Where do we get supervision?

L101: Machine Learning for Language Processing

Andreas Vlachos



Let’s start with some questions

What is supervised learning?

What is unsupervised learning?

Is language model pre-training supervised or unsupervised?

- Unsupervised: the training data is unlabeled
- Supervised: the training objective assumes we have correct answers, i.e. the next token (same objective as logistic regression classification)

Is there unsupervised learning without having a notion of correct answers?

Clustering

Many good algorithms:

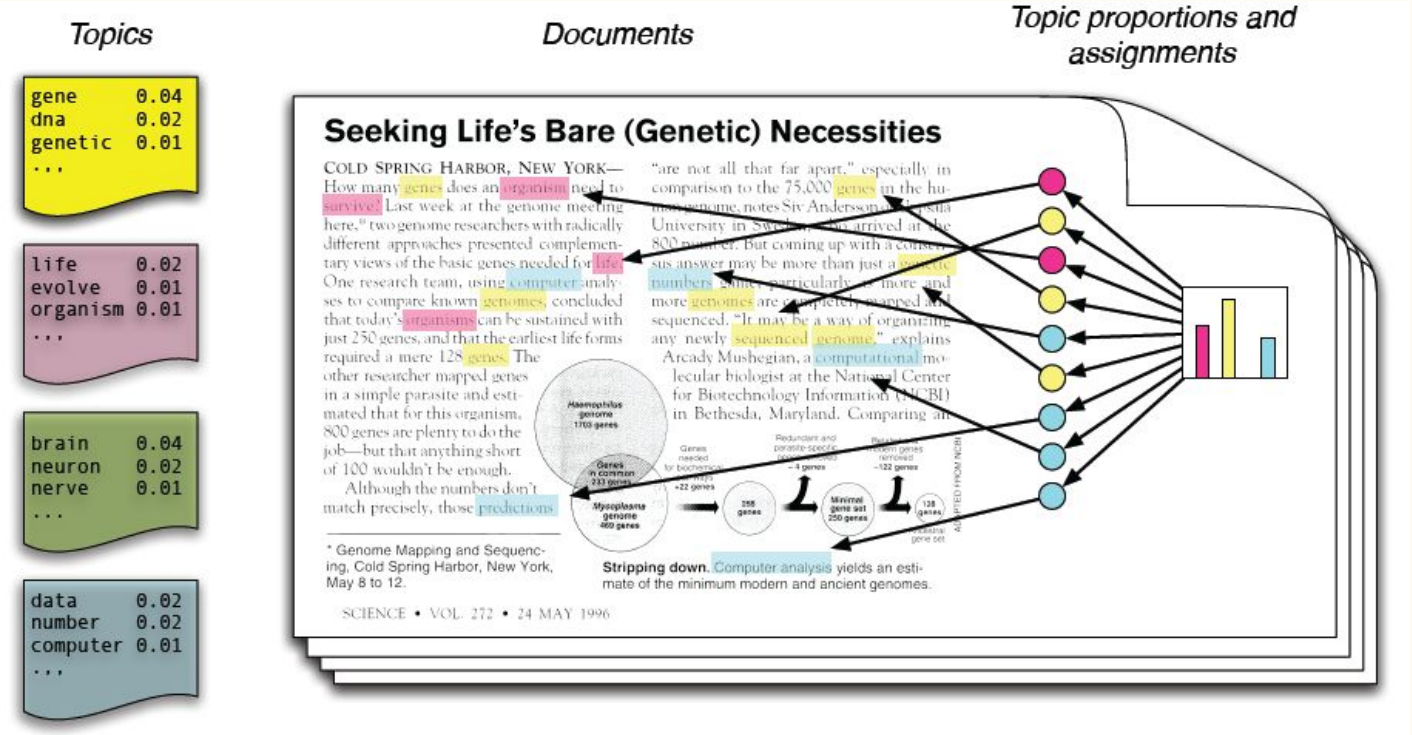
- K-means (++)
- Gaussian mixture models
- Spectral clustering
- Topic models can be thought of as soft clustering

They are great to explore unlabeled data and learn about theirs properties, but

Hard to evaluate...

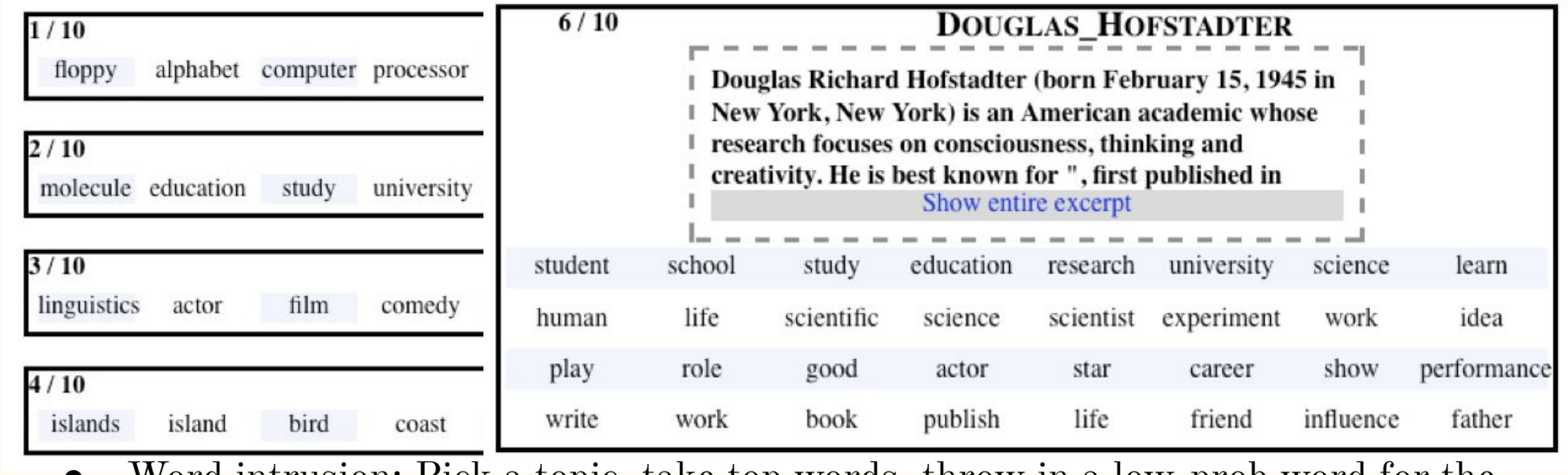
Clustering: Science or Art: “the major obstacle is the difficulty in evaluating a clustering algorithm without taking into account the context”

Topic models



Blei (2011)

Topic modeling evaluation (reading tea leaves)



- Word intrusion: Pick a topic, take top words, throw in a low-prob word for the topic, ask humans to spot the intruder
- Topic intrusion: Pick a document, take its top topics, throw in a low-prob topic for the document, ask humans to spot the intruder
- Avoid automatic evaluation if you can

But isn’t unsupervised learning common?

Yes! But often this is not unsupervised in the ML sense (no labeled data, e.g. in clustering no cluster info, in topic models no topic info):

- Some supervision gets in through dictionaries, mapping to labels, etc.
- Some dev set was most likely used (hopefully not the test set!)
- Nothing wrong with this; in fact it is a great way to solve tasks
- But if we want labels as output we need to provide them to the model

Embeddings

- Once a by-product of learning RNN-based language models, now a goal in itself
- That's the key insight of the word2vec paper: stop worrying about trying to build a language model, focus on the embeddings
- Supervised or unsupervised?
- What is a good embedding for a word?
- Embeddings for isolated words are fine-grained cluster ids; hard to evaluate
- BERT extended the idea to sentence embeddings
    - Became the standard in text classification and easy to evaluate as the feature extraction step, not in isolation!
    - Larger models pushed it document-level

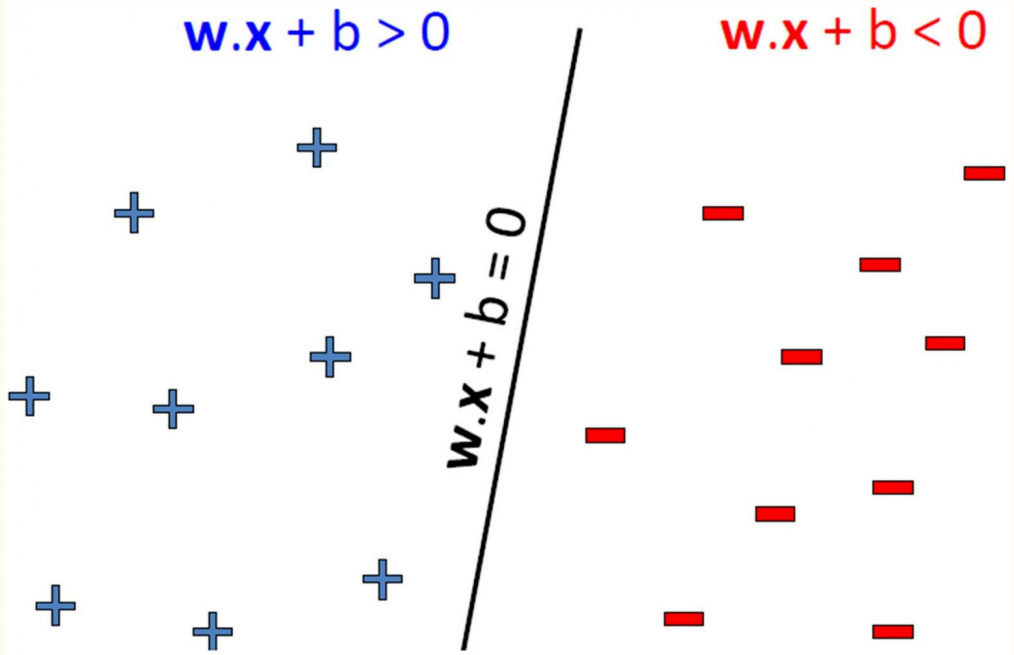
Return to supervised learning

Why?

- Training models using real outputs for the task you want to solve has better chances of success
- Even if we didn't need labeled data for training, we still want to know how good our model is before deploying it (and not tune hyperparams on the test data!)

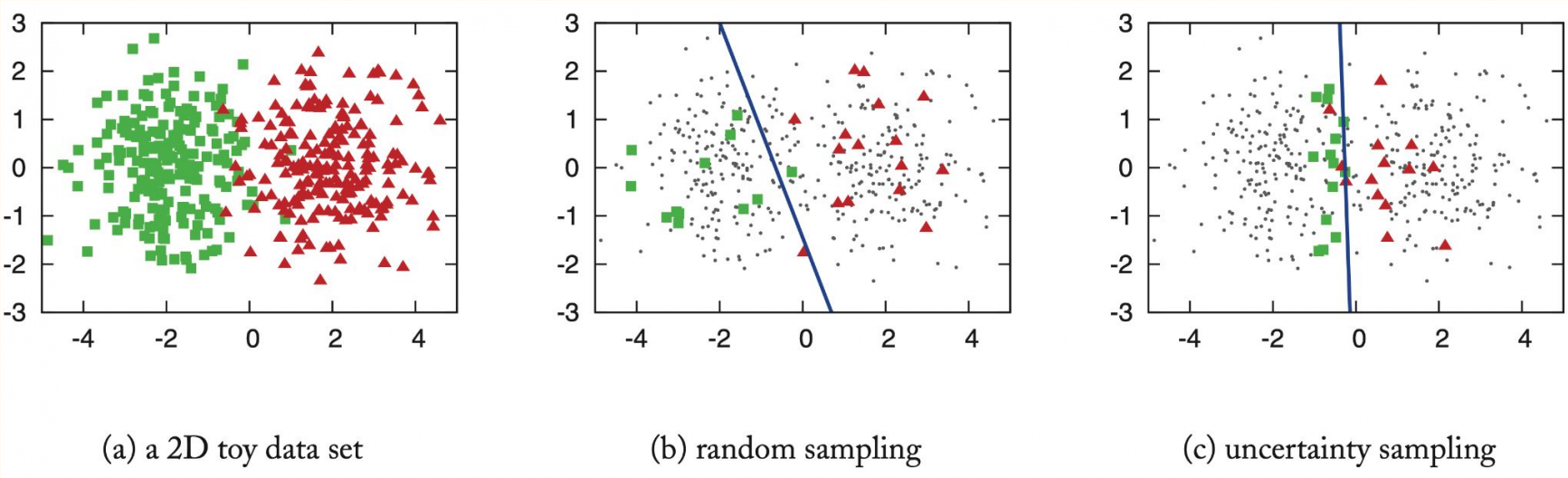
But it needs labeled data for training, how can we help ourselves given that we can only have unlabeled data for free (usually!)

Do we need all that labeled data for training?



- Are all instances equally useful in learning a model?
- Let the algorithm decide!
  - Typically select a few instances to label, update model, repeat
- Active learning in education refers to the student asking questions

Active learning works



(a) a 2D toy data set (b) random sampling (c) uncertainty sampling

- Savings against randomly selected training data can be impressive
  - Even if you are fine-tuning BERT
- In fact less can be better (sometimes)
- But this is not always the case (more later)

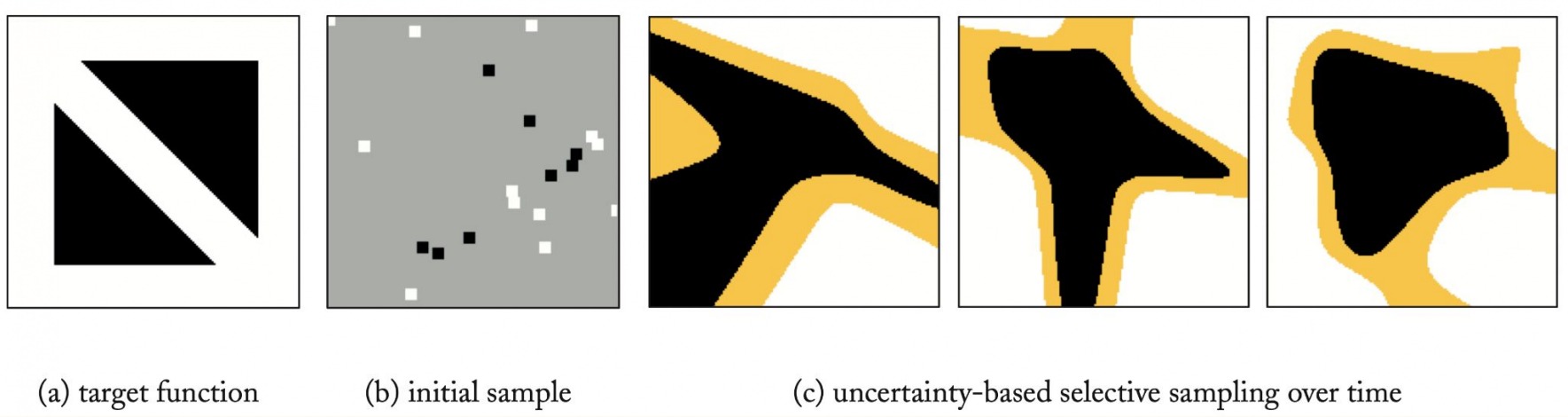
Active learning setups

- Pool-based
  - we have a fixed pool of unlabeled data and iterate through it
  - most common: we get some unlabelled instances and need to build a model
- Streaming/online:
  - decide whether to ask for a label now or never
  - ask a customer if the transaction is fraudulent, if the e-mail is spam, etc.
- Constructing instances:
  - generate instances for labeling
  - used to be rare in NLP; quite common now
- Feature-based active learning
  - label features instead of instances

How to choose informative instances?

- Uncertainty based sampling
  - Least confident: pick the instances with the lowest score by the model for any label
  - Entropy in the label distribution (for probabilistic models)
- Query by committee: train a few models and select the instances where they disagree the most
- Meta-learning: learn a model to select the most useful instances (pool, stream)
- Discriminate in favour of instances that don't look like the labeled data
- Bayesian Active Learning by Disagreement: combine uncertainty with informativity on model parameters
  - Selecting instances in batches is a common practical consideration

Things can go wrong



(a) target function (b) initial sample (c) uncertainty-based selective sampling over time

What would you do to avoid this?

- Make sure your initial sample is representative?
- Select instances at random too?

When not to active learn

- If we need data for evaluation
  - it should be obvious that AL biases the data, not guaranteed to be representative of the task
  - need to approach the data selection differently (active testing)
- If we don't think the model(s) we have can give good estimates of uncertainty
- If we are planning to change models later
  - Data selected by one model can be worse than random for another

Bandit learning

Some times obtaining a complete label for an instance is impossible: e.g. for a search query we need to check all webpages

Repeat:

- pick the most promising handle (= label)
- get a reward for that handle
- Update using the reward



The key here is how to define promising: a balance of exploration/exploitation
Can be adaptive: first explore, then exploit; sounds familiar?
See Kreutzer et al. (2017) for an application to NMT

Sometimes we get lucky

In one of their early technical reports (2019) OpenAI attributed part of their success to task demonstrations found in the web.

Of course there is more to it (e.g. model scale), but (our!) data matters

”I’m not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I’m not a fool].**

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: ”**Mentez mentez, il en restera toujours quelque chose**,” which translates as, ”**Lie lie and something will always remain.**”

“I hate the word ‘**perfume**,”’ Burr says. ‘It’s somewhat better in French: ‘**parfum**.’

If listened carefully at 29:55, a conversation can be heard between two guys in French: “**-Comment on fait pour aller de l’autre coté? -Quel autre coté?**”, which means “**- How do you get to the other side? - What side?**”.

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

“**Brevet Sans Garantie Du Gouvernement**”, translated to English: “**Patented without government warranty**”.

Pre-training data selection



NVIDIA recently put together a good paper on this

AI2 proposed that topic and format are good properties to sample for and can be inferred automatically

Our datasets are biased: not everyone's English is considered high quality

Language models as models for task X?

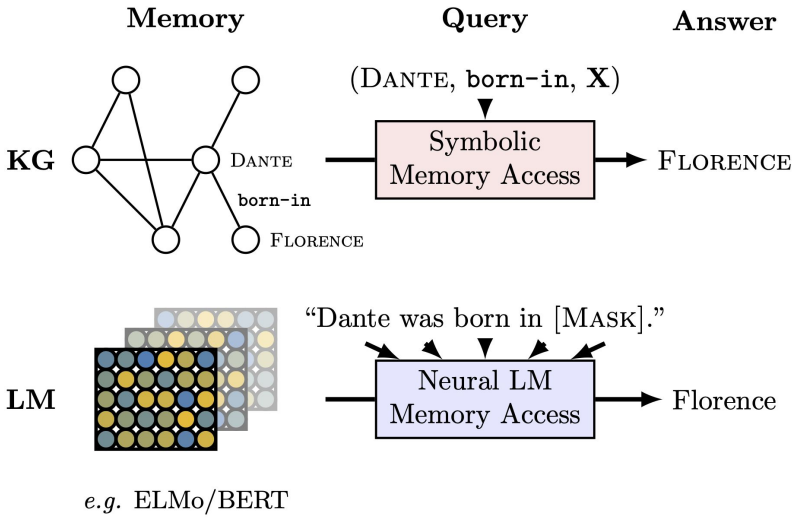
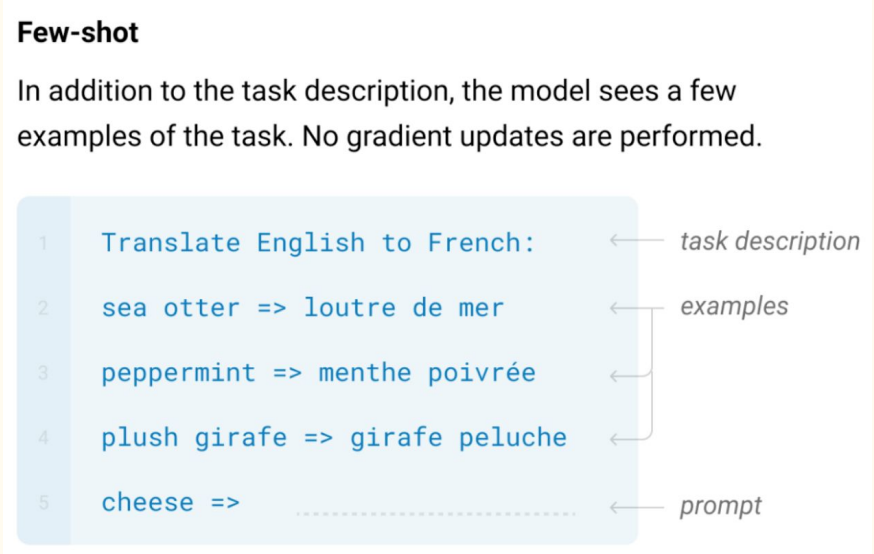


Figure 1: Querying knowledge bases (KB) and language models (LM) for factual knowledge.

- Yes! But sometimes we need to add a bit task-specific supervision (fine-tuning)
- If our task can be modelled as an LM, we can take advantage of a lot of data and pre-trained models
- If we are building task-specific models, we'd better improve on it!
- If we developing a task/dataset, make sure a LM can't solve it (easily), e.g. don't use an LM (alone) to construct it

Petroni et al. (2019)

Prompting large language models



Brown et al. (2020)

- Look ma, no training, just a few examples!
- From feature engineering, to architecture, and now to prompt engineering?
- Not so fast:
  - Needs a lot of dev data for model/ prompt selection (Perez et al., 2021)
  - Fine-tuning is more reliable and not necessarily more expensive (Logan IV et al., 2021)
- And it is brittle!

Bibliography

Active learning book (Burr Settles, where a lot of images where taken from)

Bandit learning book (Slivkins)

Foundations of LLMs (Xiao and Zhu, 2025)