# Revisit the NLP Pipeline

Bonan Min

bonanmin@gmail.com

Some slides are based on class materials from Ralph Grishman

# The Pipeline

Extracting information from natural language text is a complex process

We have been able to make it manageable by dividing it into many separate stages, each realized by its own (relatively simple) model

# Pipeline Problems

The pipeline raises two serious problems:

Error compounding

- because stages are computed separately

Need to define intermediate representations

- laborious
- suboptimal
- task-specific

# Introducing Errors

Each stage introduces some errors because

Model is an oversimplification of linguistic phenomenon

(Hand-prepared) training data may be noisy

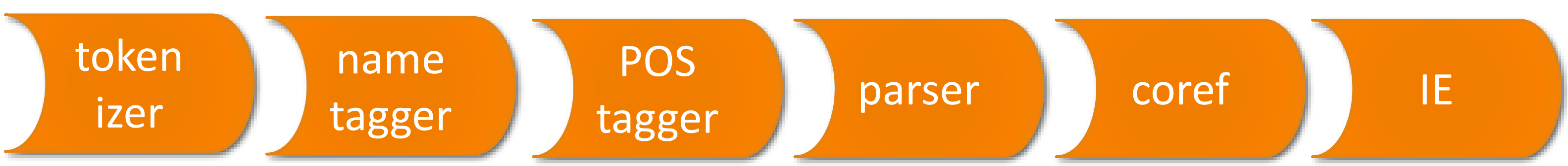
Typically 10% error rate per stage

- error rates range from 3% (POS) to 15% (name tagging)

# Compounding Errors

Errors in output of stage =
errors due to faulty input +
errors introduced by stage

Final output error rate > 50% (Is this useful?)




10% 20% 30% 40% 50% 60%

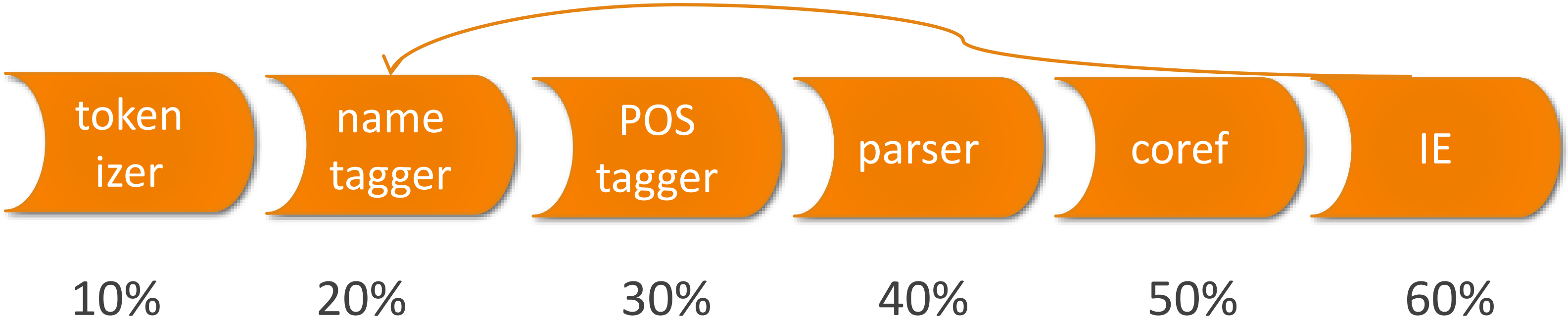
# Helpful Feedback

We will reduce the error rate by in effect providing feedback from later stages to earlier ones

Example: *"Roger Park began to work for IBM."*

NE tagger says "*Roger Park*" is most likely a *location* but could also be a *person*

relation extraction pattern (in IE stage) indicates a preference for "*person* works" over "*location* works", fixing the error



# Joint Inference

To perform joint inference between stages A and B,

- we define an objective function combining A and B
- we search the combined space of A and B
  - where possible B outputs may depend on A output
  - seeking to maximize combined objective
  - much larger search space than with independent components

# More Examples

"Meet me in front of the White House."

- "White House" may refer to the building or the organization therein
  - 1-token context used by NE doesn't help resolve ambiguity
- The relation extractor determines this is a reference to building

"Ford employs 4000 in Detroit."

- The event extractor determines that 4000 is of type people

# Implementation and Benefits

Implementations

- A joint framework based on structured prediction so that the local predictions can be mutually improved
  - Extracts event triggers and arguments together (Li et al., ACL 2013)
  - Extract entity mentions and relations together (Li et al., ACL 2014)
- Efficient beam search; maintaining multiple hypothesis along the pipeline

**Benefit**: For 2 or 3 stages, reductions of 2 - 3% (absolute) in error rate are reported

- can only correct errors which change a valid (more likely) input to second stage to an invalid (less likely) input

# Cost

Much larger space to search

- Full search of product space infeasible
- Joint token-by-token scan updating multiple models (NE, relation, event) concurrently
  - use beam search to limit search space
    - follow only top n hypotheses at each token

    OR

    - follow only hypotheses within m% of best hypothesis

OR

- Build graphical model connecting stages
    - soft constraints linking stages

# Deep Learning

Instead of training separate models for each stage and then coupling them, can we train a unified model

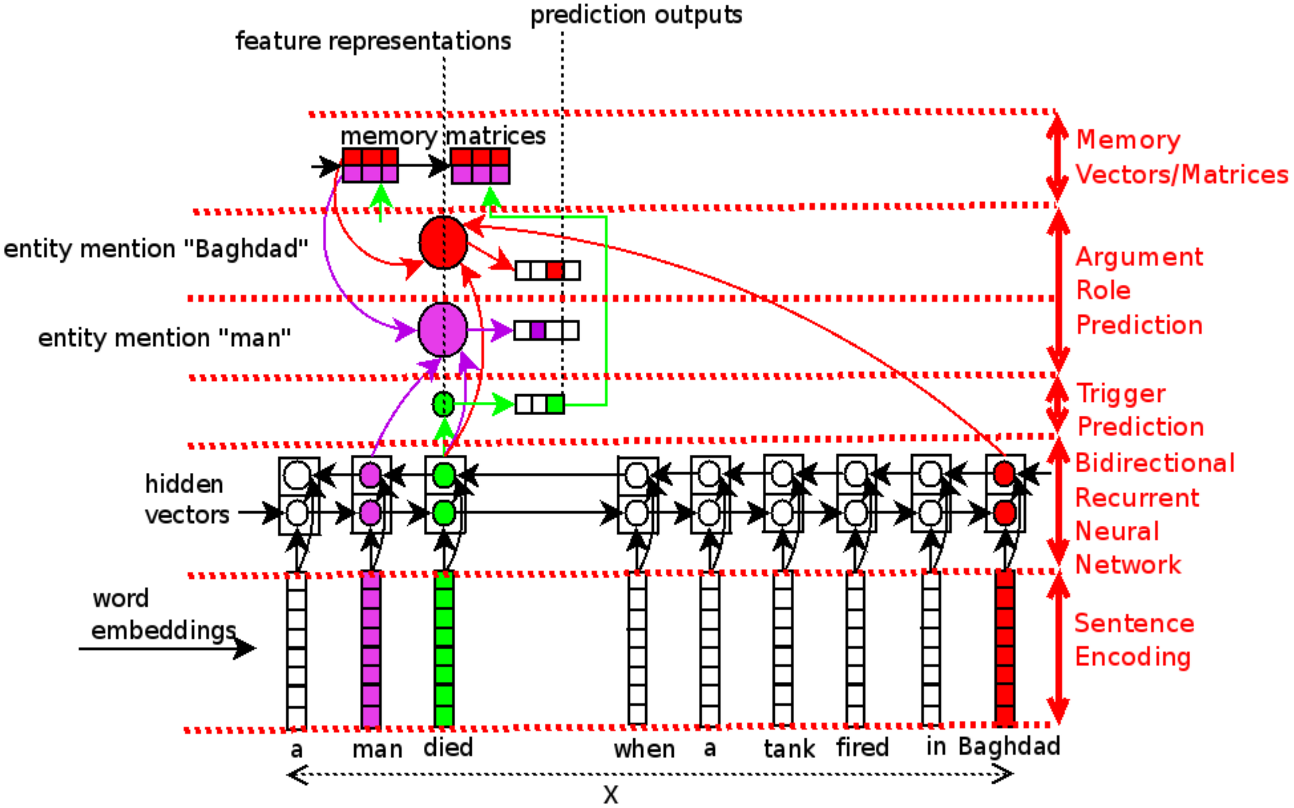
- to perform the entire analysis starting from a sequence of tokens
- To tie all tasks together with multi-layer neural networks

A more powerful model

- A multi-level network can represent a wide range of models
  - compared to the log-linear models

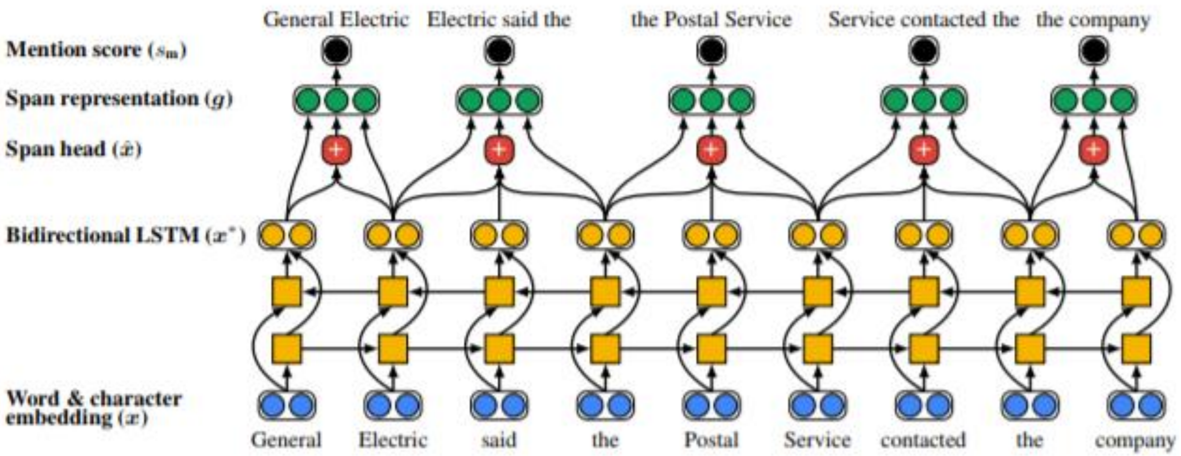
# Example 1

Joint Trigger and Argument Extraction via RNN (Nguyen et al., NAACL-HLT 2016)

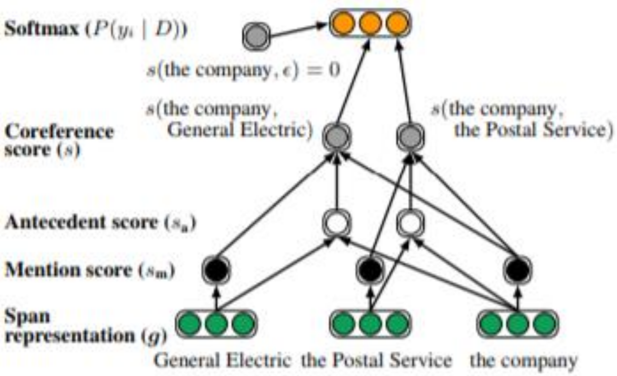


# Example 2

End-to-end neural coreference resolution (Lee et al., EMNLP 2017)



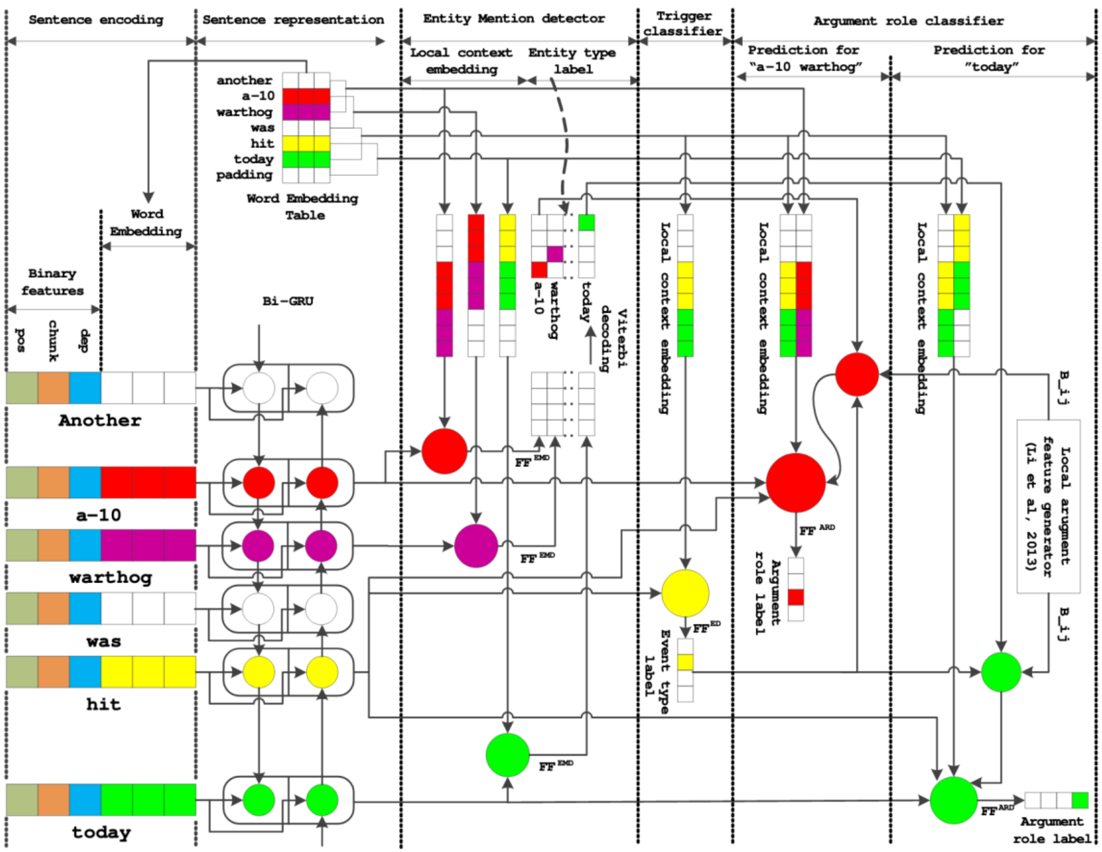

Span representation




Coreference

# Example 3

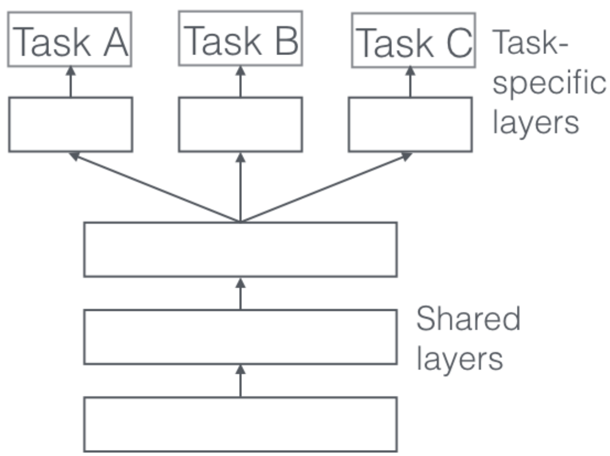
Neural Joint Modeling of Entities and Events (Nguyen and Nguyen, AAAI 2019)



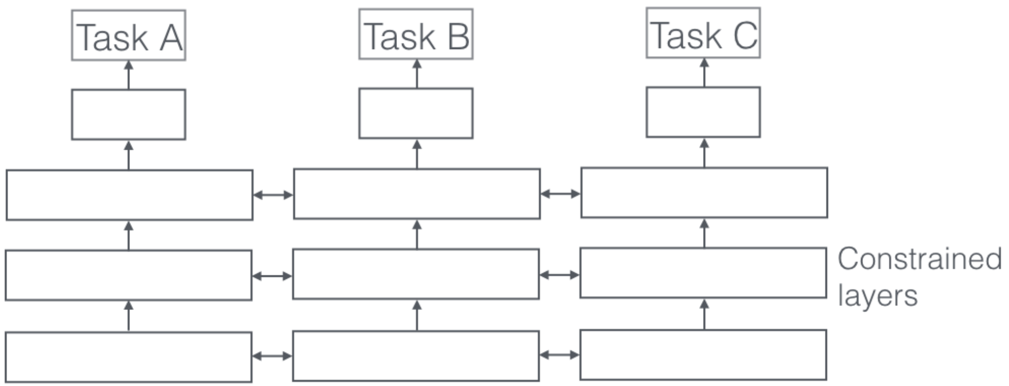
# Multi-task Learning in DNN

Related tasks:

- POS, Entity recognition, coreference, relation extraction, events extraction, event argument attachment, Semantic Role Labeling (SRL), etc.




Hard parameter sharing




Soft parameter sharing

# More Successes

Improved speech recognition

- combining acoustic and language models

Integrated NLP pipelines

- *Natural Language Processing (Almost) from Scratch* (Collobert et al JMLR 2011)
- competitive performance with less feature engineering

Better relation extraction

- with less feature engineering

Better machine translation

# The Future?

Robust systems

Little or no feature engineering

Large-scale self-supervised LM pre-training + task-specific tuning
- BERT is everywhere

Input: characters
- Zhang et al., Character-level convolutional networks for text classification. NIPS 2015.

Output: triples linked to KB
- Lots of work in (entity) linking, knowledge representation, KB completion