COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Introduction to Natural Language Processing

CMSC 470

Marine Carpuat

Final Exam

- Friday December 13, 1:30-3:30pm, EGR 1104

- You can bring one sheet of notes (double sided okay)
- Exam structure
  - True/False or short answer problem similar to homework quizzes
  - 2 or 3 longer problems where you are expected to show your work

- Cumulative exam, but with more focus on topics covered after the midterm

Topics

- Words and their meanings
  - Distributional semantics and word sense disambiguation
  - Fundamentals of supervised classification
- Sequences
  - N-gram and neural language models
  - Sequence labeling tasks
  - Structured prediction and search algorithms
- Application: Machine Translation
- Trees
  - Syntax and grammars
  - Parsing

What you should know: Dense word embeddings

- Dense vs. sparse word embeddings

- How to generating word embeddings with Word2vec
  - Skip-gram model
  - Training

- How to evaluate word embeddings
  - Word similarity
  - Word relations
  - Analysis of biases

What you should know
Machine Translation

- Context: Historical Background
  - Machine Translation is an old idea, its history mirrors history of AI
  - Why is machine translation difficult?
    - Translation ambiguity
    - Word order changes across languages
  - Translation model history: rule-based -> statistical -> neural

- Machine Translation Evaluation
  - What are adequacy and fluency
  - Pros and cons of human vs automatic evaluation
  - How to compute automatic scores: Precision/Recall and BLEU

What you should know: Recurrent Neural Network Languge Models

- Mathematical definition of an RNN language model
- How to train them

- Their strengths and weaknesses
  - Have all the strengths of feedforward language model
  - And do a better job at modeling long distance context
  - However
    - Training is trickier due to vanishing/exploding gradients
    - Performance on test sets is still sensitive to distance from training data

What you should know: Neural Machine Translation

- How to formulate machine translation as a sequence-to-sequence transformation task
- How to model P(E|F) using RNN encoder-decoder models, with and without attention
- Algorithms for producing translations
  - Ancestral sampling, greedy search, beam search
- How to train models
  - Computation graph, batch vs. online vs. minibatch training
- Examples of weaknesses of neural MT models and how to address them
  - Bidirectional encoder, length bias
- Determine whether a NLP task can be addressed with neural sequence-to-sequence models

What you should know: POS tagging & sequence labeling

- POS tagging as an example of sequence labeling task
- Requires a predefined set of POS tags
  - Penn Treebank commonly used for English
  - Encodes some distinctions and not others
- How to train and predict with the structured perceptron
  - constraints on feature structure make efficient algorithms possible
  - Unary and markov features => Viterbi algorithm
- Extensions:
  - How to frame other problems as sequence labeling tasks
  - Viterbi is not the only way to solve the argmax: Integer Linear Programming is a more general solution

What you should know: Dependency Parsing

- Interpreting dependency trees
- Transition-based dependency parsing
  - Shift-reduce parsing
  - Transition systems: arc standard, arc eager
  - Oracle algorithm: how to obtain a transition sequence given a tree
  - How to construct a multiclass classifier to predict parsing actions
  - What transition-based parsers can and cannot do
  - That transition-based parsers provide a flexible framework that allows many extensions
    - such as RNNs vs feature engineering, non-projectivity (but I don’t expect you to memorize these algorithms)
- Graph-based dependency parsing
  - Chu-Liu-Edmonds algorithm
  - Stuctured perceptron

Where we started on the 1st day of class

- Levels of linguistic analysis in NLP
  - Morphology, syntax, semantics, discourse

- Why is NLP hard?
  - Ambiguity
  - Sparse data
    - Zipf's law, corpus, word types and tokens
    - Variation and expressivity
  - Social Impact

Ambiguity and Sparsity

- What are examples of NLP challenges due to ambiguity/sparsity?
- What are techniques for addressing ambiguity/sparsity in NLP systems?

Linguistic Knowledge

- How is linguistic knowledge incorporated in NLP systems?

Example: Adding attention in an encoder-decoder model

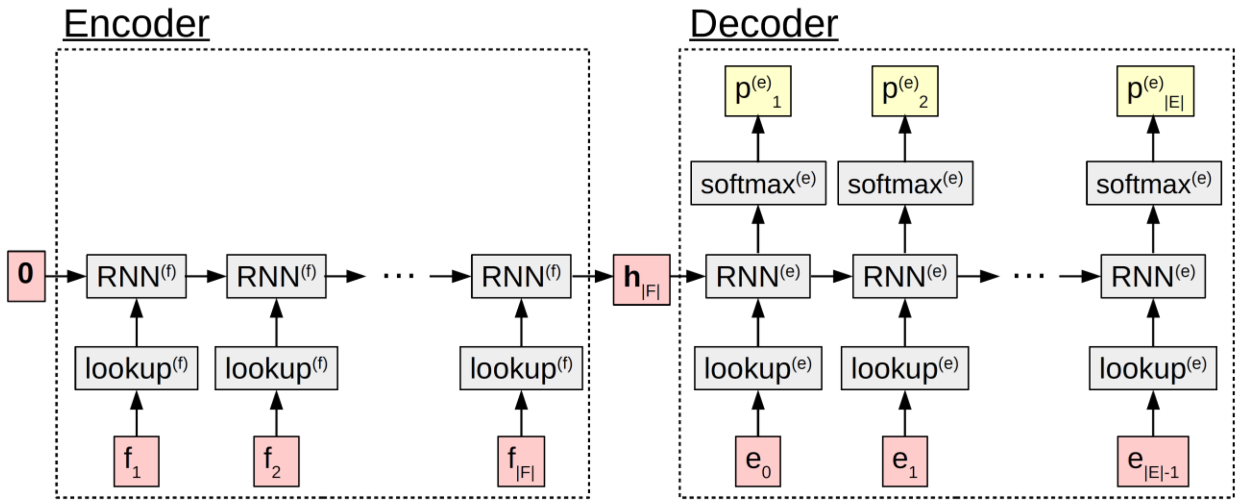
$$\boldsymbol{m}_t^{(f)} = M_{\cdot,f_t}^{(f)}$$

$$\boldsymbol{h}_t^{(f)} = \begin{cases} \text{RNN}^{(f)}(\boldsymbol{m}_t^{(f)}, \boldsymbol{h}_{t-1}^{(f)}) & t \geq 1, \\ \boldsymbol{0} & \text{otherwise.} \end{cases}$$

$$\boldsymbol{m}_t^{(e)} = M_{\cdot,e_{t-1}}^{(e)}$$

$$\boldsymbol{h}_t^{(e)} = \begin{cases} \text{RNN}^{(e)}(\boldsymbol{m}_t^{(e)}, \boldsymbol{h}_{t-1}^{(e)}) & t \geq 1, \\ \boldsymbol{h}_{|F|}^{(f)} & \text{otherwise.} \end{cases}$$

$$\boldsymbol{p}_t^{(e)} = \text{softmax}(W_{hs}\boldsymbol{h}_t^{(e)} + b_s)$$



Attention model: Create a source context vector for each time step t

$$\boldsymbol{c}_t = H^{(f)} \boldsymbol{\alpha}_t.$$

Context vector

Attention vector

- Attention vector:
  - Entries between 0 and 1
  - Interpreted as weight given to each source word when generating output at time step t

Attention model
How to calculate attention scores

$$\boldsymbol{h}_t^{(e)} = \mathrm{enc}([\mathrm{embed}(e_{t-1}); \boldsymbol{c}_{t-1}], \boldsymbol{h}_{t-1}^{(e)}).$$

$$a_{t,j} = \mathrm{attn\_score}(\boldsymbol{h}_j^{(f)}, \boldsymbol{h}_t^{(e)}).$$

$$\boldsymbol{\alpha}_t = \mathrm{softmax}(\boldsymbol{a}_t).$$

$$\boldsymbol{p}_t^{(e)} = \mathrm{softmax}(W_{hs}[\boldsymbol{h}_t^{(e)}; \boldsymbol{c}_t] + b_s).$$

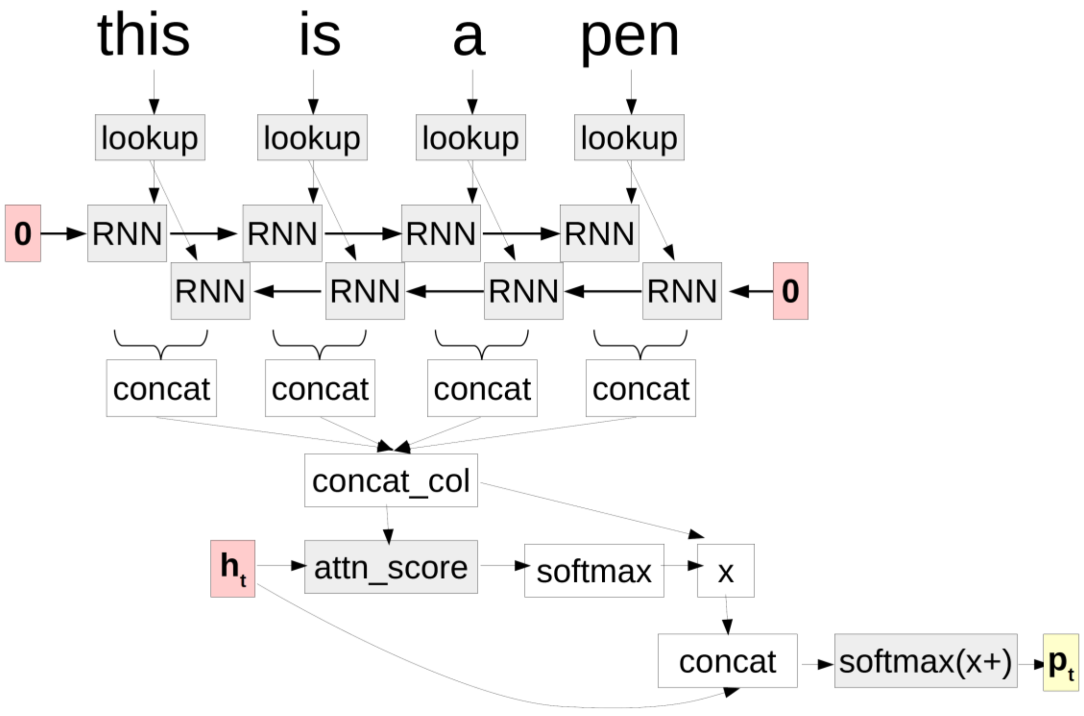


Figure 28: A computation graph for attention.

Attention model
Various ways of calculating attention score

- Dot product

$$\mathrm{attn\_score}(\boldsymbol{h}_j^{(f)}, \boldsymbol{h}_t^{(e)}) := \boldsymbol{h}_j^{(f)\intercal} \boldsymbol{h}_t^{(e)}.$$

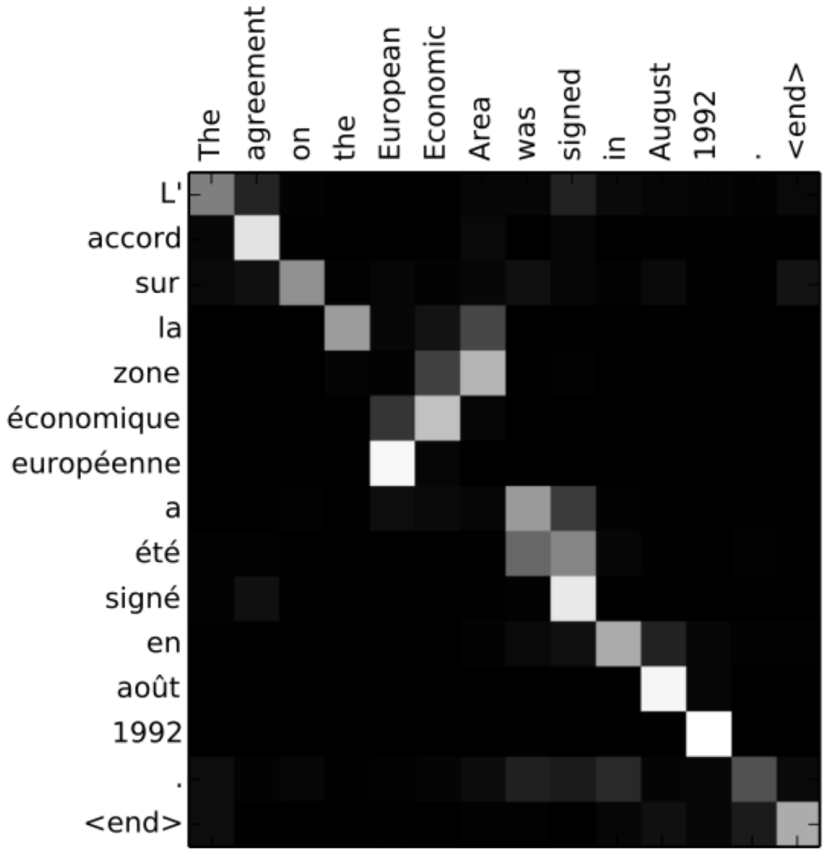
- Bilinear function

$$\mathrm{attn\_score}(\boldsymbol{h}_j^{(f)}, \boldsymbol{h}_t^{(e)}) := \boldsymbol{h}_j^{(f)\intercal} W_a \boldsymbol{h}_t^{(e)}.$$

- Multi-layer perceptron (original formulation in Bahdanau et al.)

$$\mathrm{attn\_score}(\boldsymbol{h}_t^{(e)}, \boldsymbol{h}_j^{(f)}) := \boldsymbol{w}_{a2}^{\intercal} \tanh(W_{a1}[\boldsymbol{h}_t^{(e)}; \boldsymbol{h}_j^{(f)}])$$

Attention model
Illustrating attention weights



NLP tasks often require predicting structured outputs

- What kind of output structures?
- Why is predicting structures challenging from a ML perspective?
- What techniques have we learned for addressing these challenges?

Structured prediction trade-offs in dependency parsing

Transition-based

- Locally trained
- Use greedy search algorithms
- Define features over a rich history of parsing decisions

Graph-based

- Globally trained
- Use exact (or near exact) search algorithms
- Define features over a limited history of parsing decisions

Structured prediction trade-offs in sequence labeling

Multiclass Classification at each time step

- Locally trained
- Make predictions greedily
- Can define features over history of tag predictions

Sequence labeling with structured perceptron

- Globally trained
- Use exact search algorithms
- Define features over a limited history of predictions

Consider this new NLP task

How would you build a system for this task?

- Goal: verify information using evidence from Wikipedia.
- Input: a factual claim involving one or more entities (resolvable to Wikipedia pages)
- Outputs:
  - the system must extract textual evidence (sets of sentences from Wikipedia pages) that support or refute the claim.
  - Using this evidence, label the claim as **Supported**, **Refuted** given the evidence or **NotEnoughInfo**.

Claim: The Rodney King riots took place in the most populous county in the USA.

`[wiki/Los_Angeles_Riots]`

The 1992 Los Angeles riots, also known as the Rodney King riots were a series of riots, lootings, arsons, and civil disturbances that occurred in Los Angeles County, California in April and May 1992.

`[wiki/Los_Angeles_County]`

Los Angeles County, officially the County of Los Angeles, is the most populous county in the USA.

Verdict: Supported

This is the shared task of the Fact Extraction and Verification (FEVER) workshop

You can see what solutions researchers came up with here:

http://fever.ai/task.html

Social Impact

- NLP experiments and applications can have a direct effect on individual users' lives

- Some issues
  - Privacy
  - Exclusion
  - Overgeneralization
  - Dual-use problems

- What are examples of each of these issues in NLP systems?

[Hovy & Spruit ACL 2016]

Some ways to keep learning

- CLIP talks (Wed 11am) http://go.umd.edu/cliptalks
- Language Science Center http://lsc.umd.edu
- Read research papers (e.g., from ACL and EMNLP conferences)
  - ACL anthology is a good starting point to search NLP papers
- Build your own system for shared tasks
  - E.g., yearly SemEval evaluations, Kaggle
- Podcasts:
  - NLP Highlights covers recent papers and trends in NLP research
  - Lingthusiam covers a very wide range of linguistic topics https://lingthusiasm.com/
  - Talking Machines: "Human Conversations about Machine Learning" https://www.thetalkingmachines.com