
Technical Report TR-10-98, Computer Science Group, Harvard University.


John Duchi, Elad Hazan, and Yoram Singer.
2011.

Adaptive subgradient methods for online learning and stochastic optimization.
J. Mach. Learn. Res., 12:2121–2159, July.

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

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom.
2014.
A convolutional neural network for modelling sentences.
In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 655–665, Baltimore, Maryland, June. Association for Computational Linguistics.

Matthieu Labeau, Kevin Löser, and Alexandre Allauzen.
2015.
Non-lexical neural architecture for fine-grained pos tagging.
In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 232–237, Lisbon, Portugal, September. Association for Computational Linguistics.

Hai-Son Le, Ilya Oparin, Alexandre Allauzen, Jean-Luc Gauvain, and François Yvon.
2011.
Structured output layer neural network language model.

In Proceedings of ICASSP, pages 5524–5527.

Hai-Son Le, Alexandre Allauzen, and François Yvon.

2012.

Continuous space translation models with neural networks.

In Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pages 39–48, Montréal, Canada, June. Association for Computational Linguistics.

Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur.

2011.

Extensions of recurrent neural network language model.

In Proceedings of ICASSP, pages 5528–5531.

Andriy Mnih and Geoffrey E Hinton.

2008.

A scalable hierarchical distributed language model.

In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, Advances in Neural Information Processing Systems 21, volume 21, pages 1081–1088.

Thomas Mueller, Helmut Schmid, and Hinrich Schütze.

2013.

Efficient higher-order CRFs for morphological tagging.

In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 322–332, Seattle, Washington, USA, October. Association for Computational Linguistics.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams.

1986.

Parallel distributed processing: explorations in the microstructure of cognition, vol. 1.

chapter Learning internal representations by error propagation, pages 318–362. MIT Press, Cambridge, MA, USA.

Cicero D. Santos and Bianca Zadrozny.

2014.

Learning character-level representations for part-of-speech tagging.

In Tony Jebara and Eric P. Xing, editors, Proceedings of the 31st International Conference on Machine Learning (ICML-14), pages 1818–1826. JMLR Workshop and Conference Proceedings.

Holger Schwenk and Jean-Luc Gauvain.

2002.

Connectionist Language Modeling for Large Vocabulary Continuous Speech Recognition.

In Proceedings of ICASSP, pages 765–768, Orlando, May.