LM Evaluation and Neural LMs

Sandeep Soni

02/27/2024

Language model

- Language modeling is a task of estimating a probabilistic model over words
- If x is a sequence, we're trying to find P(x)

Language model

P(“to be or not to be”)

P(“or not to be to be”)

- LM can answer:
- P(x)
- Is P(x) > P(y)
- Is x fluent?

N-gram language models

$$P(x) = \prod_i P(x_i)$$

bigram

$$P(x) = \prod_i P(x_i \mid x_{i-1})$$

trigram

$$P(x) = \prod_i P(x_i \mid x_{i-2}, x_{i-1})$$

unigram

N-gram language models

$$P(x) = \prod_i P(x_i)$$

unigram

$$P(x) = \prod_i P(x_i \mid x_{i-1})$$

bigram

$$P(x) = \prod_i P(x_i \mid x_{i-2}, x_{i-1})$$

trigram

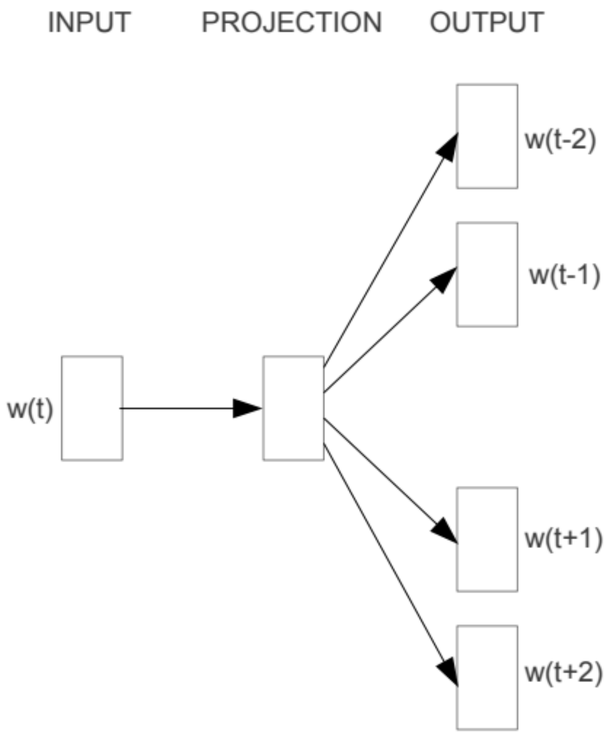
Evaluation

- A good language model
  - assigns high probability to unseen plausible sequences
  - should be less perplexed or surprised
- Perplexity = inverse probability of test data, averaged over words

Perplexity

- Perplexity = $2^{-\frac{l(w)}{M}}$, where M is the total number of unseen tokens
- $$l(w) = \sum_{m=1}^{M} \log P(w_m \mid w_{m-1}, \ldots, w_1)$$
- You'll calculate l(w) based on the LM of your choice
- Smaller perplexity is better!

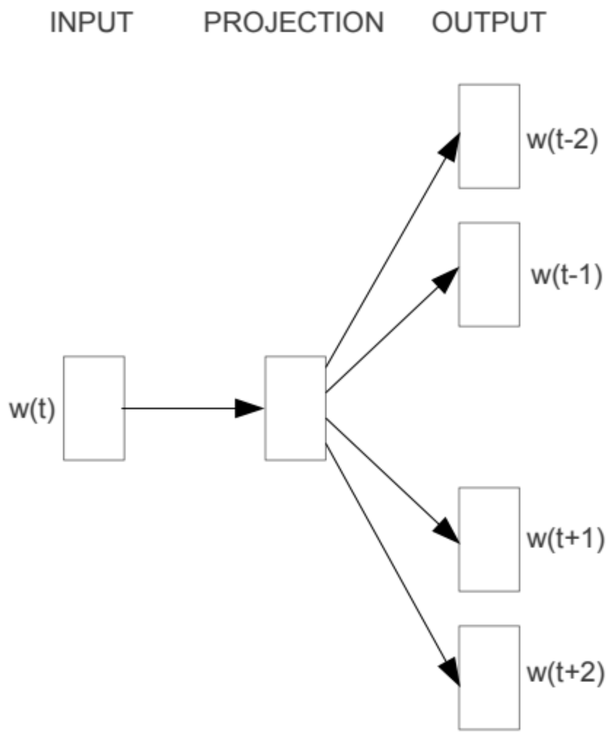
Skipgram



Skip-gram

- In one of the most sensational comebacks in the champions league Liverpool defeated Barcelona by mounting a strong performance in the second half

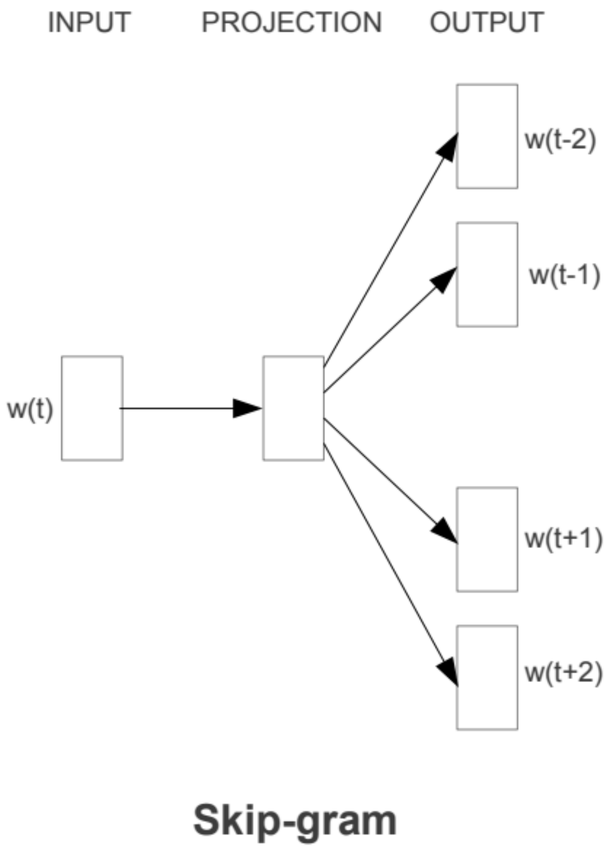
Skipgram



Skip-gram

- In one of the most sensational comebacks in the champions league Liverpool defeated Barcelona by mounting a strong _____ in the second half

Skipgram



Skip-gram

- In one of the most sensational comebacks in the champions league Liverpool defeated Barcelona by mounting a strong _____ in the second half

Word2vec learned word vectors using the skipgram language model

In class

- lm exploration

No Country for Old Members: User Lifecycle and Linguistic Change in Online Communities

Cristian Danescu-Niculescu-Mizil
Stanford University
Max Planck Institute SWS
cristiand@cs.stanford.edu

Robert West
Stanford University
west@cs.stanford.edu

Dan Jurafsky
Stanford University
jurafsky@stanford.edu

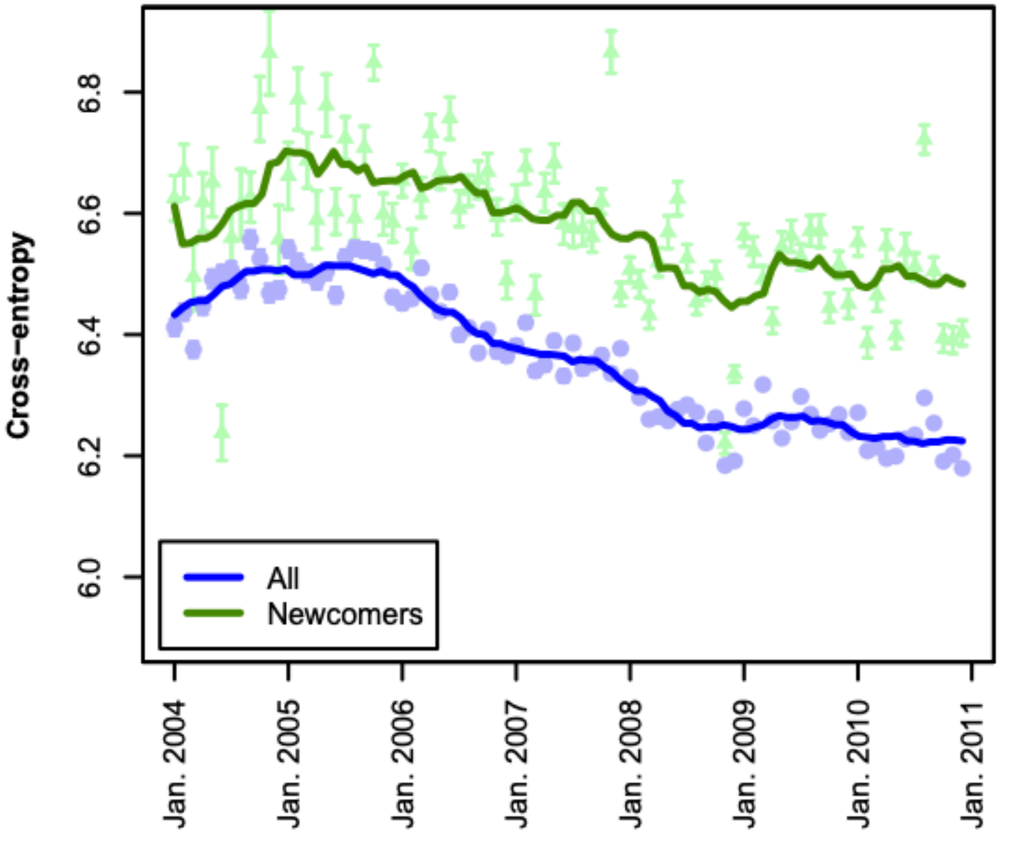
Jure Leskovec
Stanford University
jure@cs.stanford.edu

Christopher Potts
Stanford University
cgpotts@stanford.edu

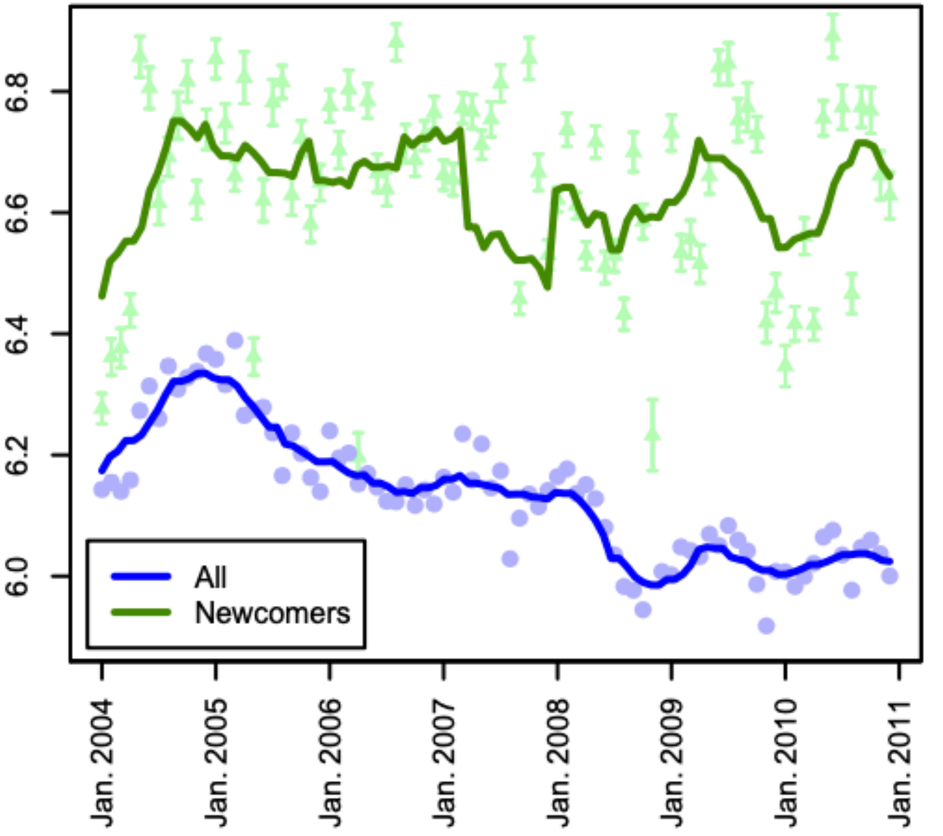
Who contributes to changing linguistic norms?

Snapshot language models

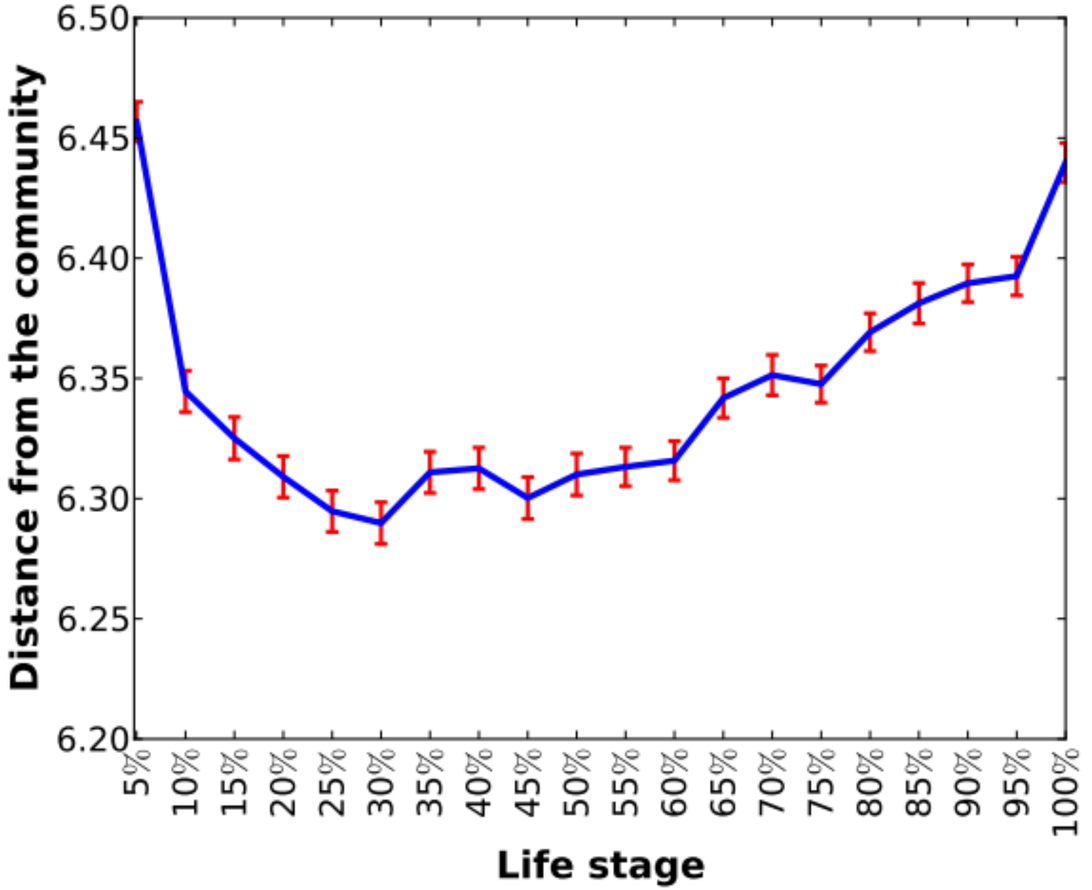
- Create a bigram language model SLM_m for every month m; this is called a snapshot language model
- Q. How surprising is a document with respect to any month?
- Calculate cross-entropy $H(p, SLM_m) = -\frac{1}{N}\sum_i \log P_{SLM_m}(b_i)$, where $b_1, b_2, \ldots, b_N$ are bigrams from the post

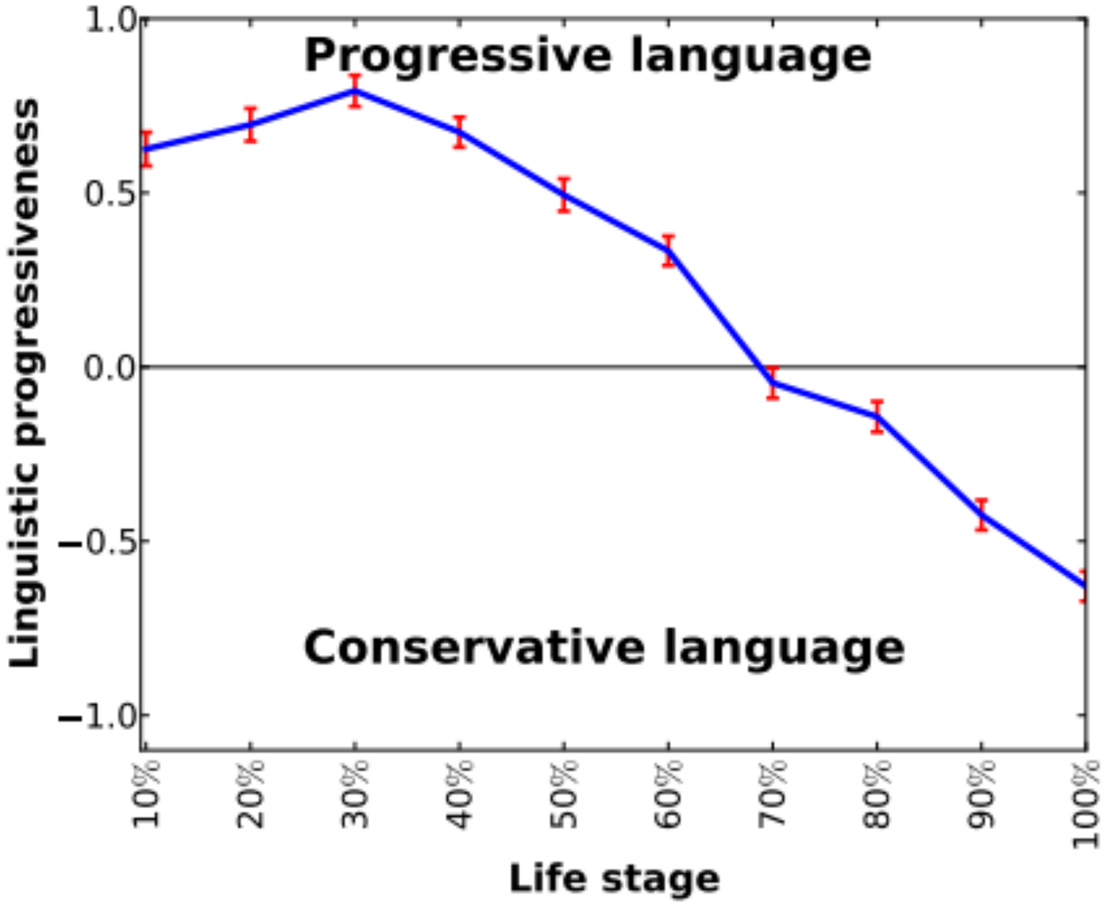


(a) BeerAdvocate



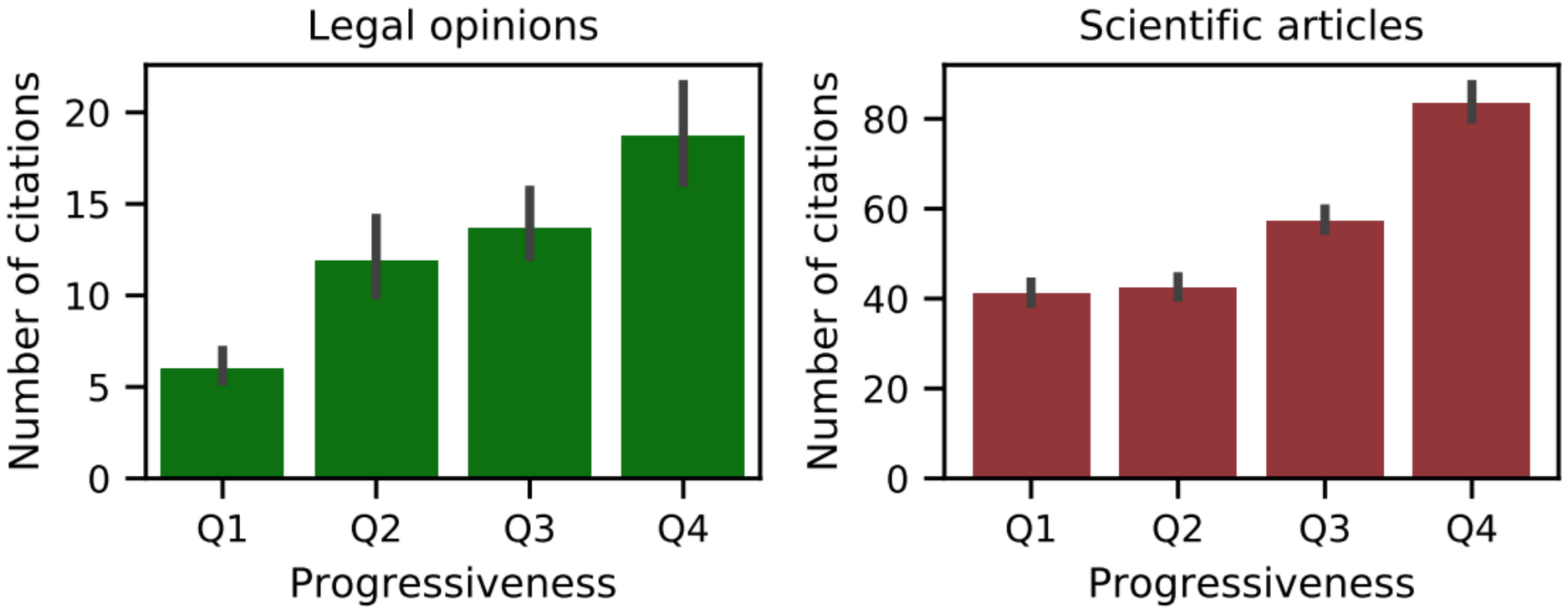
(b) RateBeer





- Users conform to the community's language initially but then stop adapting after some point

- Users are more innovative and trend-setting with their language use initially but then stabilize to rely increasingly more on past language



- A legal opinion or a scientific article is progressive if it uses words with meanings from the future than the past
- Learn two skipgram language models, one trained on future data and one trained on the past; then compare word senses with respect to both the models

Soni, Sandeep, Kristina Lerman, and Jacob Eisenstein. "Follow the leader: Documents on the leading edge of semantic change get more citations." *Journal of the Association for Information Science and Technology* 72.4 (2021): 478-492.

Can we do better than N-gram language models?

Generative Vs Discriminative

	Naive Bayes	Logistic Regression
Type of classifier	Generative	Discriminative
Model	$P(x,y)$	$P(y \mid x)$
Objective	Generate the data and label jointly	Predict the label from the data

Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors

Marco Baroni and **Georgiana Dinu** and **Germán Kruszewski**
Center for Mind/Brain Sciences (University of Trento, Italy)
(marco.baroni|georgiana.dinu|german.kruszewski)@unitn.it

Abstract

Context-predicting models (more commonly known as embeddings or neural language models) are the new kids on the distributional semantics block. Despite the buzz surrounding these models, the literature is still lacking a systematic comparison of the predictive models with classic, count-vector-based distributional semantic approaches. In this paper, we perform such an extensive evaluation, on a wide range of lexical semantics tasks and across many parameter settings. The results, to our own surprise, show that the buzz is fully justified, as the context-predicting models obtain a thorough and resounding victory against their count-based counterparts.

1 Introduction

A long tradition in computational linguistics has shown that contextual information provides a good approximation to word meaning, since semanti-

optimization process is generally unsupervised, and based on independent considerations (for example, context reweighting is often justified by information-theoretic considerations, dimensionality reduction optimizes the amount of preserved variance, etc.). Occasionally, some kind of indirect supervision is used: Several parameter settings are tried, and the best setting is chosen based on performance on a semantic task that has been selected for tuning.

The last few years have seen the development of a new generation of DSMs that frame the vector estimation problem directly as a supervised task, where the weights in a word vector are set to maximize the probability of the contexts in which the word is observed in the corpus (Bengio et al., 2003; Collobert and Weston, 2008; Collobert et al., 2011; Huang et al., 2012; Mikolov et al., 2013a; Turian et al., 2010). The traditional construction of context vectors is turned on its head: Instead of first collecting context vectors and then reweighting these vectors based on various criteria, the vector weights are directly set to optimally predict the contexts in which the corresponding

Learn representations from data by predicting parts of the data, instead of counting

Language modeling

- Instead of modeling P(x), why not model P(w|c)?
- Rather than directly estimating the word probabilities from relative frequencies, this hints at language modeling as a learning task

Language modeling

Context | Word

May the Force be with | ____

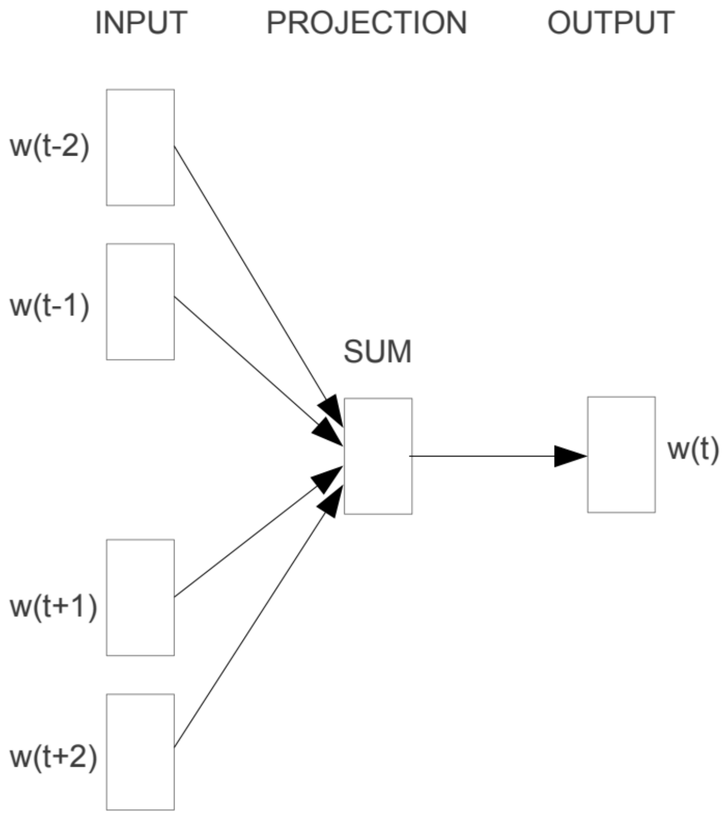
- Reparametrize the probability P(w|c) to depend on dense vectors
- $P(w \mid c) = \frac{\exp(\beta_w v_c)}{\sum_{w'} \exp(\beta_{w'} v_c)}$, where β_w is a vector representation of w and
- v_c is the vector representation of the context

Word2Vec (CBoW)

- $$P(w \mid c) = \frac{\exp(\beta_w v_c)}{\sum_{w'} \exp(\beta_{w'} v_c)}$$

- In CBoW model of word2vec, w is a word and c are words on the left and right of w

- v_c was calculated as a sum of output vectors



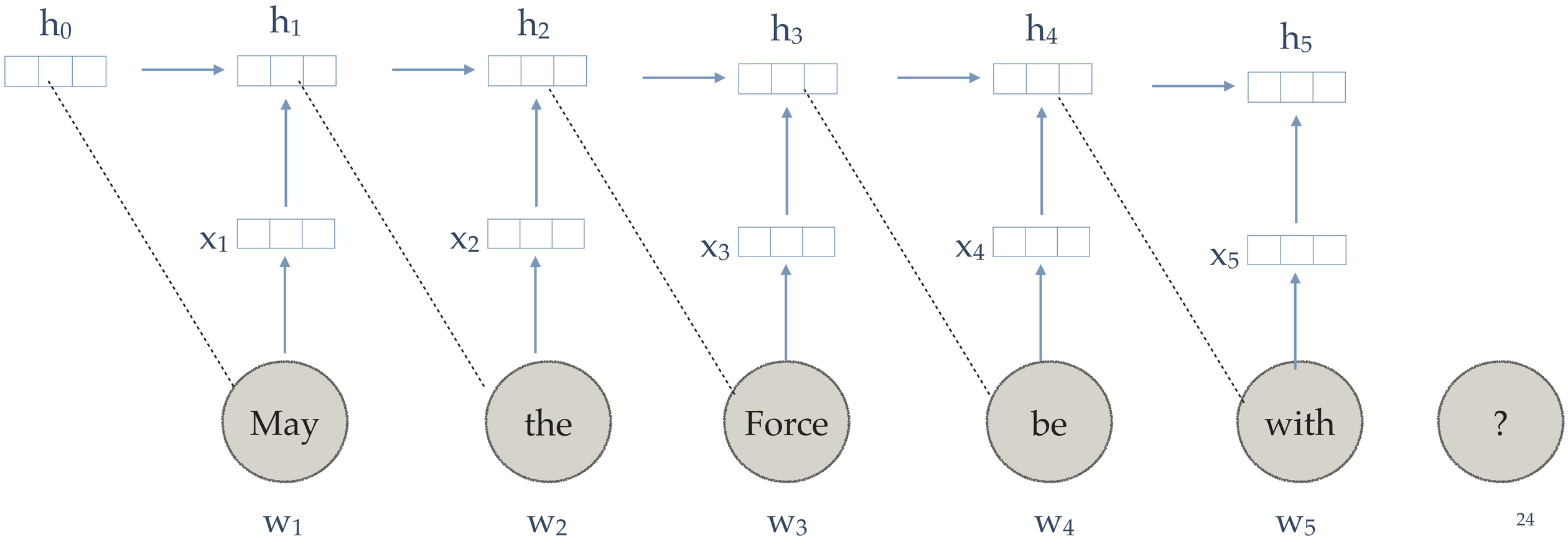
Mikolov et. al. 2013

What can be better ways of coming up with a vector representation of the context?

Context

Word

May the Force be with ____



Recurrent Neural Network LM

At every position m:

$$x_m = \text{Lookup}(\phi, w_m)$$

$$h_m = \text{RNN}(x_m, h_{m-1})$$

$$\boxed{h_m = g(\Theta h_{m-1} + x_m)} \quad \text{Elman unit}$$

$$P(w_{m+1} \mid w_1, w_2, \ldots, w_m) = \text{softmax}(\beta_{w_m}, \mathbf{h}_m)$$

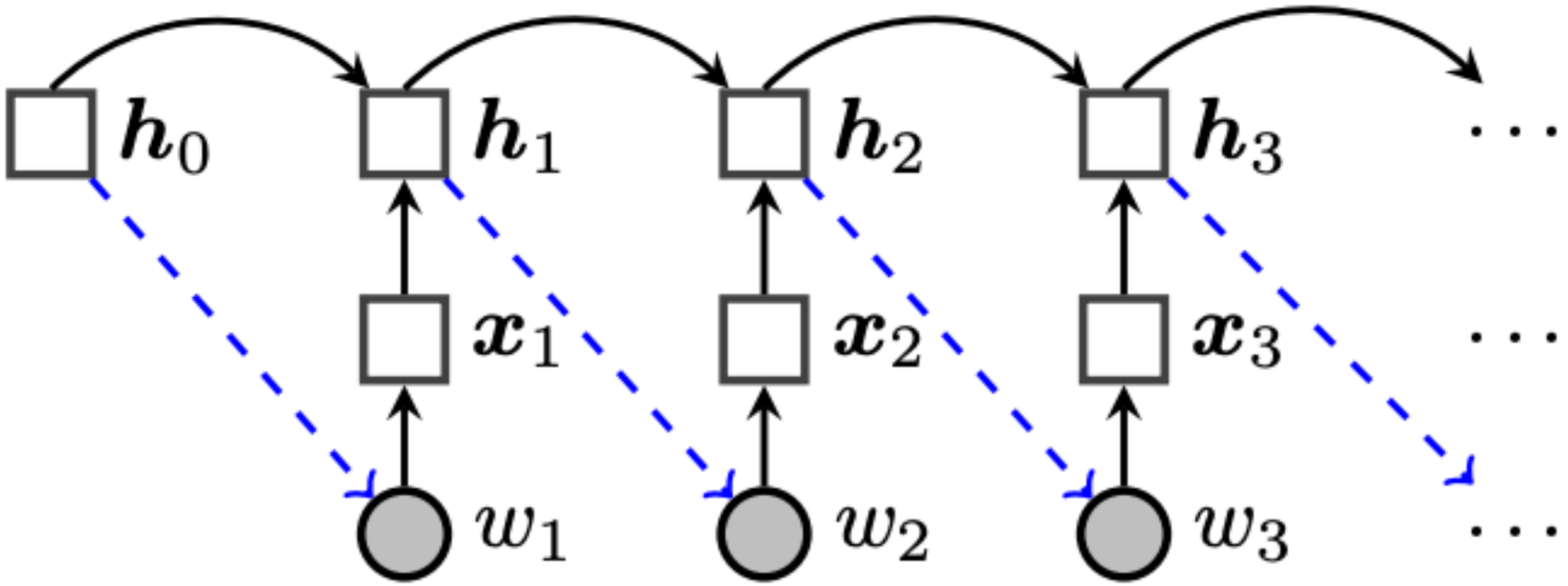
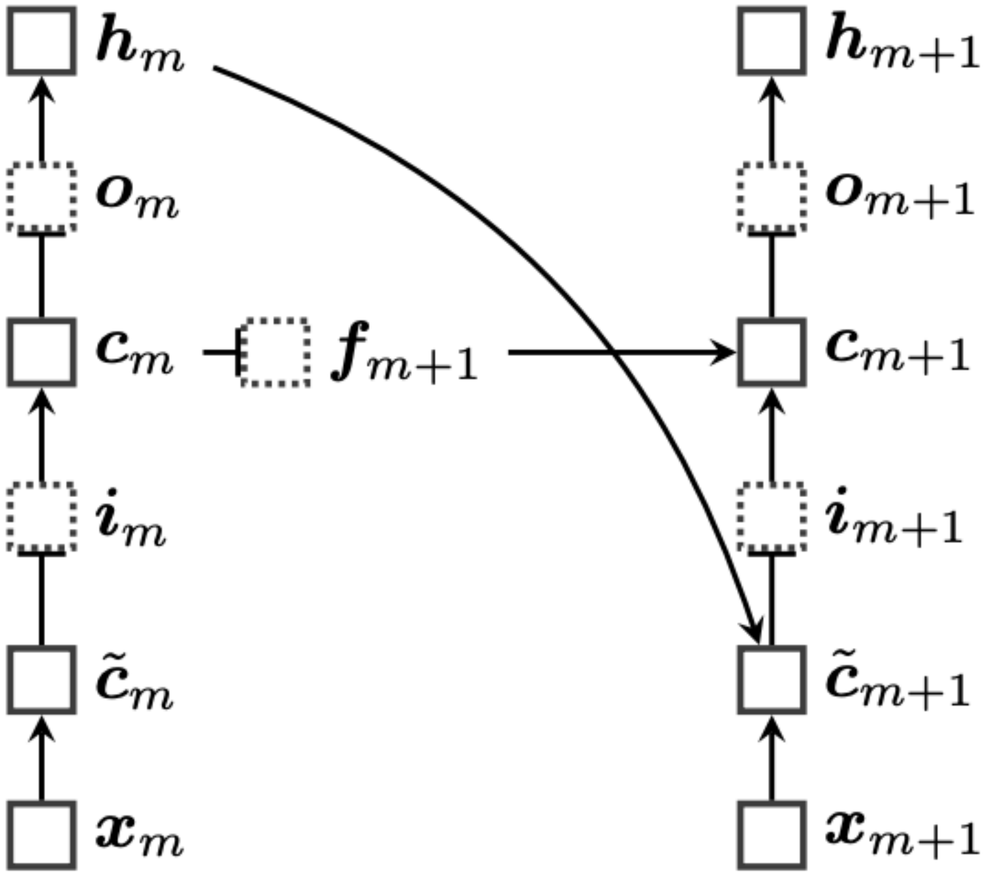


Figure taken from Eisenstein 2018

Long short-term Memories (LSTM)



- Transform x to h by passing x through gating units
- Preserves information propagation over long distances and downweights unimportant contexts in the past

Summary

- Language Modeling is a foundational task in text processing
- Count based language models are easy to interpret but not very powerful for longer sequences
- Neural LMs (RNNs, LSTMs) are extremely powerful
- Can we improve any further?