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 Language 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
  - Structured 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

\[ m_t^{(f)} = M_{:,f_t} \]

\[ h_t^{(f)} = \begin{cases} 
\text{RNN}^{(f)}(m_t^{(f)}, h_{t-1}^{(f)}) & t \geq 1, \\
0 & \text{otherwise.} 
\end{cases} \]

\[ m_t^{(e)} = M_{:,c_{t-1}} \]

\[ h_t^{(e)} = \begin{cases} 
\text{RNN}^{(e)}(m_t^{(e)}, h_{t-1}^{(e)}) & t \geq 1, \\
h_{t-1}^{(F)} & \text{otherwise.} 
\end{cases} \]

\[ p_t^{(e)} = \text{softmax}(W_h h_t^{(e)} + b_s) \]
Attention model: Create a source context vector for each time step \( t \)

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

\[
c_t = H^{(f)} \alpha_t.
\]
Attention model
How to calculate attention scores

\[ h_t^{(e)} = \text{enc}([\text{embed}(e_{t-1}); c_{t-1}], h_{t-1}^{(e)}). \]

\[ a_{t,j} = \text{attn\_score}(h_j^{(f)}, h_t^{(e)}). \]

\[ \alpha_t = \text{softmax}(a_t). \]

\[ p_t^{(e)} = \text{softmax}(W_{hs}[h_t^{(e)}; c_t] + b_s). \]

Figure 28: A computation graph for attention.
Attention model
Various ways of calculating attention score

• Dot product

$$\text{attn\_score}(h_j^{(f)}, h_t^{(e)}) := h_j^{(f)\top} h_t^{(e)}.$$ 

• Bilinear function

$$\text{attn\_score}(h_j^{(f)}, h_t^{(e)}) := h_j^{(f)\top} W_a h_t^{(e)}.$$ 

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

$$\text{attn\_score}(h_t^{(e)}, h_j^{(f)}) := w_{a2}^\top \tanh(W_{a1}[h_t^{(e)}; h_j^{(f)}]).$$
Attention model
Illustrating attention weights

The agreement on the European Economic Area was signed in August 1992.

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

- 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