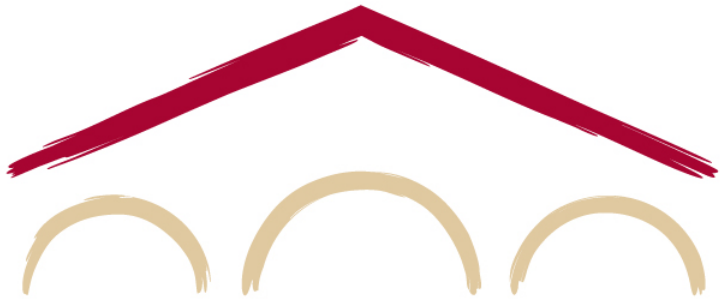
Natural Language Processing with Deep Learning
CS224N/Ling284



Christopher Manning

Lecture 8: Final Projects; Practical Tips

3. Finding Research Topics

Two basic starting points, for all of science:

- [Nails] Start with a (domain) problem of interest and try to find good/better ways to address it than are currently known/used
- [Hammers] Start with a technical method/approach of interest, and work out good ways to extend or improve it or new ways to apply it

Project types

This is not an exhaustive list, but most projects are one of

1. Find an application/task of interest and explore how to approach/solve it effectively, often with an existing model
   - Could be task in the wild or some existing Kaggle/bake-off/shared task
2. Implement a complex neural architecture and demonstrate its performance on some data
3. Come up with a new or variant neural network model and explore its empirical success
4. Analysis project. Analyze the behavior of a model: how it represents linguistic knowledge or what kinds of phenomena it can handle or errors that it makes
5. Rare theoretical project: Show some interesting, non-trivial properties of a model type, data, or a data representation

Deep Poetry: Word-Level and Character-Level Language Models for Shakespearean Sonnet Generation

Stanley Xie, Ruchir Rastogi and Max Chang

Gated LSTM

Thy youth 's time and face his form shall cover?
Now all fresh beauty, my love there
Will ever Time to greet, forget each, like ever decease,
But in a best at worship his glory die.

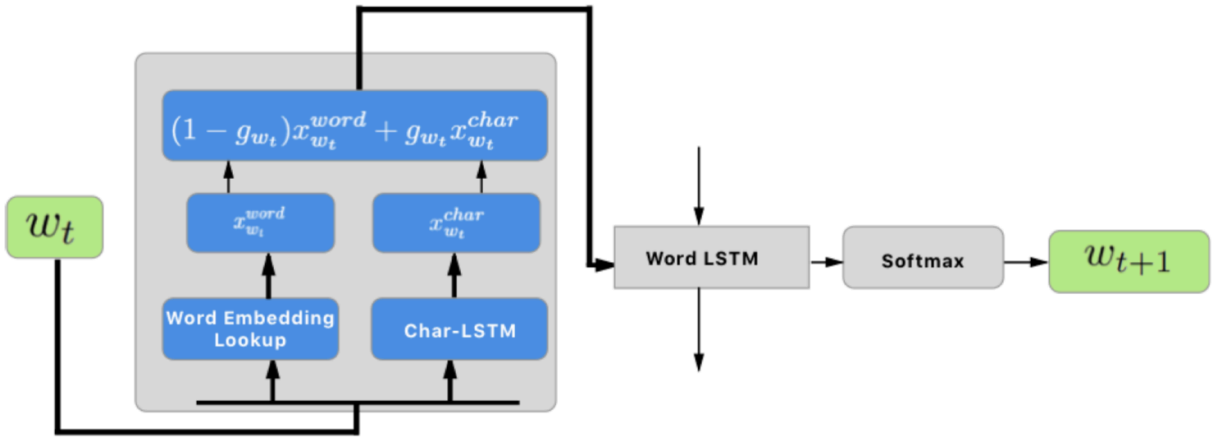
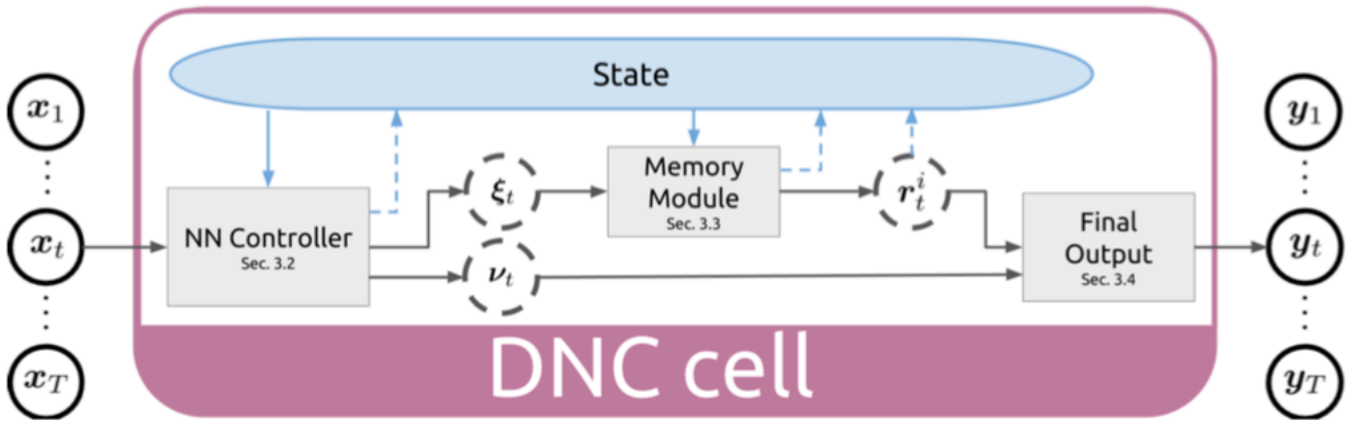


Figure 1: Architecture of the Gated LSTM

Implementation and Optimization of Differentiable Neural Computers

Carol Hsin
Graduate Student in Computational & Mathematical Engineering

We implemented and optimized Differentiable Neural Computers (DNCs) as described in the Oct. 2016 DNC paper [1] on the bAbI dataset [25] and on copy tasks that were described in the Neural Turning Machine paper [12]. This paper will give the reader a better understanding of this new and promising architecture through the documentation of the approach in our DNC implementation and our experience of the challenges of optimizing DNCs.



Improved Learning through Augmenting the Loss

Hakan Inan
inanh@stanford.edu

Khashayar Khosravi
khosravi@stanford.edu

We present two improvements to the well-known Recurrent Neural Network Language Models(RNNLM). First, we use the word embedding matrix to project the RNN output onto the output space and already achieve a large reduction in the number of free parameters while still improving performance. Second, instead of merely minimizing the standard cross entropy loss between the prediction distribution and the "one-hot" target distribution, we minimize an additional loss term which takes into account the inherent metric similarity between the target word and other words. We show with experiments on the Penn Treebank Dataset that our proposed model (1) achieves significantly lower average word perplexity than previous models with the same network size and (2) achieves the new state of the art by using much fewer parameters than used in the previous best work.

Published as a conference paper at ICLR 2017

Word2Bits - Quantized Word Vectors

Maximilian Lam
maxlam@stanford.edu

Abstract

Word vectors require significant amounts of memory and storage, posing issues to resource limited devices like mobile phones and GPUs. We show that high quality quantized word vectors using 1-2 bits per parameter can be learned by introducing a quantization function into Word2Vec. We furthermore show that training with the quantization function acts as a regularizer. We train word vectors on English Wikipedia (2017) and evaluate them on standard word similarity and analogy tasks and on question answering (SQuAD). Our quantized word vectors not only take 8-16x less space than full precision (32 bit) word vectors but also outperform them on word similarity tasks and question answering.

How to find an interesting place to start?

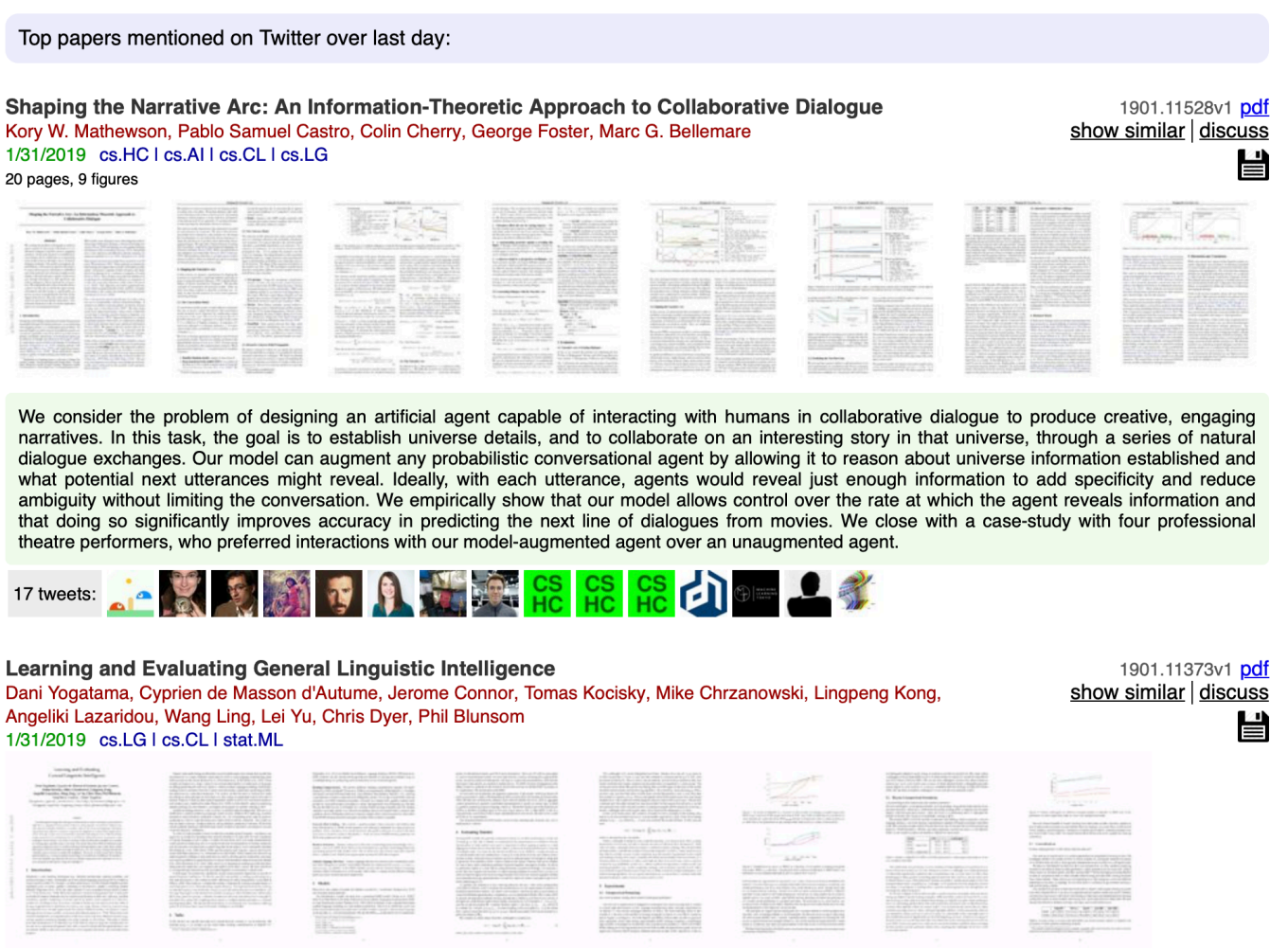
- Look at ACL anthology for NLP papers:
  - https://www.aclweb.org/anthology/
- Also look at the online proceedings of major ML conferences:
  - NeurIPS https://papers.nips.cc, ICML, ICLR
- Look at past cs224n projects
  - See the class website
- Look at online preprint servers, especially:
  - https://arxiv.org

- Even better: look for an interesting problem in the world!
  - Hal Varian: How to Build an Economic Model in Your Spare Time https://people.ischool.berkeley.edu/~hal/Papers/how.pdf

How to find an interesting place to start?

Arxiv Sanity Preserver by Stanford grad Andrej Karpathy of cs231n fame

http://www.arxiv-sanity.com



Want to beat the state of the art on something?

Great new sites that try to collate info on the state of the art

- Not always correct, though

https://paperswithcode.com/sota
https://nlpprogress.com/

Specific tasks/topics. Many, e.g.:
https://gluebenchmark.com/leaderboard/
https://www.conll.org/previous-tasks/

wse > Natural Language Processing > Machine Translation

Machine Translation

223 papers with code · Natural Language Processing

chine translation is the task of translating a sentence in a source language to a differe guage.

ate-of-the-art leaderboards

rend	Dataset	Best Method	Paper title	Paper	Code
	WMT2014 English-French	Transformer Big + BT	Understanding Back-Translation at Scale		
	WMT2014 English-German	Transformer Big + BT	Understanding Back-Translation at Scale		
	IWSLT2015 German-English	Transformer	Attention Is All You Need		
	WMT2016 English-Romanian	ConvS2S BPE40k	Convolutional Sequence to Sequence Learning		

Finding a topic

- Turing award winner and Stanford CS emeritus professor Ed Feigenbaum says to follow the advice of his advisor, AI pioneer, and Turing and Nobel prize winner Herb Simon:
  - “If you see a research area where many people are working, go somewhere else.”
- But where to go? Wayne Gretzky:
  - “I skate to where the puck is going, not where it has been.”

Old Deep Learning (NLP), new Deep Learning NLP

- In the early days of the Deep Learning revival (2010-2018), most of the work was in defining and exploring better deep learning architectures
- Typical paper:
  - I can improve a summarization system by not only using attention standardly, but allowing copying attention – where you use additional attention calculations and an additional probabilistic gate to simply copy a word from the input to the output
- That's what a lot of good CS 224N projects did too
- In 2019–2021, that approach is dead
  - Well, that's too strong, but it's difficult and much rarer
- By and large, most work downloads a big pre-trained model and works from there
  - Action is in fine-tuning, or domain adaptation followed by fine-tuning, etc., etc.

2021 NLP … recommended for all your practical projects ☺

```
pip install transformers  # By Huggingface 🤗

# not quite runnable code but gives the general idea….

from transformers import BertForSequenceClassification, AutoTokenizer

model = BertForSequenceClassification.from_pretrained('bert-base-uncased’)

model.train()

tokenizer = AutoTokenizer.from_pretrained('bert-base-uncased’)

fine_tuner = Trainer( model=model,  args=training_args, train_dataset=train_dataset,
    eval_dataset=test_dataset )

fine_tuner.train()

eval_dataset = load_and_cache_examples(args, eval_task, tokenizer, evaluate=True)

results = evaluate(model, tokenizer, eval_dataset, args)
```

Exciting areas 2021

A lot of what is exciting now is problems that work within or around this world

- Evaluating and improving models for something other than accuracy
  - Robustness to domain shift
  - Evaluating the robustness of models in general (someone could hack on this new project as their final project!): https://robustnessgym.com
- Doing empirical work looking at what large pre-trained models have learned
- Working out how to get knowledge and good task performance from large models for particular tasks without much data (transfer learning, etc.)
- Looking at the bias, trustworthiness, and explainability of large models
- Working on how to augment the data for models to improve performance
- Low resource languages
- Improving performance on the tail of rare stuff

Exciting areas 2021

- Scaling models up and down
  - Building big models is BIG: GPT-2 and GPT-3 … but just not possible for a cs224n project
  - Building small, performant models is also BIG. This could be a great project
    - Model pruning, e.g.: https://papers.nips.cc/paper/2020/file/eae15aabaa768ae4a5993a8a4f4fa6e4-Paper.pdf
    - Model quantization, e.g.: https://arxiv.org/pdf/2004.07320.pdf
    - How well can you do QA in 6GB or 500MB? https://efficientqa.github.io
- Looking to achieve more advanced functionalities
  - E.g., compositionality, systematic generalization, fast learning (e.g., meta-learning) on smaller problems and amounts of data, and more quickly
    - BabyAI: https://arxiv.org/abs/2007.12770
    - gSCAN: https://arxiv.org/abs/2003.05161

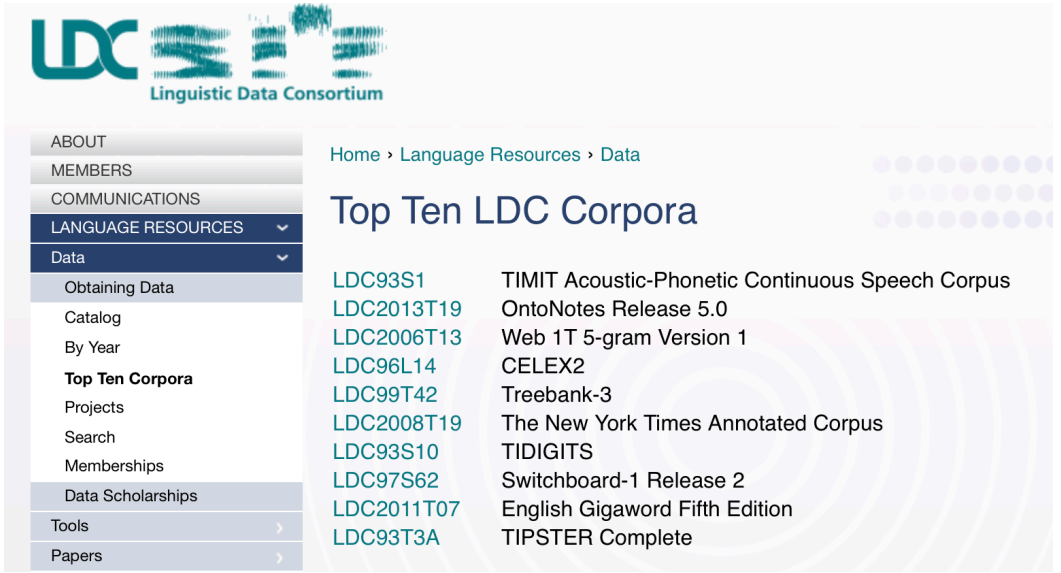
4. Finding data

- Some people collect their own data for a project – **we like that!**
  - You may have a project that uses “unsupervised” data
  - You can annotate a small amount of data
  - You can find a website that effectively provides annotations, such as likes, stars, ratings, responses, etc.
    - Let’s you learn about real word challenges of applying ML/NLP!
- Some people have existing data from a research project or company
  - Fine to use providing you can provide data samples for submission, report, etc.

- **Most people make use of an existing, curated dataset built by previous researchers**
  - You get a fast start and there is obvious prior work and baselines

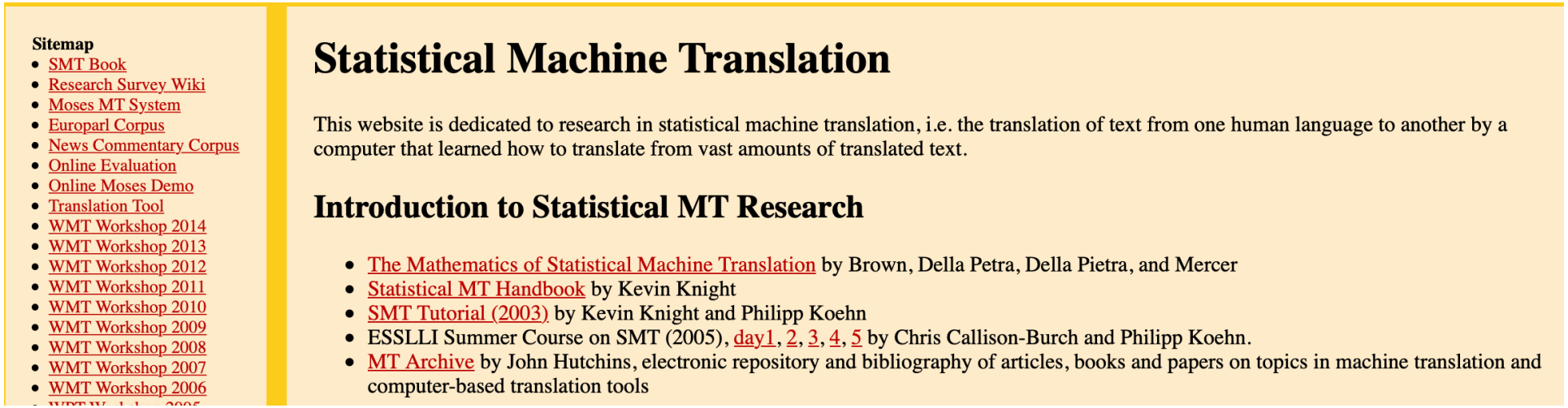
Linguistic Data Consortium

- https://catalog.ldc.upenn.edu/
- Stanford licenses data; you can get access by signing up at: https://linguistics.stanford.edu/resources/resources-corpora
- Treebanks, named entities, coreference data, lots of clean newswire text, lots of speech with transcription, parallel MT data, etc.
  - Look at their catalog
  - Don't use for non-Stanford purposes!



Machine translation

- http://statmt.org
- Look in particular at the various WMT shared tasks



Sitemap

- SMT Book
- Research Survey Wiki
- Moses MT System
- Europarl Corpus
- News Commentary Corpus
- Online Evaluation
- Online Moses Demo
- Translation Tool
- WMT Workshop 2014
- WMT Workshop 2013
- WMT Workshop 2012
- WMT Workshop 2011
- WMT Workshop 2010
- WMT Workshop 2009
- WMT Workshop 2008
- WMT Workshop 2007
- WMT Workshop 2006

Statistical Machine Translation

This website is dedicated to research in statistical machine translation, i.e. the translation of text from one human language to another by a computer that learned how to translate from vast amounts of translated text.

Introduction to Statistical MT Research

- The Mathematics of Statistical Machine Translation by Brown, Della Petra, Della Pietra, and Mercer
- Statistical MT Handbook by Kevin Knight
- SMT Tutorial (2003) by Kevin Knight and Philipp Koehn
- ESSLLI Summer Course on SMT (2005), day1, 2, 3, 4, 5 by Chris Callison-Burch and Philipp Koehn.
- MT Archive by John Hutchins, electronic repository and bibliography of articles, books and papers on topics in machine translation and computer-based translation tools

Dependency parsing: Universal Dependencies

- https://universaldependencies.org

Universal Dependencies

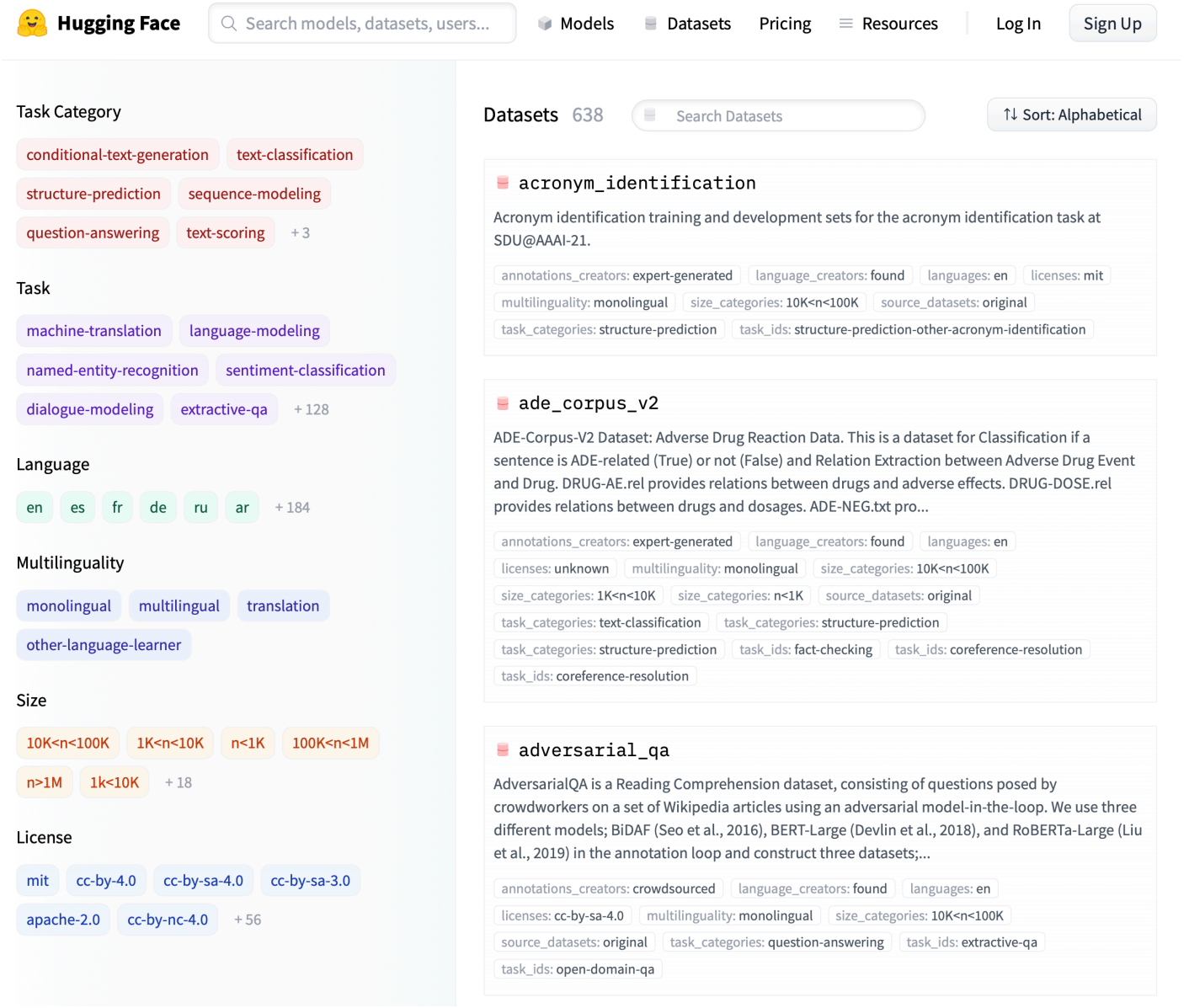
Universal Dependencies (UD) is a framework for cross-linguistically consistent grammatical annotation and an open community effort with over 200 contributors producing more than 100 treebanks in over 70 languages.

- Short introduction to UD
- UD annotation guidelines
- More information on UD:
  - How to contribute to UD
  - Tools for working with UD
  - Discussion on UD
  - UD-related events
- Query UD treebanks online:
  - SETS treebank search maintained by the University of Turku
  - PML Tree Query maintained by the Charles University in Prague
  - Kontext maintained by the Charles University in Prague
  - Grew-match maintained by Inria in Nancy
- Download UD treebanks

If you want to receive news about Universal Dependencies, you can subscribe to the UD mailing list. If you want to discuss individual annotation questions, use the Github issue tracker.

Huggingface Datasets

- https://huggingface.co/datasets



Paperswithcode Datasets

- https://www.paperswithcode.com/datasets?mod=texts&page=1

835 dataset results for Texts ×

Penn Treebank

The English Penn Treebank corpus, and in particular the section of the corpus corresponding to the articles of Wall Street Journal (WSJ), is one of the most known and used corpus for t...

1,545 PAPERS • 10 BENCHMARKS

SQuAD (Stanford Question Answering Dataset)

The Stanford Question Answering Dataset (SQuAD) is a collection of question-answer pairs derived from Wikipedia articles. In SQuAD, the correct answers of questions can be any se-...

1,254 PAPERS • 7 BENCHMARKS

Visual Genome

Visual Genome contains Visual Question Answering data in a multi-choice setting. It consists of 101,174 images from MSCOCO with 1.7 million QA pairs, 17 questions per image on aver-...

903 PAPERS • 11 BENCHMARKS

GLUE (General Language Understanding Evaluation benchmark)

General Language Understanding Evaluation (GLUE) benchmark is a collection of nine natural language understanding tasks, including single-sentence tasks CoLA and SST-2, similarity...

847 PAPERS • 14 BENCHMARKS

SNLI (Stanford Natural Language Inference)

The SNLI dataset (Stanford Natural Language Inference) consists of 570k sentence-pairs manually labeled as entailment, contradiction, and neutral. Premises are image captions fro...

743 PAPERS • 1 BENCHMARK

CLEVR (Compositional Language and Elementary Visual Reasoning)

CLEVR (Compositional Language and Elementary Visual Reasoning) is a synthetic Visual Question Answering dataset. It contains images of 3D-rendered objects; each image comes...

528 PAPERS • 1 BENCHMARK

Visual Question Answering (VQA)

Visual Question Answering (VQA) is a dataset containing open-ended questions about images. These questions require an understanding of vision, language and commonsense...

435 PAPERS • 2 BENCHMARKS

Billion Word Benchmark

The One Billion Word dataset is a dataset for language modeling. The training/held-out data was produced from the WMT 2011 News Crawl data using a combination of Bash shell and...

417 PAPERS • 1 BENCHMARK

Many, many more

- There are now many other datasets available online for all sorts of purposes
  - Look at Kaggle
  - Look at research papers to see what data they use
  - Look at lists of datasets
    - https://machinelearningmastery.com/datasets-natural-language-processing/
    - https://github.com/niderhoff/nlp-datasets
  - Lots of particular things:
    - https://gluebenchmark.com/tasks
    - https://nlp.stanford.edu/sentiment/
    - https://research.fb.com/downloads/babi/ (Facebook bAbI-related)
  - Ask on Ed or talk to course staff

5. Doing your research example: Straightforward Class Project: Apply NNets to Task

1. Define Task:
   - Example: **Summarization**

2. Define Dataset
   1. Search for academic datasets
      - They already have baselines
      - E.g.: Newsroom Summarization Dataset: http://lil.nlp.cornell.edu/newsroom/
   2. Define your own data (harder, need new baselines)
      - Allows connection to your research
      - A fresh problem provides fresh opportunities!
      - Be creative: E.g., can you generate advertising tweet from a news story?
      - There are lots of neat websites which provide creative opportunities for new tasks

Straightforward Class Project: Apply NNets to Task

3. Dataset hygiene
   - Right at the beginning, separate off devtest and test data splits
     - Discussed more next

4. Define your metric(s)
   - 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
   - Human evaluation is still much better for summarization
     - You may be able to do at least a very small scale human eval – ask some friends

Straightforward Class Project: Apply NNets to Task

5. Establish a baseline
   - Implement the simplest model first (e.g., logistic regression on unigrams and bigrams or averaging word vectors)
     - For summarization: See LEAD-3 baseline
   - Compute metrics on train AND dev NOT test
   - Analyze errors
   - If metrics are amazing and no errors:
     - Done! Problem was too easy. Need to restart. ☺/☹

6. Implement existing neural net model
   - Compute metric on train and dev
   - Analyze output and errors
   - Minimum bar for this class

Straightforward Class Project: Apply NNets to Task

7. Always be close to your data! (Except for the final test set!)
   - Visualize the dataset
   - Collect summary statistics
   - Look at errors
   - Analyze how different hyperparameters affect performance

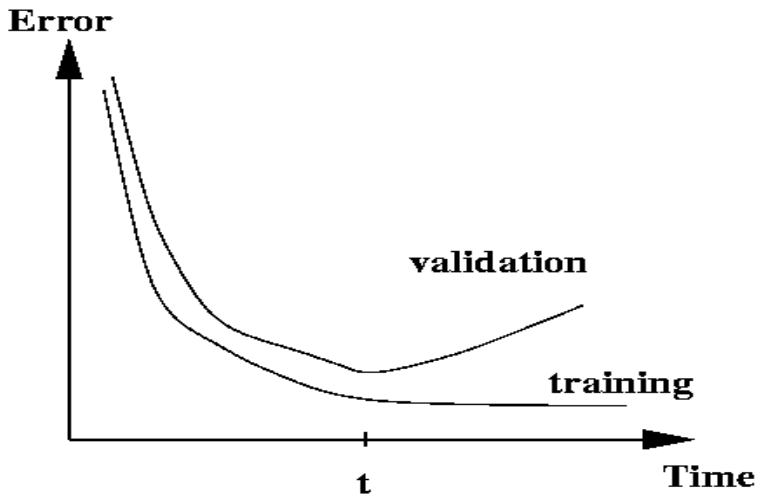
8. Try out different models and model variants
   Aim to iterate quickly via having a good experimental setup
   - Fixed window neural model
   - Recurrent neural network
   - Recursive neural network
   - Convolutional neural network
   - Attention-based model/transformer
   - …

Pots of data

- Many publicly available datasets are released with a **train/dev/test** structure.
- **We're all on the honor system to do test-set runs only when development is complete.**
- Splits like this presuppose a fairly large dataset.
- If there is no dev set or you want a separate tune set, then you create one by splitting the training data
  - We have to weigh the usefulness of it being a certain size against the reduction in train-set size.
  - Cross-validation (q.v.) is a technique for maximizing data when you don't have much
- Having a fixed test set ensures that all systems are assessed against the same gold data. This is generally good, but it is problematic where the test set turns out to have unusual properties that distort progress on the task.

Training models and pots of data

- When training, models **overfit** to what you are training on
  - The model correctly describes what happened to occur in particular data you trained on, but the patterns are not general enough patterns to be likely to apply to new data
- The way to monitor and avoid problematic overfitting is using **independent** validation and test sets …



Training models and pots of data

- You build (estimate/train) a model on a **training set**.
- Often, you then set further hyperparameters on another, independent set of data, the **tuning set**
  - The tuning set is the training set for the hyperparameters!
- You measure progress as you go on a **dev set** (development test set or validation set)
  - If you do that a lot you overfit to the dev set so it can be good to have a second dev set, the **dev2** set
- **Only at the end**, you evaluate and present final numbers on a **test set**
  - Use the final test set **extremely** few times … ideally only once

Training models and pots of data

- The **train**, **tune**, **dev**, and **test** sets need to be completely distinct
- It is invalid to test on material you have trained on
  - You will get a falsely good performance.
  - We almost always overfit on train
- You need an independent tuning set
  - The hyperparameters won't be set right if tune is same as train
- If you keep running on the same evaluation set, you begin to overfit to that evaluation set
  - Effectively you are "training" on the evaluation set … you are learning things that do and don't work on that particular eval set and using the info
- To get a valid measure of system performance you need another untrained on, **independent** test set … hence dev2 and final test

Getting your neural network to train

- Start with a positive attitude!
  - **Neural networks want to learn!**
    - If the network isn't learning, you're doing something to prevent it from learning successfully
- Realize the grim reality:
  - **There are lots of things that can cause neural nets to not learn at all or to not learn very well**
    - Finding and fixing them ("debugging and tuning") can often take more time than implementing your model
- It's hard to work out what these things are
  - But experience, experimental care, and rules of thumb help!

Details matter!

- Look at your data, collect summary statistics
- Look at your model's outputs, do error analysis
- Tuning hyperparameters is **often** important to the successes of NNets

6. The Default Final Project

Reading Comprehension

a.k.a. Question Answering

over documents

Google

who won the 2020 iowa caucus?

Top stories

Live Iowa Caucus Results 2020

LIVE The New York Times · 54 mins ago

'It's a total meltdown': Confusion grips Iowa with no official results in sight

Politico · 1 hour ago

www.politico.com › 2020/02/03 › iowa-caucus-2020-election-110600

'It's a total meltdown': Confusion grips Iowa with no official ...

2 hours ago - The **Iowa caucus** results appear to be indefinitely delayed, as the state party blames ... The biggest "**winner**" might have been Joe Biden.

Technical note: This is a “featured snippet” answer extracted from a web page, not a question answered using the (structured) Google Knowledge Graph (formerly known as Freebase).

Motivation: Question answering

- With massive collections of full-text documents, i.e., the web ☺, simply returning relevant documents is of limited use
- Rather, we often want **answers** to our **questions**
- Especially on mobile
- Or using a digital assistant device, like Alexa, Google Assistant, …

- We can factor this into two parts:
  1. Finding documents that (might) contain an answer
     - Which can be handled by traditional information retrieval/web search
  2. Finding an answer in a paragraph or a document
     - This problem is often termed **Reading Comprehension**

Stanford Question Answering Dataset (SQuAD)

(Rajpurkar et al., 2016)

Question: Which team won Super Bowl 50?

Passage

Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California.

100k examples

Answer must be a span in the passage

A.k.a. extractive question answering

Stanford Question Answering Dataset (SQuAD)

Private schools, also known as independent schools, non-governmental, or nonstate schools, are not administered by local, state or national governments; thus, they retain the right to select their students and are funded in whole or in part by charging their students tuition, rather than relying on mandatory taxation through public (government) funding; at some private schools students may be able to get a scholarship, which makes the cost cheaper, depending on a talent the student may have (e.g. sport scholarship, art scholarship, academic scholarship), financial need, or tax credit scholarships that might be available.

Along with non-governmental and nonstate schools, what is another name for private schools?

Gold answers: ① independent ② independent schools ③ independent schools

Along with sport and art, what is a type of talent scholarship?

Gold answers: ① academic ② academic ③ academic

Rather than taxation, what are private schools largely funded by?

Gold answers: ① tuition ② charging their students tuition ③ tuition

SQuAD evaluation, v1.1

- Authors collected 3 gold answers
- Systems are scored on two metrics:
  - Exact match: 1/0 accuracy on whether you match one of the 3 answers
  - F1: Take system and each gold answer as bag of words, evaluate
    Precision = $\frac{TP}{TP+FP}$, Recall = $\frac{TP}{TP+FN}$, harmonic mean F1 = $\frac{2PR}{P+R}$
    Score is (macro-)average of per-question F1 scores
- F1 measure is seen as more reliable and taken as primary
  - It's less based on choosing exactly the same span that humans chose, which is susceptible to various effects, including line breaks
- Both metrics ignore punctuation and articles (**a**, **an**, **the** only)

SQuAD 2.0 Example: Adds unanswerable questions

Genghis Khan united the Mongol and Turkic tribes of the steppes and became Great Khan in 1206. He and his successors expanded the Mongol empire across Asia. Under the reign of Genghis' third son, Ögedei Khan, the Mongols destroyed the weakened Jin dynasty in 1234, conquering most of northern China. Ögedei offered his nephew Kublai a position in Xingzhou, Hebei. Kublai was unable to read Chinese but had several Han Chinese teachers attached to him since his early years by his mother Sorghaghtani. He sought the counsel of Chinese Buddhist and Confucian advisers. Möngke Khan succeeded Ögedei's son, Güyük, as Great Khan in 1251. He

When did Genghis Khan kill Great Khan?

Gold Answers: <No Answer>

Prediction: 1234 [from Microsoft nlnet]