Statistics for Text Analysis

NTV 2026

Today's lecturer: Wilker Aziz

w.aziz@uva.nl

Outline

In NTV, you will learn to design and estimate models that **analyse and generate language**.

As you saw last class, language and language use are fairly intricate phenomena. So, at face value, this looks like a tall order.

To get there (and we will),

- we begin with a focus on **analysing population-level**
  - properties of language and language use [module 1]
- equipped with that ability, we will then combine modelling ideas into more complex models in order to **tackle instance-level**
  - language understanding [modules 2–3 and 5–6]
  - and generation [modules 4–6].

Goals

ILOs After this class the student is able to

- recognise and motivate a choice of statistical model for text/corpus analysis
- estimate the parameters of a statistical model using observed text/corpora

Experimental Uncertainty

When we collect measurements about phenomena of interest, these measurements are subject to:

- randomness in the world (e.g., different opinions)
- randomness in how we measure things (e.g., star-ratings can be ambiguous)
- our incomplete knowledge of relevant information (e.g., situational context).

Because of all of that (and more), there is a great deal of uncertainty about these measurements.

Uncertainty, Probability and Statistics

When we are uncertain about things, a mathematical representation of this ‘state of incomplete knowledge’ can help us with reasoning and decision-making.

Probability is the framework of choice:[1]

- a preference order over the possible outcomes of the phenomenon under study.

Statistics gives us tools to fit probabilistic representations of uncertainty to experimental data.[2]

[1] Other frameworks exist [2], but they are not nearly as common in AI.
[2] I can hardly think of a better textbook on statistics than McElreath [3].

The statistical approach or probabilistic pipeline

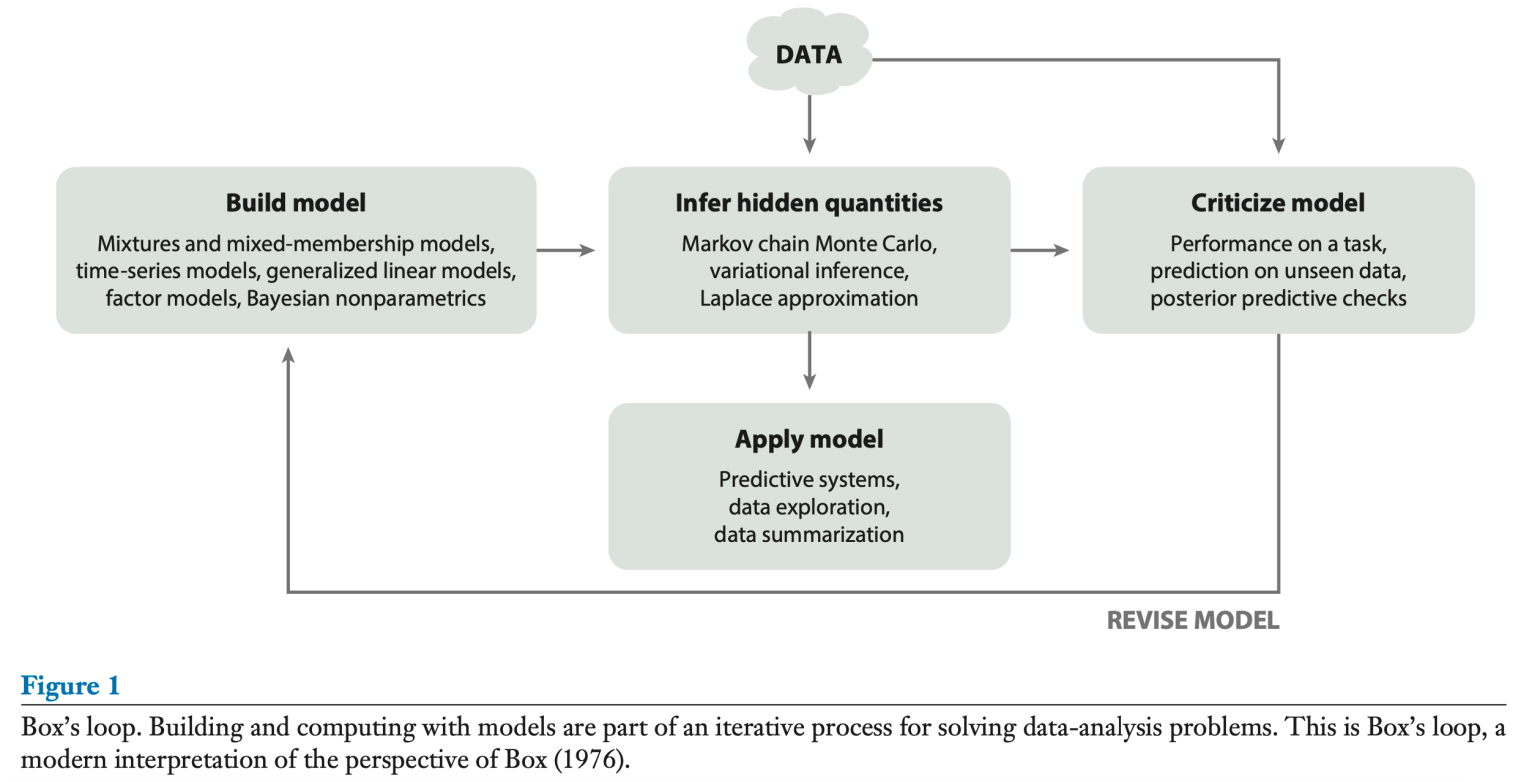


Figure 1

Box's loop. Building and computing with models are part of an iterative process for solving data-analysis problems. This is Box's loop, a modern interpretation of the perspective of Box (1976).

Figure 1: Figure from [1] (pdf)

Table of contents

Data

Written Language

In NTV, we focus on written language, in the formal of digital text.

We may be interested in things like

- properties of a specific language (e.g., structure of words or sentences), properties of a collection of texts (e.g., topics, style);
- a person' belief state (e.g., sentiment analysis);
- mapping between different texts (e.g., translation, question answering);
- many other things (e.g., product demand, social network analysis, etc.)

which involve understanding and/or generating text.

Basic manipulation of digital text in computers: section 1 of T1 is a good introduction/recap.

What types of measurements are there in NLP?

Intrinsic attributes of linguistic data:

- length of a sentence
- syntactic category of a word
- grammatical number (e.g., singular vs. plural)

Statistical attributes of linguistic data:

- word count (e.g., how many times ‘learning’ occurs in BKI course materials?)
- word rank (e.g., what is the most frequent word in BKI course materials?)

Extrinsic attributes:

- the demand for a course (say, based on description in the catalogue and what students post on social media)
- written feedback on a course, possibly accompanied by a numerical or categorical degree of appreciation for the course

What types of measurements are there in NLP?

Intrinsic attributes of linguistic data:

- length of a sentence
- syntactic category of a word
- grammatical number (e.g., singular vs. plural)

Statistical attributes of linguistic data:

- word count (e.g., how many times ‘learning’ occurs in BKI course materials?)
- word rank (e.g., what is the most frequent word in BKI course materials?)

Extrinsic attributes: [modules 2–6]

- the demand for a course (say, based on description in the catalogue and what students post on social media)
- written feedback on a course, possibly accompanied by a numerical or categorical degree of appreciation for the course

Numerical data

Discrete:

- Number of words: 1, 2, 3, ...
- Number of thumbs-up: 1000, 2000000, etc.
- Star ratings: 1, 2, 3, 4, 5.

Numerical data

Discrete:

- Number of words: 1, 2, 3, ...
- Number of thumbs-up: 1000, 2000000, etc.
- Star ratings: 1, 2, 3, 4, 5.

Continuous:

- Market value of book: 10.25, 23.99, etc.
- Temperature: 37.6 C, etc.
- Rate of frequency decay over time: 2.16.

Non-numerical data

Nominal/unstructured

- characters, word parts (e.g., syllables, morphemes)
- words (if we regard a word as indivisible)
- named categories (e.g., sentiment levels, topics, syntactic functions, semantic roles)
- named entities (e.g., Amsterdam, The_Beatles, Angela_Davis)

Non-numerical data

Nominal/unstructured

- characters, word parts (e.g., syllables, morphemes)
- words (if we regard a word as indivisible)
- named categories (e.g., sentiment levels, topics, syntactic functions, semantic roles)
- named entities (e.g., Amsterdam, The_Beatles, Angela_Davis)

Combinatorial/structured

- sequences (e.g., characters forming a word, morphemes forming a word, words forming a sentence)
- trees (e.g., syntactic structure of a sentence)
- graphs (e.g., syntactic and/or semantic relations in text, posts connected by topic in a social media platform)

Some elementary properties of data types

Bounded vs. unbounded:

- Bounded data types vary within a pre-specified finite range. This may be a matter of fact or convenience. For example, we may regard sentiment as having 3 levels (negative, neutral, positive) or 5 levels, or a continuous value from -1 to 1.

Some elementary properties of data types

Bounded vs. unbounded:

- Bounded data types vary within a pre-specified finite range. This may be a matter of fact or convenience. For example, we may regard sentiment as having 3 levels (negative, neutral, positive) or 5 levels, or a continuous value from -1 to 1.

Dimensionality:

- univariate measurements (e.g., length of a document, frequency of a word, market value of a book)
- multivariate measurements are collections of numbers
  - fixed-dimensionality (e.g., frequencies of V words, voting intentions for K candidates)
  - variable dimensionality (e.g., a word represented as a sequence of characters, a sentence represented as a sequence of words, a stream of posts in a social network platform).

What are the main properties of my data type?

A good understanding of the data we are modelling will help us make good modelling choices. Often there is flexibility in how we conceive of our data.

For example, we are modelling the demand for certain books based on written reviews. The demand for a book can be

- an integer (e.g., 12, 120, ...);
- a real value (e.g., 12.6, 120.98, ...);
- an application-specific category such as 'low'/'regular'/'high', or 'tens'/'hundreds'/'thousands'.

Strategy. Find arguments

- in favour of a choice
- against alternative choices

Next: *model*

These first decisions (concerning how we conceive of the data) already count as modelling choices.

Yet the part we typically think of as ‘the model design’ is the part that embraces the probabilistic and statistical aspects of the modelling problem.

That’s what we turn to next.

We will start with two key skills:

1. being able to read out a model design from a structured description of it;
2. being able to motivate a model design from properties of the data;

From generative story to model

Data Generating Process

A procedure that describes how data points are constructed.

Word Slots

Words in English are made of 3 parts: a root (core meaning of the word), a *prefix* and a ***suffix*** (modify the meaning or function of the root). Example: *un*-reason-***ably***.

- All words have roots.
- With probability p, a word also has a prefix.
- Independently of a word having a prefix or not, with probability q, a word gets a suffix.

This procedure might be based on theoretical knowledge, empirical knowledge or just a design choice.

First steps: recognise the data being modelled

Word Slots

Words in English are made of 3 parts: a root (core meaning of the word), a *prefix* and a ***`suffix`*** (modify the meaning or function of the root).
Example: *un*-reason-***`ably`***.

- All words have roots.
- With probability p, a word also has a prefix.
- Independently of a word having a prefix or not, with probability q, a word gets a suffix.

This procedure

- treats a word as a data type made of three slots each of which has a certain kind of slot filler;
- is focused on modelling *which slots* get to be filled, hence we have no probabilistic account of the content of any slot beyond it being 'filled' vs. 'unfilled'.

Next: recognise the *generative story*

The procedure that tells us how to construct data is also known as the **generative story**.

The steps of the procedure are stochastic (but not arbitrary), their stochasticity is controlled by certain *parameters*.

For example, ‘with probability p’ do such and such, otherwise (that is, ‘with probability $1 - p$) do something else.

The steps are chained (we also say they are hierarchically organised).

As such, the generative story prescribes a probability distribution.

A good way to recognise this probability distribution is to imagine the ‘garden of forking paths’ which the generative story describes.

Garden of forking paths [Exercise]

Imagine the generative story as a garden of forking paths. A complete path is a chain of steps taking the process from its initial state (○) to a terminating state (●).

Word Slots

Words in English are made of 3 parts: a root (core meaning of the word), a *prefix* and a ***suffix*** (modify the meaning or function of the root). Example: *un*-reason-***ably***.

- All words have roots.
- With probability p, a word also has a prefix. *(thus, with prob $1 - p$ the prefix is empty*
- Independently of a word having a prefix or not, with prob q, a word gets a suffix. *(thus, with prob $1 - q$ the suffix is empty)*

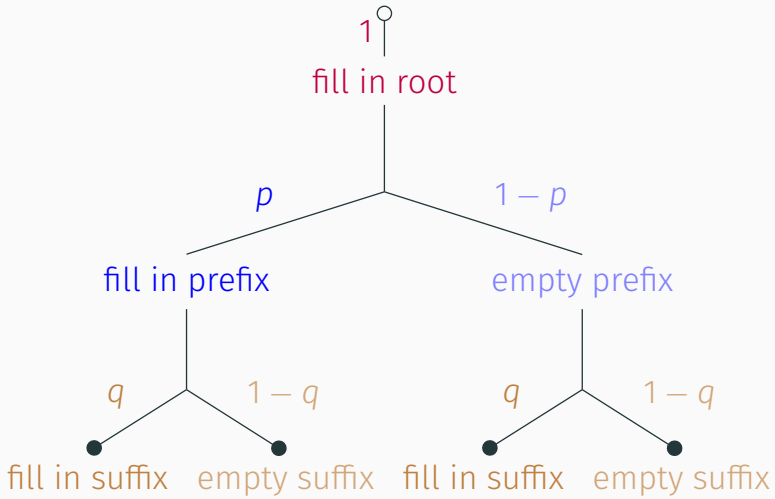
Garden of forking paths [Exercise]

Imagine the generative story as a garden of forking paths. A complete path is a chain of steps taking the process from its initial state (○) to a terminating state (●).

Word Slots

Words in English are made of 3 parts: a root (core meaning of the word), a *prefix* and a ***suffix*** (modify the meaning or function of the root). Example: *un*-reason-***ably***.

- All words have roots.
- With probability p, a word also has a prefix. *(thus, with prob $1 - p$ the prefix is empty*
- Independently of a word having a prefix or not, with prob q, a word gets a suffix. *(thus, with prob $1 - q$ the suffix is empty)*



Each step is assigned a probability. A path's probability is the product of its steps' probabilities.

Probability of ⟨ fill in root, fill in prefix, empty suffix ⟩?

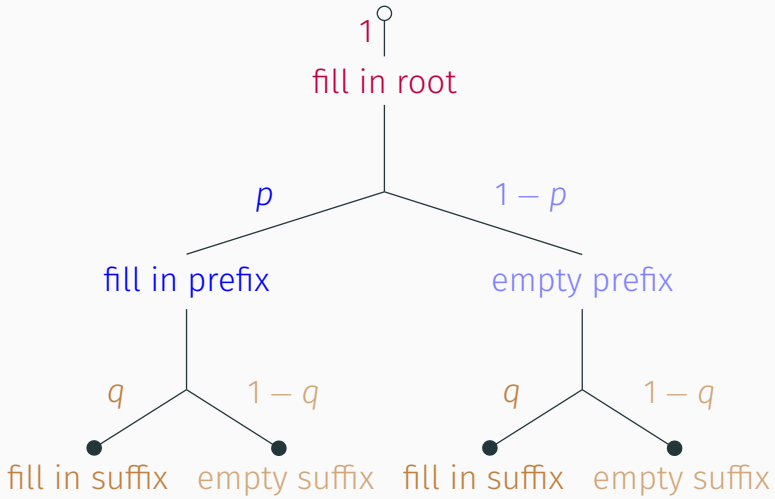
Garden of forking paths [Exercise]

Imagine the generative story as a garden of forking paths. A complete path is a chain of steps taking the process from its initial state (○) to a terminating state (●).

Word Slots

Words in English are made of 3 parts: a root (core meaning of the word), a *prefix* and a ***suffix*** (modify the meaning or function of the root). Example: *un*-reason-***ably***.

- All words have roots.
- With probability p, a word also has a prefix. *(thus, with prob $1-p$ the prefix is empty*
- Independently of a word having a prefix or not, with prob q, a word gets a suffix. *(thus, with prob $1-q$ the suffix is empty)*



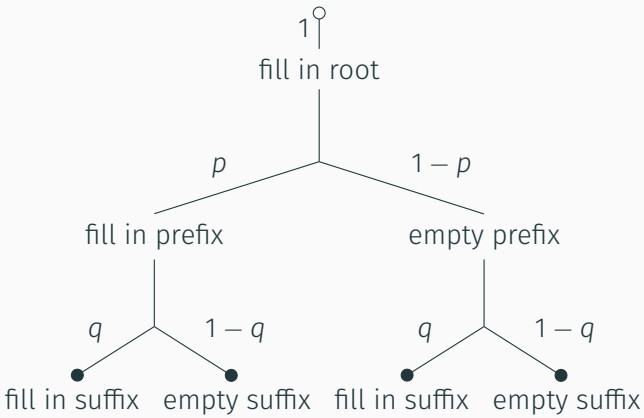
Each step is assigned a probability. A path's probability is the product of its steps' probabilities.

Probability of ⟨ fill in root, fill in prefix, empty suffix ⟩? $1 \times p \times (1-q)$.

Events

We often use the generative story to reason about attributes shared by many outcomes, aka *events*.

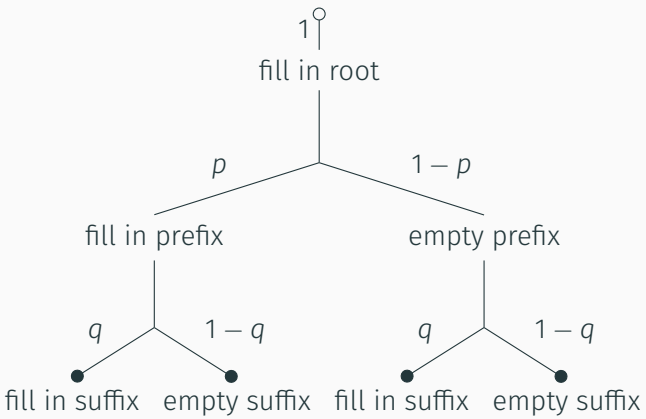
For example, we may be interested in the probability that a word has exactly 2 slots filled in.



Events

We often use the generative story to reason about attributes shared by many outcomes, aka *events*.

For example, we may be interested in the probability that a word has exactly 2 slots filled in.



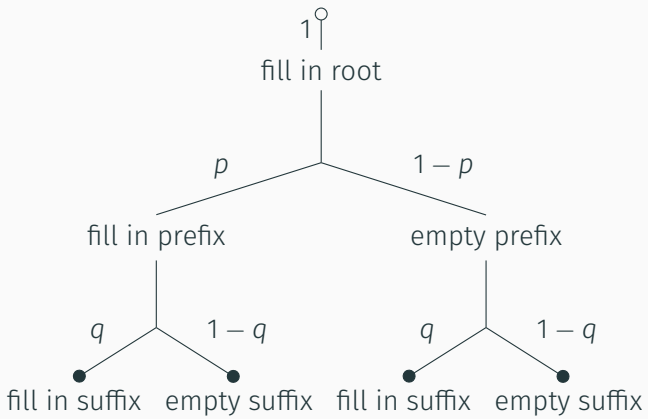
Outcomes with exactly two slots filled in:

- a word with prefix but without suffix,
- *or* a word with suffix but without prefix.

Events

We often use the generative story to reason about attributes shared by many outcomes, aka *events*.

For example, we may be interested in the probability that a word has exactly 2 slots filled in.



Outcomes with exactly two slots filled in:

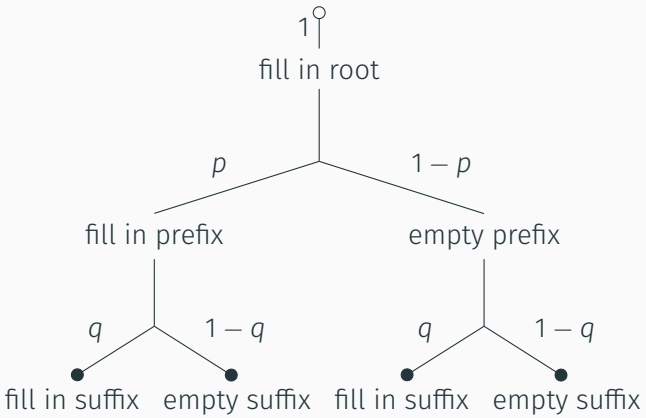
- a word with prefix but without suffix,
- *or* a word with suffix but without prefix.

Sum the probabilities of the paths above:

Root	Prefix	Suffix	Prob
y	y	n	$1 \times p \times (1 - q)$
y	n	y	$1 \times (1 - p) \times q$
		Sum	$p \times (1 - q) + (1 - p) \times q$

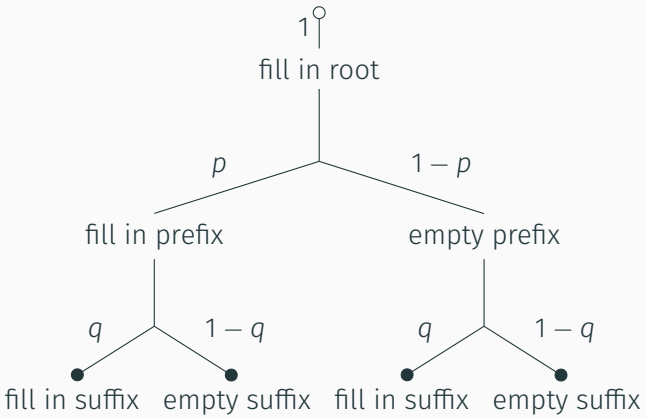
Exercise

According to the *Word Slots* model, what is the probability distribution of the number of filled-in slots in an English word?



Exercise

According to the *Word Slots* model, what is the probability distribution of the number of filled-in slots in an English word?



- Let X denote an outcome of the model.
- Let N denote the number of filled in slots.
- Relevant outcome probabilities

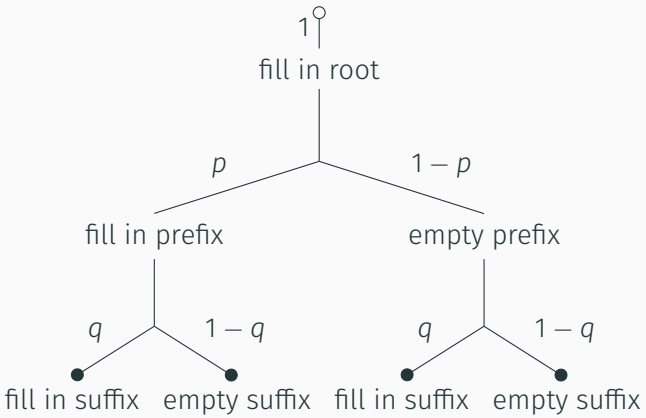
X	$P(X)$	N
(root, -, -)	$(1-p)(1-q)$	1
(root, prefix, -)	$p(1-q)$	2
(root, -, suffix)	$(1-p)q$	2
(root, prefix, suffix)	pq	3

- Distribution of N under this model

N	$P(N)$
1	$(1-p)(1-q)$
2	$p(1-q)+(1-p)q$
3	pq

Exercise

According to the *Word Slots* model, what is the probability distribution of the number of filled-in slots in an English word?



- Let X denote an outcome of the model.
- Let N denote the number of filled in slots.
- Relevant outcome probabilities

X	$P(X)$	N
(root, -, -)	$(1-p)(1-q)$	1
(root, prefix, -)	$p(1-q)$	2
(root, -, suffix)	$(1-p)q$	2
(root, prefix, suffix)	pq	3

- Distribution of N under this model

N	$P(N)$
1	$(1-p)(1-q)$
2	$p(1-q)+(1-p)q$
3	pq

This is a rather common ‘trick’: model one thing (e.g., what slots get filled) to reason about another thing (e.g., number of parts in a word).

From data to model

Data samples

In many cases we do not know the data generating process, yet we can interact with it to obtain data samples.

For example,

I do not know the mechanisms by which a person forms verbal reports in response to a perceptual experience, but I can elicit a verbal report by asking them *'how did you like the class?'*.

Their response (e.g., *I found it exciting, the topic is challenging and relevant and the lecturer's energy was spot on*) is a **data sample** from an otherwise rather *opaque* data generating process.

Analysing the report might shed light into things I care about (e.g., the person's state of mind or opinion about the class). For example, I may associate certain word choices with the person being more or less satisfied with the class.

Data analysis

We can analyse data samples to discover something about how they come about (e.g., understanding something about the mechanisms by which data samples are generated, or how to answer other questions that depend directly or indirectly on that understanding).

For example, studying corpora of written language (at first, mostly English), linguists realised that the law (known as Zipf's law)

$$\text{word frequency} \propto \frac{1}{\text{word rank}}$$

holds rather robustly.

How would you go about studying whether or not that's true of language use in general?

Tools for data analysis

Data analysis techniques help us test generalisations as well as ‘uncover’ (candidate) generalisations.

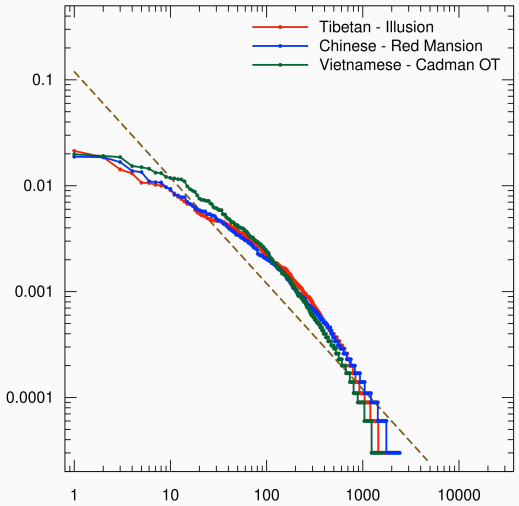
Important tools include:

- Data visualisation: instance-level and population-level (e.g., plots).
- Descriptive statistics
- Hypothesis testing
- Statistical inference

Search for (better) explanations

When we plot word frequency against rank for languages other than English, we may find deviations from the pattern predicted by Zipf's law.

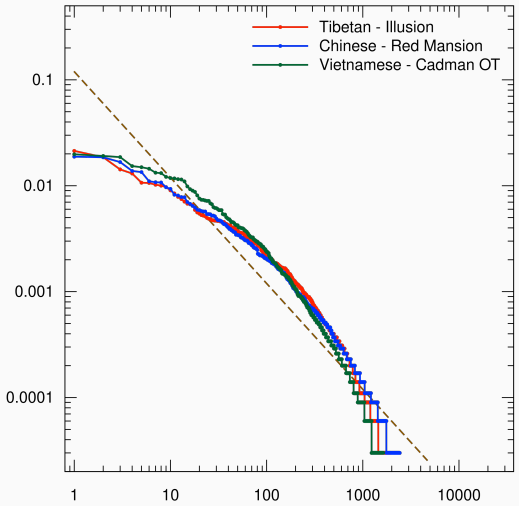
Is the pattern then an illusion?



Search for (better) explanations

When we plot word frequency against rank for languages other than English, we may find deviations from the pattern predicted by Zipf's law.

Is the pattern then an illusion?

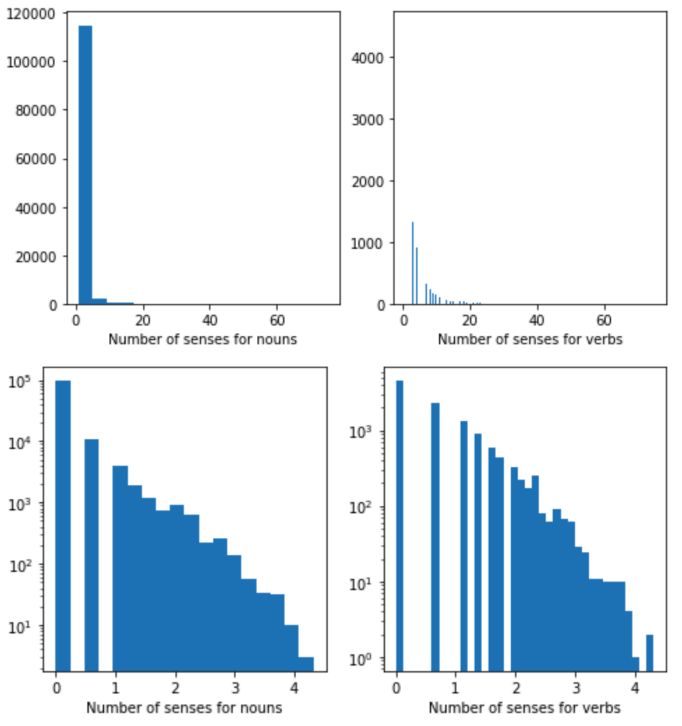


The empirical method is a process of iterative refinement of the available explanations. A better question to ask is: **can we explain what went wrong and perhaps improve our model of how frequency and rank relate?**

Factors known to affect the original prediction: translated text, spelling reforms, morphological complexity (e.g., words marked for agreement of grammatical gender).

Number of noun/verb senses [Exercise]

Suppose we are interested in learning about polysemy in English. We collect text and annotate the occurrences of nouns and verbs with information about their senses.[4] See histograms below.



Top: occurrences vs. number of senses (left/nouns, right/verbs). Bottom: same as top but with log axes.

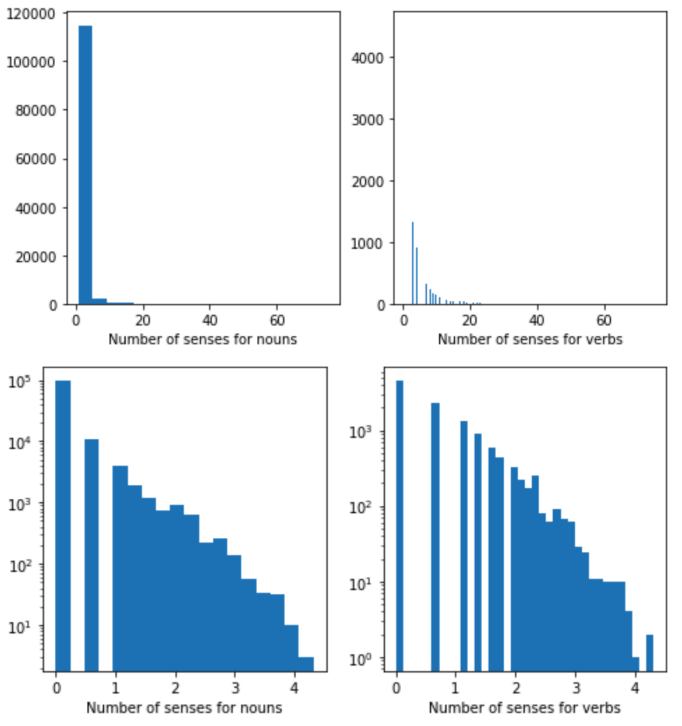
Can the observe pattern of number of senses of verbs in English be captured by a known statistical law? Here are some options

- Binomial
- Poisson
- Geometric
- Zipf

[4] Example: look-v (7 senses)

Number of noun/verb senses [Exercise]

Suppose we are interested in learning about polysemy in English. We collect text and annotate the occurrences of nouns and verbs with information about their senses.[4] See histograms below.



Top: occurrences vs. number of senses (left/nouns, right/verbs). Bottom: same as top but with log axes.

Can the observe pattern of number of senses of verbs in English be captured by a known statistical law? Here are some options

- Binomial
- Poisson
- Geometric
- Zipf

Would this choice also work well for nouns?

[4] Example: look-v (7 senses)

Number of noun/verb senses [Solution]

We need to look for a law that (1) supports integers starting from 1, (2) the pmf decays (roughly) exponentially quickly, and (3) the variance is not too high. We can contrast properties of known laws against these objectives.

The Binomial distribution does not seem appropriate: its generative story involves a known number of fixed draws, which we don't have here. The Geometric, the Poisson, and the Zipf distributions are possibly appropriate in terms of goal number 1.

It looks like the mode of the data samples is always at 1. The Poisson distribution seems less adequate now: the only Poisson with a mode at 1 is Poisson(1), if we pick another parameter hoping to better match the spread/variance, we will change the position of the mode.

The Geometric and the Zipf have their modes fixed at 1, for any choice of parameter. Both have pmfs that decay (roughly) exponentially quickly. The Zipf though has extremely heavy tails: draws from Zipf will always keep deviating dramatically far from 1. This kind of behaviour does not seem to be present in the data, so amongst these options, Geometric seems the most defensible.

While the Geometric might work well for verbs, the situation is less clear for nouns. It looks like nouns concentrate much more than the Geometric can express (or in other words, the Geometric variance cannot be adjusted well to the available data)

Limitations

Often no known model is adequate and we need to design new ones [all of NTV beyond module 1].

Strategies include

- composing known models in novel ways [PGMs]
- developing flexible tools for function approximation [NNs] and data representation [DL]
- combining PGMs and NNs/DL [NTV from module 3]

Next: *learning from data*

So far the data we have (incl. our knowledge about the problems we want to tackle) informed our choices as model designers:

- how to better conceive of the data type
- what patterns do we want to capture or uncover
- what known statistical law can come to help

Now, we use the data samples we have access to as a means to **fit** our probabilistic models to historical/observed data.

Parameter estimation

Unknown parameters

Most our models are so-called **parametric models** built upon a mathematical relationship between outcomes of a random variable (rv) and some numerical **parameters**.

For example,

- Given a strictly positive parameter $\lambda \in \mathbb{R}_{>0}$, the Poisson model assigns probability $\frac{\lambda^k \exp(-k)}{k!}$ to outcome $k \in \mathbb{N}_0$ of an rv counting events of a certain kind (specified by the Poisson generative story).
- Given three parameters $(\pi_1, \pi_2, \pi_3)^\top \in \Delta$, the Categorical model assigns probability π_i to outcome $i \in \{1, 2, 3\}$ of an rv with three levels (e.g., negative/neutral/positive).

Unknown parameters

Most our models are so-called **parametric models** built upon a mathematical relationship between outcomes of a random variable (rv) and some numerical **parameters**.

For example,

- Given a strictly positive parameter $\lambda \in \mathbb{R}_{>0}$, the Poisson model assigns probability $\frac{\lambda^k \exp(-k)}{k!}$ to outcome $k \in \mathbb{N}_0$ of an rv counting events of a certain kind (specified by the Poisson generative story).
- Given three parameters $(\pi_1, \pi_2, \pi_3)^\top \in \Delta$, the Categorical model assigns probability π_i to outcome $i \in \{1, 2, 3\}$ of an rv with three levels (e.g., negative/neutral/positive).

When we pick a model family (such as Poisson or Categorical), we still need to specify a parameter value in order to single out a concrete member of that family (e.g., $\mathsf{Poisson}(2.5)$ or $\mathsf{Categorical}(0.3, 0.1, 0.6)$).

Maximum likelihood estimation

Our strategy of choice will be to use the available data ('training data') to optimise our choice of parameter value.

We will pick the parameter value θ (in the space of valid parameter values for our model) which leads to our model assigning highest probability (mass or density) to the observed data.

That is, given N observations $\mathcal{D} = \{x^{(1)}, \ldots, x^{(N)}\}$, and a model with probability mass/density function $p(x; \theta)$, we choose the value of θ whose likelihood

$$L(\theta|\mathcal{D}) = \prod_{n=1}^{N} p(x^{(n)}; \theta)$$

is maximum.

Strategies

In T1

- Exact: compute the right statistic (usually scalable, if you have the data), and apply a formula.
- Grid search: simple procedure, but generally intractable (esp. for multivariate parameters).

Scalable solutions for multivariate problems (from module 2): gradient-based optimisation.

Example: the Categorical case

The Categorical model is the arguably the most important building block in NLP. We will use it to design text classifiers and language models.

In its simplest form, the Categorical MLE captures the rate a which we observe a category or another out of a finite countable set.

For K categories, it is specified via a K-dimensional probability vector.

Categorical model of sentiment in a given review dataset

	⋆	⋆⋆	⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆⋆
Counts	217	250	772	2084	6938
MLE					

Example: the Categorical case

The Categorical model is the arguably the most important building block in NLP. We will use it to design text classifiers and language models.

In its simplest form, the Categorical MLE captures the rate a which we observe a category or another out of a finite countable set.

For K categories, it is specified via a K-dimensional probability vector.

Categorical model of sentiment in a given review dataset

	⋆	⋆⋆	⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆⋆
Counts	217	250	772	2084	6938
MLE	0.021	0.024	0.075	0.203	0.676

Note this is as *population-level* analysis supporting claims about the dataset (as opposed to specific reviews in it).

Example: the conditional Categorical case

In conditional form, a Categorical model can support claims about specific *conditions*. Here, MLE captures the rate a which we observe a category or another out of a finite countable set **in a given observed condition (or context)**.

Categorical model of sentiment given review summary

Context	*great strings*				
	⋆	⋆⋆	⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆⋆
Counts	0	0	0	2	37
MLE					

Example: the conditional Categorical case

In conditional form, a Categorical model can support claims about specific *conditions*. Here, MLE captures the rate a which we observe a category or another out of a finite countable set **in a given observed condition (or context)**.

Categorical model of sentiment given review summary

Context	*great strings*				
	⋆	⋆⋆	⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆⋆
Counts	0	0	0	2	37
MLE	0	0	0	0.051	0.949

Context	*does the job*				
	⋆	⋆⋆	⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆⋆
Counts	0	0	2	14	12
MLE					

Example: the conditional Categorical case

In conditional form, a Categorical model can support claims about specific *conditions*. Here, MLE captures the rate a which we observe a category or another out of a finite countable set **in a given observed condition (or context)**.

Categorical model of sentiment given review summary

Context	*great strings*				
	⋆	⋆⋆	⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆⋆
Counts	0	0	0	2	37
MLE	0	0	0	0.051	0.949

Context	*does the job*				
	⋆	⋆⋆	⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆⋆
Counts	0	0	2	14	12
MLE	0	0	0.071	0.5	0.429

Limitations

How many reviews do you expect to find with summary *does as promised, okay if you got a lot of axes and a fav strap*?

In general, the more verbose the review, the more information we expect to be available for comprehension. But, due to the overly-simplistic assumptions of our conditional categorical model, we run into data sparsity issues: **we simply do not have enough data to estimate all parameters**.

Key limitation: simple statistical models often treat outcomes and conditions as unrelated to one another

- 4-stars is categorically distinct from 5-stars
- *does the job* is categorically distinct from *does as promised, okay if you got a lot of axes and a fav strap*

Limitations

How many reviews do you expect to find with summary *does as promised, okay if you got a lot of axes and a fav strap*?

In general, the more verbose the review, the more information we expect to be available for comprehension. But, due to the overly-simplistic assumptions of our conditional categorical model, we run into data sparsity issues: **we simply do not have enough data to estimate all parameters**.

Key limitation: simple statistical models often treat outcomes and conditions as unrelated to one another

- 4-stars is categorically distinct from 5-stars
- *does the job* is categorically distinct from *does as promised, okay if you got a lot of axes and a fav strap*

Strategy (from module 2): treat data as if made of decomposable parts and model them with a hierarchy of simple model components.

Wrap-up

Summary

We can readily apply our knowledge of probability and statistics to model population-level attributes of linguistic data (e.g., length, rank-frequency, number of parts, etc.).

But in NLP more generally, we want to reason about (and/or discover) patterns that are far more specific. For example, we may need to reason about patterns about how people's sentiments get expressed in words, so that we can infer one's sentiment based on the fresh reviews they write.

This requires more than identifying and fitting simple statistical laws and/or generative stories. It will require putting together a number of known statistical laws / building our own generative stories (open the PGMs toolbox) and exploiting flexible data representations with statistically efficient parameter sharing (open the NNs/DL toolbox).

What Next?

Beyond its graded exercises, **T1** teaches you *a lot* of important practical data analysis skills.

This week's **reading** is a notebook with a collection of distributions. You don't need to memorise those distributions, but you need to familiarise yourself with working with distributions in general.

Exam-like 1 (on ANS) has an exercise for you to practice mapping from data to model. TAs will be solving this in class at WC1.

In **module 2**, we start modelling input-output pairs. The input is text (e.g., a review) and the output is an attribute of it (e.g., sentiment).

References

[1] David M Blei. Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, 1(1):203–232, 2014.

[2] Joseph Y Halpern. *Reasoning about uncertainty*. MIT press, 2017.

[3] Richard McElreath. *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman and Hall/CRC, 2018.