Consolidation week
Word Embeddings and End-to-end NLP

Venelin Kovatchev

Lecturer in Computer Science

v.o.kovatchev@bham.ac.uk

Outline

- Semantics
- Word embeddings
  - Count based
  - Word2vec
  - Elmo
- Compositionality and end-to-end

Mid-term exam

- Takes place on Monday (26th) in the lab session
- 10 multiple choice questions
- Cover weeks 1-5
- 20% of the final grade

Textbooks and materials for next few weeks

- Some parts of Speech and Language Processing
- Natural Language Processing with transformers (https://transformersbook.com/)
- Individual journal articles and conference papers

Semantics

What do the words mean?

- What does “A blue cat sat on the red mat” mean?
- What does “cat” mean?
- What does “a blue cat” mean?
- What does “the blue cat” mean?
- Now consider the sentence “Joanna sat on the red mat?”
  - What does “Joanna” mean?
  - Does the meaning change for someone who knows her compared with someone who doesn’t?

Some concepts of semantics

- The meaning of a linguistic unit can be
  - A list of properties (small, carnivore, four legged, has fur)
  - A set of all entities that contain said properties (“A cat” can refer to any and all cats)
  - An abstract concept: some undetermined cat performing the act of sitting
  - A concrete and determined entity (Simon’s cat, your friend Joanna)
  - A set of other related entities (e.g., a “cat” is a type of “animal”)

- And humans still manage to talk to and understand each other, talk about concepts they have never seen or entities that don’t exist (e.g., unicorns)

But wait! There is more

- How to go from the meaning of [“a blue cat”, “sat”, “on the red mat”] to a single meaning?
- What does “a blue cat sat on the red mat” mean?
  - It is not a cat, or a mat, or the color blue
  - It could be truth or a lie. Perhaps a blue cat has never sat on a red mat. Or blue cats don’t exist

What we have learned so far

- Semantics is the study of meaning
- How to represent meaning for algorithms?
- Do we need full theory of meaning?
  - Goal oriented representations

We are yet to find a way to get a full theory,
so we will restrict the condition.

Pop quiz

- Which of the following are meaning representations
  - Part-of-speech tag
  - Word2vec √
  - Syntactic tree → ignoring the meaning, syntax,
  - Tf-idf vector √ it's sparse
  - ELMO √
  - Wordnet synset

Vector representations and word embeddings

The distributional hypothesis and word embeddings

- The distributional hypothesis
  - What does it state?
  - How does it affect NLP?
- Word embeddings
  - Encoding words as “semantic” vectors

Count based vector representations

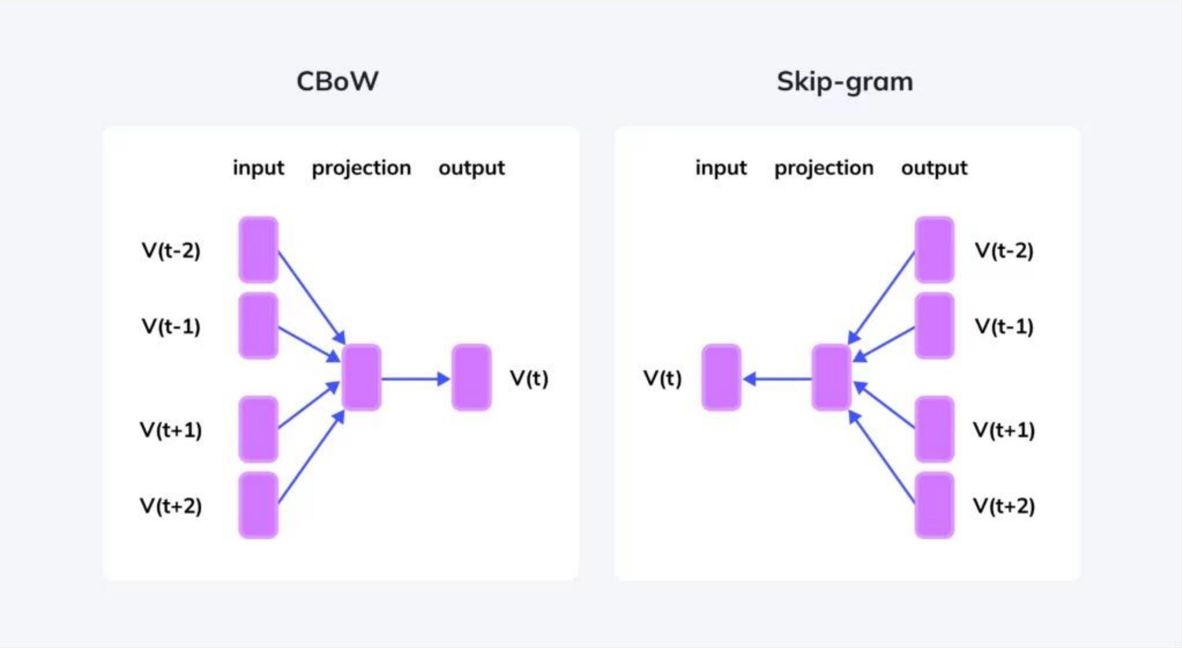
- Obtain a large corpus in the language/domain of interest
- Define a context of co-occurrence
  - What contexts can you think of?

the same word occurs. / single units
same document
sentence

- Count and fill in a co-occurrence matrix
- Apply transformations (which?)

Word2Vec

- Learning embeddings directly from text
- A simple neural architecture
- Two algorithms
  - What is the difference between them?



Pop Quiz

- Negative sampling is used to:
  - Improve fairness in classifiers
  - Process negation in text
  - Improve computational efficiency
  - Create dictionaries of negation
  - Reduce the effect of positive sampling

Negative Sampling

- Global objective: maximize the log probability of the dataset of size T with a context size c

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

- Calculate the probability of every word given context (or the other way around)

$$p(w_O|w_I) = \frac{\exp({v'_{w_O}}^T v_{w_I})}{\sum_{w=1}^{W}\exp({v'_w}^T v_{w_I})}$$

← This is computationally expensive

- Training with a softmax over the whole vocabulary is expensive

↓ so you change this.

- Convert the task into "classifying the correct objective" using a logistic

$$P(+|w,c) = \sigma(\mathbf{c}\cdot\mathbf{w}) = \frac{1}{1+\exp(-\mathbf{c}\cdot\mathbf{w})}$$

Calculating similarity

- How do we calculate similarity between vectors?

- Dot product

$$\text{dot product}(\mathbf{v},\mathbf{w}) = \mathbf{v}\cdot\mathbf{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N$$

- Cosine

$$\text{cosine}(\mathbf{v},\mathbf{w}) = \frac{\mathbf{v}\cdot\mathbf{w}}{|\mathbf{v}||\mathbf{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

Difference between count-based vectors and w2v

- Are count based vectors word embeddings?
- Are tf-idf vectors word embeddings?

↳ lots of dimensions, very sparse, this is $\boxed{\text{vector representation}}$

- Can you list some differences between count-based vectors and word embeddings?

ELMO, Word2Vec

ELMO

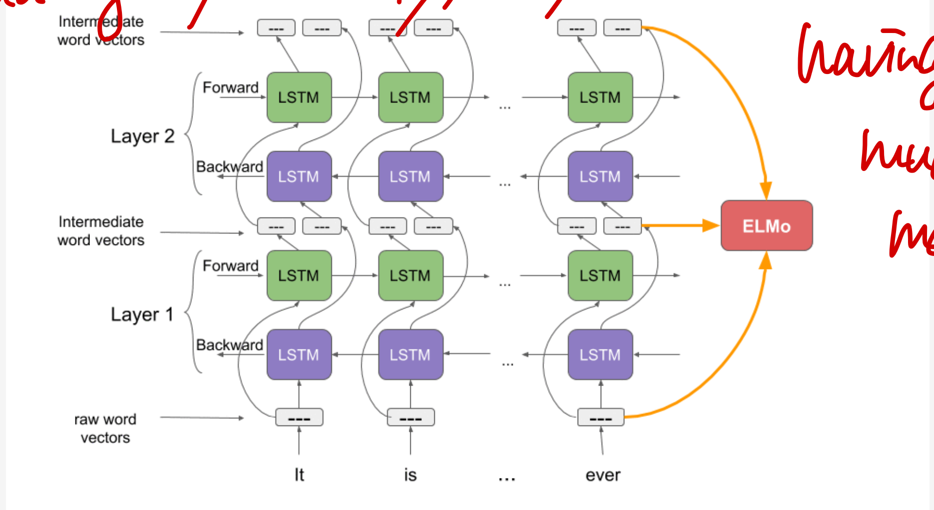
- What is the most important difference between ELMO and word2vec?

The embedding in the word2Vec is only for one word, because the vector is static. ELMO can handle the embedding dynamically, they can handle words having multiple meaning

- What makes ELMO representations "deep"?

ELMO uses all the representation

- What is the training objective behind ELMO?



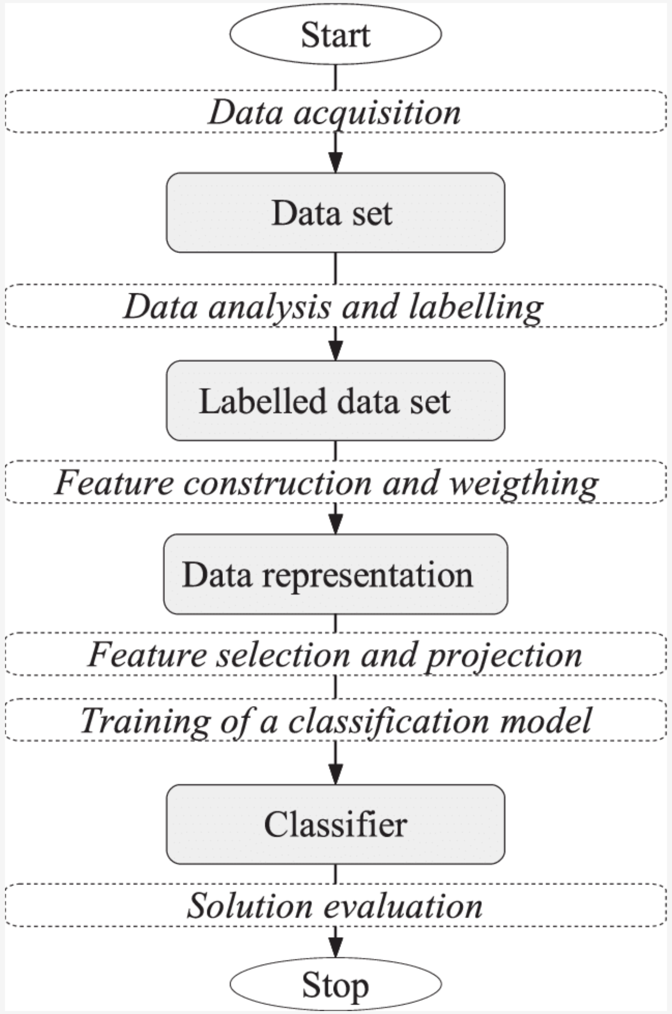
Compositionality and End-to-end

Feature engineering

- Analyze the problem, the input, and the desired outcome
- Explore existing resources and processing techniques
- Select the most relevant features and feature-extraction methods
- Empirically test what works best

Text classification using features

- Step by step process
- Involves active human engagement
  - Feature selection and extraction
- Data is fed into a classifier (Logistic, NB, SVM)
- Iteratively improve feature selection and model (hyper) parameters



Why not go end to end?

- Do we need full pipelines?
- Embeddings make it possible to "feed" text directly into models

Assign vectors as "features"

- Is it possible to go fully end-to-end and eliminate
  - Accumulation of errors
  - Human labor and supervision
  - (In)compatibility issues between elements?

Embeddings and the problem of compositionality

合成性

- Embeddings represent individual words
- NLP deals with processing texts

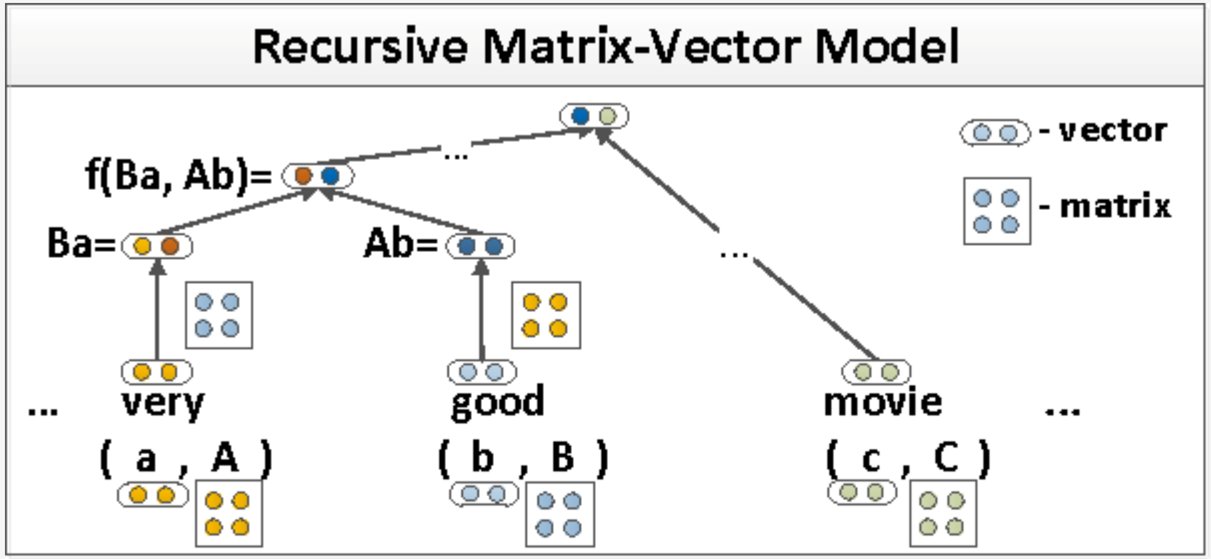
Concatenate

- How to go from word representations to text representations?
  - Can you list different compositional approaches?

Vector operations

- Vector addition
- Pointwise vector multiplication
- Vector concatenation
- Complex matrix-vector operations
- Which of these operations consider text structure?

Vector operations in general



concatenation

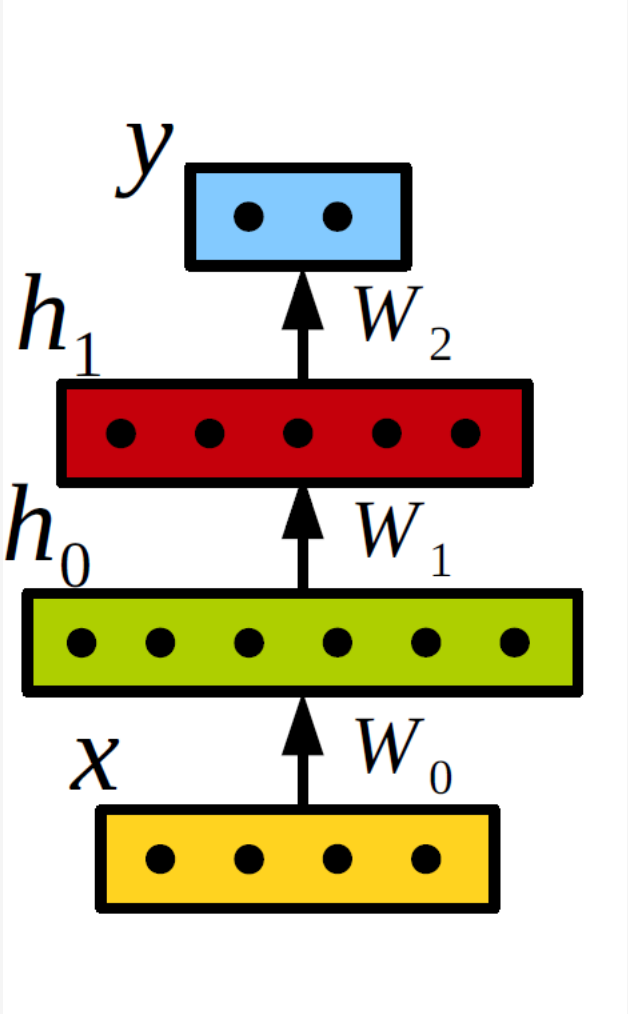
Compositionality using neural networks

- Convert words into word vectors
- Let the neural network learn compositionality
- What architectures have we discussed?

MLP, feed forward network

- What determines the “weights” of the compositionality?

Multi-layer perceptron

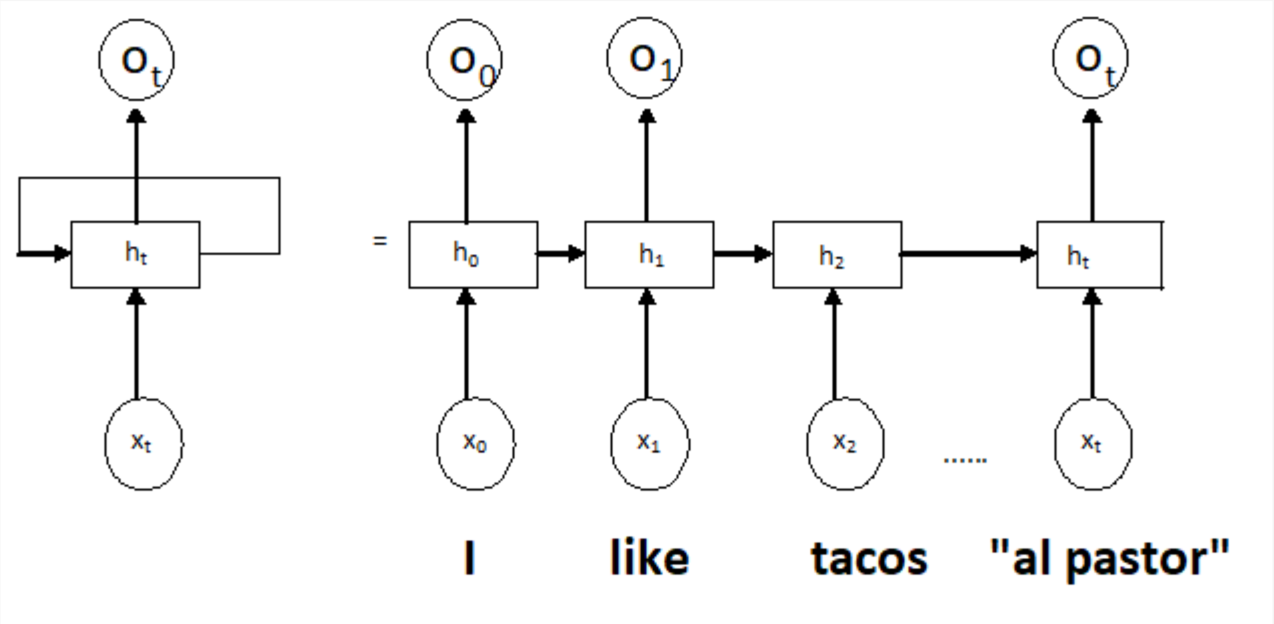


- $y = \text{softmax}(h_1 \cdot W_2 + b_2)$
- $h_1 = f(h_0 \cdot W_1 + b_1)$
- $h_0 = f(x \cdot W_0 + b_0)$
- Non-linear functions f:
  - Sigmoid: $\sigma(x) = \frac{1}{\exp(-x)}$
  - Hyperbolic: $\tanh(x) = \frac{1-\exp(-2x)}{1+\exp(-2x)}$
  - ReLU: $\text{rect}(x) = \max(0, x)$

Recurrent neural networks

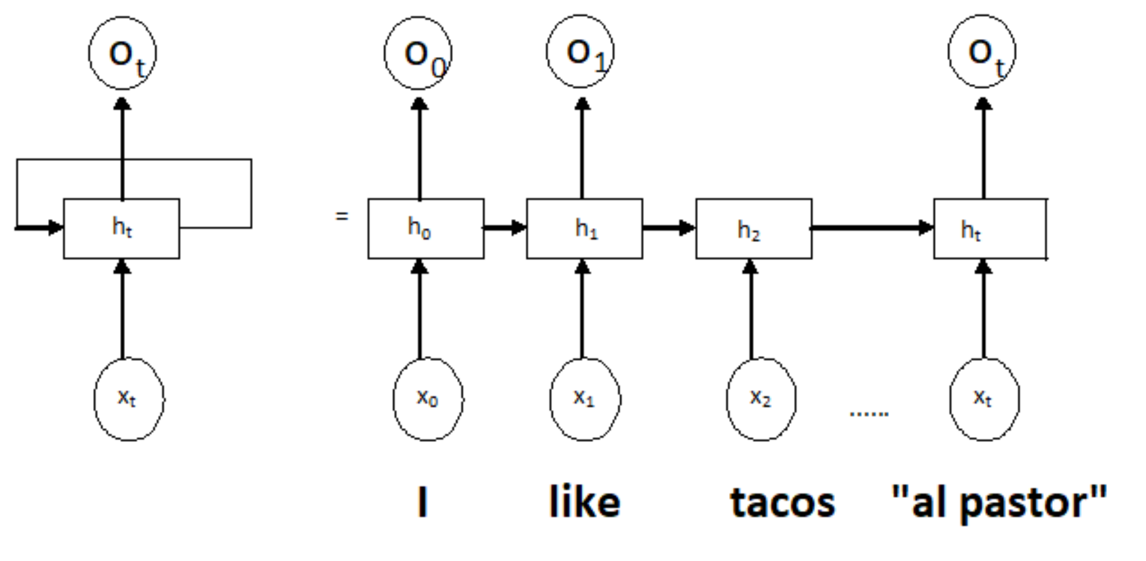
- How to represent text of varying length?
- Copy the network for each word
- At each timestep t, input is X_t and $h_{(t-1)}$
- I + like + tacos + "al pastor"
- Left to right, combining words one at a time
- Sequence classification
- Sequence to sequence

if a new word comes out, just copy it.



RNNs (formally)

- At each timestep
  - Input x_t, and previous hidden state $h_{(t-1)}$
  - Three sets of weights:
    - input (W_x), hidden (W_h) , and output (W_o)
  - $h_t = f_h(W_x x_t + W_h h_{(t-1)})$
  - $O_t = f_o(W_o h_t)$
  - $J_t = f(O_t, y_t)$
- Pop-quiz: What are we predicting at O?



tagging, labelling,

Neural language modeling with RNNs

- More formally:
  - $h_t = \tanh(W_x x_t + W_h h_{(t-1)})$
  - $\hat{y}_t = \text{softmax}(W_y h_t)$
  - $J_t = -\log \hat{y}_{t,\text{ correct}}$ (- log prob the word at t+1)
- $J_{sent} = \frac{1}{T}\sum_{t=1}^{T} -log\ \hat{y}_{t,\text{ correct}}$
- Pop quiz: What is "weight tying" in RNNs?

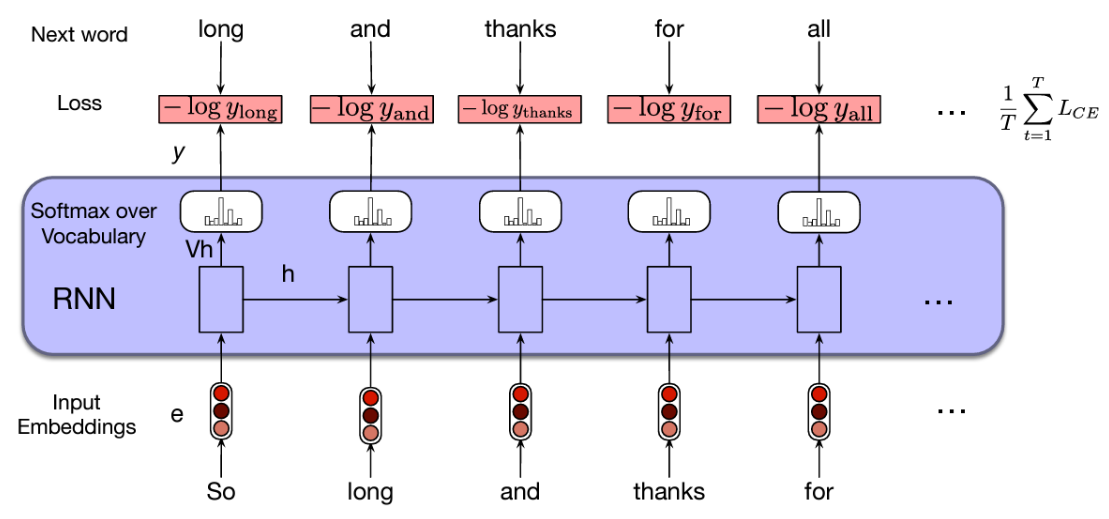


Figure 9.6 from SLP, chapter 9

Stacked and bidirectional RNNs

- The basic RNN is a powerful tool
- We can go deeper
- Stacking RNNs
- Bidirectional RNNs

LSTM

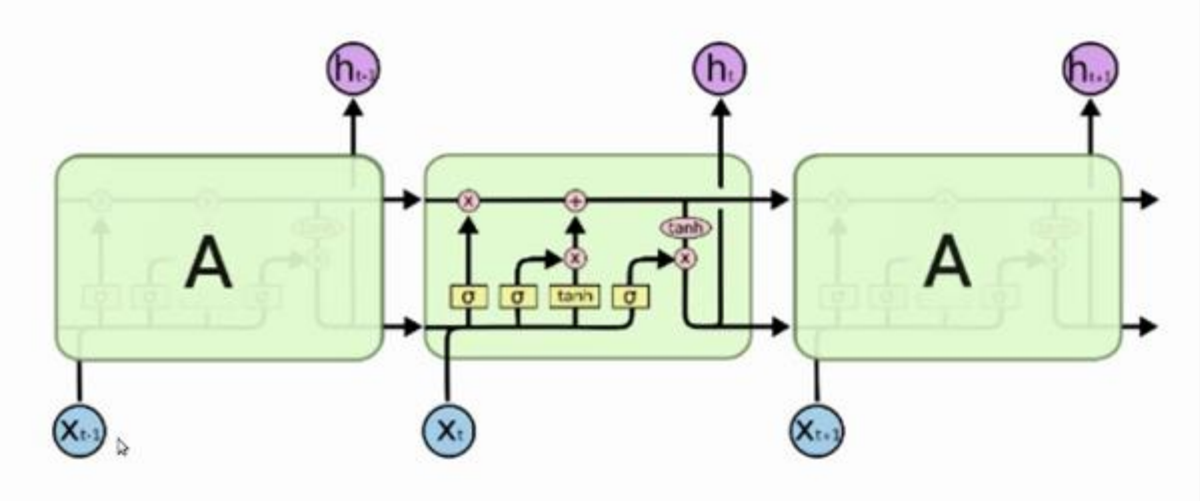
- Popular variation of RNN that address native limitations
- Same recurrent concept (copy the network)
- Three different "gates":
  - Forget gate
  - Input gate
  - Output gate

distinguishing the info

- Gates manipulate the flow of information through time
  - Addressing conflicting objectives

Understanding the gates

- All gates have the same format:
  - A feedforward layer followed by a sigmoid
- The gates are filtering out information at a certain part of the network
- Each gate has two important aspects:
  - How to calculate the filter
  - What is the input that the filter is applied to
- Pop quiz: which memory we use to calculate gates?



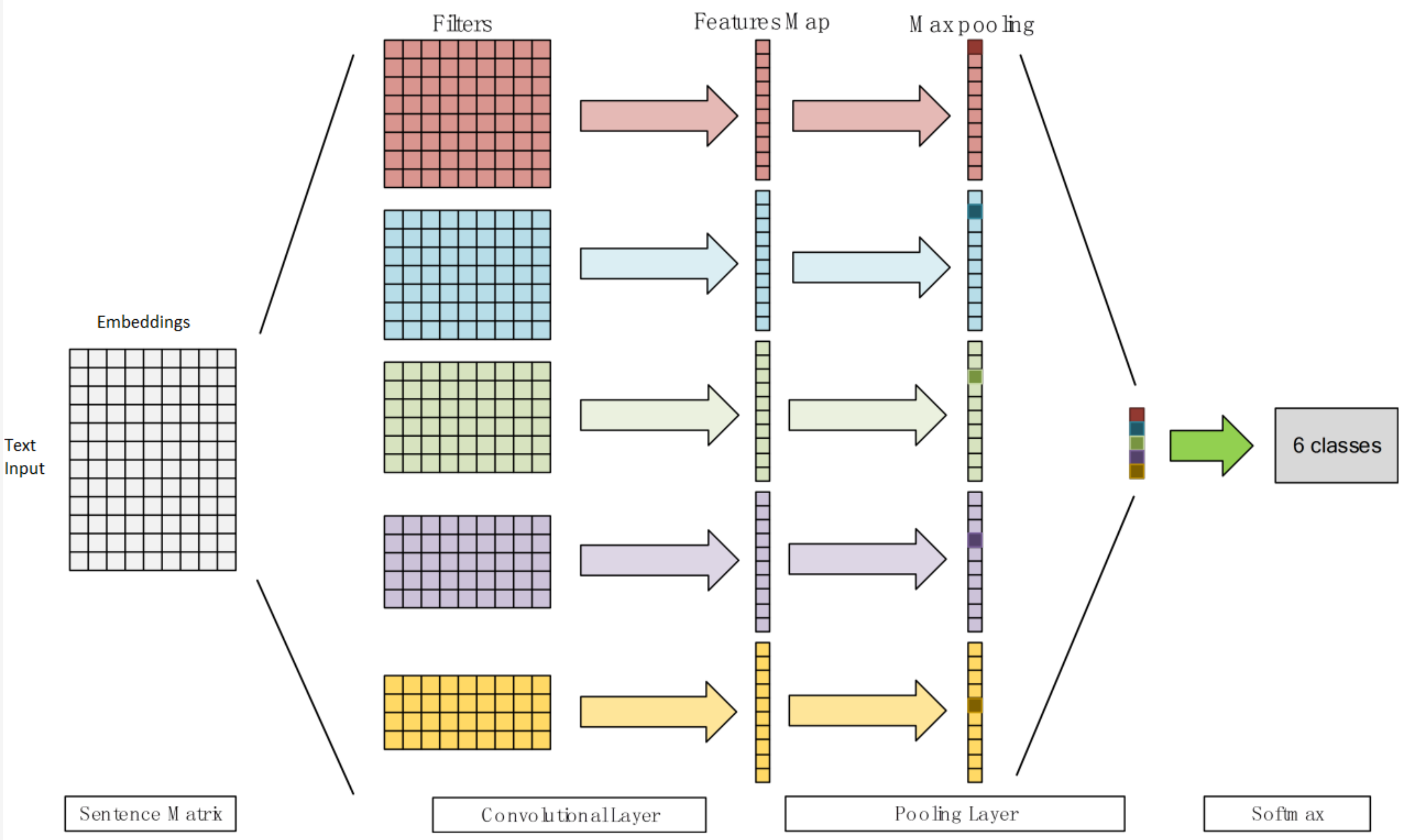
Short memory .

What we have learned so far

Convolutional neural networks (CNN)

- Another popular architecture for NLP
  - Deals with text of various size
- Inspired by computer vision success
- Applying filters of different size to the input (2,3,4) + pooling
  - Analogous to n-grams

Convolutional neural networks (CNN)



Pop quiz: input size

- Can each of these architecture process text of infinite size
  - RNN
  - MLP
  - CNN
  - LSTM
  - Bi-LSTM

you have to define the input

Compositionality

- How each network "composes" meaning
  - Feed-forward network

    doesn't care about the word order

    - Linear combination of words (no order) + activation function
  - RNN/LSTM
    - Recursively, word by word (linear order)
  - CNN
    - Locally, similar to n-gram (proximity window, limited order importance)

    it cares only words within a couple of words can interact.

Multimodality and multilinguality

- Embeddings and multilinguality
  - Words of any language can be mapped to vectors
  - Words of different languages can be mapped to the same space
  - Mapping and similarity across languuages
- Embeddings and multimodality
  - Images, sound, and other modalities can also be mapped to vectors (e.g., using CNNs)
  - Shared multimodal spaces (vision + language)

The first “should I worry about my job”

- Rapid change in technologies can be stressful

- Three “major” milestones in NLP in the past 10 years
  - Word2Vec and end-to-end models (BiLSTM, CNN) – 2013
  - BERT (and the transformer family) – 2018
  - Large generative language models (GPT3, ChatGPT, Bard) – 2022 - 2023

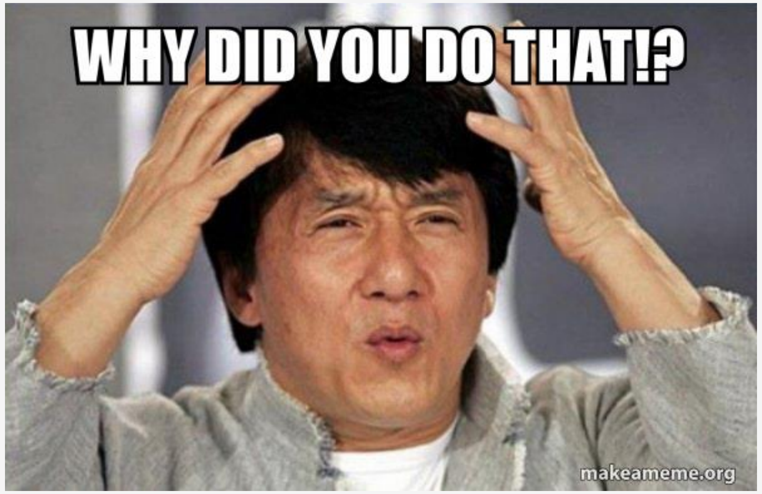
- There are still many unsolved problems

Some concerns

Explainability and Interpretability

- Interpreting feature-based models
  - Feature values (“v1agra”) + weights = prediction (“spam”)

- Interpreting end-to-end neural networks
  - Feature values (300d dense vector)
  - weights (input, forget, output gates)
  - different types of nonlinearity



Explainability and Interpretability

- Why did you do that?
- How did you do that? What makes you say that?
- How can I verify your results?

Bias, Guarantees, and Robustness

- Does the algorithm discriminate?
- Does the algorithm contain bias with respect to race, gender, religion, sexual orientation?
- Does the algorithm guarantee consistent and robust performance?
- Is the algorithm secure from adversarial attacks?

New fields of study in NLP in the are of deep learning

- Explainability of neural networks
- Algorithmic fairness
- Evaluation, unit testing, and adversarial attacks for NLP
- Data centric AI

Conclusions

Embeddings in NLP

- Embeddings changed the way we do NLP
- Valuable stand-alone resource
  - Can be used to query lexical information
  - Can be used as automatically extracted features
  - Can be used for a simple text representation
  - Static and dynamic embeddings, polisemy
- Enable end-to-end neural models

End-to-end neural models

- Minimizing human interaction and supervision
  - Remove the need of feature engineering
  - Only require labeled data for classification tasks
- Improving efficiency and removing accumulation of errors in pipeline
- Enabling full training without depending on external resources

End-to-end neural models

- Introducing new challenges and problems
  - Increased computational complexity
  - Need for more data
  - Error accumulation becomes internal
  - Difficult to interpret and debug
  - Potentially containing biases
  - Ultimately re-invent many of the pipeline parts during training