

# Project types

1. Apply existing neural network model to a new task
2. Implement a complex neural architecture
3. Come up with a new neural network model
4. Theory of deep learning, e.g. optimization

# Class Project: Apply Existing NNets to Tasks

## 1. Define Task:

- Example: **Summarization**

## 2. Define Dataset

### 1. Search for academic datasets

- They already have baselines
- E.g.: Document Understanding Conference (DUC)

### 2. Define your own (harder, need more new baselines)

- If you're a graduate student: connect to your research
- Summarization, Wikipedia: Intro paragraph and rest of large article
- Be creative: Twitter, Blogs, News

# Class Project: Apply Existing NNets to Tasks

## 3. Define your metric

- Search online for well established metrics on this task
- Summarization: Rouge (Recall-Oriented Understudy for Gisting Evaluation) which defines n-gram overlap to human summaries

## 4. Split your dataset!

- Train/Dev/Test
- Academic dataset often come pre-split
- Don't look at the test split until ~1 week before deadline!  
(or at most once a week)

# Class Project: Apply Existing NNets to Tasks

## 5. Establish a baseline

- Implement the simplest model (often logistic regression on unigrams and bigrams) first
- Compute metrics on train AND dev
- Analyze errors
- If metrics are amazing and no errors: done, problem was too easy, restart :)

## 6. Implement existing neural net model

- Compute metric on train and dev
- Analyze output and errors
- Minimum bar for this class

# Class Project: Apply Existing NNets to Tasks

## 7. Always be close to your data!

- Visualize the dataset
- Collect summary statistics
- Look at errors
- Analyze how different hyperparameters affect performance

## 8. Try out different model variants

- Soon you will have more options
  - Word vector averaging model (neural bag of words)
  - Fixed window neural model
  - Recurrent neural network
  - Recursive neural network
  - Convolutional neural network

# Class Project: A New Model -- Advanced Option

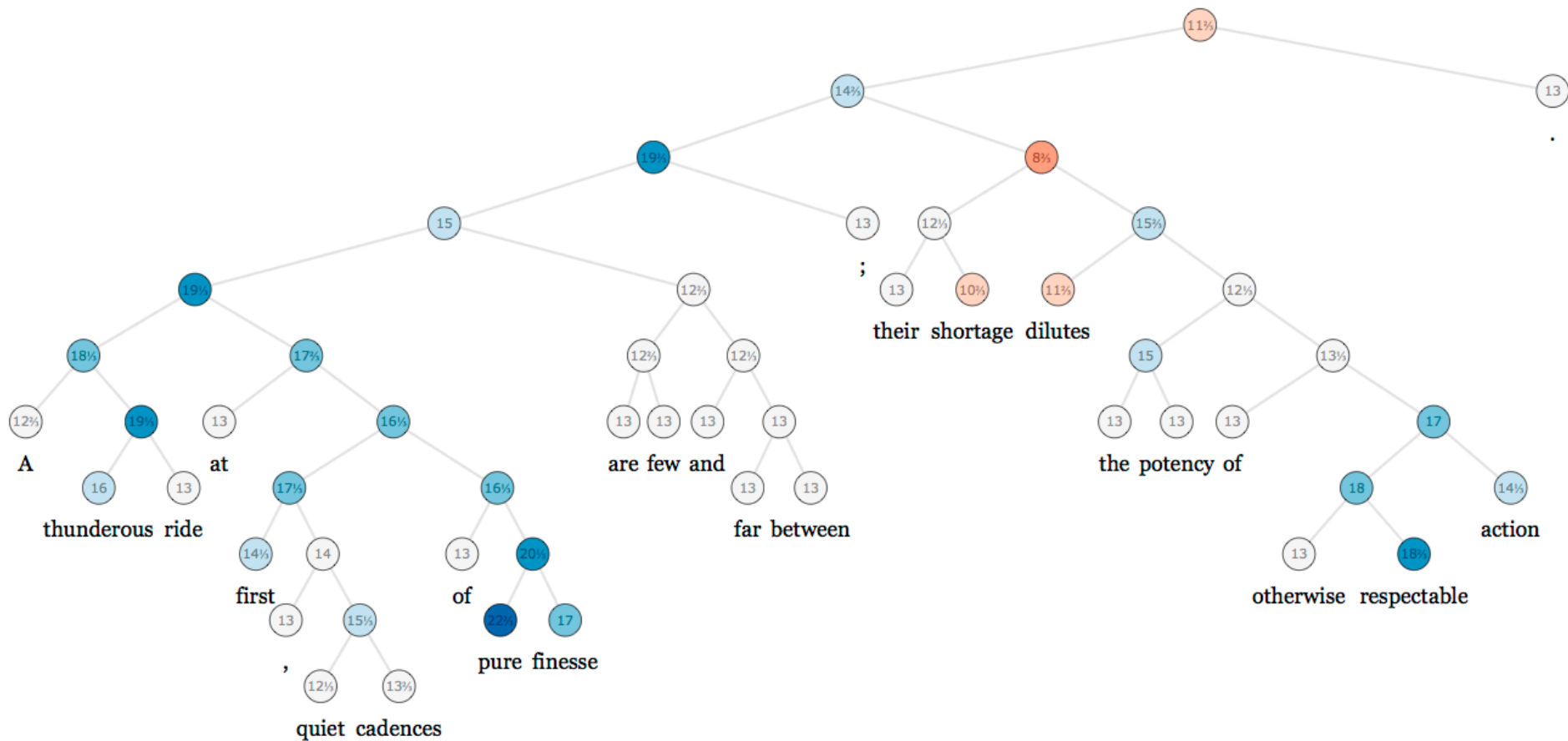
- Do all other steps first (Start early!)
- Gain intuition of why existing models are flawed
- Talk to researcher/mentor, come to project office hours a lot
- Implement new models and iterate quickly over ideas
- Set up efficient experimental framework
- Build simpler new models first
- Example Summarization:
  - Average word vectors per paragraph, then greedy search
  - Implement language model (introduced later)
  - Stretch goal: Generate summary with seq2seq!

# Project Ideas

- Summarization
- NER, like PSet 2 but with larger data  
Natural Language Processing (almost) from Scratch, Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, Pavel Kuksa, <http://arxiv.org/abs/1103.0398>
- Simple question answering, [A Neural Network for Factoid Question Answering over Paragraphs](#), Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher and Hal Daumé III (**EMNLP 2014**)
- Image to text mapping or generation,  
[Grounded Compositional Semantics for Finding and Describing Images with Sentences](#), Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, Andrew Y. Ng. (**TACL 2014**)  
or  
Deep Visual-Semantic Alignments for Generating Image Descriptions, Andrej Karpathy, Li Fei-Fei
- Entity level sentiment
- Use DL to solve an NLP challenge on kaggle,  
Develop a scoring algorithm for student-written short-answer responses, <https://www.kaggle.com/c/asap-sas>

## Another example project: Sentiment

- Sentiment on movie reviews: <http://nlp.stanford.edu/sentiment/>
- Lots of deep learning baselines and methods have been tried





## Next up

- Some fun and fundamental linguistics with syntactic parsing
- TensorFlow lecture (for Ass.2 ) – also useful for projects and life : )