Week 5: Scoping an NLP project

Text Analytics and Natural Language Processing
Instructor: Benjamin Batorsky

Final project outline (Assignment 4)

- Now on Canvas!
- Due (**latest!**): Week 6, Monday, 7/27, 6:30pm ET
- Earlier submission = earlier feedback
  - NOTE: Covering transformer models this week and scoping next week
- Must include
  - Names of project members (no more than 3!)
  - Overview of sections
- Can include
  - Any questions you have

Outline must detail your strategy around the following sections of the project

- A clear research question or problem to address
- A description of the dataset and justification of why it is appropriate to solve the problem
- Exploratory analyses of the dataset
- A description of the methodology to be applied and justification of its use
- Deployment strategy
  - (optional, if appropriate)

Final project **DUE 8/7 (11:59pm)**

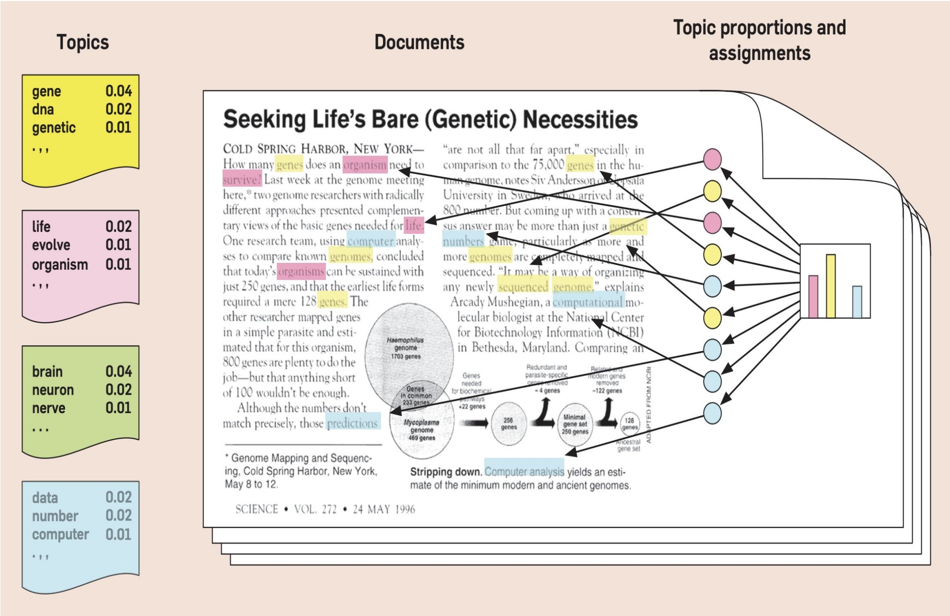
- I had mentioned 8/10, but the school requires 8/7 - Let me know if this is any issue
- Submission format
  - Github repo or set of files
    - Code for exploration, processing, modeling (notebooks, scripts)
    - Data files
    - Write-up
      - All project participants
      - Link to repo (make it public!) or reference to specific files
      - All sections from outline, but expanded with results, conclusions, etc
  - MUST include a runnable version
    - Example: Subsample of full data, simplified code
- On Canvas: Submit write-up
- Presentation
  - If you/your group is willing: Present to the class on the last day
  - This will take the place of a full write-up (though you will still need to document your code!)
  - *Advised for group projects!*

Exciting stuff: Tentative speaker schedule

- Weeks 5, 6 and 7 will have some speakers talking about real-world NLP
- Week 5:
  - Monday @ 7pm-8pm: Andrew Thierrault, former Chief Data Officer, City of Boston
  - Wednesday @ 6:30-7pm: Matthew Honnibal, SpaCy author, Explosion.ai founder
- Week 6:
  - Wednesday @ 7-8pm: Mady Mantha, Senior Technical Evangelist, Rasa AI
- Week 7:
  - Monday: Maryam Jahanshahi, Research Scientist, TapRecruit

Review of what we've covered

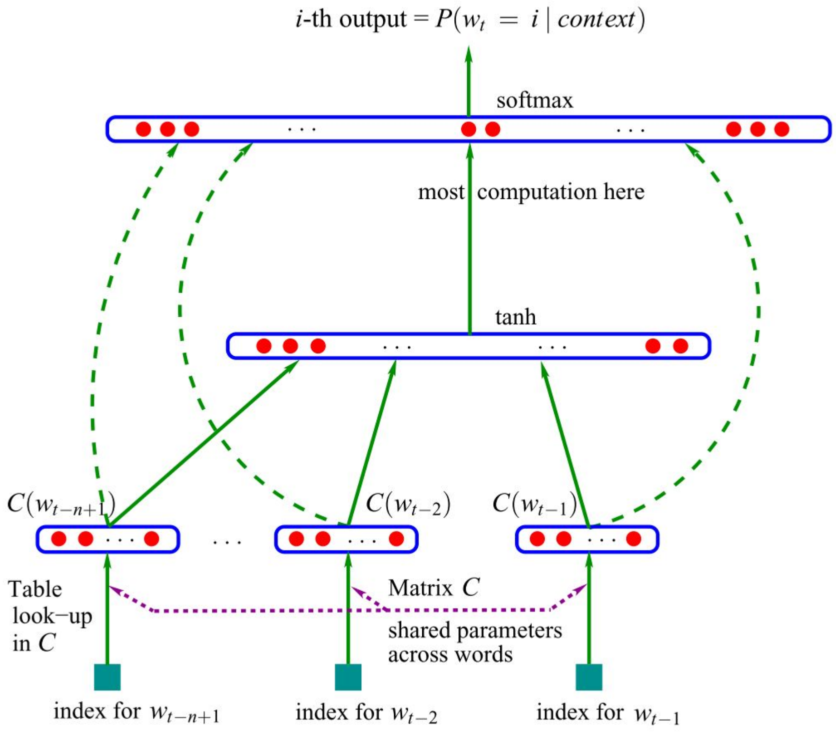
- 40s-50s: Machine translation era
- 60-70s: Shift towards semantic-driven processing
- 70s to 80s: Community expansion
- 90s-00s: Probabilistic/Statistical models
- 2000s: Neural Language models
- 2008: Multi-task learning
- 2013: Word embeddings
- 2014: Expansion of Neural models
- 2015: Attention
- 2018 and beyond: Language model advancements



http://www.cs.columbia.edu/~blei/papers/Blei2012.pdf

Review of what we've covered

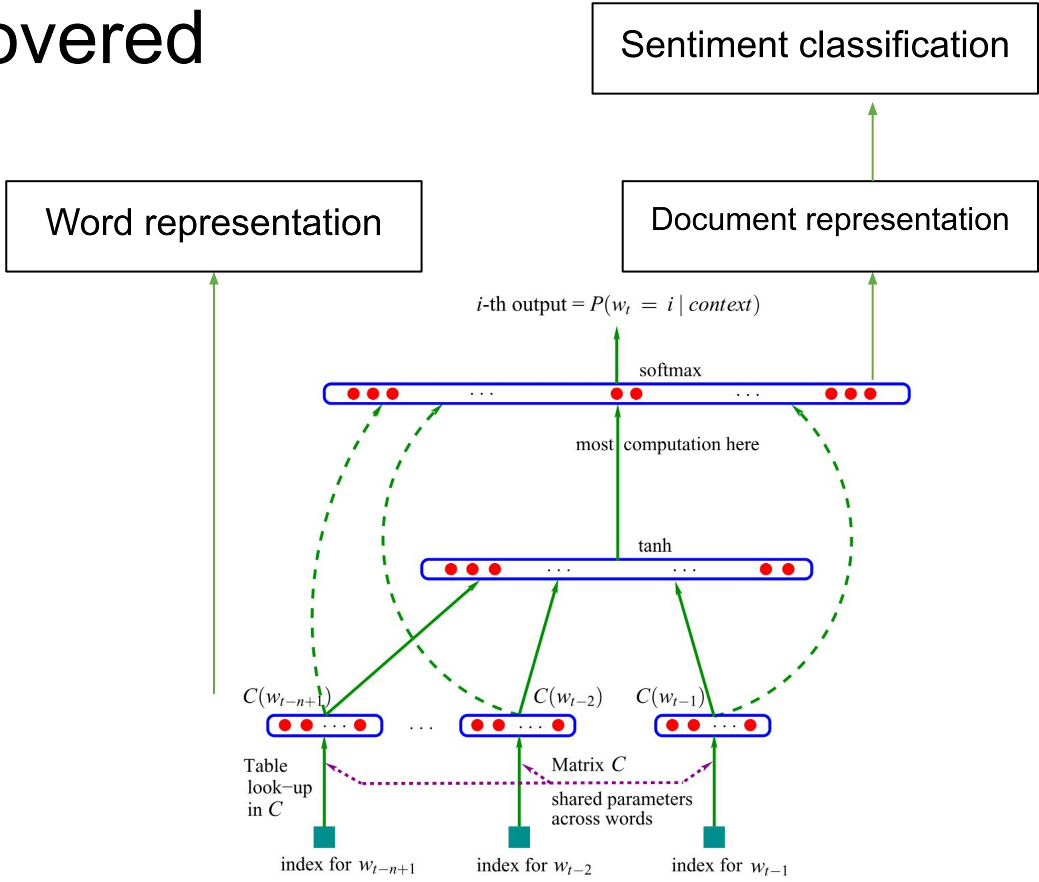
- 40s-50s: Machine translation era
- 60-70s: Shift towards semantic-driven processing
- 70s to 80s: Community expansion
- 90s-00s: Probabilistic/Statistical models
- 2000s: Neural Language models
- 2008: Multi-task learning
- 2013: Word embeddings
- 2014: Expansion of Neural models
- 2015: Attention
- 2018 and beyond: Language model advancements



[A Neural Probabilistic Language Model](https://papers.nips.cc/paper/1839-a-neural-probabilistic-language-model.pdf)
(https://papers.nips.cc/paper/1839-a-neural-probabilistic-language-model.pdf)

Review of what we’ve covered

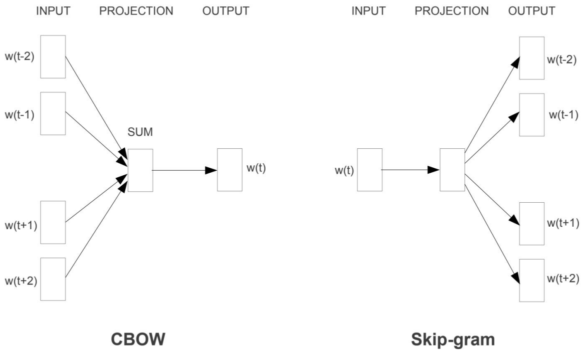
- 40s-50s: Machine translation era
- 60-70s: Shift towards semantic-driven processing
- 70s to 80s: Community expansion
- 90s-00s: Probabilistic/Statistical models
- 2000s: Neural Language models
- 2008: Multi-task learning
- 2013: Word embeddings
- 2014: Expansion of Neural models
- 2015: Attention
- 2018 and beyond: Language model advancements



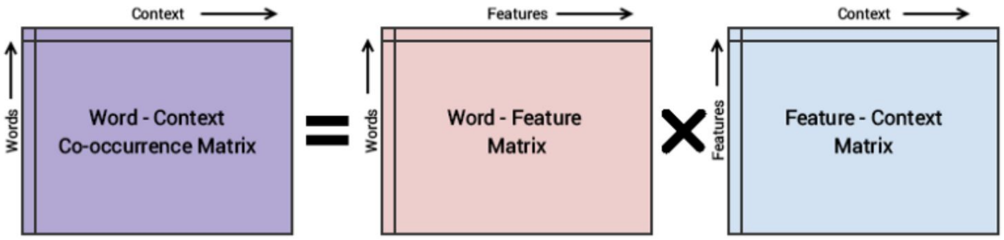
[A Neural Probabilistic Language Model](https://papers.nips.cc/paper/1839-a-neural-probabilistic-language-model.pdf)
(https://papers.nips.cc/paper/1839-a-neural-probabilistic-language-model.pdf)

Review of what we've covered

- 40s-50s: Machine translation era
- 60-70s: Shift towards semantic-driven processing
- 70s to 80s: Community expansion
- 90s-00s: Probabilistic/Statistical models
- 2000s: Neural Language models
- 2008: Multi-task learning
- 2013: Word embeddings
- 2014: Expansion of Neural models
- 2015: Attention
- 2018 and beyond: Language model advancements



https://arxiv.org/pdf/1301.3781.pdf

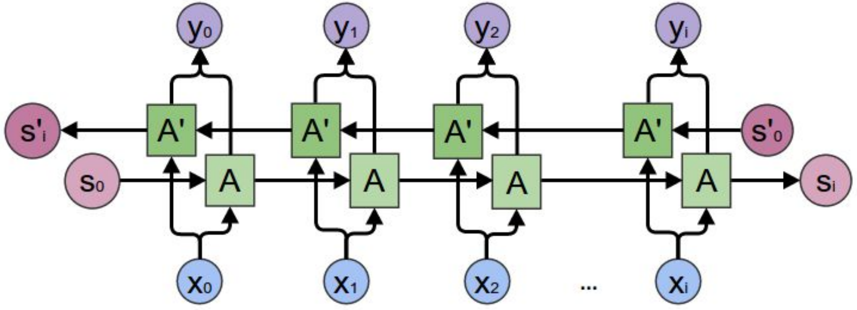


Conceptual model for the GloVe model's implementation

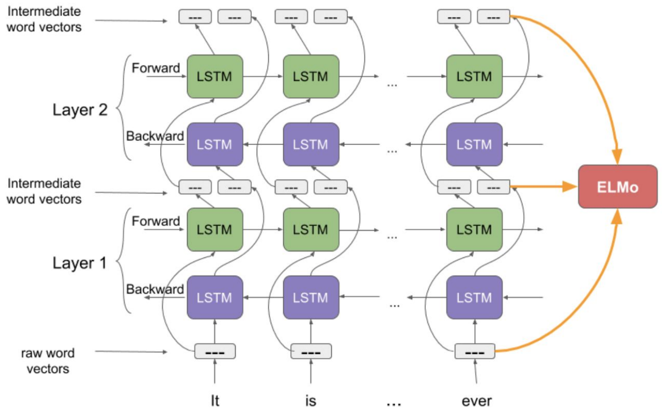
Implementing Deep Learning Methods and Feature Engineering for Text Data: The GloVe Model

Review of what we’ve covered

- 40s-50s: Machine translation era
- 60-70s: Shift towards semantic-driven processing
- 70s to 80s: Community expansion
- 90s-00s: Probabilistic/Statistical models
- 2000s: Neural Language models
- 2008: Multi-task learning
- 2013: Word embeddings
- 2014: Expansion of Neural models
- 2015: Attention
- 2018 and beyond: Language model advancements



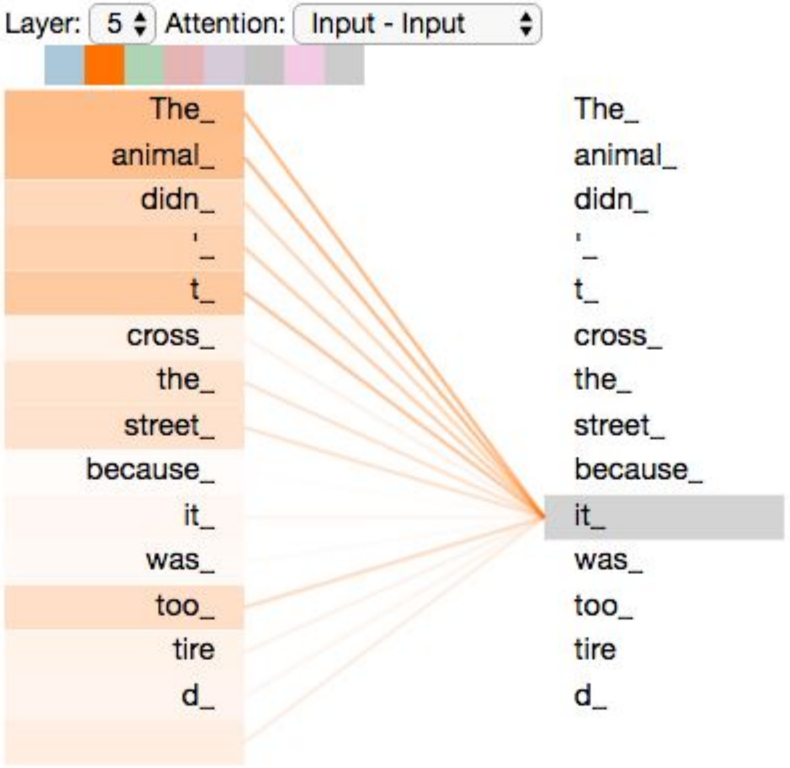
http://colah.github.io/posts/2015-09-NN-Types-FP/



https://www.analyticsvidhya.com/blog/2019/03/learn-to-use-elmo-to-extract-features-from-text/

Review of what we've covered

- 40s-50s: Machine translation era
- 60-70s: Shift towards semantic-driven processing
- 70s to 80s: Community expansion
- 90s-00s: Probabilistic/Statistical models
- 2000s: Neural Language models
- 2008: Multi-task learning
- 2013: Word embeddings
- 2014: Expansion of Neural models
- 2015: Attention
- 2018 and beyond: Language model advancements



http://jalammar.github.io/illustrated-transformer/

Review of what we've covered

- 40s-50s: Machine translation era
- 60-70s: Shift towards semantic-driven processing
- 70s to 80s: Community expansion
- 90s-00s: Probabilistic/Statistical models
- 2000s: Neural Language models
- 2008: Multi-task learning
- 2013: Word embeddings
- 2014: Expansion of Neural models
- 2015: Attention
- 2018 and beyond: Language model advancements

https://www.gettyimages.com/photos/paul-morigi

Which is the *best* approach?

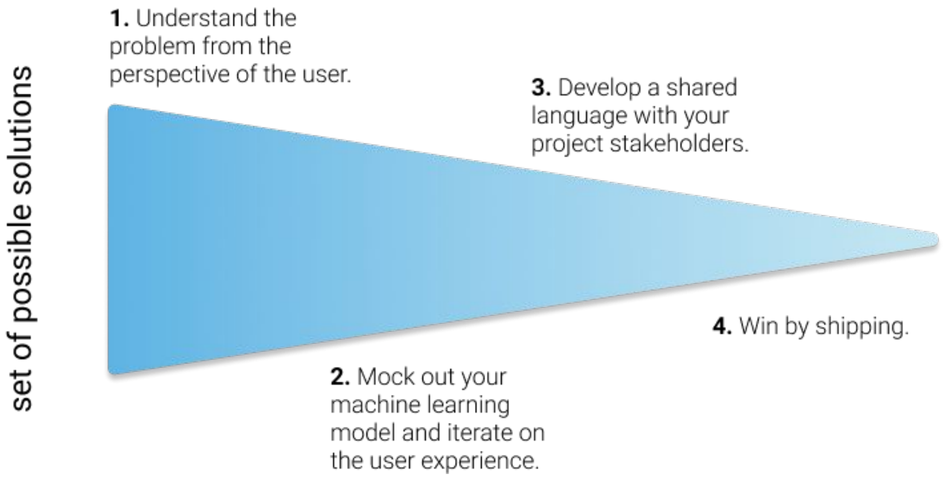
Question	**Type of task**	**Word counts**	**Topic models**	**Word vectors**	**LSTM model**	**Transformer model**
Is a review positive or negative?						
What are the different types of reviews?						
How do I tell whether people liked my movie?						

Which is the *best* approach?

Question	Type of task	Word counts	Topic models	Linear classifier	LSTM model	Transformer model
Is a review positive or negative?	Classification	Yes	Maybe	Yes	Yes	Maybe
What are the different types of reviews?	Clustering	Yes	Yes	No	No	No
Can we provide a review filtering mechanism?	Research, analysis	Yes	Maybe	Maybe	Maybe	Maybe

Scoping an NLP project

- Setup
  - Understand the problem
  - Inventory of solutions
    - Impact
    - Feasibility
    - Requirements
  - Setting up code base
- Data collection/labelling/sourcing
- Model exploration
- Deployment
- (throughout) Debugging and testing

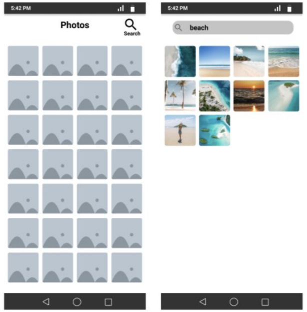


https://www.jeremyjordan.me/ml-requirements/

Understanding the problem

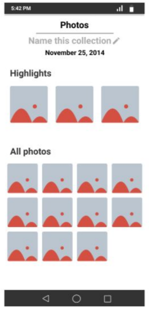
Review filtering mechanism

- Who are the end users?
    - Viewers
    - Creators
    - Internal
- Is this an NLP problem?
    - Are there text features?
    - Are the text features informative?
- What are some possible solutions?
    - Text query
    - Sentiment
    - Topic
    - Relevance



Text query

Topic-centric query



Group photos and select highlights

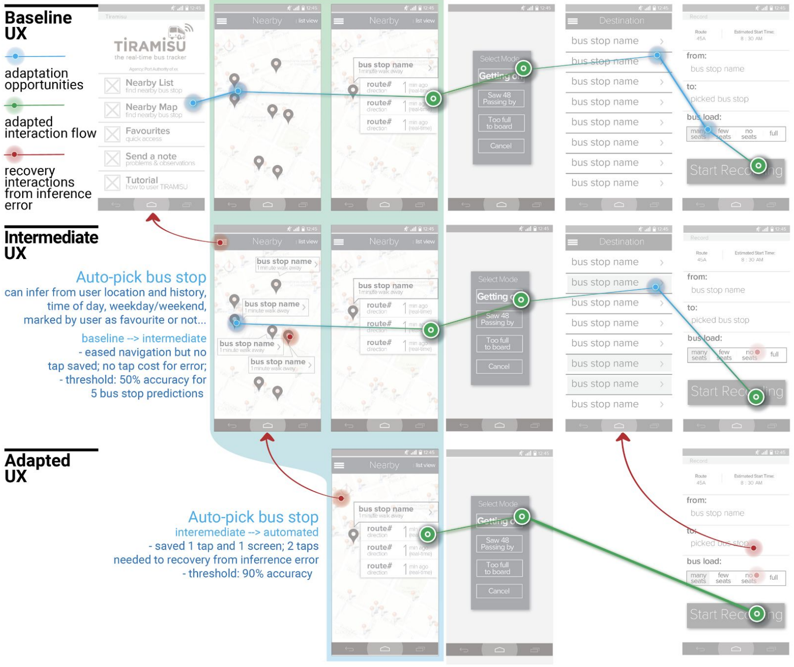
https://www.jeremyjordan.me/ml-requirements/

Clear research questions/problems for projects

- Research question: What are you actually trying to answer?
    - In problem framing: What are you actually trying to solve?
- Example problem: Netflix users are interested in looking at positive and negative reviews (see Amazon “top positive”, “top critical”)
- Clear
    - Question framing: Can we develop a methodology to enable users to sort movie reviews by sentiment?
    - Problem framing: Movie viewers are interested in being able to read positive and negative reviews. We will develop a method to enable them to sort by positivity of review.
- Not clear:
    - Predict the sentiment of movie reviews
    - Classify reviews as positive or negative

Testing possible solutions

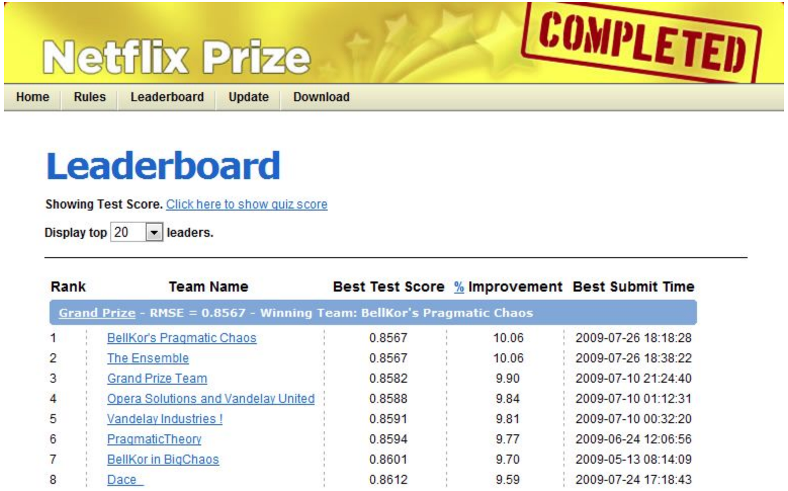
- Assessing value: “Coming soon” feature
  - Provide a way for users to select the feature
  - Track the number of people that select it
- “Wizard of Oz” design
  - “Simulating” the solution: Manually providing the output the model would provide
- Wireframes
  - Simple visualization of what the output would be



https://www.behance.net/gallery/34746505/machine-learning-ready-UI

Impact/Feasibility assessment: Motivating examples

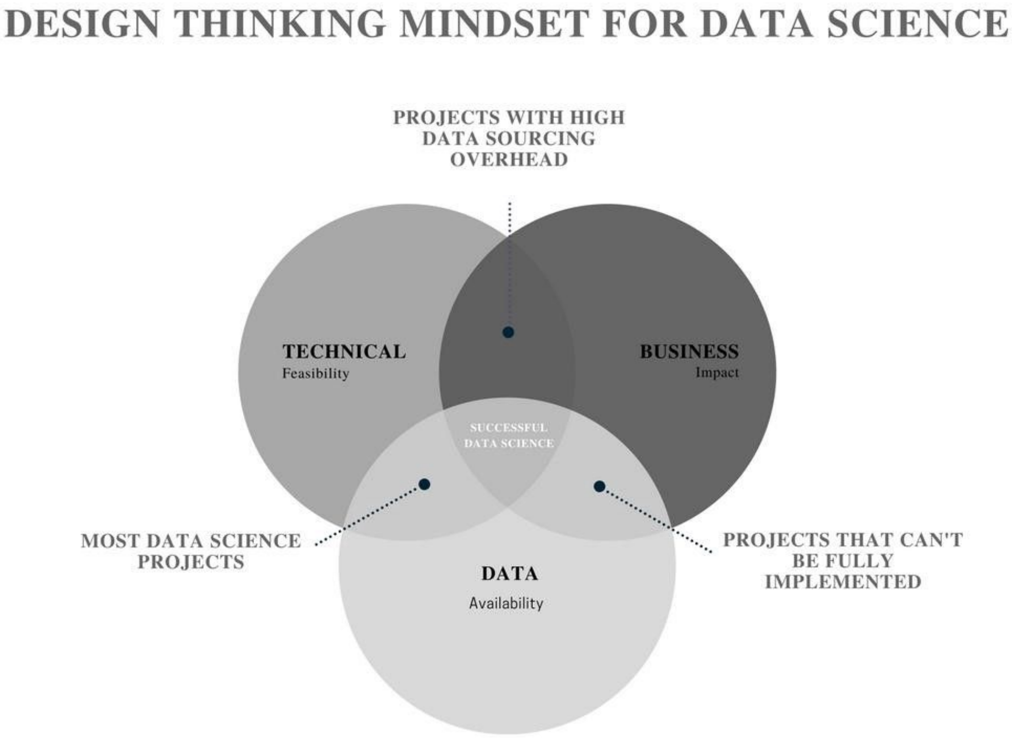
- Netflix prize (2006)
  - Netflix's assessment that their recommendation system extremely profitable
  - $1M to algorithm that improved the recommendation system
  - Never actually implemented: Difficulty of implementation outweighed benefit
  - Trained on DVD rental, not relevant for streaming
- Counterexample: Apple purchased Siri technology for $200M, continues to be major part of company
- Generally: Consider performance/feasibility trade-offs



Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace_	0.8612	9.59	2009-07-24 17:18:43

https://netflixtechblog.com/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429

Main areas of consideration when targeting your project



https://towardsdatascience.com/a-design-thinking-mindset-for-data-science-f94f1e27f90

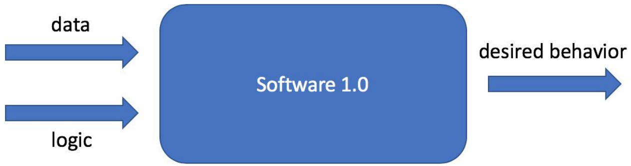
Movie spoilers example

Goal: Can we create method for identifying whether a review is a spoiler?

- Who is the user of this product?	Imdb/other companies, browser/google play users, imdb end user, social media companies, production companies
- What data might we need?	Movie synopsis, reviews tagged, screenplay
- What data/features do we have to work with?	Reviews, movie synopsis
- Is the a simple heuristic here that might be preferable over a model?	
- How can we iterate here and find improvement?	
- What is the benefit from getting more elaborate with our design? What is the cost?	

Assessing impact: Software 1.0

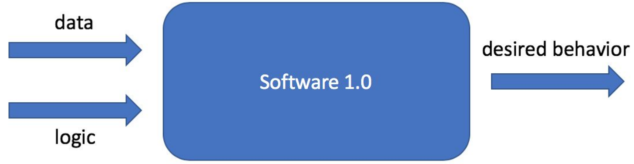
- Software 1.0
  - Data+Logic = Behavior
  - Logic
    - Sets of rules, functions, etc
    - Similar to
  - Low-complexity (at first), transparent
  - Likely less performant, does not optimize
- Examples
  - Identifying stopwords based on a list
  - Inventory/pattern-based named-entity recognition

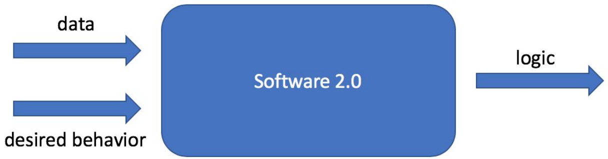


https://www.jeremyjordan.me/ml-projects-guide/

Moving to Software 2.0

- Software 1.0
  - Data+Logic = Behavior
  - Logic
    - Sets of rules, functions, etc
    - Similar to
  - Low-complexity (at first), transparent
  - Likely less performant, does not optimize
- Software 2.0
  - Data+Behavior = Logic
  - Behavior
    - Dataset labels
    - Existing clusters/seed items
  - Logic “learned” by model
  - More performant, dynamic extraction of relationships





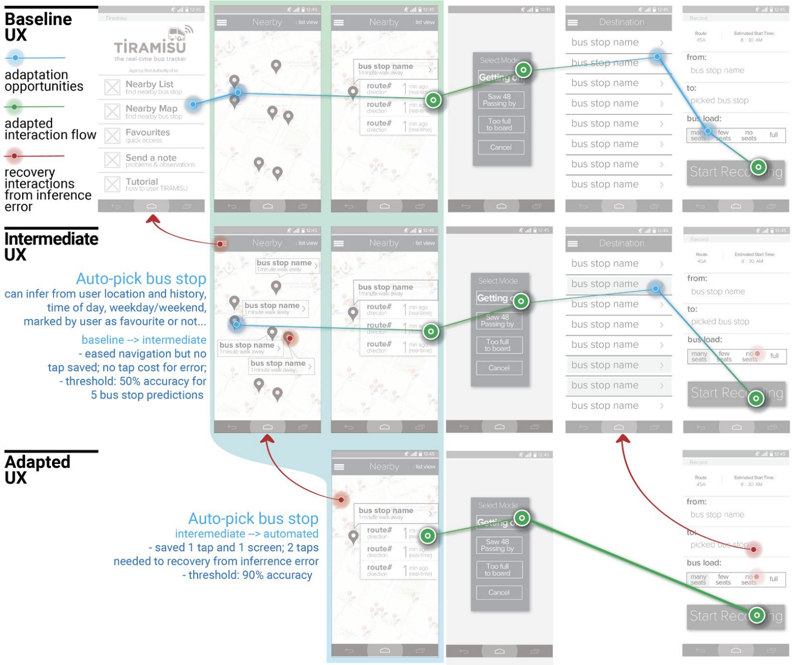
https://www.jeremyjordan.me/ml-projects-guide/

Software 1.0 vs Software 2.0

Use-case	Software 1.0	Software 2.0
Removing stopwords	Based on list/set of heuristics as in SpaCy/Sckit-learn, pruning the vocabulary, inventories	Feature importance/univariate models, embeddings of stopwords/non-stopwords
Named-entity recognition	List or pattern-matching	NER Model/seq-to-seq
Sentiment analysis	Dictionary-based	Transformer model, LSTM, classification model
Translation		
Movie spoilers identification	Match between synopsis and review, dictionary of words for plot, look for “spoilers”, cosine similarity	Classification model

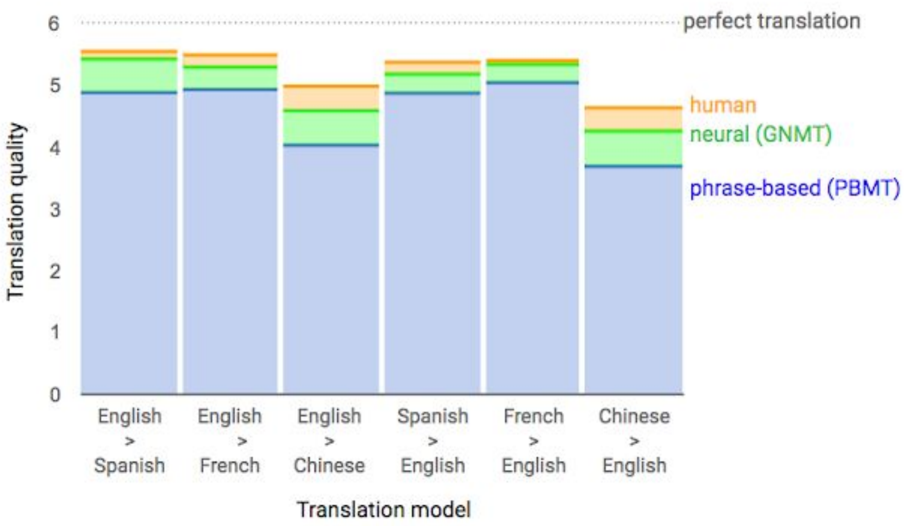
Software 1.0 v 2.0 solutions to movie spoilers

Thinking about impact: Bus route selection



https://www.behance.net/gallery/34746505/machine-learning-ready-UI

Impact: Translation



Input sentence:	Translation (PBMT):	Translation (GNMT):	Translation (human):
李克強此行將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers.	Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada.

Spanish->English	Uno no es lo que es por lo que escribe, sino por lo que ha leído.	One is not what is for what he writes, but for what he has read.	You are not what you write, but what you have read.	You are who you are not because of what you have written, but because of what you have read.

https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html

Software 1.0 vs Software 2.0

Use-case	Software 1.0	Software 2.0	impact
Removing stopwords	Based on list/set of heuristics as in SpaCy/Sckit-learn, pruning the vocabulary, inventories	Feature importance/univariate models, embeddings of stopwords/non-stop words	Transparency can decrease (trust), Might lessen the bias, more generalizable
Named-entity recognition	List or pattern-matching	NER Model/seq-to-seq	
Sentiment analysis	Dictionary-based	Transformer model, LSTM, classification model	
Translation			
Movie spoilers identification	Match between synopsis and	Classification model	Uncovers patterns/relationship

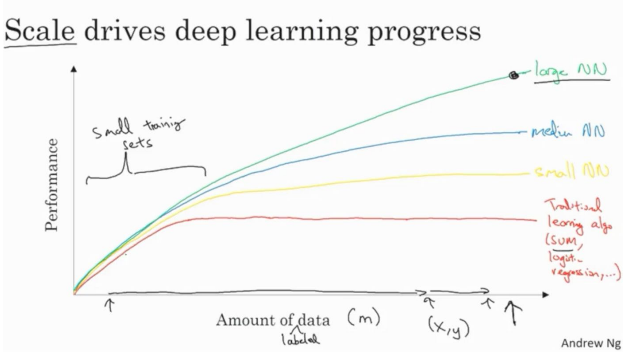
Impact of Software 1.0 vs Software 2.0

Use-case	Software 1.0	Software 2.0	Impact
Removing stopwords	Based on list/set of heuristics as in SpaCy/Sckit-learn		
Named-entity recognition	List or pattern-matching		
Sentiment analysis			
Translation			
Movie spoilers			

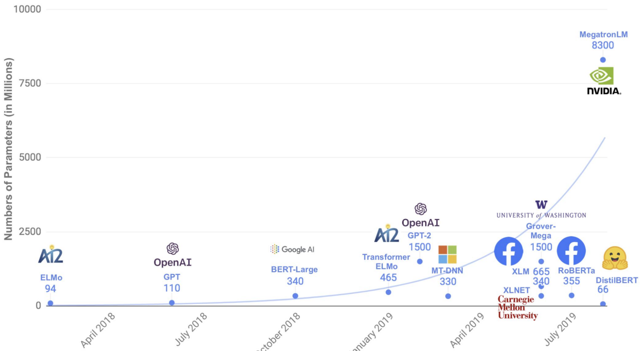
Deep learning for movie spoilers

Feasibility assessment: Computational complexity

- Rules of thumb for deep learning
  - Acceptable performance: 5k labelled examples
  - Near-human performance: 10M examples
- Training time for transformer models
  - Pre-training BERT: 4 days on 4-16 TPUs
  - Fine-tuning BERT (freezing layers, re-training)
    - Hours on a GPU, not optimized/possible on CPU
  - Pre-training DistilBERT: Hours on a GPU, not optimized/possible on CPU
- For most use cases
  - Starting with DL models likely doesn't make sense
  - DL model-based solution may be too complex



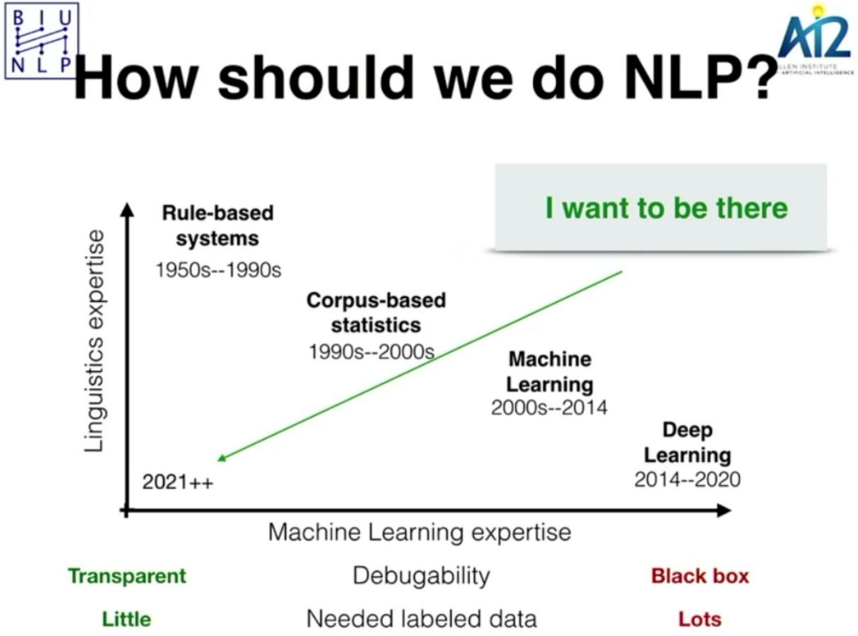
Andrew Ng. Neural Networks and Deep Learning Coursera Course.



https://arxiv.org/pdf/1910.01108.pdf

Deep learning and transparency

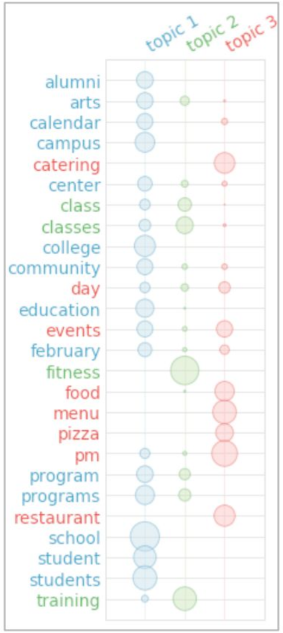
- Movement of NLP towards a "discipline" within ML
  - Less focus on linguistics, more focus on model architectures
  - Less transparency/debugability
  - Greater dependency on data
- Better performance?
  - Vulnerability to "attacks"
  - Dependence on particular text elements

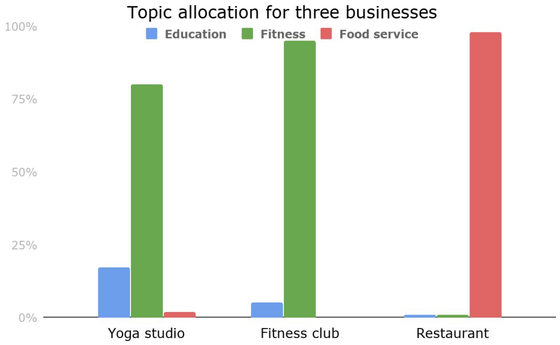


[Yoav Goldberg: The missing elements in NLP (spaCy IRL 2019)]

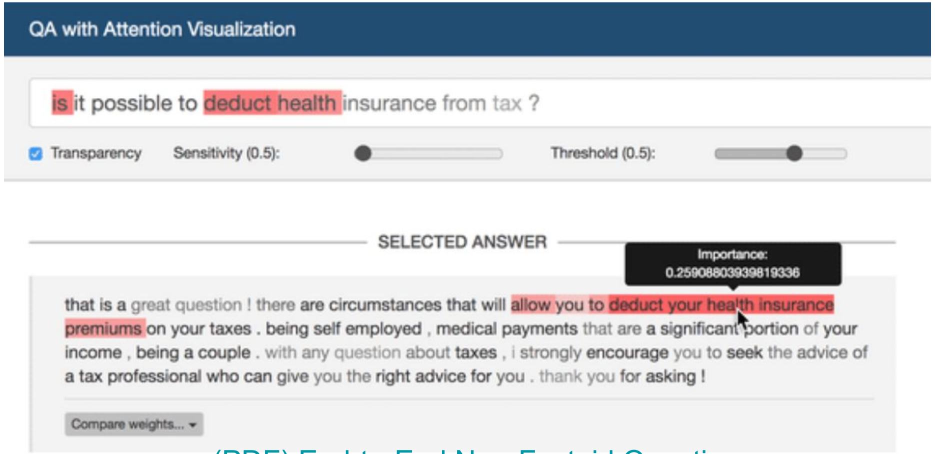
“Shallow” learning vs Deep learning

Product similarity





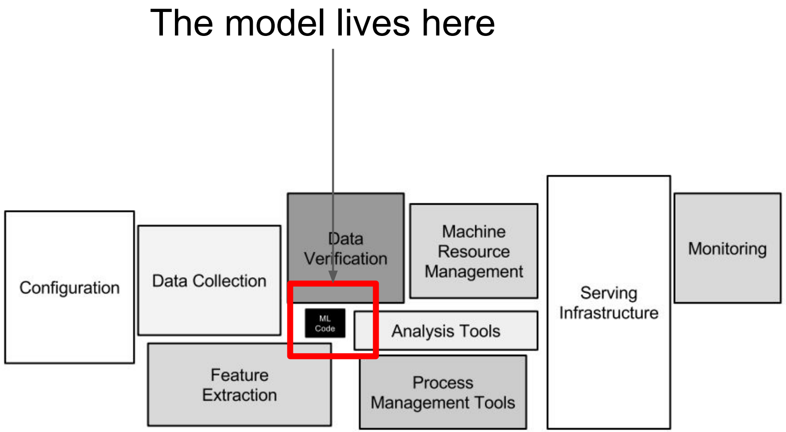
Circles are sized according to “relevance” to each topic



[(PDF) End-to-End Non-Factoid Question Answering with an Interactive Visualization of Neural Attention Weights]()

Technical debt

- Incurred by moving quickly in engineering
  - Similar to financial debt: May not be “bad”, but needs to be handled
- Technical debt in ML products
  - Actual model very small part of product codebase
  - Model has many dependencies, logic needs to be in place
  - Correction cascades
  - Undeclared customers
- Addressing debt
  - Think about the complexity of your system
  - Maybe design custom solution rather than adding dependencies (e.g. SpaCy)
  - Important initiative, but does not add features



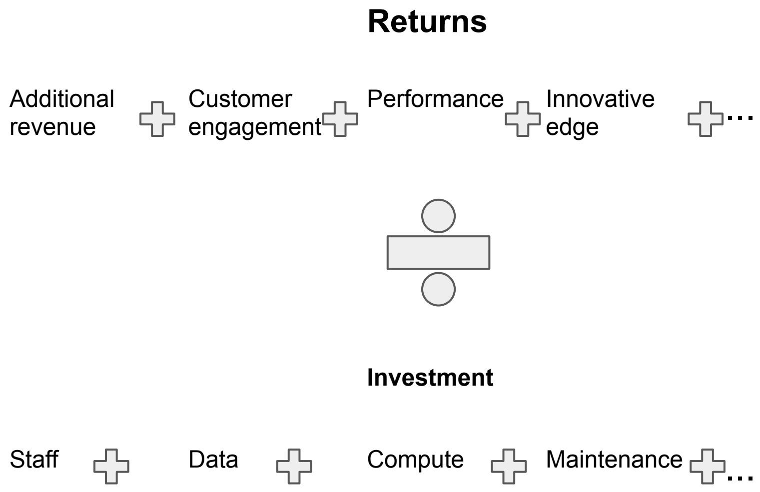
Hidden Technical Debt in Machine Learning Systems (https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf)

Feasibility Software 1.0 vs Software 2.0

Use-case	Software 1.0	Software 2.0	Feasibility
Removing stopwords	Based on list/set of heuristics as in SpaCy/Sckit-learn		
Named-entity recognition	List or pattern-matching		
Sentiment analysis			
Translation			
Movie spoilers			

Assessing impact/feasibility trade-off: Return on Investment

- Returns: What the initiative is likely to provide
- Investment: What the initiative is likely to need
- Project options
  - Baseline: The current process
  - Minimum viable product: Solution where RoI > 1
  - Stage 2, 3, etc
- Needs careful design/assessment of measures



RoI Software 1.0 vs Software 2.0

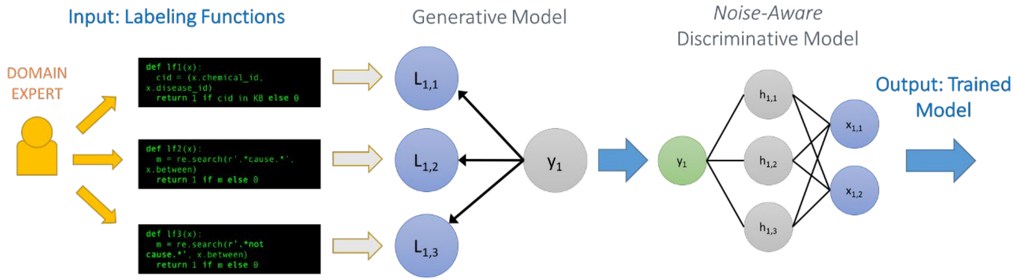
Use-case	Software 1.0	Software 2.0	Impact	Feasibility	RoI
Removing stopwords	Based on list/set of heuristics as in SpaCy/Sckit-learn				
Named-entity recognition	List or pattern-matching				
Sentiment analysis					
Translation					
Movie spoilers					

Choosing a methodology for your final project

- List the possible solutions based on your question/problem
  - Example: Movie review sentiment sorting
  - Dictionary-based scoring, sort high/low
  - Classification based on bag-of-words test features, sort by positive class probability
  - Fine-tuning BERT trained on web text
  - Training BERT from scratch
- Assess the impact, feasibility and compare
- Not sufficient
  - Pre-trained model/class code and run on new data
- Sufficient
  - Testing a baseline model versus a more advanced model, assessing results
  - Deployment of an appropriate methodology (even if it’s a simple one)

Labelling "types"

- Pre-labeled
  - Question answering (SQUaD)
  - Sentiment analysis (IMDB sentiment)
  - Others
- Self-labeled
  - Interactions with social media feed
  - Word/sentence sequence (e.g. for language modelling)
- Labels derived from heuristics ("weak" supervision)
    - Rules engines (Drools, SpaCy's matchers)
    - "Weak" supervision software (Snorkel)
- "Active learning"
- Unlabelled
  - Most use-cases, this is where you're at!



https://towardsdatascience.com/snorkel-a-weak-supervision-system-a8943c9b639f

Collection process

- Observational
  - Data collected by observing, limited/no control over groups/intervention
  - Cohort study: Follow a group over time
    - Example: User activity after a website change (post-mortem)
  - Case-control: Observing two groups who were exposed to different conditions
    - Example: User activity using two different versions of software
- Experimental
  - Control over groups/intervention
  - Controlled trial: Control over intervention, less control over allocation
    - Example: User activity after a controlled website change (A-A testing)
  - Randomized controlled trial: Complete control over experimental/control allocation and administration of intervention
    - Example: A/B testing with random allocation

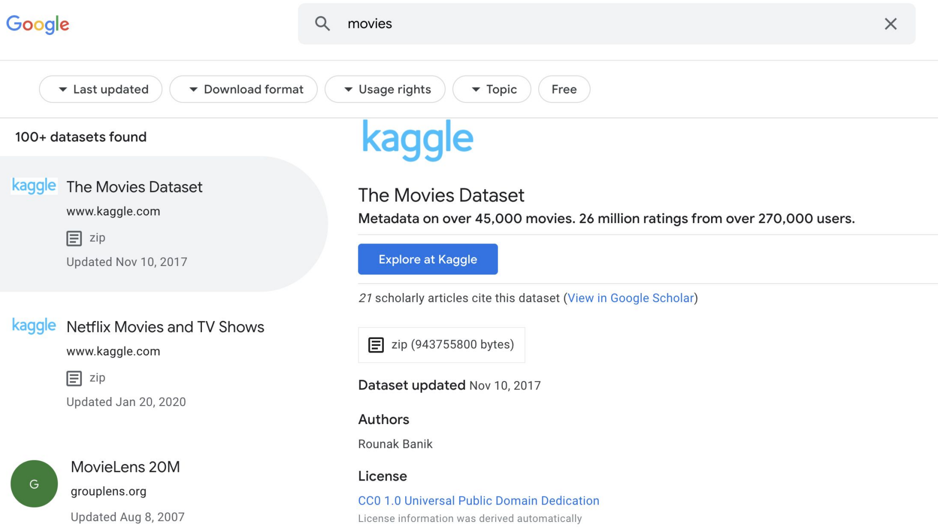
Other data considerations

- Size
  - Near-human performance for NN: 10M observations
  - Per-character unicode ~2-4 bytes, 10M x 1000-character = 3 GB
- Domain
  - Language
    - Do learnings generalize across languages?
  - Subject area
    - Wikipedia vs Biology papers
  - Source
    - Do learnings generalize from text messages to scientific literature?
- Cost

Potential data sources

Google dataset search

https://datasetsearch.research.google.com/



Public databases

https://github.com/awesomedata/awesome-public-datasets#naturallanguage

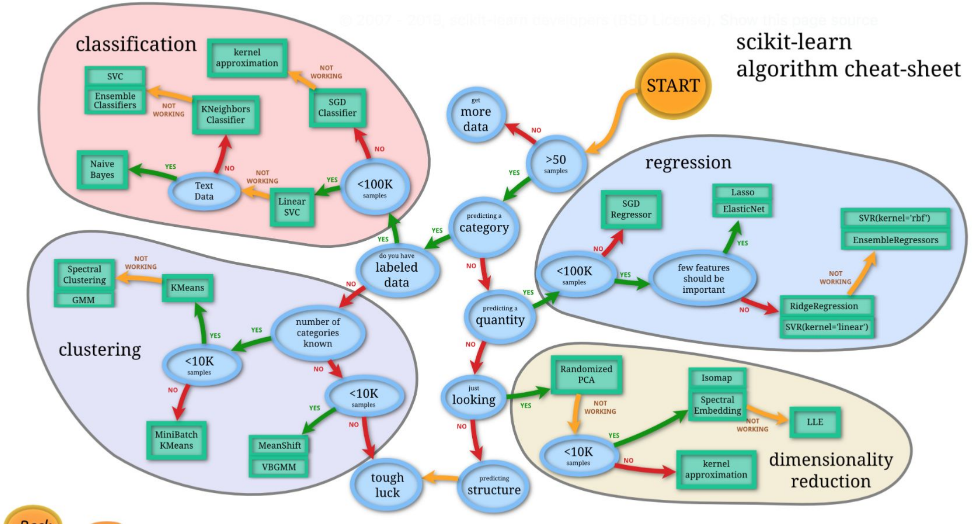
- Automatic Keyphrase Extraction
- The Big Bad NLP Database
- Blizzard Challenge Speech - The speech + text data comes from [...]
- Blogger Corpus
- CLiPS Stylometry Investigation Corpus [fixme]
- ClueWeb09 FACC
- ClueWeb12 FACC
- DBpedia - 4.58M things with 583M facts
- Flickr Personal Taxonomies
- Freebase of people, places, and things [fixme]
- German Political Speeches Corpus - Collection of political speeches from [...]
- Google Books Ngrams (2.2TB)
- Google MC-AFP - Generated based on the public available Gigaword dataset [...]
- Google Web 5gram (1TB, 2006)
- Gutenberg eBooks List
- Hansards text chunks of Canadian Parliament
- LJ Speech - Speech dataset consisting of 13,100 short audio clips of a [...]

Choosing a dataset for the final project

- Supervised problem (e.g. classification)
    - Choose a labelled, structured dataset, avoid trying to create your own labelled set
- Unsupervised problem (e.g. clustering)
    - Think carefully about the problem, how you will evaluate and how the final product looks
- Size considerations
    - Need AT LEAST 1k observations
    - Deep learning should have at least 5k observations
    - You’ll likely not be able to do this work in collab, at least not on the full data
    - For larger datasets (>5GB), think about sensible subsets
- Design your exploration to understand important elements, potential discriminative ability, etc

Choosing models

- Identify performance metric and expected performance (next slide)
- Start simple
    - Average probabilities
    - Word count + SVM/Regression
- Survey literature for existing implementations
- Train, validation and test performance
    - Train >> Validation, examine overfitting
    - Validation >> Test, examine the split of your data



https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Performance evaluation

- Performance
  - Classification
    - Accuracy (pred y == y)
    - Precision
    - Recall
    - F1-score
  - Ranking
    - ROC AUC
  - Clustering
    - Silhouette score
    - Inertia (K-Means)
- Stability
  - Sensitivity analysis
  - Prediction distribution

`metrics.accuracy_score`(y_true, y_pred, *[, ...])	Accuracy classification score.
`metrics.auc`(x, y)	Compute Area Under the Curve (AUC) using the trapezoidal rule
`metrics.average_precision_score`(y_true, ...)	Compute average precision (AP) from prediction scores
`metrics.balanced_accuracy_score`(y_true, ...)	Compute the balanced accuracy
`metrics.brier_score_loss`(y_true, y_prob, *)	Compute the Brier score.
`metrics.classification_report`(y_true, y_pred, *)	Build a text report showing the main classification metrics.
`metrics.cohen_kappa_score`(y1, y2, *[, ...])	Cohen's kappa: a statistic that measures inter-annotator agreement.
`metrics.confusion_matrix`(y_true, y_pred, *)	Compute confusion matrix to evaluate the accuracy of a classification.
`metrics.dcg_score`(y_true, y_score, *[, k, ...])	Compute Discounted Cumulative Gain.
`metrics.f1_score`(y_true, y_pred, *[, ...])	Compute the F1 score, also known as balanced F-score or F-measure
`metrics.fbeta_score`(y_true, y_pred, *, beta)	Compute the F-beta score
`metrics.hamming_loss`(y_true, y_pred, *[, ...])	Compute the average Hamming loss.
`metrics.hinge_loss`(y_true, pred_decision, *)	Average hinge loss (non-regularized)
`metrics.jaccard_score`(y_true, y_pred, *[, ...])	Jaccard similarity coefficient score
`metrics.log_loss`(y_true, y_pred, *[, eps, ...])	Log loss, aka logistic loss or cross-entropy loss.
`metrics.matthews_corrcoef`(y_true, y_pred, *)	Compute the Matthews correlation coefficient (MCC)
`metrics.multilabel_confusion_matrix`(y_true, ...)	Compute a confusion matrix for each class or sample
`metrics.ndcg_score`(y_true, y_score, *[, k, ...])	Compute Normalized Discounted Cumulative Gain.
`metrics.precision_recall_curve`(y_true, ...)	Compute precision-recall pairs for different probability thresholds
`metrics.precision_recall_fscore_support`(...)	Compute precision, recall, F-measure and support for each class
`metrics.precision_score`(y_true, y_pred, *)	Compute the precision
`metrics.recall_score`(y_true, y_pred, *[, ...])	Compute the recall
`metrics.roc_auc_score`(y_true, y_score, *[, ...])	Compute Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.
`metrics.roc_curve`(y_true, y_score, *[, ...])	Compute Receiver operating characteristic (ROC)
`metrics.zero_one_loss`(y_true, y_pred, *[, ...])	Zero-one classification loss.

https://scikit-learn.org/stable/modules/classes.html#sklearn-metrics-metrics

Model sources

Huggingface transformers

https://huggingface.co/models

HUGGING FACE

Back to home

All Models and checkpoints

Also check out our list of **Community contributors** 🏆 and **Organizations** 🌍.

Search models... | Tags: All | Sort: Default

- DeepPavlov/bert-base-bg-cs-pl-ru-cased
- DeepPavlov/bert-base-cased-conversational
- DeepPavlov/bert-base-multilingual-cased-sentence
- DeepPavlov/rubert-base-cased-conversational
- DeepPavlov/rubert-base-cased-sentence
- DeepPavlov/rubert-base-cased
- Hate-speech-CNERG/dehatebert-mono-arabic
- Hate-speech-CNERG/dehatebert-mono-english

PyTorch Hub

https://pytorch.org/hub/

FOR RESEARCHERS —

EXPLORE AND EXTEND MODELS FROM THE LATEST CUTTING EDGE RESEARCH

All | Audio | Generative | Nlp | Scriptable | Vision

SpaCy-transformers

https://explosion.ai/blog/spacy-transformers

Package name	Pretrained model	Language	Author	Size	Release
en_trf_bertbaseuncased_lg	bert-base-uncased	English	Google Research	387MB	
de_trf_bertbasecased_lg	bert-base-german-cased	German	deepset	386MB	
en_trf_xlnetbasecased_lg	xlnet-base-cased	English	CMU/Google Brain	413MB	
en_trf_robertabase_lg	roberta-base	English	Facebook	278MB	
en_trf_distilbertbaseuncased_lg	distilbert-base-uncased	English	Hugging Face	233MB	

Choosing model(s) for the final project

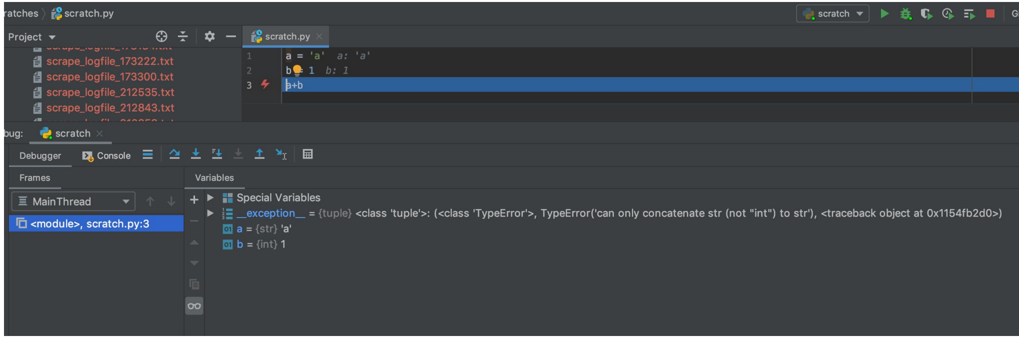
- Refer to the slide on methodology (i.e. assessing feasibility/impact)
- Understand the domain on which the model was trained
- Avoid building a lot of custom architecture
- Training a Transformer model is probably not appropriate
    - Consider fine-tuning: Retraining layers or training an additional layer on top

Debugging

- General code debugging
    - Start with minimal, working example
    - Incremental changes to code
        - E.g. wrapping in functions/loops
    - Proper debugger (e.g. PyCharm)
    - Try/Except catching (not recommended)
- Performance debugging
    - Start with minimal, working example!
    - Checks/assertions for format
    - Clustering of errors/misclassifications, examine

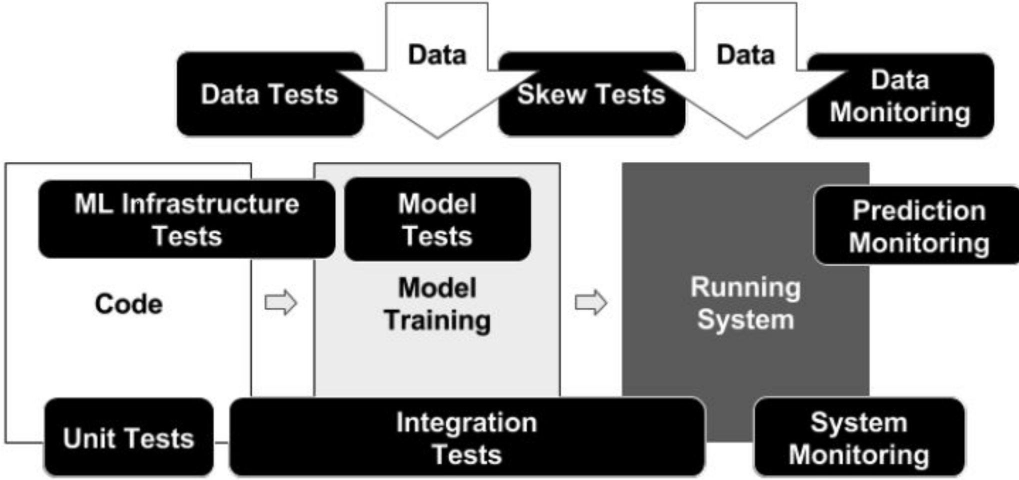
Pycharm debugger

(Pro is free for students: https://www.jetbrains.com/community/education/#students))



System testing

- Purpose of testing
  - Ensure system stability
  - Facilitate debugging
  - Early warning for problems
- Unit testing
  - Tests of individual components
- Integration tests
  - Tests of multiple systems (e.g. code feeds the expected values to training)
- General data/prediction monitoring
  - Ensure certain features are available/stable importance
  - Ensure prediction distributions don't wildly vary



ML-Based System Testing and Monitoring

https://ai.google/research/pubs/pub46555

Deployment

- Some options for serving models
  - FastAPI
    - Backend only: Given input arguments, provides outputs
  - Flask
    - Full web application, backend + frontend
- Considerations
  - Uptime
  - Latency
  - Freshness
  - Security

```python
from fastapi import FastAPI
from pydantic import BaseModel

app = FastAPI()


class Item(BaseModel):
    name: str
    price: float
    is_offer: bool = None


@app.get("/")
def read_root():
    return {"Hello": "World"}
```

```python
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hey, we have Flask in a Docker container!'


if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0')
```

Deployment design for the final project

- At a minimum: Assess what your final product looks like
- Example deployments
  - (simple) An API for interacting with the model (e.g. provide text, get prediction)
  - (complex) An interactive application
  - (complex) A robust, re-trainable output
- Good options
  - Complex methodology + well thought-out hypothetical deployment
  - Complex methodology + simple deployment
  - Simple methodology + complex deployment

Deployment with FastAPI (simple)