# Fortuitous data

## ESSLLI 2016, Day 1

Željko Agić, Anders Johannsen, Barbara Plank

# Getting to know each other

# Željko Agić (read as: Zhelyko Aggich ☺)

IT University of Copenhagen

http://zeljkoagic.github.io/

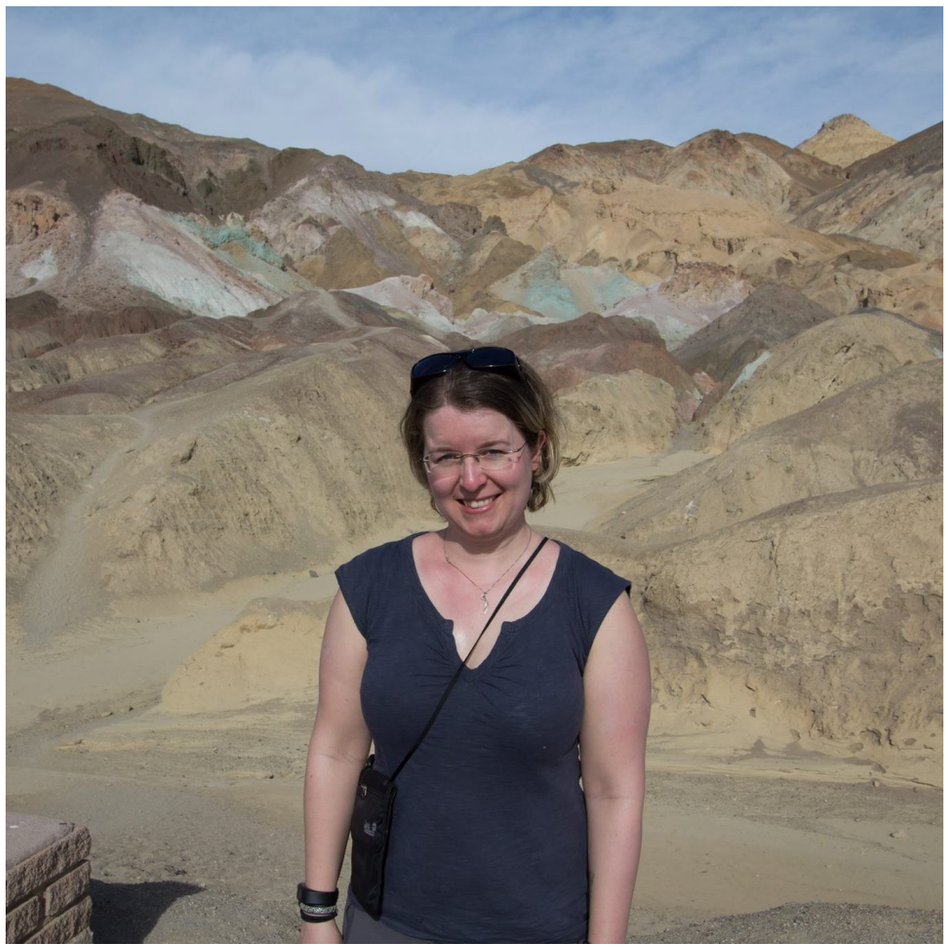
zeljko.agic@gmail.com

# Anders Johannsen

Apple

anders@johannsen.com

# Barbara Plank



University of Groningen

http://www.let.rug.nl/~bplank

bplank@gmail.com

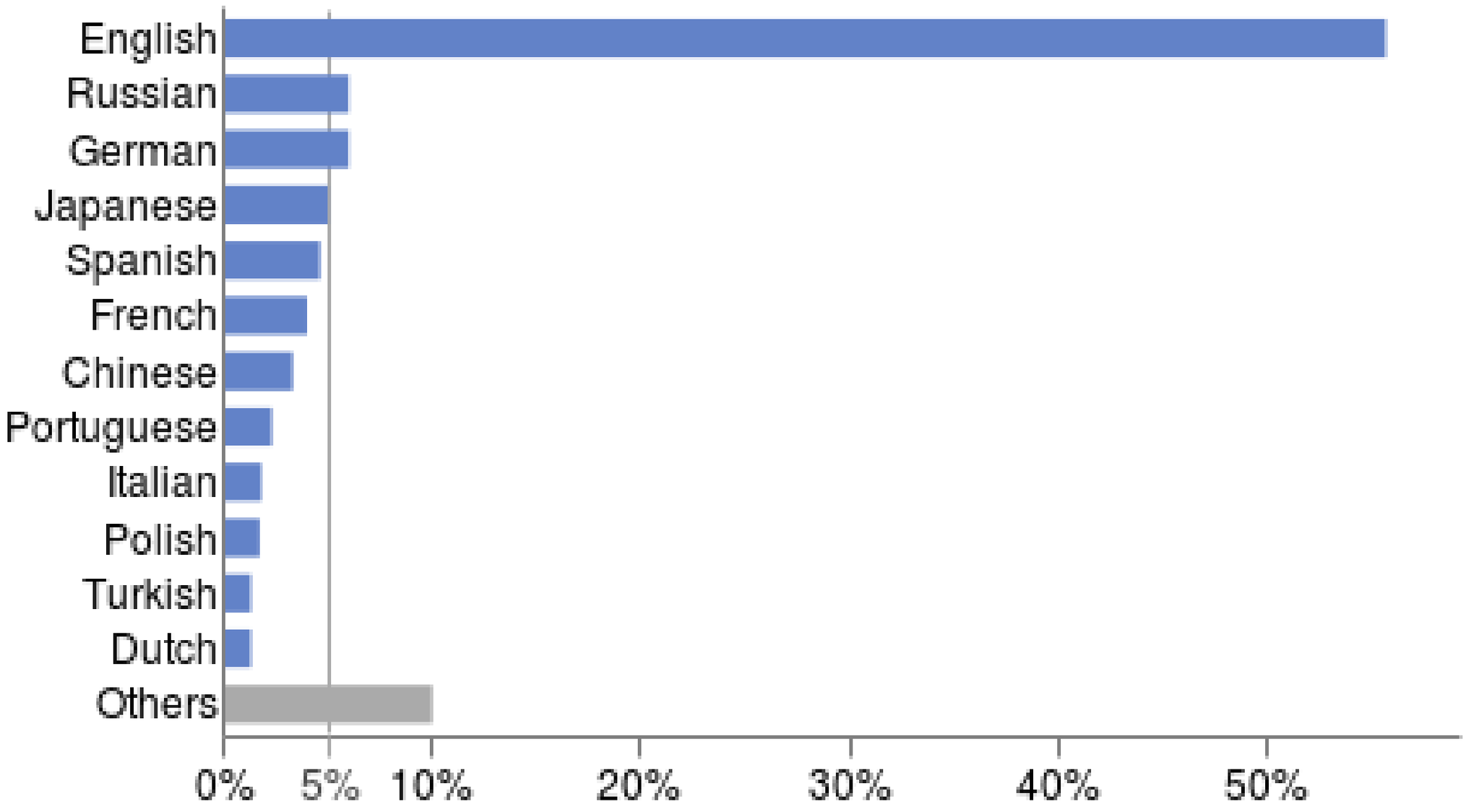
# Who are you?

http://bit.ly/2aVJVbs

# The course

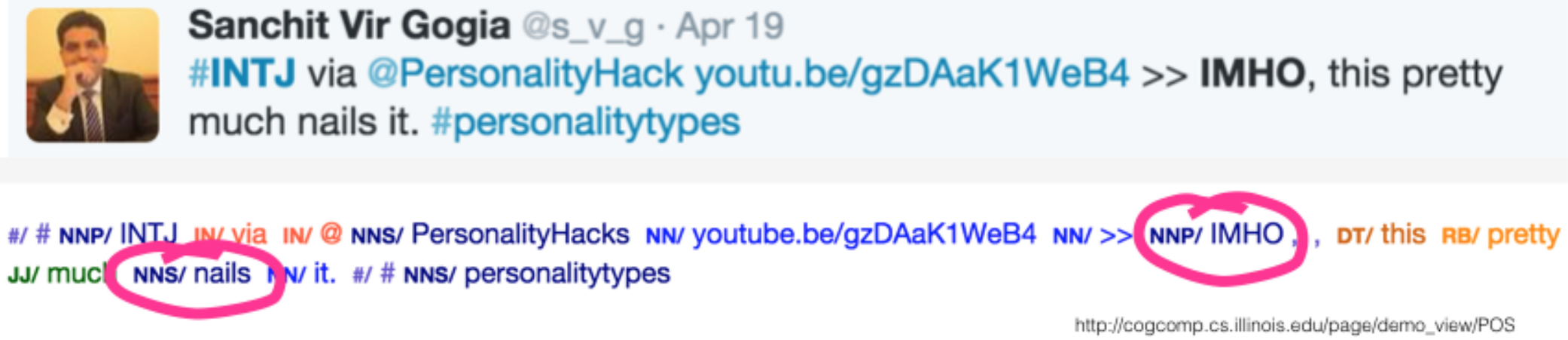
# Motivation

# Ultimate goal: NLP for everyone




Languages used on the Internet

# The problem



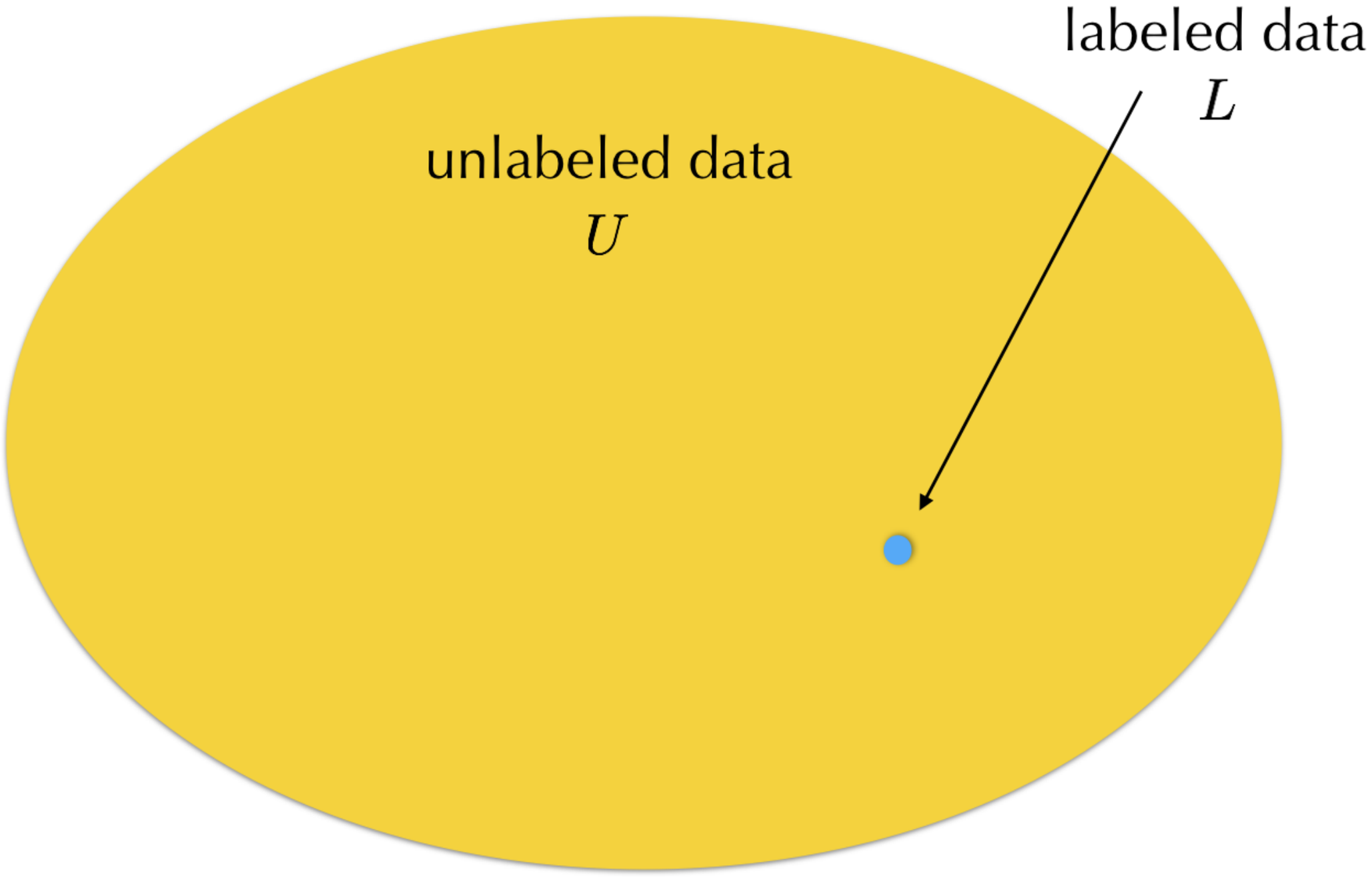

Tagging Twitter #hard

# The reason

- NLP models are trained on samples from a **limited set of canonical data**
- Mainly: English newswire


**CROSS-DOMAIN GULF**

# Labeled data is **scarce**




Training data sparsity

# Labeled data is **biased**

For a long while main resource: Wall Street Journal (WSJ), texts from late 80s.


Newswire bias

``*it is an uncomfortable fact that the text in* ***many of our most frequently-used corpora*** *was written and edited predominantly* ***by working-age white men***" (Eisenstein 2013)

# Still newswire?

| | *news* | *fiction* | *nonfiction* | *blog* | *medical* | *legal* | *social* | *spoken* | *wiki* | *web* | *reviews* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Basque** | ✓ | ✓ | | | | | | | | | |
| **Bulgarian** | ✓ | ✓ | | | | ✓ | | | | | |
| **Croatian** | ✓ | | | | | | | | ✓ | | |
| **Czech** | ✓ | | | | | | | | | | |
| **Danish** | ✓ | ✓ | ✓ | | | | | ✓ | | | |
| **Dutch** | ✓ | | ✓ | | ✓ | | | ✓ | ✓ | | |
| **English** | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ |
| **Finnish** | ✓ | ✓ | | ✓ | | ✓ | | | ✓ | | |
| **French** | ✓ | | | ✓ | ✓ | | | | ✓ | | |
| **German** | ✓ | | | | | | | | | | ✓ |
| **Greek** | ✓ | | | | | | | ✓ | ✓ | | |
| **Hungarian** | ✓ | | | | | | | | | | |
| **Irish** | ✓ | ✓ | | | | ✓ | | | | ✓ | |
| **Italian** | ✓ | | | | | ✓ | | | ✓ | | |
| **Spanish** | ✓ | | | ✓ | | | | | | | |
| **Swedish** | ✓ | | ✓ | | | | | | | | |

**Training data sparsity:** subset of treebanks from Universal Dependencies v1.2 (Nivre et al. 2016) for which domain/genre info is available (Plank 2016)

# What if language technology could start over?

- English newswire has advanced our field, but also introduced impercetible biases
- Why is newswire more **canonical** than other text types?
- If NLP could start over, what would be our canon?

# Data mismatch - dichotomy:


CROSS-DOMAIN GULF

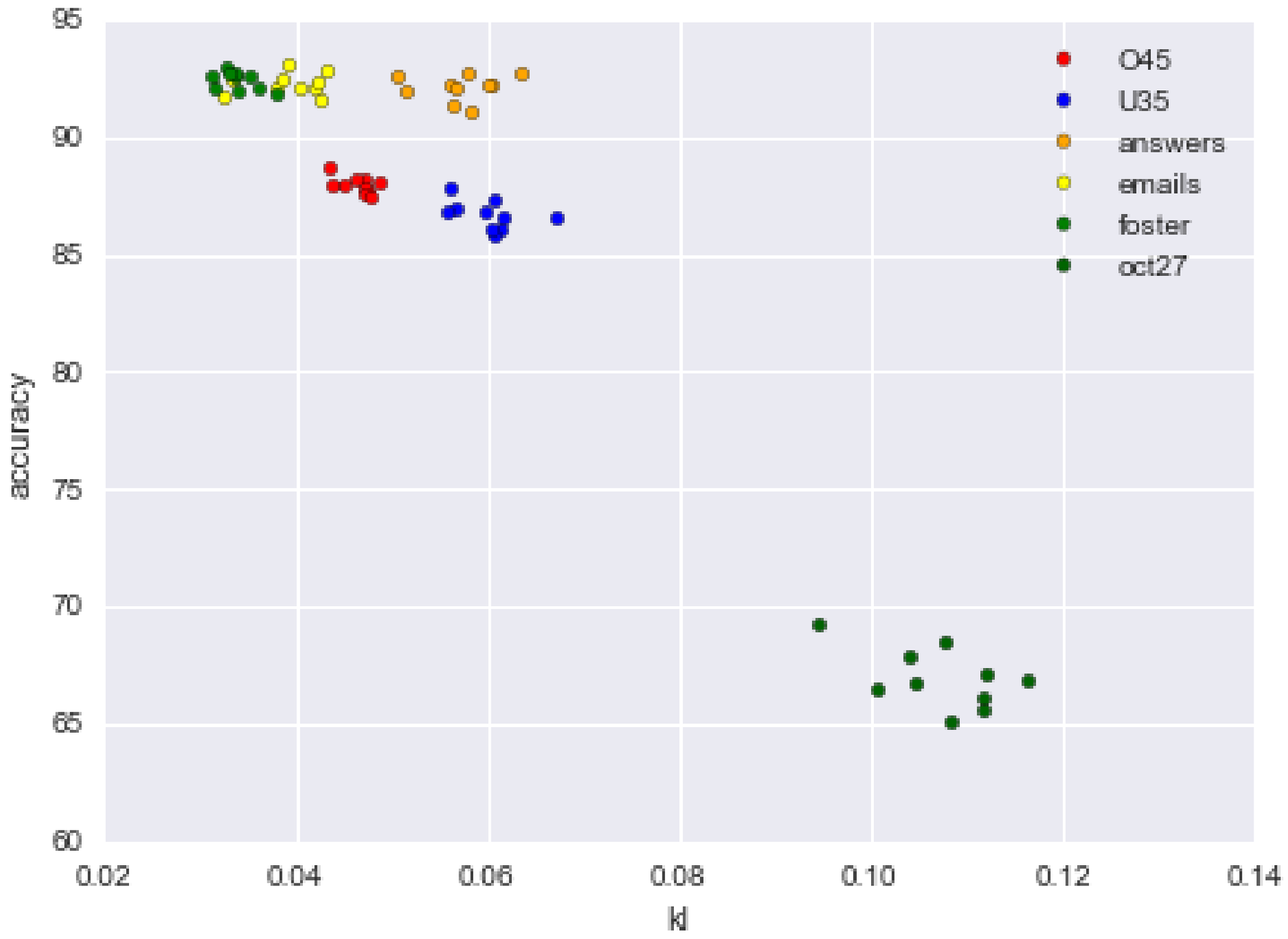
Train/application time

- Train <> Test
- Source <> Target

Really a dichotomy?

# What's in a **domain**?


POS tagging accuracies versus OOV rate




POS tagging accuracies versus POS bigram KL divergence

# The variety space

- Where does our data come from?
- *Domain* is an overloaded term.
- In NLP typically used to refer to some coherent set of data from some topic or genre.
- There are many other possible factors out there.

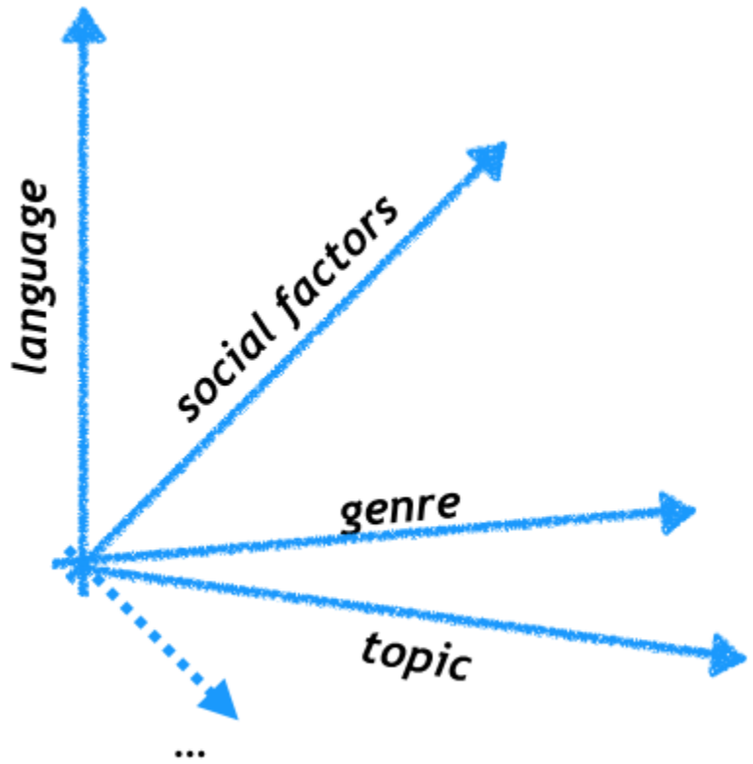
Our datasets $\mathcal{D}$ are sampled from a **variety space**

$$\mathcal{D} \sim P(X, Y|V)$$

Is there such a variety space? What would the factors be?

# The variety space - illustration

- unknown high-dimensional space
- A domain (variety) forms a region in this space, with some members more prototypical than others (prototype theory, Wittgenstein, **graded** notion of category)




The variety space

# General statement of the problem

Whatever we consider **canonical**, the challenge remains: processing non-canonical data is hard.

What are possible solutions?

# Silly problem with simple solution?

# Approach 1: Annotate more?

- Take cross-product between *domain* and *language* - huge space!
- Our ways of communication change, so does our data; social media is a moving target (Eisenstein 2013)

| | *news* | *fiction* | *nonfiction* | *blog* | *medical* | *legal* | *social* | *spoken* | *wiki* | *web* | *reviews* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Basque** | ✓ | ✓ | | | | | | | | | |
| **Bulgarian** | ✓ | ✓ | | | | ✓ | | | | | |
| **Croatian** | ✓ | | | | | | | | ✓ | | |
| **Czech** | ✓ | | | | | | | | | | |
| **Danish** | ✓ | ✓ | ✓ | | | | | ✓ | | | |
| **Dutch** | ✓ | | ✓ | | ✓ | | | ✓ | ✓ | | |
| **English** | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ |
| **Finnish** | ✓ | ✓ | | ✓ | | ✓ | | | ✓ | | |
| **French** | ✓ | | | ✓ | ✓ | | | | ✓ | | |
| **German** | ✓ | | | | | | | | | | ✓ |
| **Greek** | ✓ | | | | | | | ✓ | ✓ | | |
| **Hungarian** | ✓ | | | | | | | | | | |
| **Irish** | ✓ | ✓ | | | | ✓ | | | | ✓ | |
| **Italian** | ✓ | | | | | ✓ | | | ✓ | | |
| **Spanish** | ✓ | | | ✓ | | | | | | | |
| **Swedish** | ✓ | | ✓ | | | | | | | | |

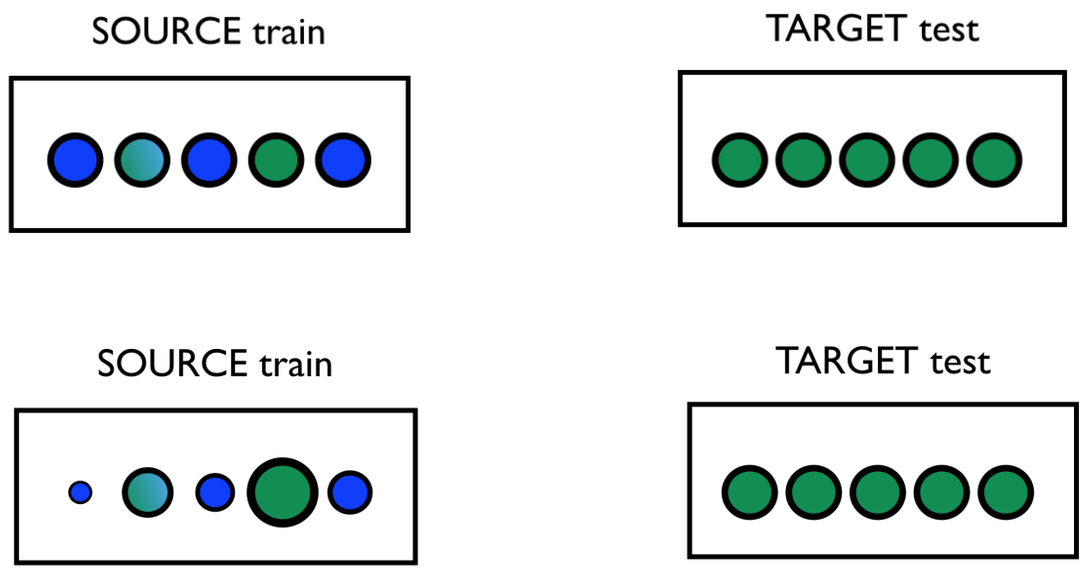
**Training data sparsity**

# Approach 2: Map to canonical form?

- Example: spelling normalisation (e.g. Han, Cook, and Baldwin 2013) ``u must be talkin bout the paper''
- However, what **norm**?

# Approach 3: Domain adaptation

Example: Importance weighting.




- Not final answer.
- Many approaches (Daume III 2007; Weiss, Khoshgoftaar, and Wang 2016), but unrealistic assumptions
- Often, in reality, we don't know the target domain.

# Fortuitous data (this course)

# Define fortuitous!

## fortuitous

/fɔːˈtjuːɪtəs/

*adjective*

happening by chance rather than intention.
"the similarity between the paintings may not be simply fortuitous"
*synonyms:* chance, unexpected, unanticipated, unpredictable, unforeseen, unlooked-for, serendipitous, casual, incidental, coincidental, haphazard, random, accidental, inadvertent, unintentional, unintended, unplanned, unpremeditated
"his success depended on entirely fortuitous events"

- happening by a lucky chance; fortunate.
  "the ball went into the goal by a fortuitous ricochet"
  *synonyms:* lucky, fortunate, providential, advantageous, timely, opportune, serendipitous, expedient, heaven-sent, auspicious, propitious, felicitous, convenient, apt; More

Fortuitous

# Fortuitous data

Data that is out there, waits to be harvested (**availability**), and can be used (relatively) easily (**readiness**)

# Fortuitous data to the rescue

- Annotate more: reuse **data that was not explicitly annotated**.
- Normalization: With sufficient data learn **invariant representations**.
- Domain adaptation: **Gather data of new varieties quickly**, or use additional signal to build more robust models.

# Typology of fortuitous data

- **Side benefit of user-generated content** (e.g., hyperlinks, HTML markup, large unlabeled data pools), availability: +, readiness: +
- **Side benefit of annotation** (e.g., annotator disagreement), availability: -, readiness: +
- **Side benefit of behavior** (e.g., cognitive processing data), availability: +, readiness: -

# Fortuitous approaches

Combine fortuitous **data** with proper **models** to enable adaptive language technology.

# Overview of the course

# Monday

## A typology of data mismatch

## Learning in the shire

# Tuesday

## Structured prediction

# Wednesday

your very own fortuitous learner (hands on).

# Thursday

## Learning from related tasks

# Friday

## Transfer learning in the extreme

# Transfer intution

# Learning to ride

How can we hope to use data from other tasks where both the input and output spaces are different?

Let's say you want to learn how to ride a motorcycle, and you already know how to drive a car.

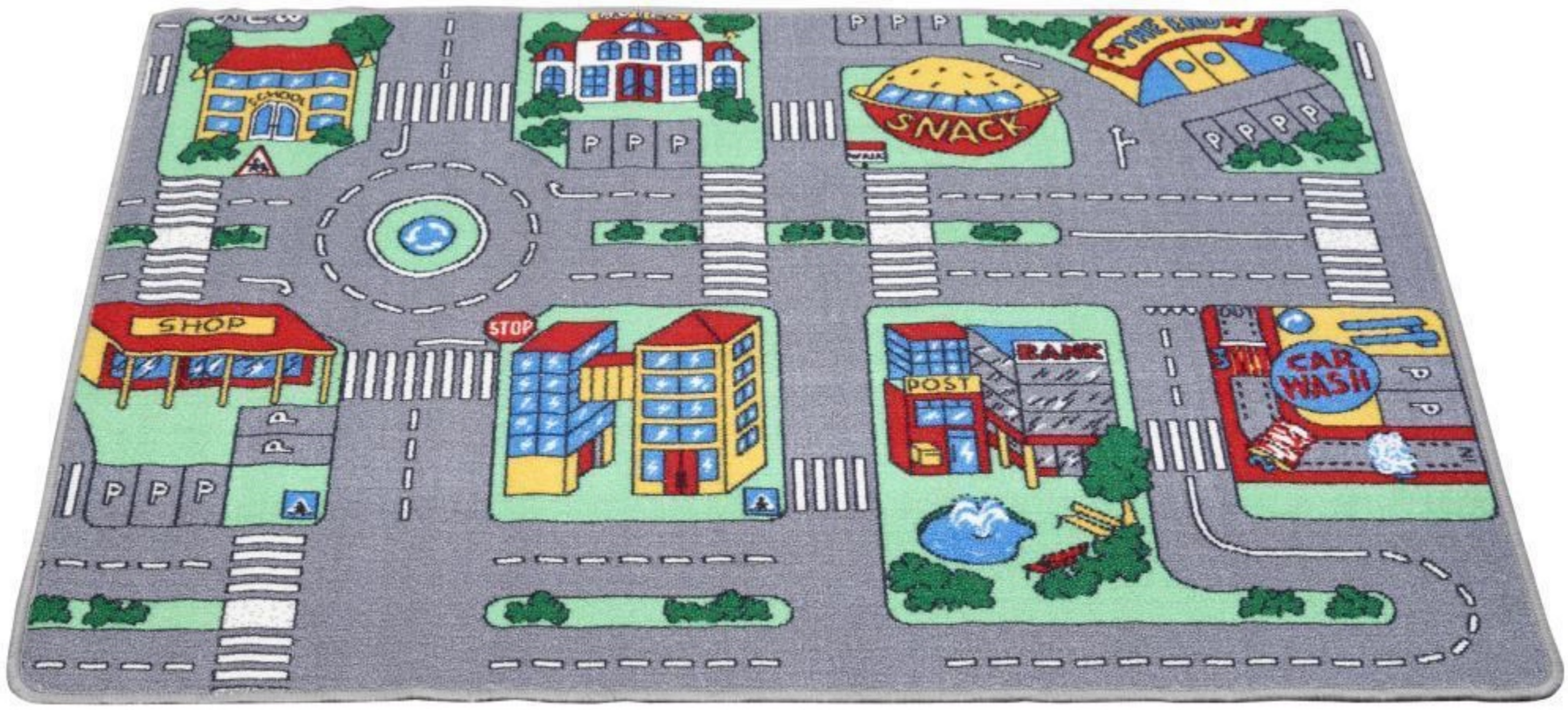
First observation: motorcycle driver's licenses are *cheaper* if you already have an ordinary driver's license.

The market believes in transferable skills!

Input space: what you observe on the road

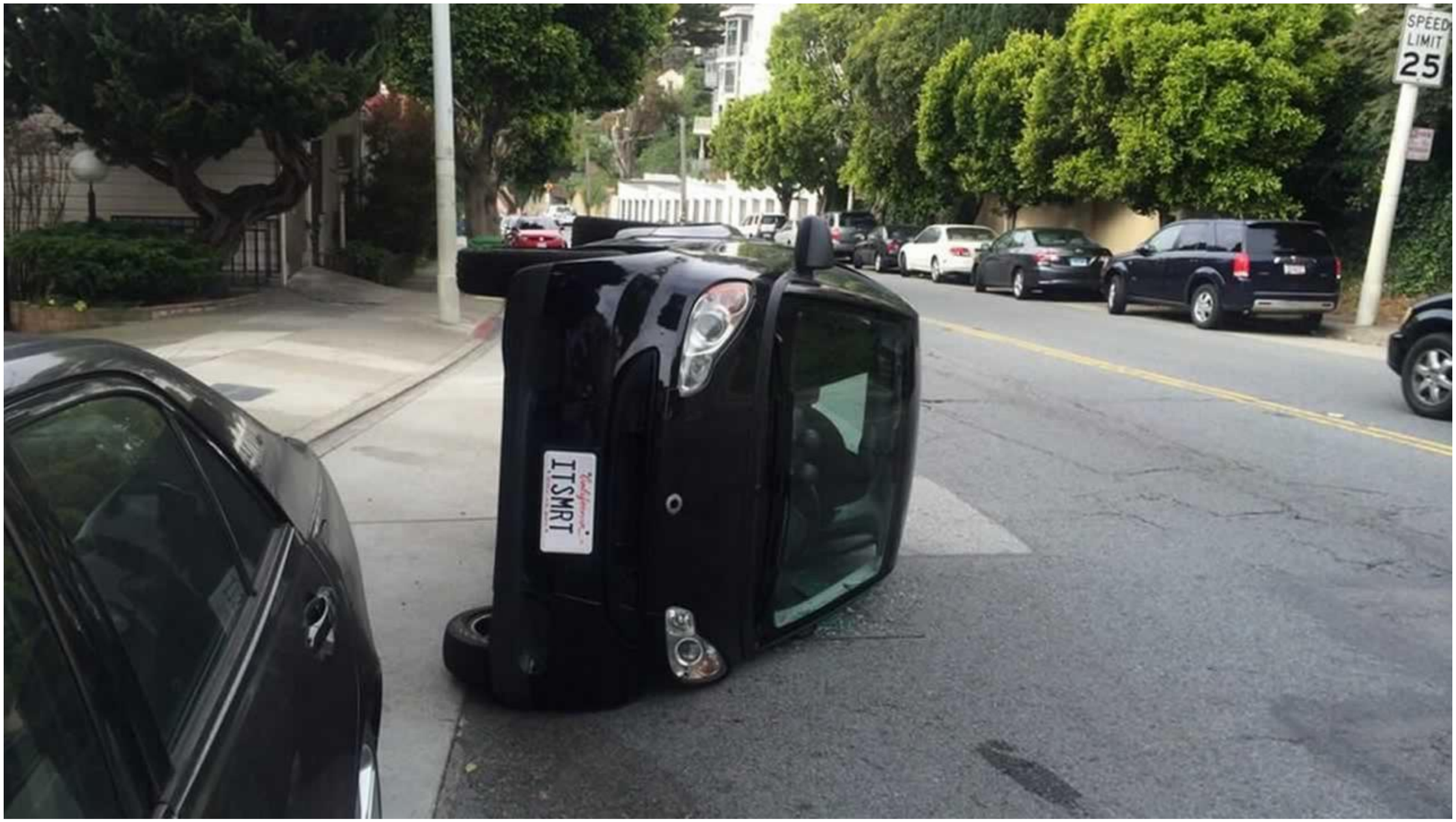
Output space: actions that you can take, like changing gears, speeding up, breaking, etc.


Some traffic skills are independent of the mode of transport.




In general, your internal model of how traffic works is transferable.

Some skills are unique to driving a motorcycle.




You typically don't have to worry about this when stopping in a car.

# Caution

For a car it’s best to stop when the light changes to yellow.

On a motorcycle suddenly applying the brakes can be fatal, because the car or truck behind you might decide to just continue.

This is a case of **negative transfer**.

# Part 1: The shire

# A static world

The shire is a quiet and wonderful place, with jolly and content people inhabiting its rolling green hills and quaint villages.

The only kind of horse that lives in the shire is the stout Hackney pony. At no point will you be asked to shoe a Belgian horse, or mend a broken bike wheel.

**Wouldn't it be great if we actually all lived in the shire?**

# The shire assumption

Machine learning theory assumes that the world behaves predictably like the shire.

This assumption is what gets us nice things like theoretical generalisation bounds.

$\Longrightarrow$ **there is no generalisation theory outside the shire.**

# Input and output

The goal of supervised machine learning is to find a function $h$ that maps from some percept or input $x$ to a label $y$.

- $x$ is a credit application, $y$ the outcome.
- $x$ is a tweet, $y$ its sentiment.
- $x$ is a sentence, $y$ the syntactic parse tree. (Tomorrow).

Let $x \in \mathcal{X}$ (input space) and $y \in \mathcal{Y}$ (label space).

NLP applications almost always have **discrete** output spaces.

In these lectures, $y$ will either be an integer (for classification) or a vector of integers (for structured prediction).

# Target and hypothesis function

There is an **unknown target function** solving our problem:

$$f : \mathcal{X} \mapsto \mathcal{Y}$$

Goal: learn a **hypothesis function** $h$ that is as close as possible to the target function.

$$h : \mathcal{X} \mapsto \mathcal{Y}$$

# Dataset

It gets worse before it gets better.

We also don't know the true distribution of our inputs. $P(X)$ is unknown.

Supervised learning rests on the idea that we can get a **finite sample**

$$x_1, \ldots, x_n \sim P(X)$$

from the unknown input distribution $P(x)$, and that we can (somehow) evaluate $f$ on these examples.

Putting this together yields the concept of a **training set**:

$$\mathcal{D}_t = \{(x_1, f(x_1)), \ldots (x_n, f(x_n))\}$$

How do we gain access to the unknown target function?

# Big data sale

- Available: large sample from $P(x)$. No labels included.
- Available: large sample from $P(x)$. Weird labels included.

The setting in which there are no labels at all is called **unsupervised learning**.

When unlabeled data is available in addition to a labeled dataset this is **semi-supervised learning**.

These concepts are a little less useful in transfer learning / multi-task learning.

# Comparable representations

We wish to learn from the past, but:

- We'll never see the same tweet twice, hopefully.
- When a failed credit application is resubmitted, the customer's circumstances have changed, and so the application isn't the same anymore.

"You cannot submit a credit application twice," as Heraclitus might have said.

We wish to learn from the past, but whatever happened will not happen *exactly* like that again. Something *similar* might happen.

# Feature space

Observations decompose into **features** in some **feature space** $\mathcal{F}$.

Each input example is transformed into a suitable **input representation** for the learning algorithm by a **feature function** $\Phi(x)$.

The feature function $\Phi(\cdot)$ maps examples from the input space to the feature space:

$$\Phi : \mathcal{X} \to \mathcal{F}$$

Typically, the $\Phi(x)$ is a real-valued vector of some fixed dimension $d$:

$$\mathcal{F} = \mathbb{R}^d$$

The $\Phi$ feature function is deterministic and not a part of the learner.

Traditional NLP: find better $\Phi$ for specific tasks by hand.

Feature representations are also a theme in this course, but the flavour will be different.

# Latent space

We've seen three kinds of spaces already: input space, feature space, and label space. Now, tada, the **latent space** where the model's *internal* representations live.

## Example: word embeddings

Embeddings are not model output. They are latent state in a model that is doing something else. A side benefit of optimising another objective.

# Notation

We introduce a latent space $\mathcal{Z}$. We use two extra functions:

- $j : \mathcal{X} \mapsto \mathcal{Z}$ from feature to latent.
- $k : \mathcal{Z} \mapsto \mathcal{Y}$ from latent to label,

and define h as the composition of $j$ and $k$:

$$h = j \circ k$$

.

# Example: $h$ is linear

The shape of $h$ depends on our choice of **hypothesis class**, that is which kind of learner we will be using.

A simple example is the linear hypothesis class for binary classification:

$$\tilde{y} = h(x; \theta, b) = \text{sign}(\theta^\top \Phi(x) + b)$$

This example shows how the parameters $\theta$ and $b$ of the model are combined with the feature representation produced by $\Phi(x)$.

# Suggestion

It’s a great summer; we’re young, and it feels like the nights extend indefinitely. Let’s use some of that time to write down a parameter vector $\theta$ that classifies all the examples in our training set *perfectly*.

Is that a good choice of parameter vector?

**Or would we become bitter as we grow old, looking back on a summer of wasted opportunity?**

Not important how $h$ performs on the training data. It could have simply remembered all of the answers.

We are interested in a system that is able to **generalize**. It needs to represent the regularities of the data **compactly**.

# Evaluate on unseen data

Evaluation uses **unseen data**. Given a new fresh $x$, $h$ makes a prediction

$$\tilde{y} = h(x)$$

The system incurs a **loss** (the cost of the prediction) $l(y, \hat{y})$ which is typically $0$ if the predicted label is correct, and $> 0$ otherwise (if $y \neq \tilde{y}$).

# Summary, inside the shire:

Given training and evaluation data:

$$\mathcal{D}_t = \{(x_1, f(x_1)), \ldots, (x_n, f(x_n))\} \sim P(X)$$
$$\mathcal{D}_e = \{(x_1, f(x_1)), \ldots, (x_m, f(x_m))\} \sim P(X)$$

Learn a parameterised function $h$ to approximate $f$.

$$\tilde{\theta} = \arg\min_{\theta} \sum_{(x,y)\in\mathcal{D}_t} l(y, h(\Phi(x); \theta))$$

Estimate generalisation error using $\mathcal{D}_e$.

# Part 2: Outside the shire


Shire.... Baggins....

# This is not what we trained for

A number of things can go wrong outside the shire.

All of a sudden the horses are not ponies anymore.

Shire assumption:

- Train: $\mathcal{D}_t \sim P(X)$
- Eval: $\mathcal{D}_e \sim P(X)$

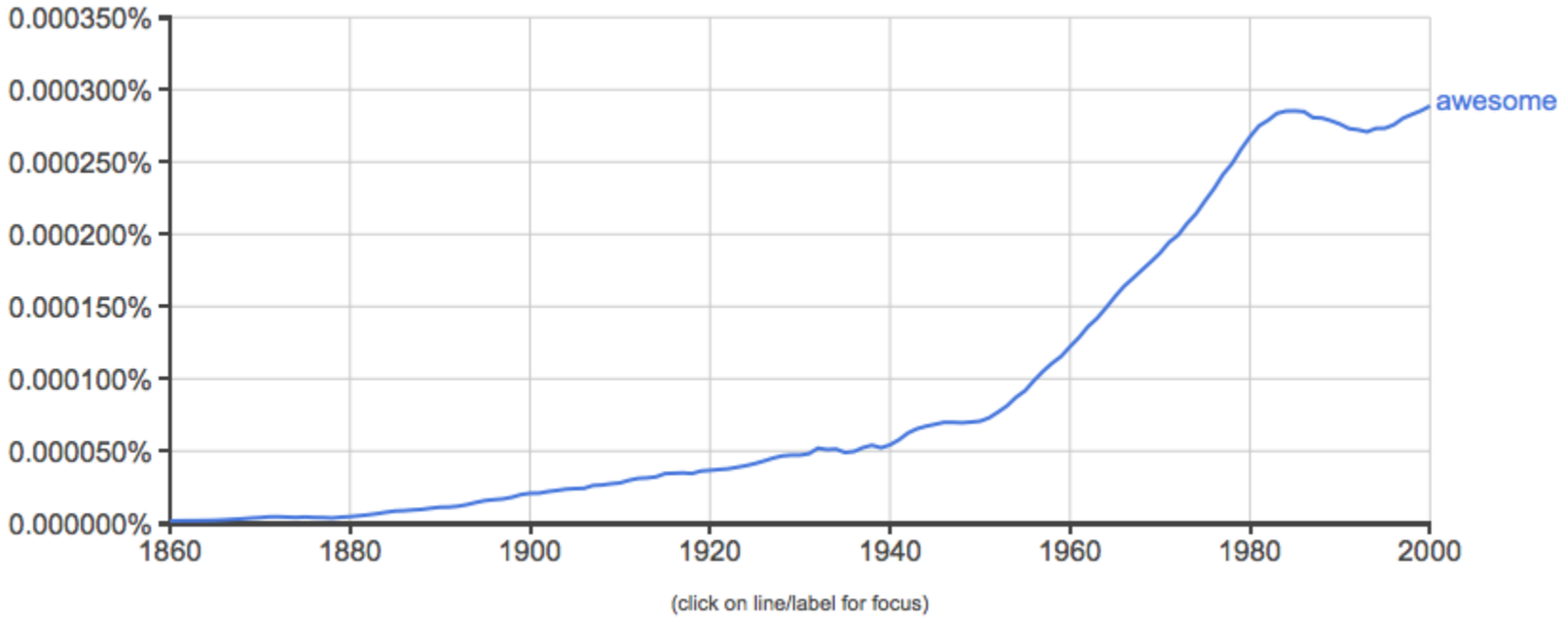
Mordor reality:

$$P_t(X) \neq P_e(X)$$

# Input distributions differ

**Condition**: $P_t(X) \neq P_e(X)$

Case: Language changes. A word like “awesome” has become much more frequent, perhaps losing some of its former oomph, but not fundamentally changing meaning.




Say you learned a sentiment model on English music reviews from 1960 and wished to apply it *now*. What would happen?

# Output distributions differ

**Condition**: $P_t(Y) \neq P_e(Y)$

Case: Corporate IT projects in banks run for a long time. Suppose you were a British bank and used your recorded credit application history from before Brexit for the loan classifier you are using today. The market is insecure, and the bank would like to approve fewer application to reduce its overall risk.

Here the label distribution has changed:

$$P_t(Y = \text{Approved}) > P_e(Y = \text{Approved})$$

This could happen without the criteria for evaluating loan risk changing.

# Conditional distributions differ

**Condition**: $P_t(Y|X) \neq P_e(Y|X)$

Case: The dust has not yet settled on Brexit. Two groups of people with particularly uncertain prospects are foreigners in Britain, and Britons in Europe. Say a British family moved to Berlin and wished to purchase a property in Prinzlauerberg. Would the fact that they are British alter their chances of getting a loan, without necessary affecting anyone else?

# General setting

- Single **target task**.
- One or more **source datasets**.

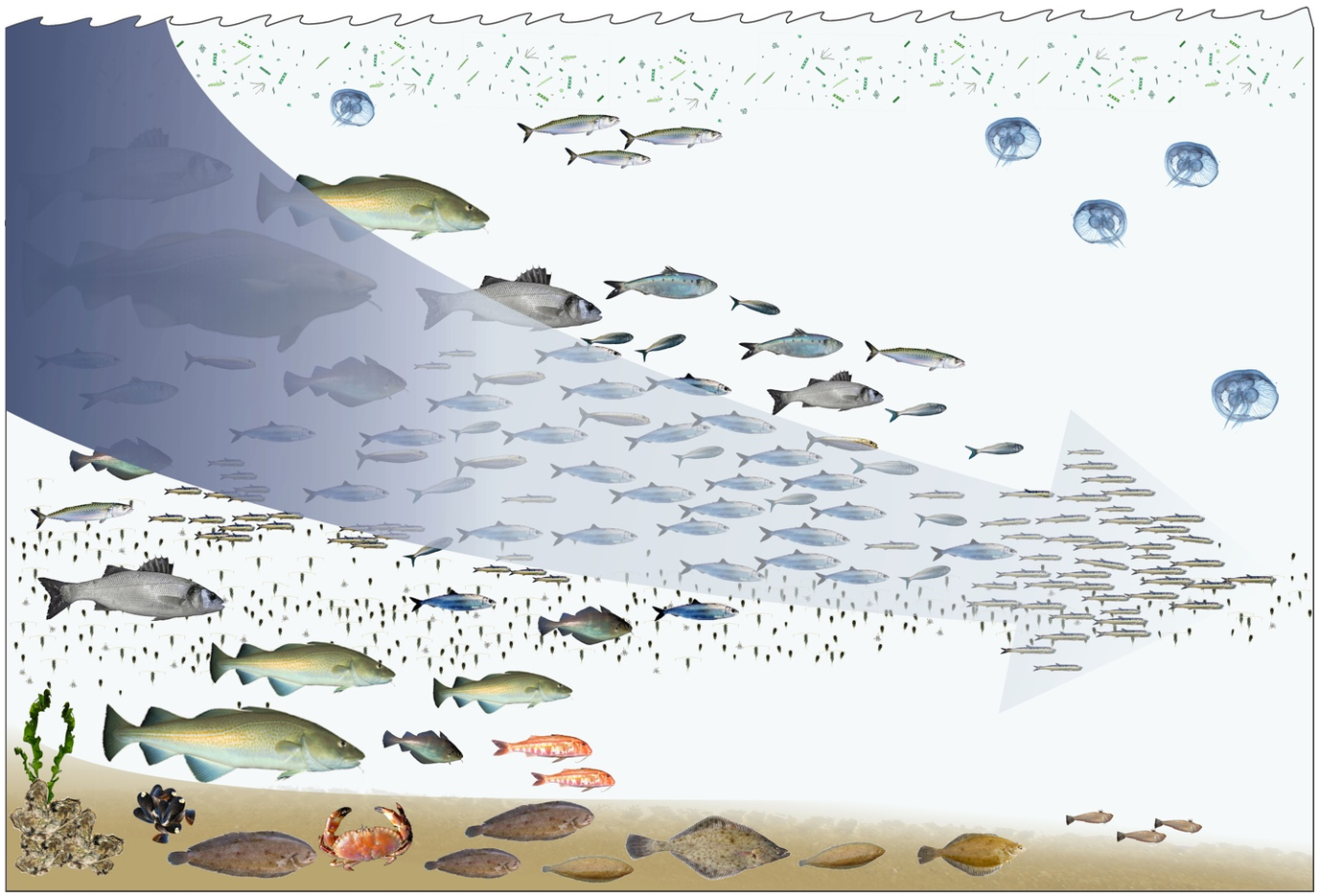
Target task: a label space $\mathcal{Y}$, an input distribution $P(x \in \mathcal{X}_t)$, and a loss $l(y, \tilde{y})$.

Source dataset: minimally a sample from $P(x \in \mathcal{X}_s)$.

But often labeled data, induced classifiers, latent representations from the classifiers, and so on.

**No requirement that $\square_t = \square_s$ or $\square_t = \square_s$.**

# Awaky



Wikipedia commons

Classic machine learning setting: **fish classification**.

What events could cause:

- $P_t(X) \neq P_e(X)$,
- $P_t(Y) \neq P_e(Y)$, and
- $P_t(Y|X) \neq P_e(Y|X)$?

# Bonus question

You're a tech-savyy fusion sushi chef and want to try out new ingredients. So you build a binary classifier to help you decide. You friend works in the factory and used this new tool `fish2vec` to extract latent fish representations (fish embeddings). Could they be helpful in your task?

Another friend trained `fish2vec` on an recipe database. Does it matter which ones you use?

# References

# References


Daume III, Hal. 2007. “Frustratingly easy domain adaptation.” In *Proceedings of Acl*.

Eisenstein, Jacob. 2013. “What to Do About Bad Language on the Internet.” In *NAACL*.

Han, Bo, Paul Cook, and Timothy Baldwin. 2013. “Lexical Normalization for Social Media Text.” *ACM Transactions on Intelligent Systems and Technology (TIST)* 4 (1). ACM: 5.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, et al. 2016. “Universal Dependencies V1: A Multilingual Treebank Collection.” *LREC*.

Plank, Barbara. 2016. “What to do about non-standard (or non-canonical) language in NLP.” In *To appear in KONVENS*.

Weiss, Karl, Taghi M Khoshgoftaar, and DingDing Wang. 2016. “A Survey of Transfer Learning.” *Journal of Big Data* 3 (1). Springer: 1–40.