NLP1 Projects 17/18

Projects

We offer **6** projects

Teams of **3** will be divided over the projects

Sign up after this lecture (link on Blackboard)

Each project requires implementing a **model** and writing a **report**

Implementations in **Python** & **PyTorch**

We cannot offer **GPUs** - but our projects can also be completed on **CPU**

List of Projects

1. Visual Question Answering
2. Image Retrieval from Dialogue
3. Neural Graph-based Dependency Parsing
4. Neural Language Modeling
5. Finding Grammar in Neural Language Models
6. What do Neural Machine Translation models encode?

Visual Question Answering

Who is wearing glasses?

man woman

Where is the child sitting?

fridge arms

Is the umbrella upside down?

yes no

How many children are in the bed?

2 1

Visual Question Answering - Given

Subset of VQA Dataset

Preprocessed Image features

PyTorch Introduction

Visual Question Answering - Deliverables

1. Analyze & Preprocess language data
2. Implement the following approaches
    a. BoW
    b. Bag of Words + NLP features
    c. RNN

    BONUS:

    d. Implementation of SoA models
    e. Your very own idea
3. Error Analysis of Models
4. Writer scientific report about your findings

Image Retrieval from Dialogue

Task Definition:

- **Input:** The MSCOCO caption of the target image, followed by a dialogue which discusses the image.
- Predict the correct image from a line up of 5/10 images.
- Levels:
  - Easy (One where it should be possible to just guess with the caption)
  - Hard? (Where the images can be chosen from the l2 distance of VGG features or have similar objects(from MSCOCO data)

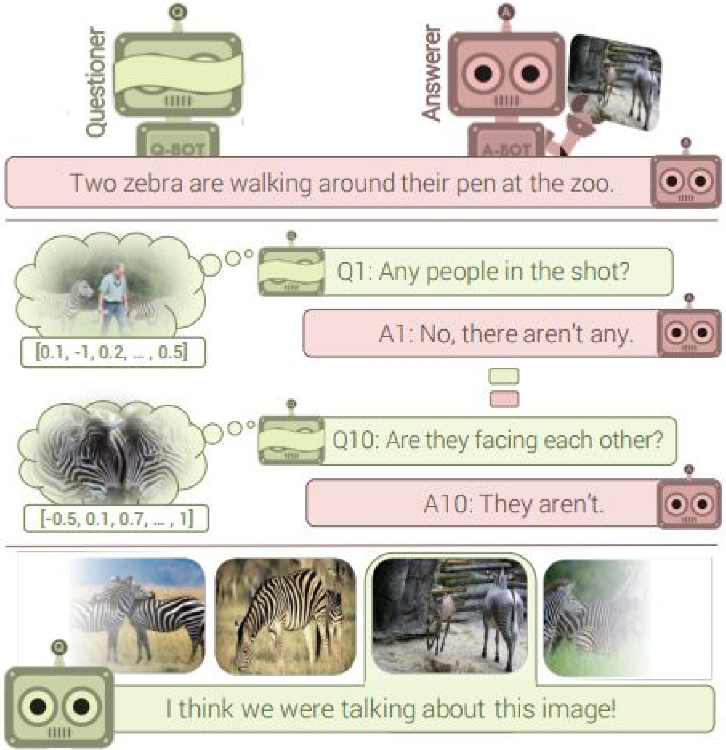


Image Retrieval from Dialogue - Given

Subset of VisDial dataset

Preprocessed Image features

PyTorch Introduction

Image Retrieval from Dialogue - Deliverables

1. Analyze & Preprocess language data
2. Implement the following approaches
   a. BoW
   b. Bag of Words + NLP features
   c. RNN

   BONUS:

   d. Explore various ways to combine the image and text features
   e. Your very own idea
3. Error Analysis of Models
4. Writer scientific report about your findings

Further Reading

VQA

http://visualqa.org/index.html

http://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html

Image Retrieval from Language

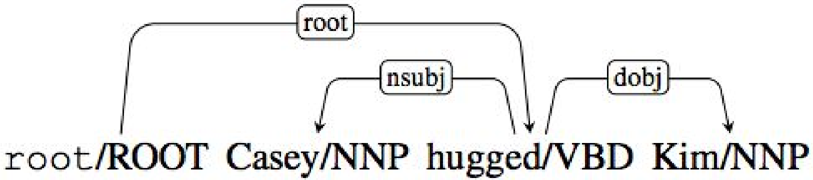
https://visualdialog.org/

Neural Graph-based Dependency Parsing

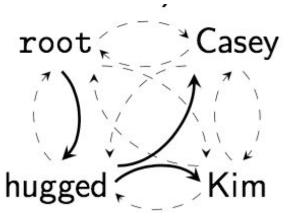
Neural Graph-based Dependency Parsing

Analyze **grammatical structure** of a sentence; relations between heads & dependants

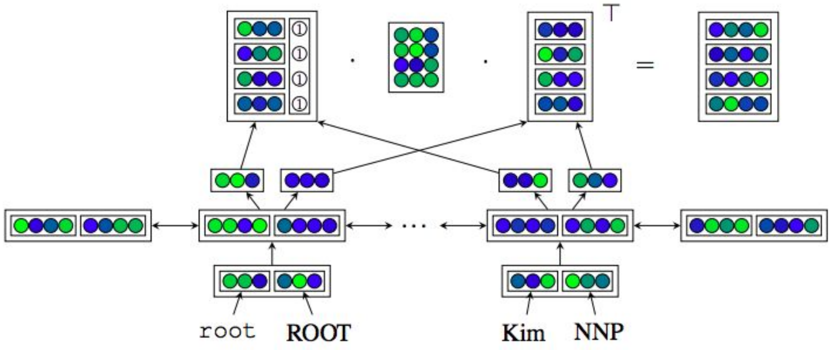
Edges go from **heads** to **dependants:**



Graph-based approach: calculate a weight for *each edge*, construct a maximum spanning tree



We will compute the weights using a **Neural Net**



In this project you will:

- Run an off-the-shelf parser (baseline)
- Build a state-of-the-art parser in PyTorch
  Ref. https://arxiv.org/abs/1611.01734
- Learn & implement 2 ways to get an MST

Further reading

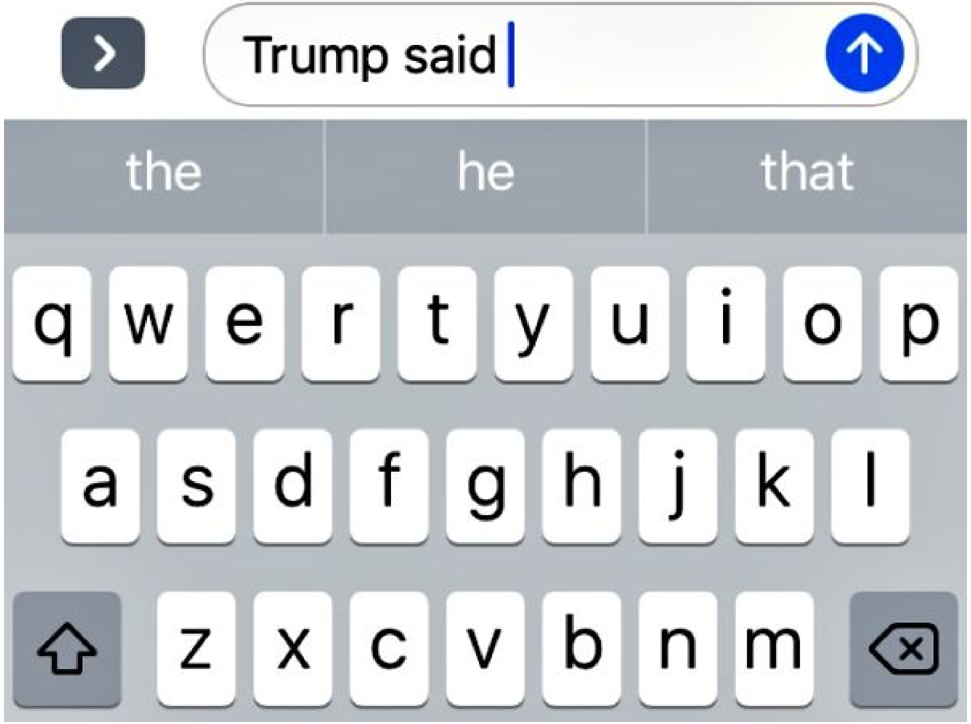
1. J&M 3rd edition Ch. 14 Dependency parsing https://web.stanford.edu/~jurafsky/slp3/14.pdf

2. Kiperwasser & Goldberg (2016)
   Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations
   https://aclweb.org/anthology/Q16-1023
   Esp. section 5 (Graph based parsing)

3. Dozat & Manning (2017)
   Deep Biaffine Attention for Neural Dependency Parsing
   https://arxiv.org/abs/1611.01734

You will need to read (1) to understand (2) to understand (3).

Neural Language Modeling



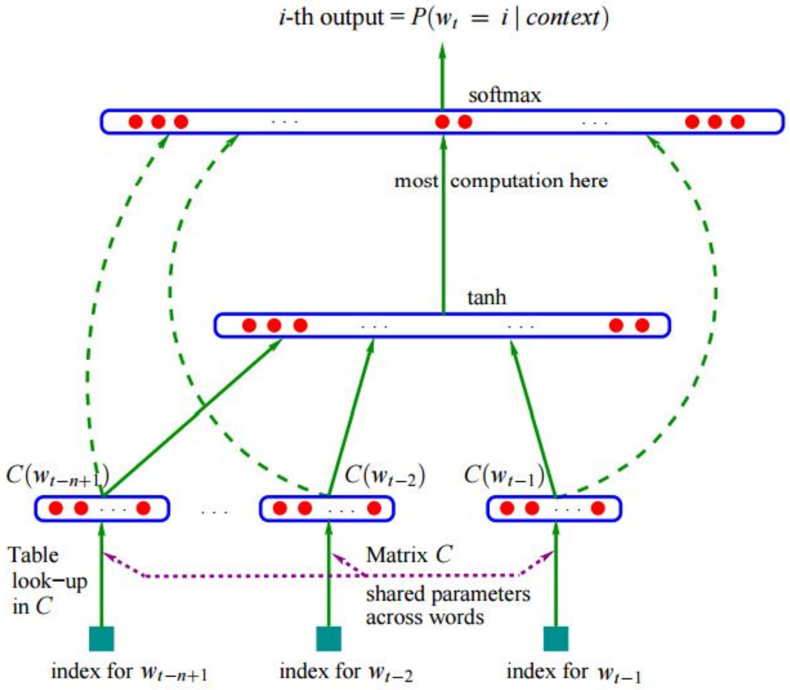
Neural Language Modeling

A Language Model gives the **probability** of the **next word** given a **history** of previous words

It can assign a probability to a **sentence**

In this project you will:

- Run an off-the-shelf LM as a baseline
- Develop your own Neural Language models in PyTorch
    - Feed-forward
    - Simple RNN
    - RAN https://arxiv.org/abs/1705.07393
    - LSTM
- Analyze your models, e.g. what word is in focus when generating the next word?



Further reading

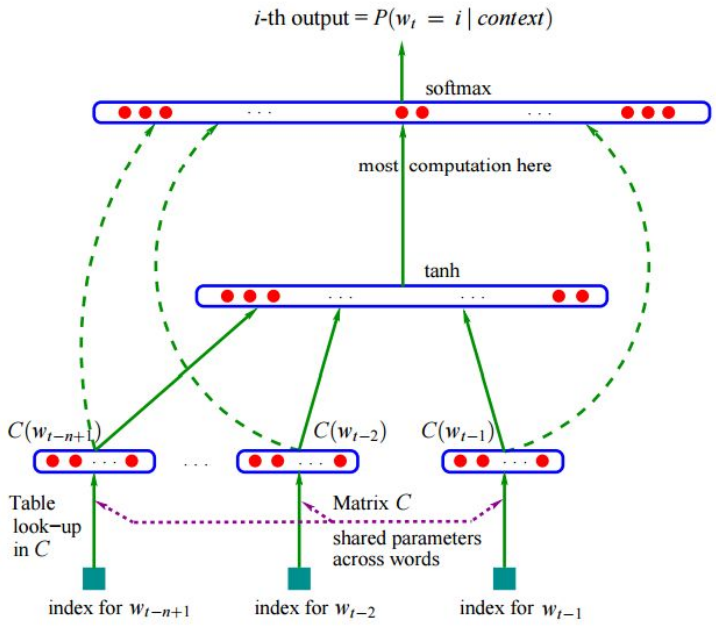
1. Bengio NNLM http://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf
   This is a feed-forward network that predicts the next word given a **fixed** amount of previous words.

2. Simple RNN LM
   http://www.fit.vutbr.cz/research/groups/speech/publi/2010/mikolov_interspeech2010_IS100722.pdf
   A recurrent NN that (in principle) models unlimited history.

3. Recurrent Additive Networks https://arxiv.org/abs/1705.07393
   RANs are a simpler version of LSTMs. This makes them not only faster to train, it also allows us to inspect which previous word was most influential in predicting the current word. Using RANs can be cool for analysis, and you can try to tweak them to get even better performance.

4. LSTM / GRUs
   You can also train an LSTM (or GRU) language model. See (3) for details. Note that, of all networks, training this will take the longest, especially on CPU.

Finding Grammar in Neural Language Models

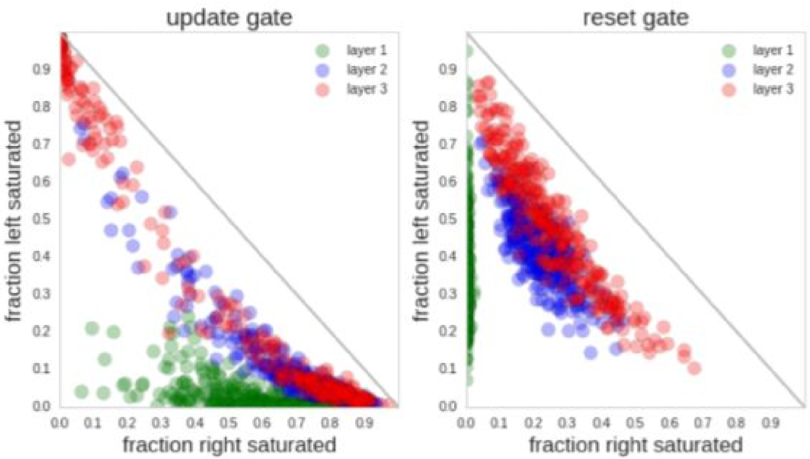


Goals

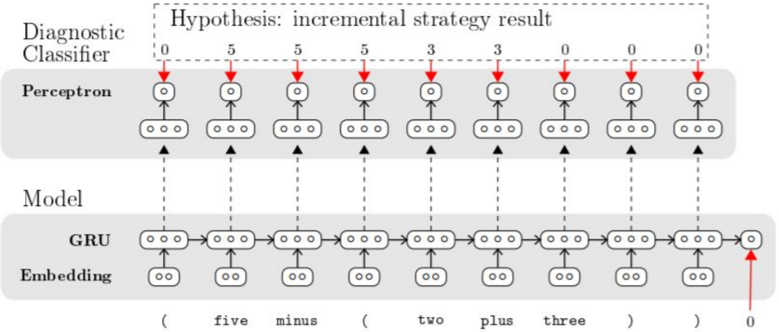
- Getting acquainted with recurrent neural network models
- Gaining a better understanding of what neural language models encode
- Testing their syntactic awareness
- Learning how to inspect and visualise the internal dynamics of recurrent networks
- Proposing methods to make them more syntactically aware

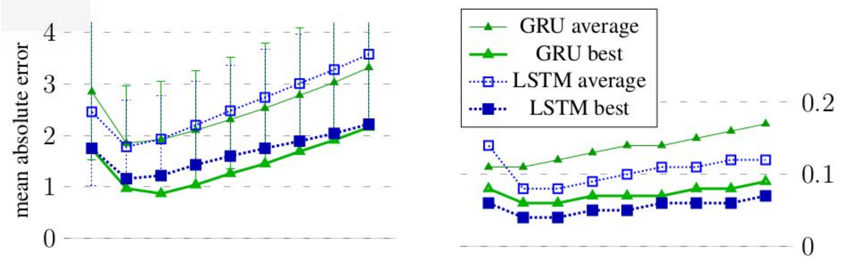
Methods

Visualising gate saturation values



Diagnosing symbolic strategies





(a) Intermediate error

(b) `minus1depth_count`

Further Reading

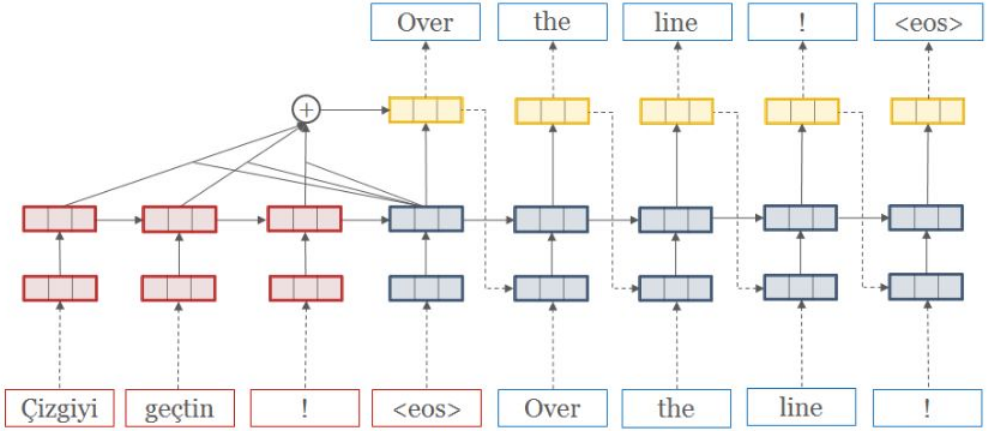
Assessing the Ability of LSTMs to learn Syntax-Sensitive Dependencies:
https://arxiv.org/pdf/1611.01368.pdf

Visualisation and diagnostic classifiers reveal how recurrent and recursive neural networks process hierarchical structure
http://dieuwkehupkes.nl/research/JAIR.pdf

Exploring the Syntactic Abiities of RNNs with Multi-task Learning
http://tallinzen.net/media/papers/enguehard_goldberg_linzen_2017_conll.pdf

Visualizing and Understanding Recurrent Networks
https://arxiv.org/pdf/1506.02078.pdf

What do Neural Machine Translation Models do?

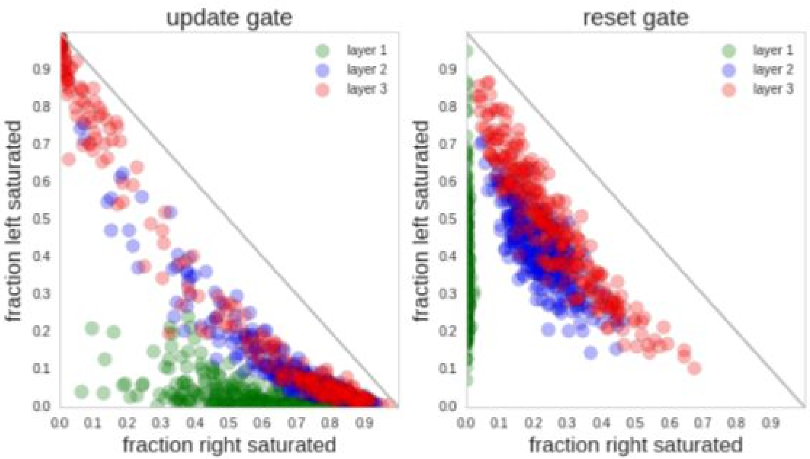


Goals

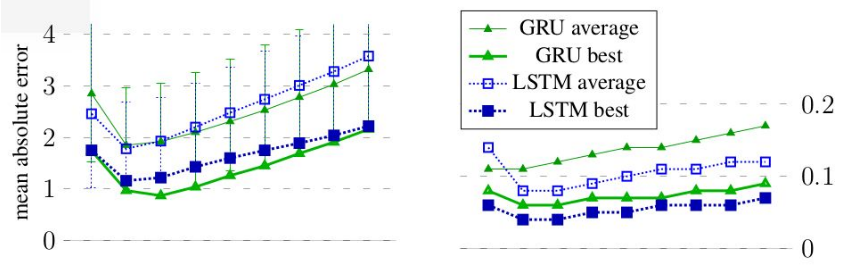
- Learning to use pretrained MT models
- Developing hypotheses about what MT models encode
- Getting an insight in what is important for neural MT models
- Learning how to inspect the dynamics of recurrent neural models

Using visualisation and Diagnostic Classifiers

Visualising gate saturation values

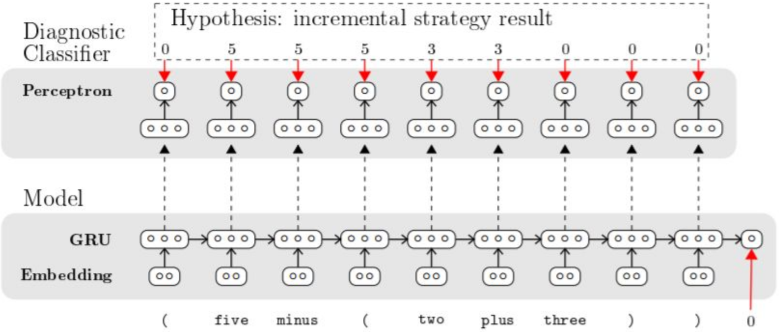


Diagnosing symbolic strategies



(a) Intermediate error

(b) `minus1depth_count`



Further Reading

Assessing the Ability of LSTMs to learn Syntax-Sensitive Dependencies:
https://arxiv.org/pdf/1611.01368.pdf

Visualisation and diagnostic classifiers reveal how recurrent and recursive neural networks process hierarchical structure
http://dieuwkehupkes.nl/research/JAIR.pdf

Visualizing and Understanding Recurrent Networks
https://arxiv.org/pdf/1506.02078.pdf