

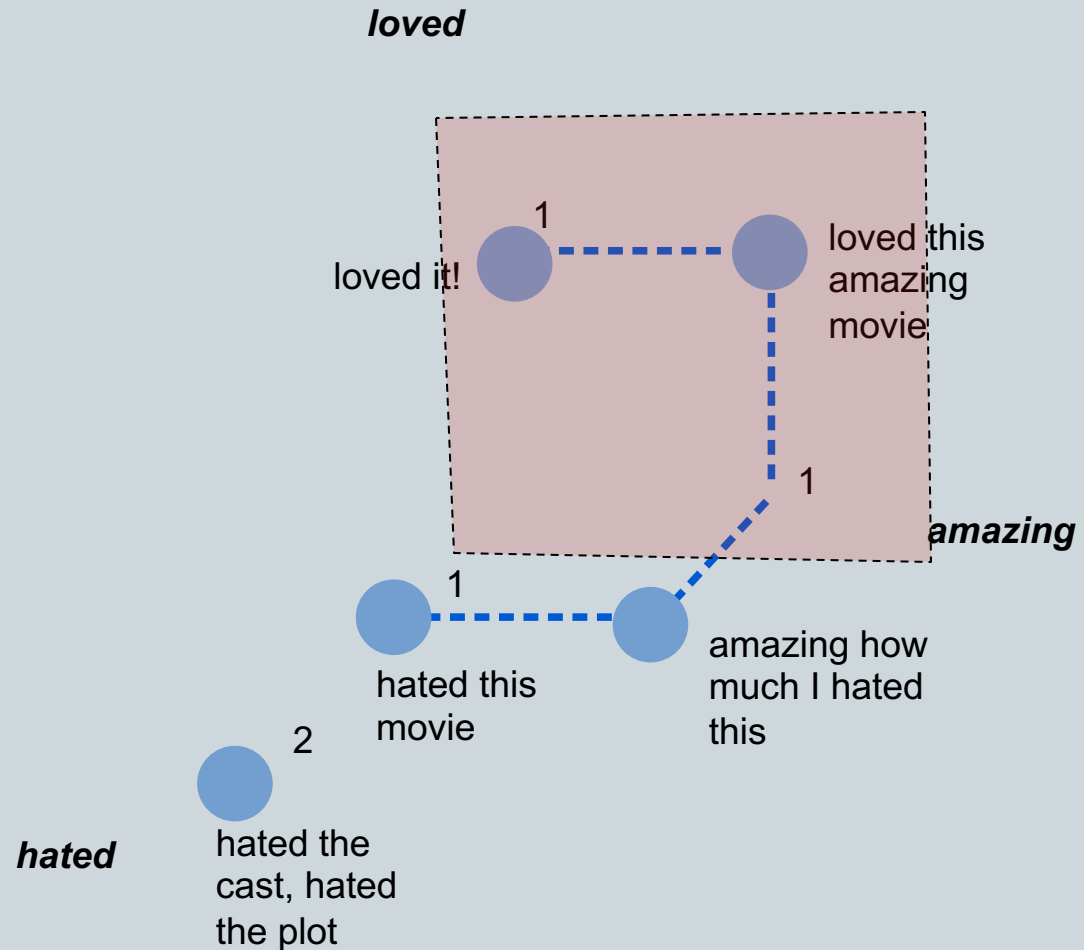
# Supervised machine learning for text classification

**KING'S**  
*College*  
**LONDON**



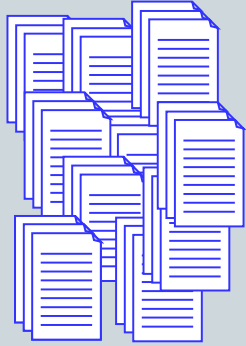
# Bag of words

- Recall our bag of words example
- We represented our documents in a vector space
- We separated out different classes of documents
- This is an example of a general technique, called *classification*
- Building a model from examples is called *training* the model, or *supervising* the model



# Supervised classification

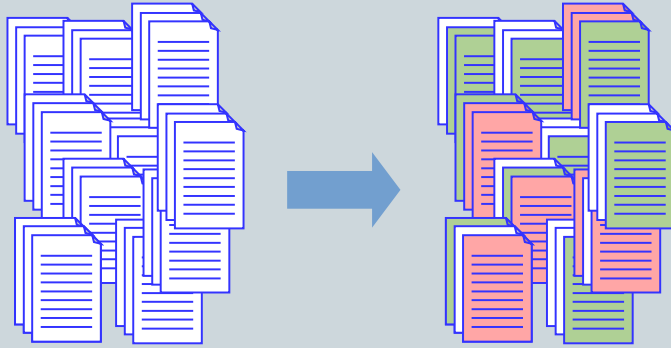
---



- Take a set of example texts.
- They might be sentences, whole documents, single words, or some other portion of text.
- This is our training corpus.

# Supervised classification

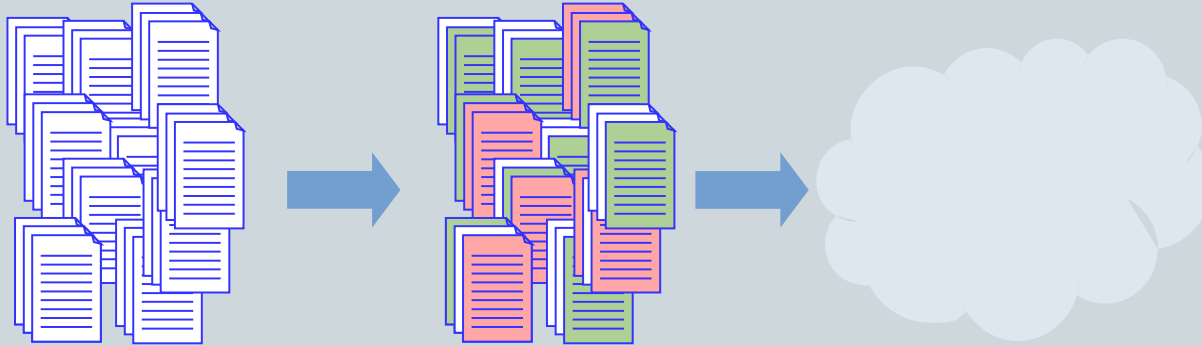
---



- Label each example, with the classes in our problem.
- Labelling will often be done by human.
- We might be lucky enough to have some existing labelled data, e.g. radiology reports with a code for tumour class attached..

# Supervised classification

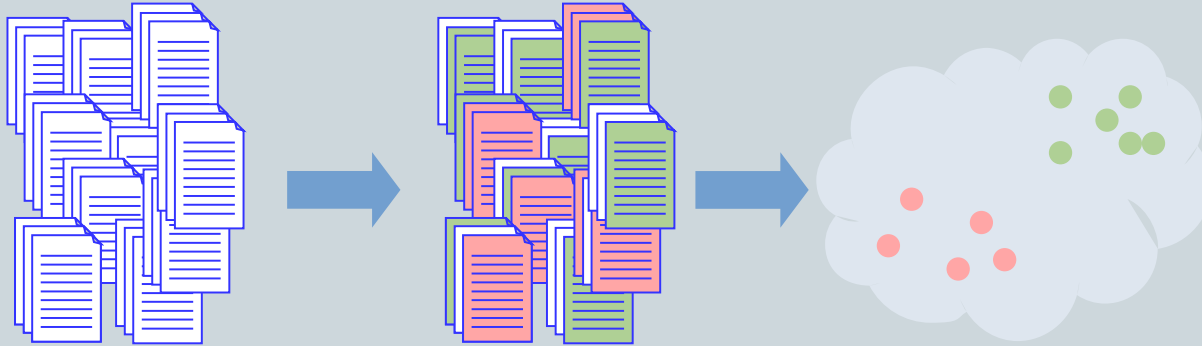
---



- Select features to represent our texts.
- These might be the presence of words, parts of speech, distances between words, word sequences (ngrams), presence of word groups, sentence lengths, etc.
- We may use numeric representations of words as features, computed in a separate step. In the state of the art, these are referred to as embeddings.

# Supervised classification

---



- Represent the texts in this feature space.

# Supervised classification

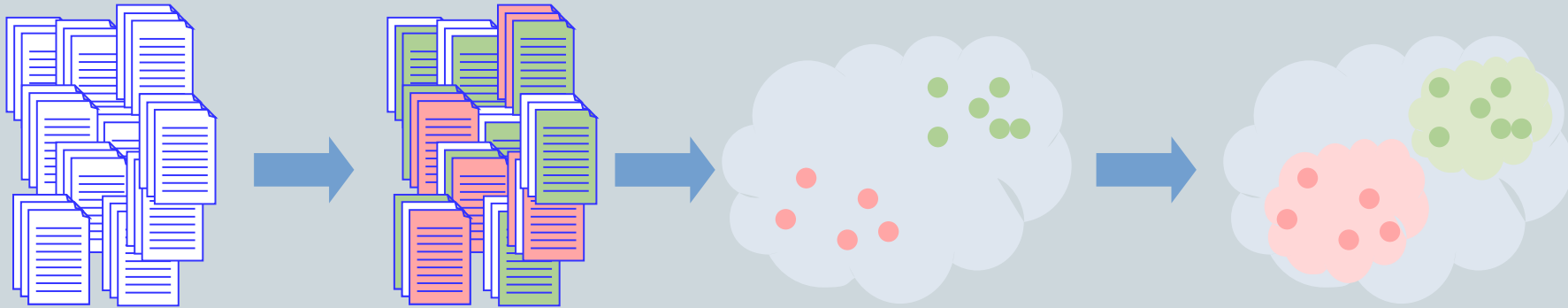
---



- Compute some separator between classes.
- This will involve measures of distance between points.
- It might also involve methods for projecting multiple dimensions into different spaces in which they are separable (kernels).

# Supervised classification

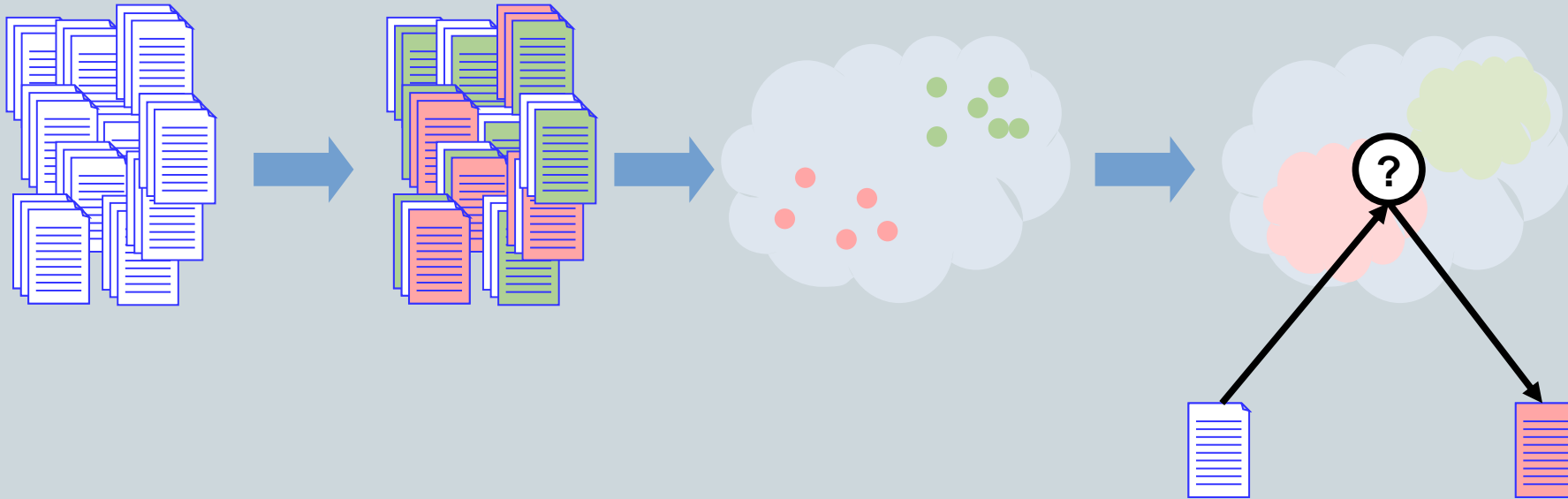
---



- Commonly used classification algorithms in NLP:
  - K Nearest Neighbours
  - Decision Trees and Random Forest
  - Naive Bayes
  - SVM (very popular)
  - CRF
  - Neural nets, e.g. CNNs, LSTMs, Transformers



# Supervised classification



- Classify / label new, previously unseen examples by representing them in the same feature space.

# Named entity recognition

# Named Entity Recognition as a classification problem

- We have looked at how we might classify document
- But what if we want to extract mentions of things from documents?
- For example, people's names, or medications, or symptoms?
- This is called Named Entity Recognition (NER)

Anna Larsson **PERSON** is a famous author from Sweden **GPE** who now lives in New York **GPE** .  
Her recent book Shadows in the Dark **WORK\_OF\_ART** was an international success.

Yesterday **DATE** at 9 a.m. the **TIME** IKEA **ORG** stock went up 30% **PERCENT**  
because of their upcoming launch in New Zealand **GPE** .

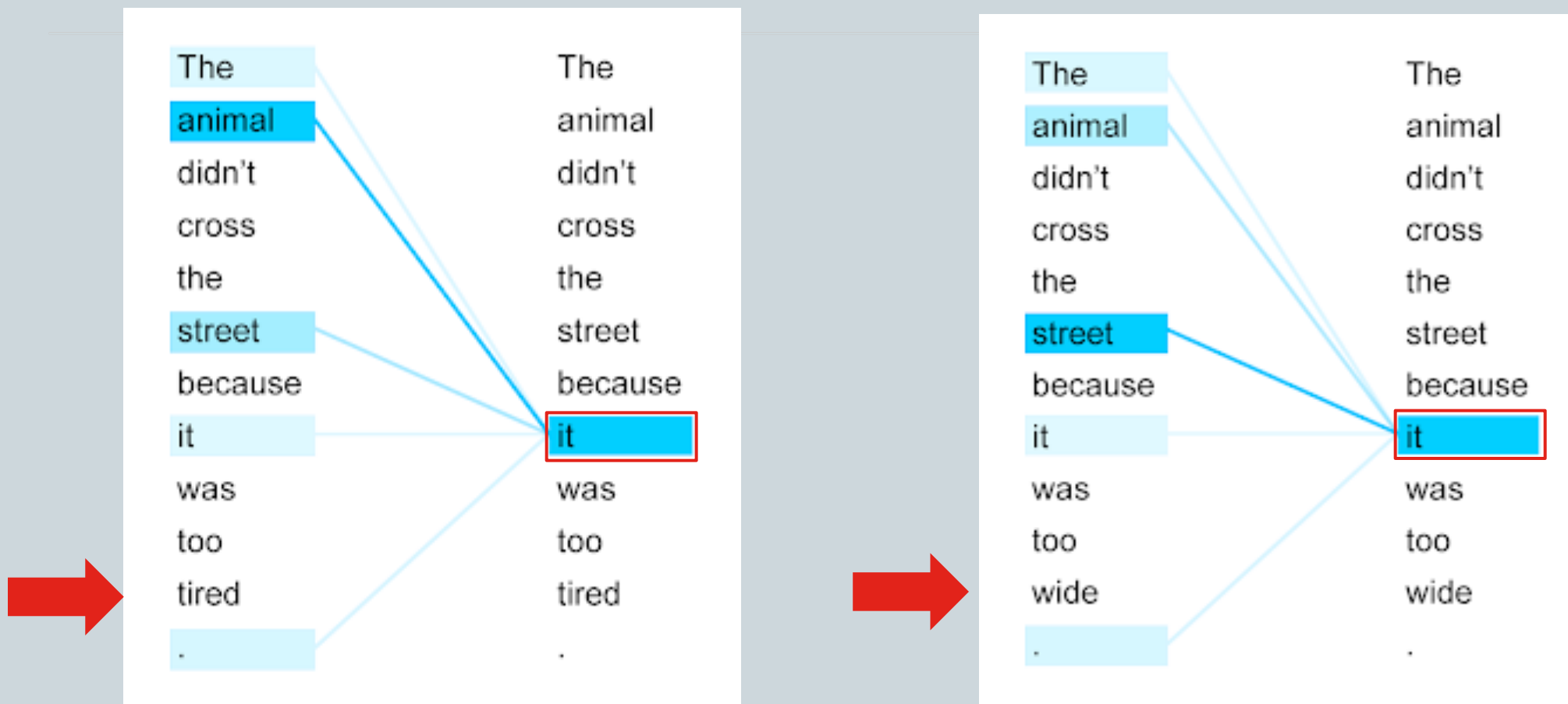
# Named Entity Recognition as a classification problem

- Instead of finding the class of each document, we find the class of each word
- But some entities are made up of multiple words
- So we have classes that represent both the entity type, and where in that entity the word is found
- (There are other ways to do this)

TEXT	IOB	ENTITY TYPE	CLASS	DESCRIPTION
Anna	B	PERSON	PER_B	beginning of an entity
Larsson	I	PERSON	PER_I	inside an entity
is	O	""	O	outside an entity
a	O	""	O	outside an entity
famous	O	""	O	outside an entity
author	O	""	O	outside an entity
from	O	""	O	outside an entity
Sweden	B	GPE	GPE_B	beginning of an entity

**Better and bigger representations**

# Self-attention – example distribution



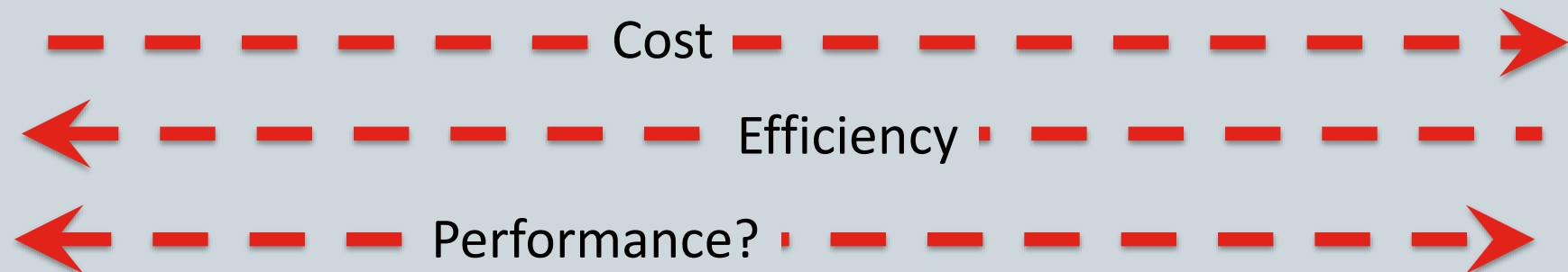
The encoder self-attention distribution for the word “it” from the 5th to the 6th layer of a Transformer trained on English to French translation (one of eight attention heads).

# Large Language Models

	BERT	GPT	GPT2	GPT3	GPT4
<b>Model layer</b>	Encoder	Decoder	Decoder	Decoder	Decoder
<b>Pre-training task</b>	MLM, NSP	Text generation	+ task conditioning	+ in-context patterns	
<b>Training data</b>	3.3 billion words	7000 books	40 GB	45 TB	1 PB ?
<b>Context window</b>	512	512	1024	2048	8000 – 32000 ?
<b>Parameters</b>	110 M	117 M	1.5 B	175 B	1 T ?
<b>Suitability</b>	Sequence tasks	Generation	Generation	Generation, adaptable	Generation, adaptable
<b>Availability</b>	Open	Open	Open	Limited, API	Limited, API

# Large Language Models

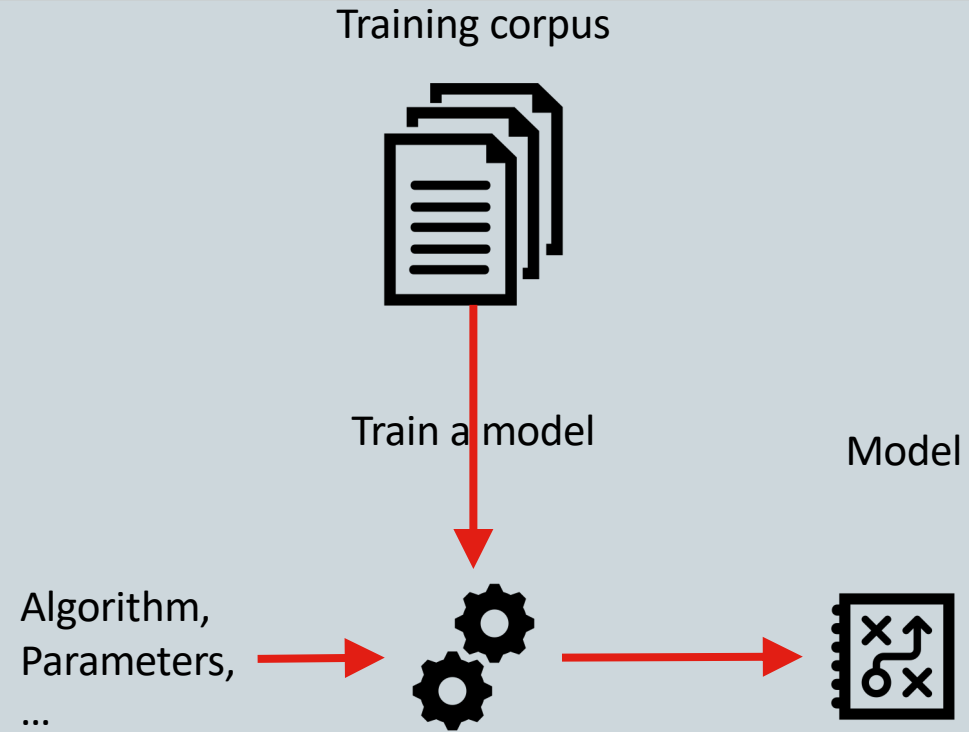
	BERT	GPT	GPT2	GPT3	GPT4
<b>Model layer</b>	Encoder	Decoder	Decoder	Decoder	Decoder
<b>Pre-training task</b>	MLM, NSP	Text generation	+ task conditioning	+ in-context patterns	
<b>Training data</b>	3.3 billion words	7000 books	40 GB	45 TB	1 PB ?
<b>Context window</b>	512	512	1024	2048	8000 – 32000 ?
<b>Parameters</b>	110 M	117 M	1.5 B	175 B	1 T ?
<b>Suitability</b>	Sequence tasks	Generation	Generation	Generation, adaptable	Generation, adaptable
<b>Availability</b>	Open	Open	Open	Limited, API	Limited, API



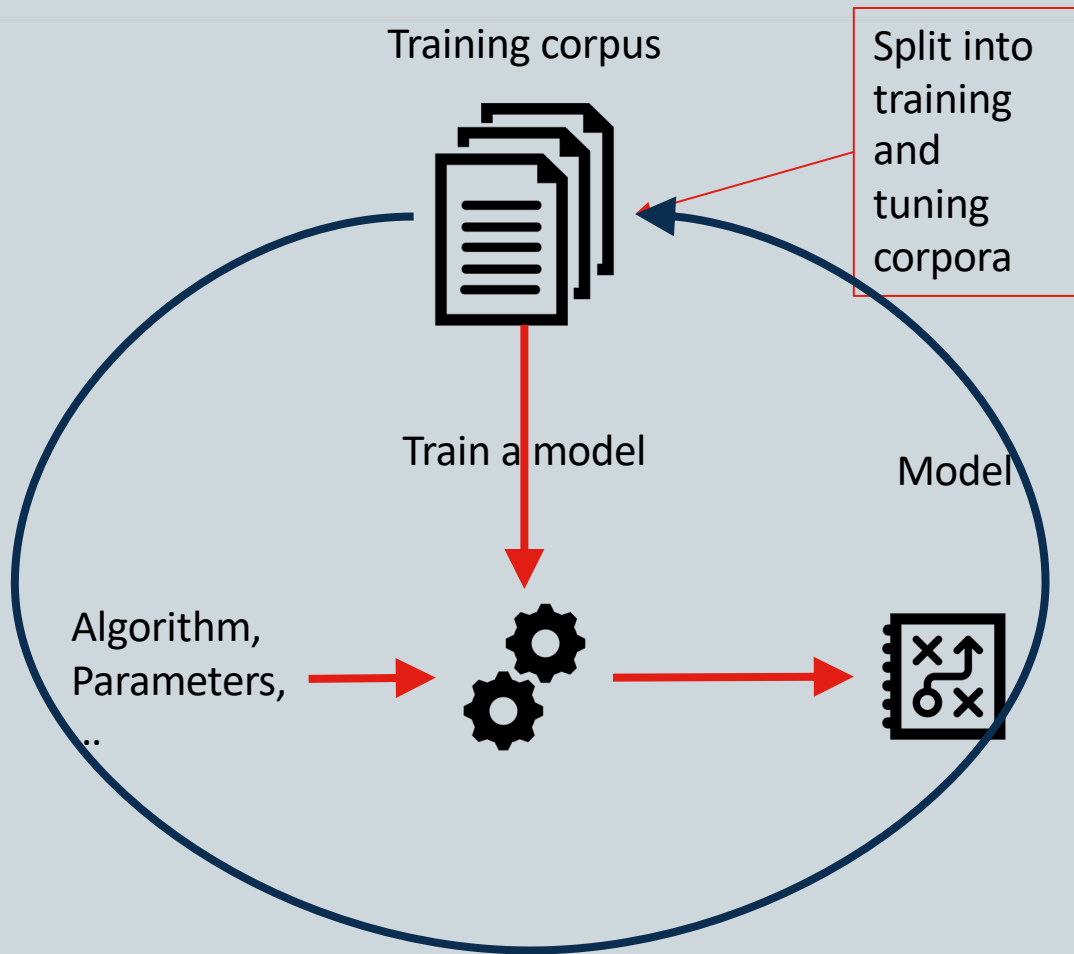


**Testing and evaluating**

# How good is the model?



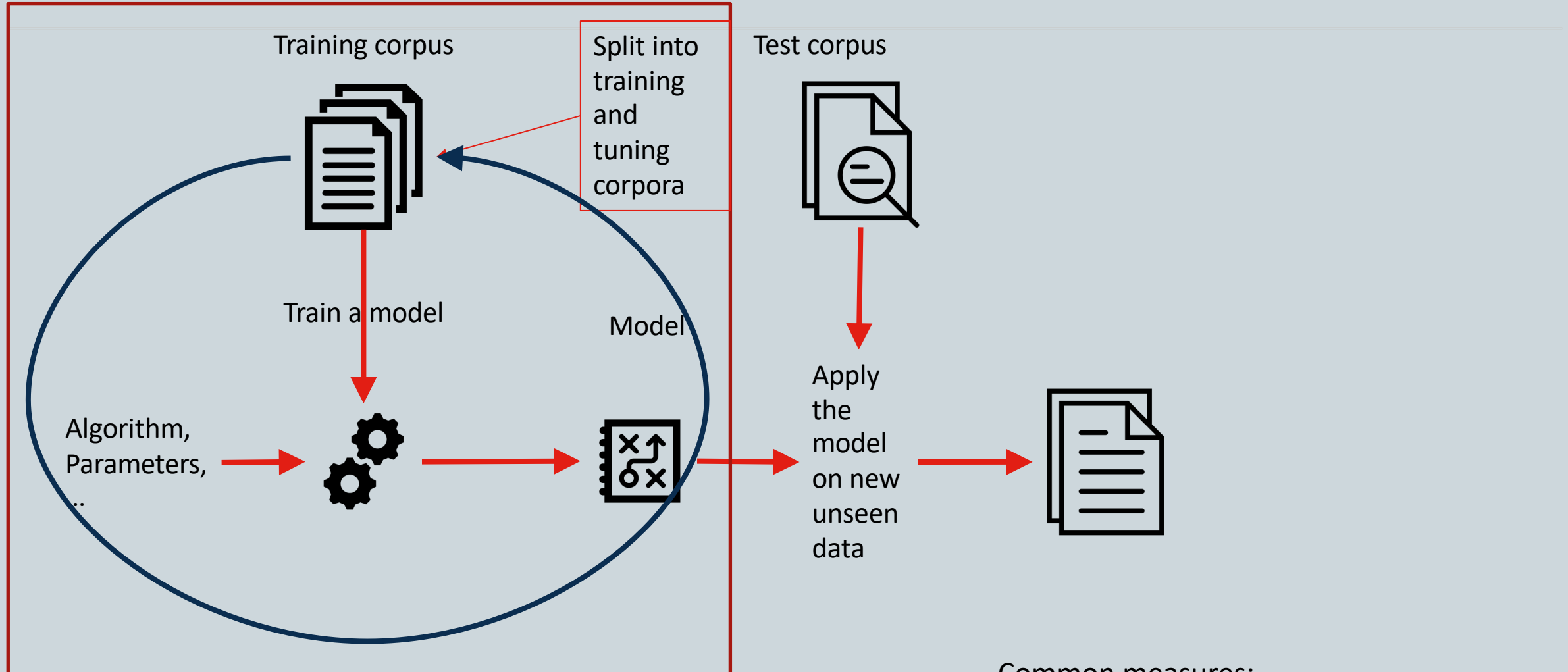
# How good is the model?



Common measures:

- Precision,  $P$  == positive predictive value
- Recall,  $R$  == sensitivity
- F1, the harmonic mean of  $P$  and  $R$

# How good is the model?



Common measures:

- Precision,  $P$  == positive predictive value
- Recall,  $R$  == sensitivity
- F1, the harmonic mean of  $P$  and  $R$

# Thank you

[angus.roberts@kcl.ac.uk](mailto:angus.roberts@kcl.ac.uk)

<https://www.kcl.ac.uk/people/angus-roberts>