Natural language Processing

Project Guidelines

Mahindra™ University
Global Thinkers. Engaged Leaders.

The Final Project

- **You can work in teams of 1–5. Being in a team is encouraged.**
  - **A larger team project or a project should be larger and often involves exploring more models or tasks**

- **You can use any language/framework for your project**
  - Though we expect nearly all of you to keep using **Python/PyTorch**
- I am very happy to talk to people about all final projects (any time beside office hours too), but
  - It is recommended to schedule a slot for discussion
  - the slight problem is that there is only one of me, so Look/use TA NLP expertise for custom final projects too

List of previous year NLP projects

These are only for your reference. These are not benchmarks for you!!

- The Spring 2024 NLP final projects are available on the website

https://lab.vlanc.co.in/course/spring-2024/projects/

Finding Project/Research Topics

1. 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 it, improve it, understand it, or find new ways to apply it

Encouragement for participation in NLP/LLM datathons

If you want to participate in any competition, do the task as a team, be on leaderboard, this will also be counted towards your contribution in project!

For example:

Datathon@IndoML 2025 – Evaluating LLM-Powered AI Tutors

Project types

- ✓ This is not an exhaustive list, but most projects are one of
  - ✓ Find an application/task of interest and explore how to approach/solve it effectively, often with an existing model
    - ✓ Could be a task in the wild or some existing Kaggle/bake-off/shared task
  - ✓ Implement a complex neural architecture and demonstrate its performance on some data
  - ✓ Come up with a new or variant neural network model or approach and explore its empirical success
    - ✓ All the above are expected to have experiments with numbers and ablations ☺
  - ✓ 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
  - ✓ Rare theoretical or linguistic project: Show some interesting, non-trivial properties of a model type, data, or a data representation

Want to beat the state of the art on something?

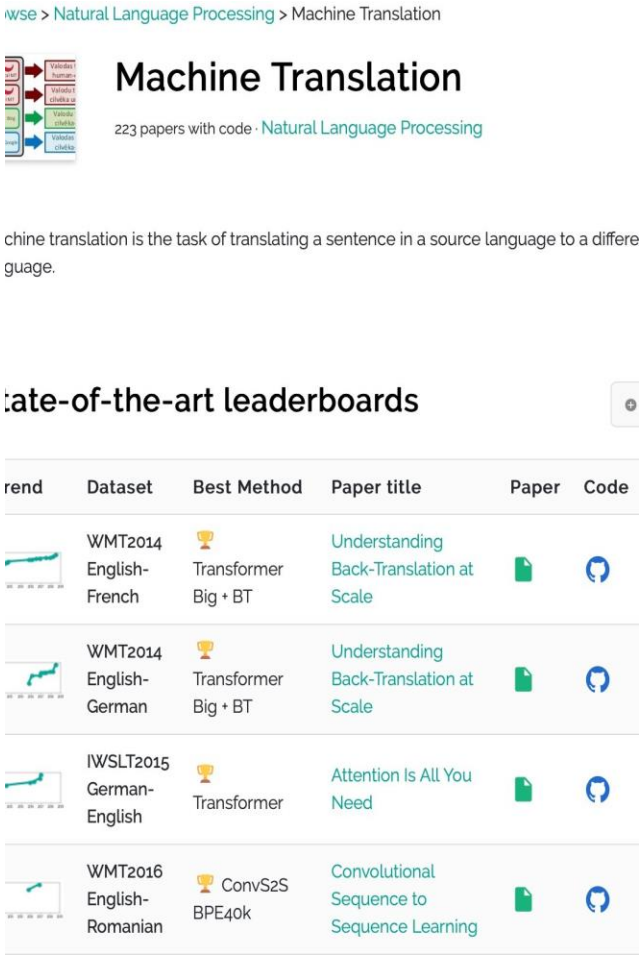
Great new sites that try to collate info on the state of the art

- Not always correct, though

https://paperswithcode.com/sot

a https://nlpprogress.com/

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



2024 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)
```

Can I use train GPT-2 or use ChatGPT to do my final project?

- You need to be very cognizant of how large a model you can train
  - You just don't have the resources to train your own GPT-2 model
  - You probably don't have the resources to even **load** T5 11B
  - But you can use : unslothai/unsloth: Finetune Llama 3.2, Mistral, Phi, Qwen 2.5 & Gemma LLMs 2-5x faster with 80% less memory
- You can use GPT-3 or ChatGPT in your final project
  - Though we can't fund your API use bills
- There are almost certainly interesting projects that you can do using them
  - Analysis projects
  - Learning good prompts
  - Chain of thought reasoning
- **But be careful to remember that you will be evaluated based on what you have done – and not on the amazing ChatGPT output that you show us ("look, it works zero shot")**

How to find an interesting place to start?

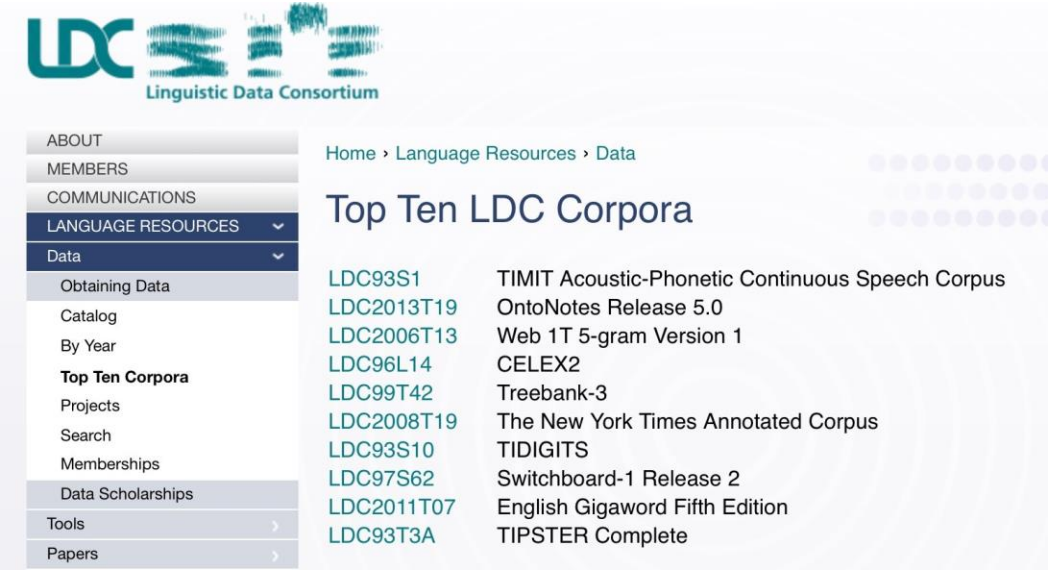
- Look at ACL anthology for NLP papers:
  - https://aclanthology.org/
- Also look at the online proceedings of major ML conferences:
  - NeurIPS https://papers.nips.cc, ICML, ICLR https://openreview.net/group?id=ICLR.cc
- Look at online preprint servers, especially:
  - https://arxiv.org
- Even better: 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

5. Finding data for your projects

- Some people collect their own data for a project – **we like that, but it's trick to do fast!**
  - 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.
    - This let's you learn about real word challenges of applying ML/NLP!
  - **But be careful on scoping things so that this doesn't take most of your time!!!**
- 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

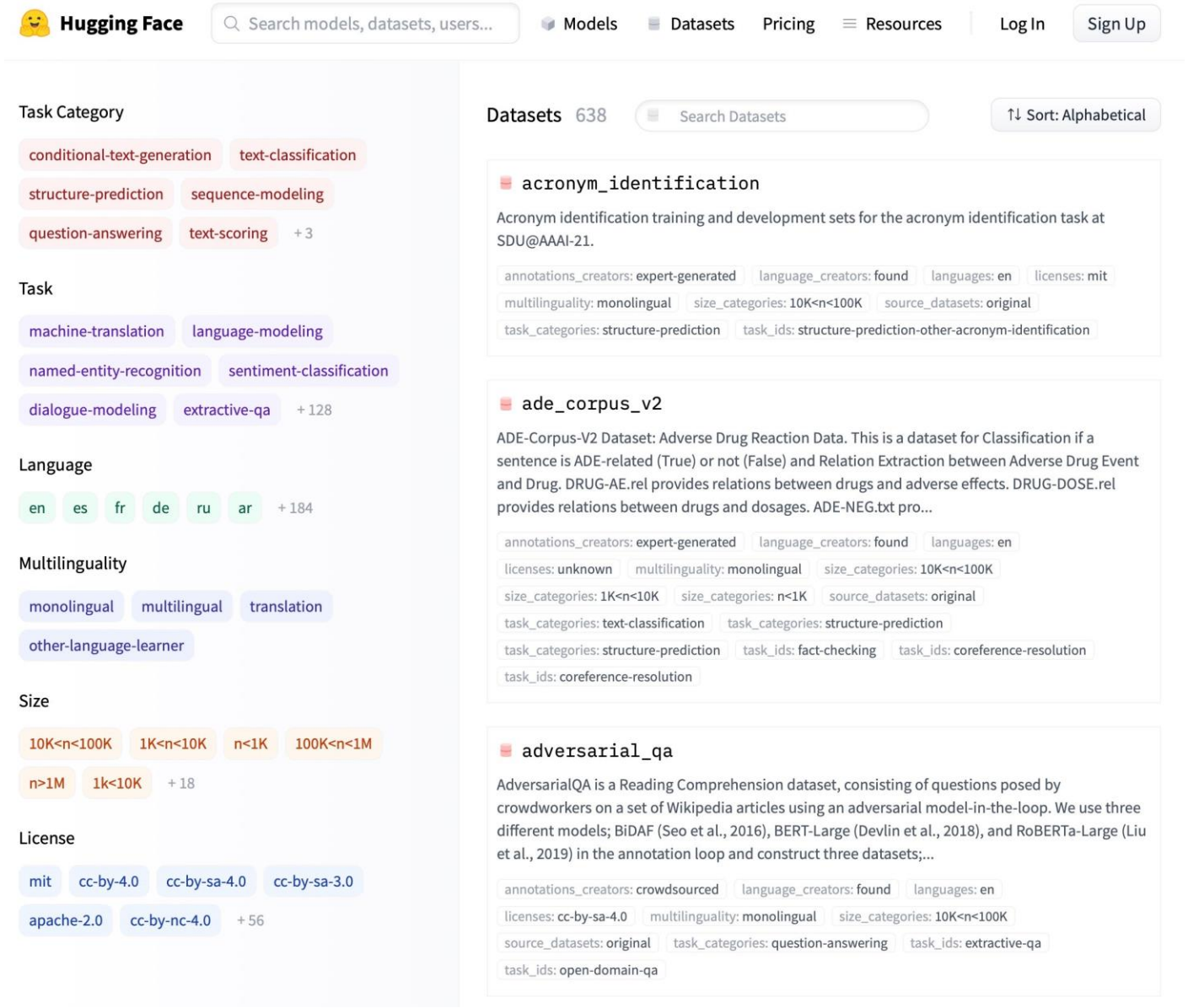
Linguistic Data Consortium

- https://catalog.ldc.upenn.edu/
- Stanford licenses this data; you can get access. Sign up/ask questions 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!



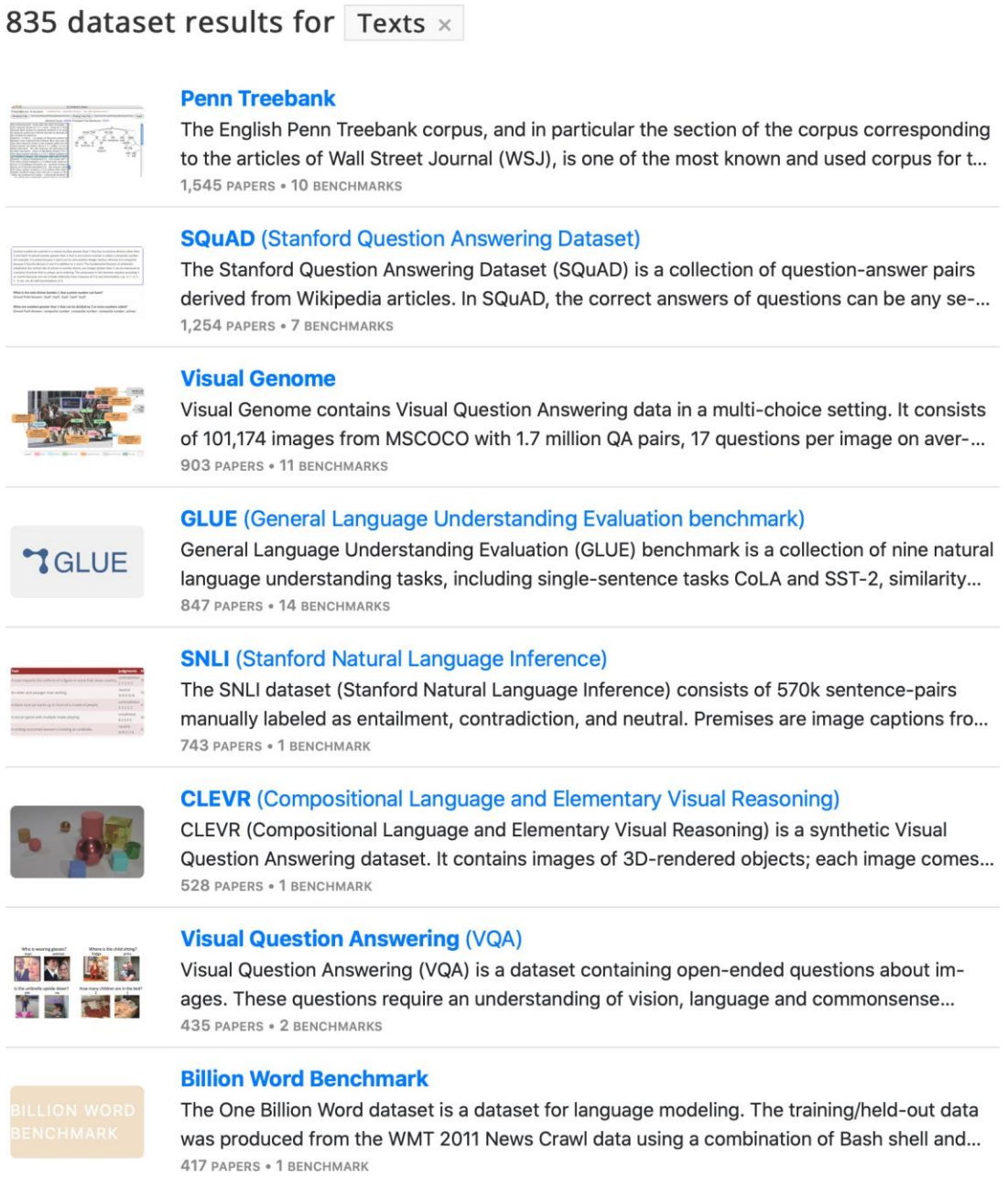
Huggingface Models and Datasets

- https://huggingface.co/datasets



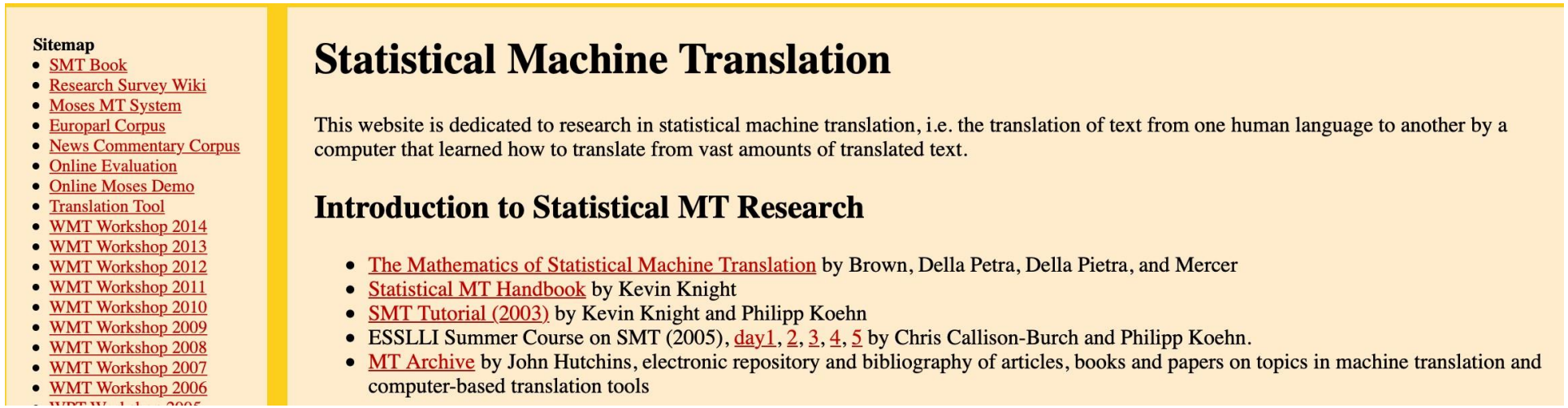
Paperswithcode Datasets

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



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.

Many, many more

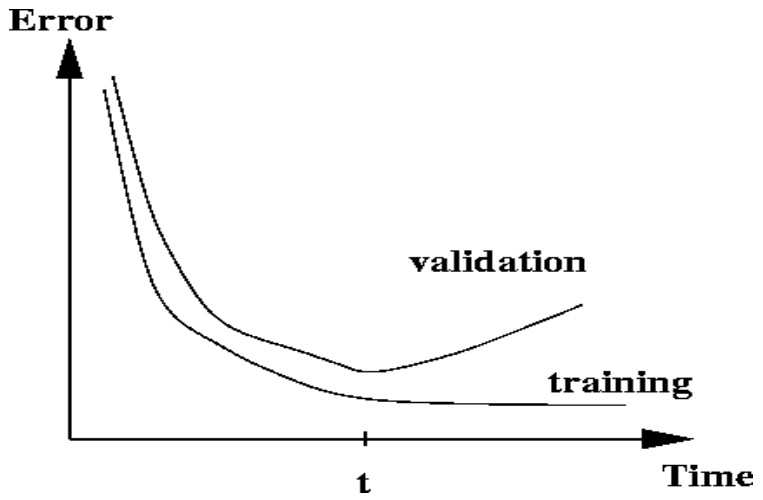
- There are now many other datasets available online for all sorts of purposes
  - Look at Kaggle
  - Look at research papers to see what data they use
  - Look at lists of datasets
    - https://machinelearningmastery.com/datasets-natural-language-processing/
    - https://github.com/niderhoff/nlp-datasets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

5. Care with datasets and in model development

- 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 when 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 give results testing 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!

Experimental strategy

- Work incrementally!
- Start with a very simple model and get it to work!
  - It’s hard to fix a complex but broken model
- Add bells and whistles one-by-one and get the model working with each of them (or abandon them)

- Initially run on a tiny amount of data
  - You will see bugs much more easily on a tiny dataset … and they train really quickly
  - Something like 4–8 examples is good
  - Often synthetic data is useful for this
  - Make sure you can get 100% on this data (testing on train)
    - Otherwise your model is definitely either not powerful enough or it is broken

Experimental strategies

- Train and run your model on a large dataset
  - It should still score close to 100% on the training data after optimization
    - Otherwise, you probably want to consider a more powerful model!
    - Overfitting to training data is **not** something to fear when doing deep learning
      - These models are usually good at generalizing because of the way distributed representations share statistical strength regardless of overfitting to training data
- But, still, you now want good generalization performance:
  - Regularize your model until it doesn't overfit on dev data
    - Strategies like L2 regularization can be useful
    - But normally **generous dropout** is the secret to success

Details matter!!!

- Look at your data, collect summary statistics
- Look at your model's outputs, do error analysis
- Tuning hyperparameters, learning rates, getting initialization right, etc. is **often** important to the successes of neural nets

Project Evaluations and guidelines

- Course Project (1–5 people): **20%**
- **Bonus: 3% (Research)**
- **Late day policy: penalty with delay in number of days**
- **Collaboration policy**: Read the First lecture slide, Academic dishonesty policy and the Honor Code!
  - For projects: It's okay to use existing code/resources, but you **must document** it, and
    - You will be graded on your value-add
  - **If multi-person: Include a brief statement on the work of each team-mate**

What to do in project (default)?

What you do (implementation):

- Finish writing an implementation of any statistical models or neural models (miniBERT, BERT, etc.)
- If you are using transformer-based model, then Fine-tune it for any task
- Extend and improve it in various ways of your choice:
  - Any component
  - Attention mechanism
  - Loss functions, etc.

Why Choose The Default Final Project?

- If you:
  - Have limited experience with research, don’t have any clear idea of what you want to do, or want guidance and a goal, and a leaderboard, even
- Then:
  - Do the default final project!
  - Many people should do it!

Why Choose The Custom Final Project?

- If you:
  - Have some research project that you're excited about (and are possibly already working on)
  - You want to try to do something different on your own
  - You want to see more of the process of defining a research goal, finding data and tools, and working out something you could do that is interesting, and how to evaluate it
- Then:
  - **Do the custom final project!**
- **But note: The final project for CS3126 must substantively involves human language, traditional techniques or neural networks** (the topics of this class)**!**

Gamesmanship

- The default final project is more guided, but it should be the same amount of work
- **It's just that you can focus on a given problem rather than coming up with your own**
- The default final project is also an open-ended project where you can explore different approaches, but to a given problem. **Strong default final projects do new things.**
- There are
  - great default final projects
  - great custom final projects
  - weak default final projects
  - weak custom final projects
  - The path to success is not to do something that looks kinda weak/ill-considered compared to what you could have done with the DFP.

- Neither option is the easy way to a good grade; we giv**e Best Project Awards**

Project Proposal – from every team 5%

Skill: How to think critically about a research paper

- What were the main novel contributions or points?
- Is what makes it work something general and reusable or a special case?
- Are there flaws or neat details in what they did?
- How does it fit with other papers on similar topics?
- Does it provoke good questions on further or different things to try?
  - Grading of research paper review is primarily summative

How to do a good job on your project plan

- You need to have an overall sensible idea (!)
- But most project plans that are lacking are lacking in nuts-and-bolts ways:
  - Do you have appropriate data or a realistic plant to be able to collect it in a short period of time
  - Do you have a realistic way to evaluate your work
  - Do you have appropriate baselines or proposed ablation studies for comparisons
  - Grading of project proposal is primarily formative

Project Proposal (and refinement) – from every team 5%

1. Find a relevant (key) research paper for your topic
   - For DFP, we provide some suggestions, but you might look elsewhere for interesting QA/reading comprehension work
2. Write a summary of that research paper and what you took away from it as key ideas that you hope to use
3. Write what you plan to work on and how you can innovate in your final project work
   - Suggest a good milestone to have achieved as a halfway point
4. Describe as needed, **especially for Custom projects**:
   - A project plan, relevant existing literature, the kind(s) of models you will use/explore; the **data** you will use (and how it is obtained), and how you will **evaluate** success

Project Milestones – from everyone 5%+5%

- **Progress to be shared in your slack project group** as per deadlines!
- Describe the analysis you have done
- Describe the experiments you have run
- Describe the preliminary results you have obtained
- Describe how you plan to spend the rest of your time

You are expected to **have implemented some system** and to **have some initial experimental results** to show by this date (except for certain unusual kinds of projects)

Project Grading

- **Project Proposal refinement (5%)**: Improvement and refinement of the initial project proposal. Incorporation of feedback from the first evaluation phase.
- **Problem identification/Motivation/Dataset collection/Literature survey (5%)**: Critical analysis of existing work and identification of gaps. Highlight your innovation/uniqueness!
- **Methodology (5%)**- Appropriateness and rigor of the proposed methodology. clarity in explanation of methods, tools, and techniques. Justification of the chosen methods in relation to the problem.
- **Timeliness/Deliverables/Preliminary Final results (5%)**- Relevance and accuracy of data. Ability to analyze and interpret the results within the project's scope.
- **BONUS: Exceptional/Going Beyond - Research (3%)-** This bonus is awarded for those who take their projects a step further by translating their work into high-quality research or publication.
- **If you can demonstrate that your project has been developed into a well-researched paper or article suitable for a reputable venue, you will earn an additional 3% on your final grade.**

Final Project Report/Research paper template

Deadline: 10 Dec 2025, 11:59 PM

- Writeup quality is very important to your grade!!!
  - Look at recent years' papers/report for examples

Abstract Introduction	Prior related work	Dataset	Methodology/ Model
Experiments	Results	Analysis & Conclusion	Demo/ System (if any)

Mahindra™ University
Global Thinkers. Engaged Leaders.

Good luck with your projects!

Acknowledgments

- Inspirations of Project guidelines from CS224N.
- These slides were adapted from the book SPEECH and LANGUAGE PROCESSING: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition
- Some modifications from CS224N presentations and resources found in the WEB by several scholars.