NIHR | Maudsley Biomedical Research Centre

Natural language processing for healthcare

Methods

Angus Roberts, Senior Lecturer in Health Informatics
Institute of Psychiatry, Psychology and Neuroscience
King’s College London

Contents

- Rationalist methods
  - rule based NLP
- Empirical approaches
  - machine learning based NLP
- NLP as a processing pipeline
- Comparing rule based NLP and machine learning

Rationalist methods and rule based NLP

Rationalism

- If we want to manipulate language computationally, and process it, we need to represent it in some way
- The first few decades of NLP concentrated on a rationalist approach to this problem
- The thinking was that language could be reasoned about in some logical way, and that the structures of language could be rationalised in to sets of rules

Pattern matching

- Rationalism is typified by using rules to match patterns in language
- In pattern matching, the representation is typically
  - the surface string itself
  - mappings from the string to grammatical and semantic categories
- We will explain the approach with an example

An example – mini mental state exam, a test of cognitive ability

```
His MMSE was 23/30 on 15 January 2008.
```

Sentence detection / splitting

`His MMSE was 23/30 on 15 January 2008.`

Sentence detection / splitting

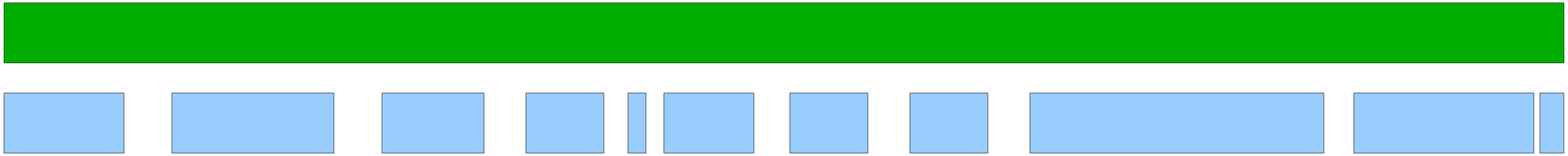
`His MMSE was 23/30 on 15 January 2008.`

Id	Type
1	sentence

We have annotated (labelled, tagged) some span of text

Tokenisation

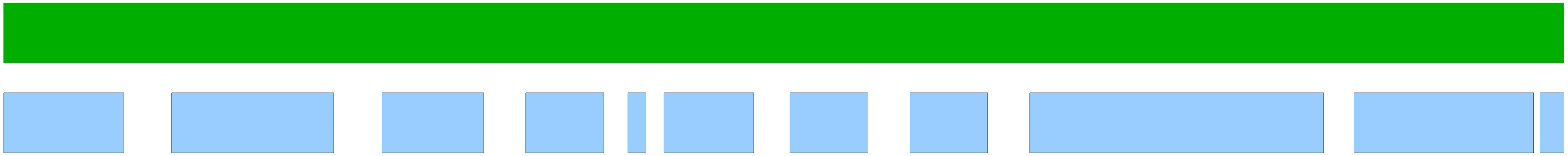
His MMSE was 23/30 on 15 January 2008.



Id	Type
1	sentence

Tokenisation

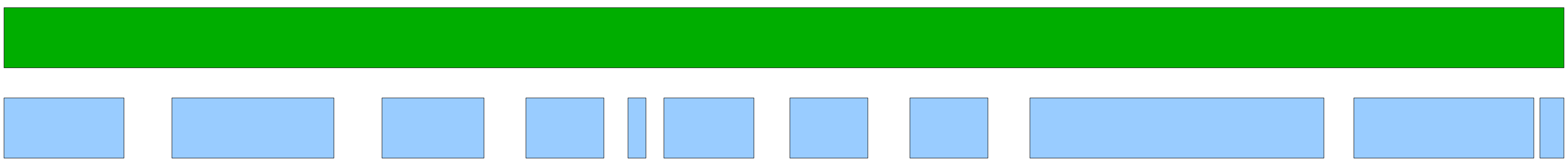
`His MMSE was 23/30 on 15 January 2008.`



Id	Type
1	sentence

Tokenisation

```
His MMSE was 23/30 on 15 January 2008.
```



Id	Type
1	sentence
2	token
3	token
4	token
5	token
6	token
7	token

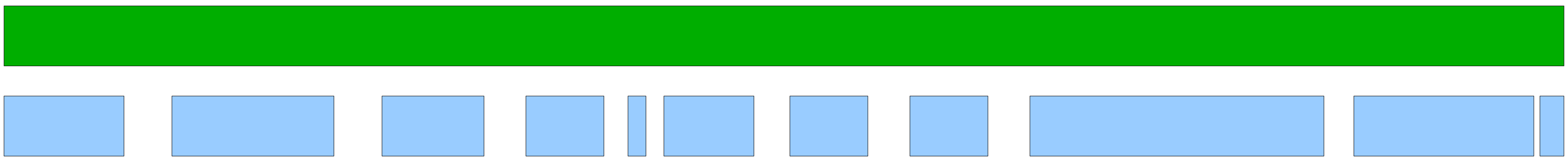
More spans of text annotated (labelled, tagged)

Part of speech (POS) tagging

- Categorise words by their grammatical function
- Allows us to
  - Generalise – e.g. from “he” / “she” to pronoun
  - Take in to account syntactic structure

Part of speech tagging

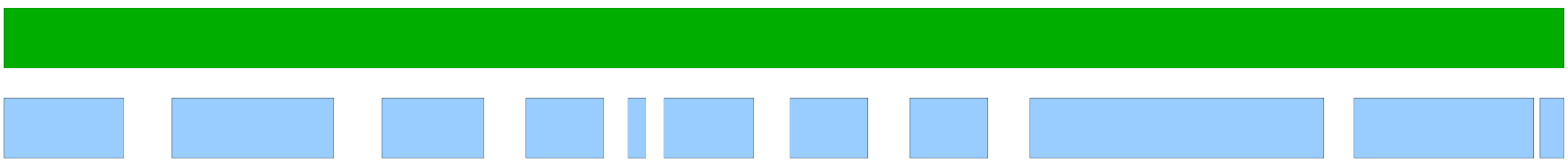
```
His MMSE was 23/30 on 15 January 2008.
```



Id	Type	Attributes
1	sentence	
2	token	pos=PP
3	token	pos=NN
4	token	pos=VB
5	token	pos=CD
6	token	pos=SM
7	token	pos=CD

Part of speech tagging

```
His MMSE was 23/30 on 15 January 2008.
```

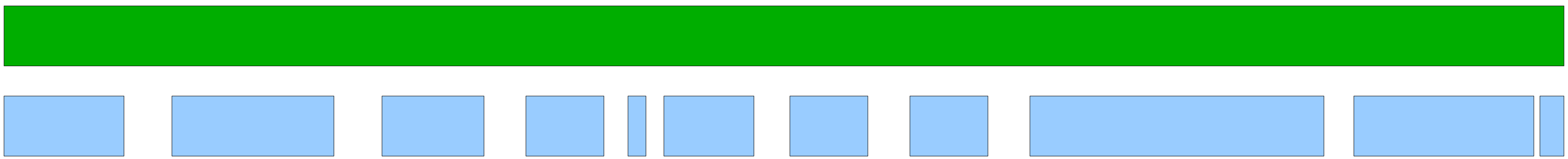


Id	Type	Attributes
1	sentence	
2	token	pos=PP
3	token	pos=NN
4	token	pos=VB
5	token	pos=CD
6	token	pos=SM
7	token	pos=CD

Adding attributes to our annotations

Other token categories

```
His MMSE was 23/30 on 15 January 2008.
```



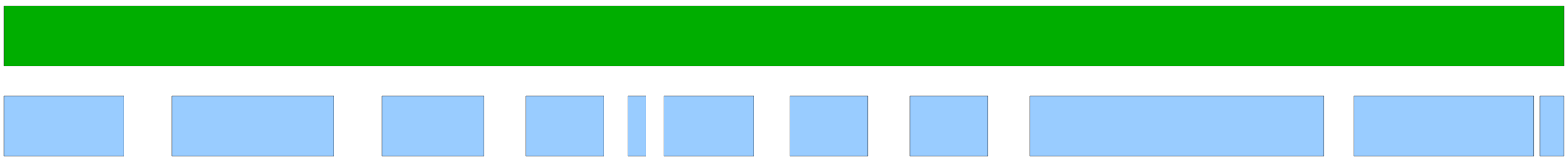
Id	Type	Attributes	
1	sentence		
2	token	pos=PP	
3	token	pos=NN	
4	token	pos=VB	
5	token	pos=CD	type=number
6	token	pos=SM	type=slash
7	token	pos=CD	type=number

Lemmatisation

- Replace a word with its canonical form
  - e.g. sleep is the lemma for
    - slept
    - sleeping
    - sleeps
- Allows us to generalise

Lemmatisation

```
His MMSE was 23/30 on 15 January 2008.
```



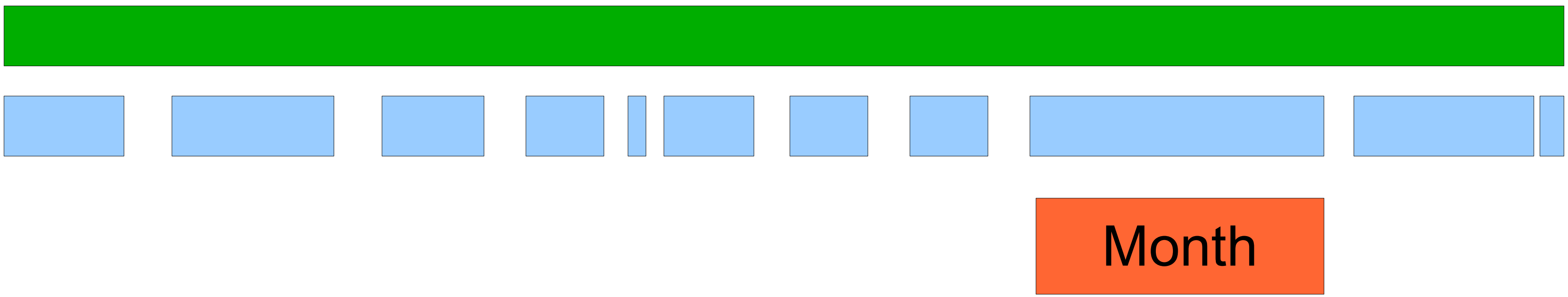
Id	Type	Attributes	
1	sentence		
2	token	pos=PP	lemma=he
3	token	pos=NN	
4	token	pos=VB	lemma=be
5	token	pos=CD	type=number
6	token	pos=SM	type=slash
7	token	pos=CD	type=number

Dictionary lookup

- The fact that a word occurs in some list of similar words is useful information
- We might have a list of month names, or of words used to describe MMSEs.
- Allows us to incorporate semantics, or meaning, in to our analysis
- We can also make use of standard lists, terminologies and knowledge bases
- e.g. SNOMED-CT or ICD-10 terms
  - But beware that these lists were not usually designed to reflect human language use

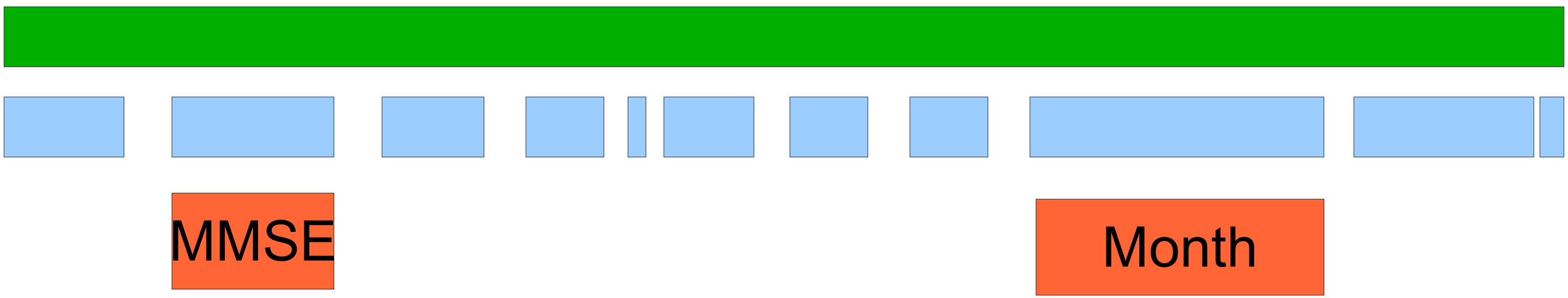
Dictionary lookup

`His MMSE was 23/30 on 15 January 2008.`



Dictionary lookup

```
His MMSE was 23/30 on 15 January 2008.
```



Limitations of dictionary lookup

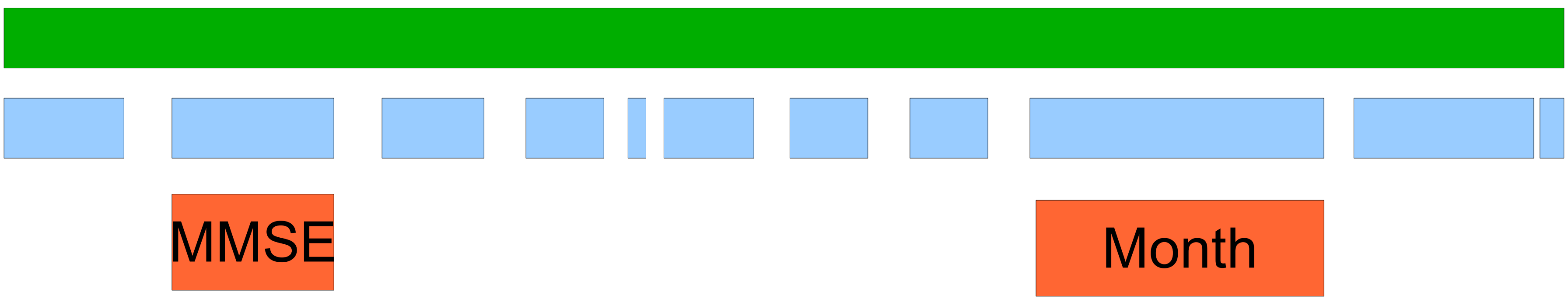
- Dictionary lookup is designed for finding simple, regular terms and features
- False positives
  - *"He may get better"*
  - *"Mother is a smoker"*
  - *"He often burns the toast, setting off the smoke alarm"*
- Cannot deal with complex patterns
  - For example, recognising e-mail addresses using just a dictionary would be impossible
- Cannot deal with ambiguity
  - I for Iodine, or I for me?

Pattern matching languages

- The early parts of our process produce simple annotations
  - Token, Sentence, Dictionary lookups
- These annotations have attributes
  - Token kind, part of speech …
- Patterns of annotations and attributes can suggest more complex information
- We can write rules in some formal language to match these patterns
  - regular expressions – common in all programming languages, e.g. Python
  - JAPE, the pattern matching language used by the GATE NLP framework

Pattern matching

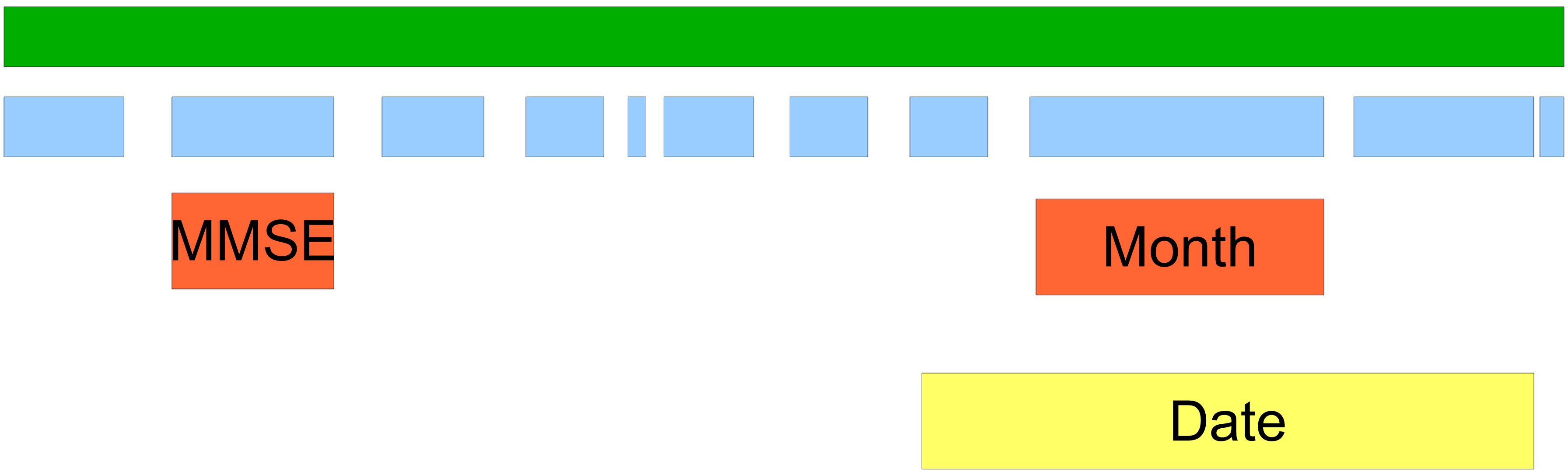
```
His MMSE was 23/30 on 15 January 2008.
```



Date = {number}{Month}{number}

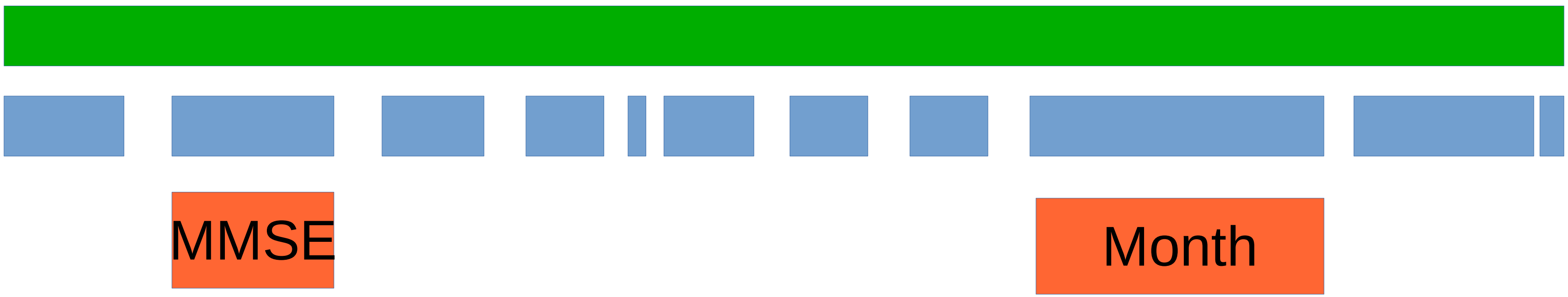
Pattern matching

`His MMSE was 23/30 on 15 January 2008.`



Pattern matching

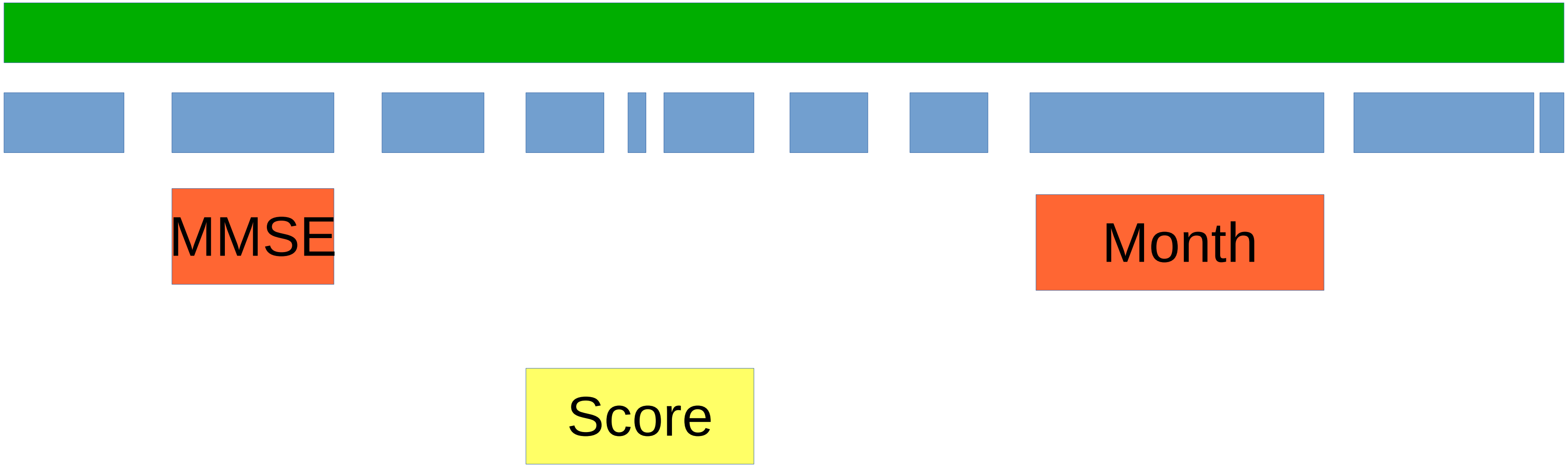
`His MMSE was 23/30 on 15 January 2008.`



Score = {number}{slash}{number}

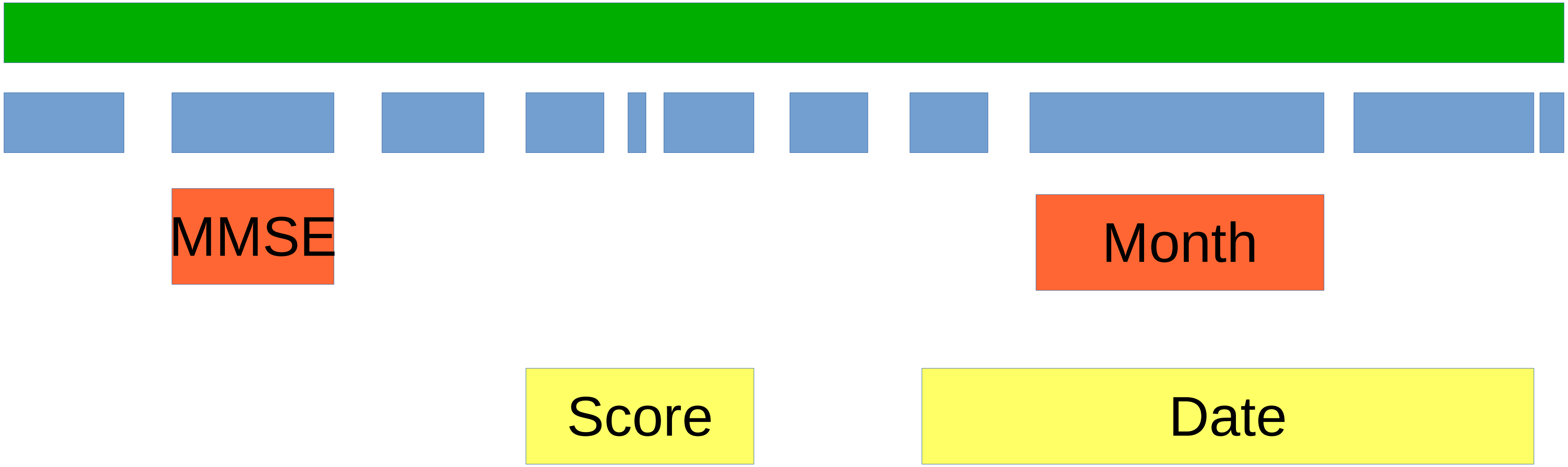
Pattern matching

```
His MMSE was 23/30 on 15 January 2008.
```



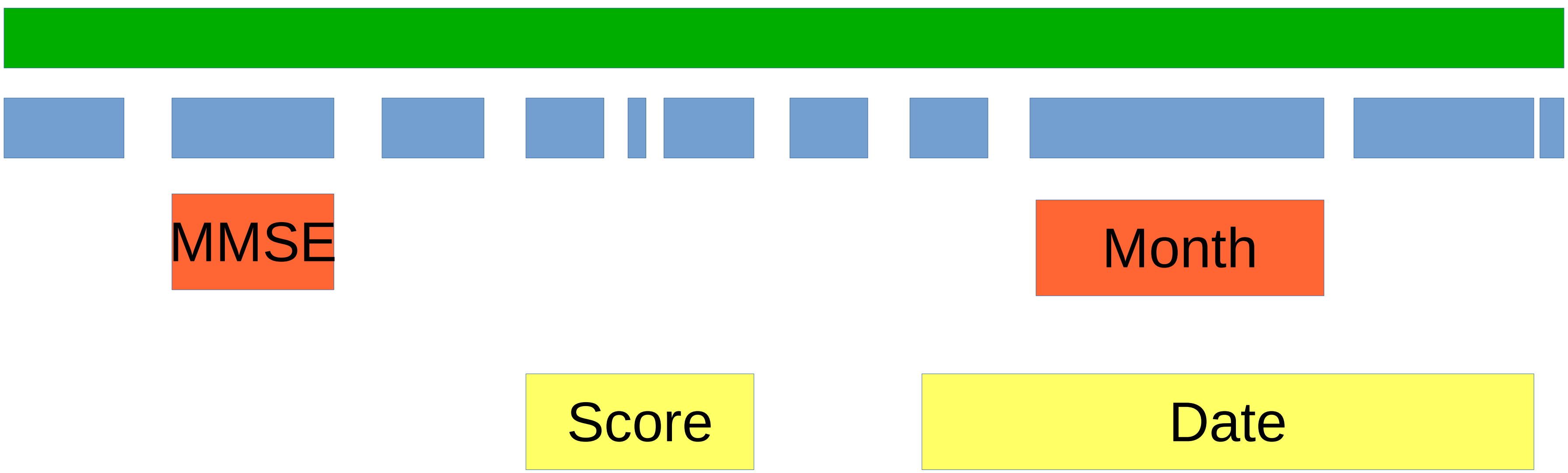
Pattern matching

`His MMSE was 23/30 on 15 January 2008.`



Pattern matching

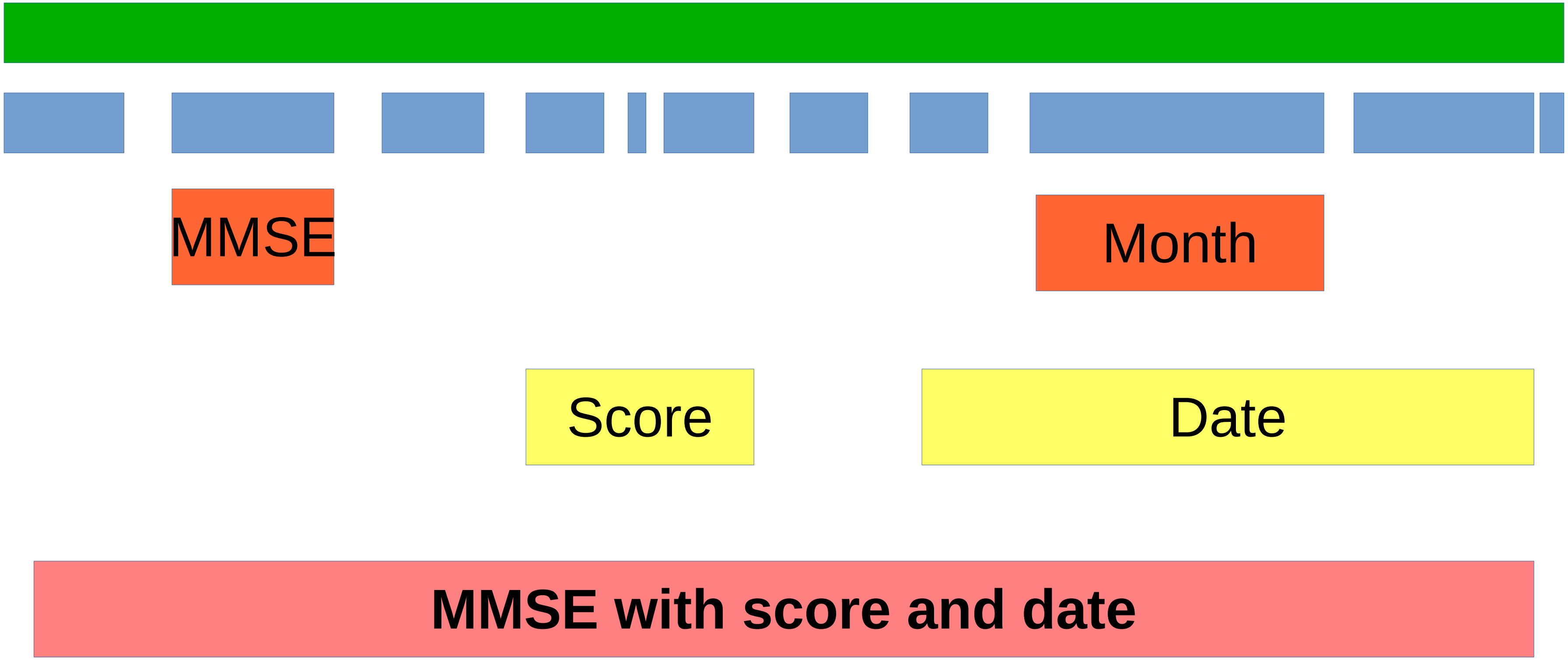
`His MMSE was 23/30 on 15 January 2008.`



MMSE = {pos=PP}{MMSE}{lemma=be}{Score}{Date}

Pattern matching

`His MMSE was 23/30 on 15 January 2008.`



Generalisability

- His MMSE was 23/30 on 15 January 2009
- Her Mini mental was 25/30 on 12/08/07
- His MMS was 25/30 last week
- Her MMSE is 25/30 today
- With adaptation
  - MMSE 25 out of 30
  - Long range dependencies on dates

Empirical methods

Rationalism vs empiricism

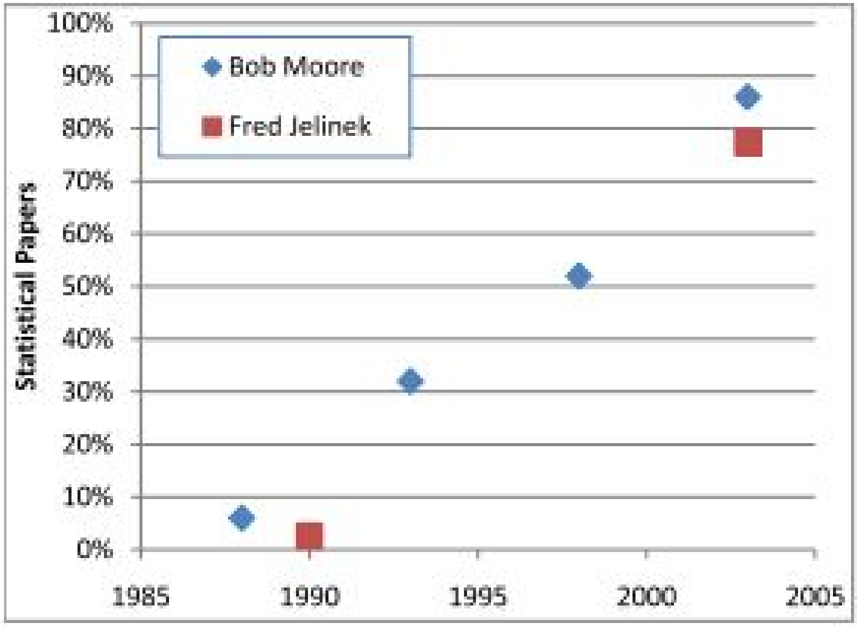


FIGURE 1 The shift from Rationalism to Empiricism is striking (and no longer controversial). This plot is based on two independent surveys of ACL meetings by Bob Moore and Fred Jelinek (personal communication).

(Church, LiLT Volume 2, Issue 4 May 2007)

Learning from examples

Example sentence				
He smokes				
Suffers from anhedonia				
She does not smoke				
20 cigarettes a day				
Blood pressure 70/120				

Learning from examples

Example sentence	Label			
He smokes	T			
Suffers from anhedonia	F			
She does not smoke	F			
20 cigarettes a day	T			
Blood pressure 70/120	F			

Learning from examples

Example sentence	Label	Features		
He smokes	T			
Suffers from anhedonia	F			
She does not smoke	F			
20 cigarettes a day	T			
Blood pressure 70/120	F			

Learning from examples

Example sentence	Label	Features		
		smok*		
He smokes	T	1		
Suffers from anhedonia	F	0		
She does not smoke	F	1		
20 cigarettes a day	T	0		
Blood pressure 70/120	F	0		

Learning from examples

Example sentence	Label	Features		
		smok*	cig*	
He smokes	T	1	0	
Suffers from anhedonia	F	0	0	
She does not smoke	F	1	0	
20 cigarettes a day	T	0	1	
Blood pressure 70/120	F	0	0	

Learning from examples

Example sentence	Label	Features		
		smok*	cig*	negation
He smokes	T	1	0	0
Suffers from anhedonia	F	0	0	0
She does not smoke	F	1	0	1
20 cigarettes a day	T	0	1	0
Blood pressure 70/120	F	0	0	0

Learning from examples

Example sentence	Label	Features		
		smok*	cig*	negation
He smokes	T	1	0	0
Suffers from anhedonia	F	0	0	0
She does not smoke	F	1	0	1
20 cigarettes a day	T	0	1	0
Blood pressure 70/120	F	0	0	0

- We could learn a model that describes a sentence as true if:
  - Sentence contains smok* OR cig*
  - AND sentence does not contain any negation

Learning from examples

Example sentence	Label	Features		
		smok*	cig*	negation
He smokes	T	1	0	0
Suffers from anhedonia	F	0	0	0
She does not smoke	F	1	0	1
20 cigarettes a day	T	0	1	0
Blood pressure 70/120	F	0	0	0

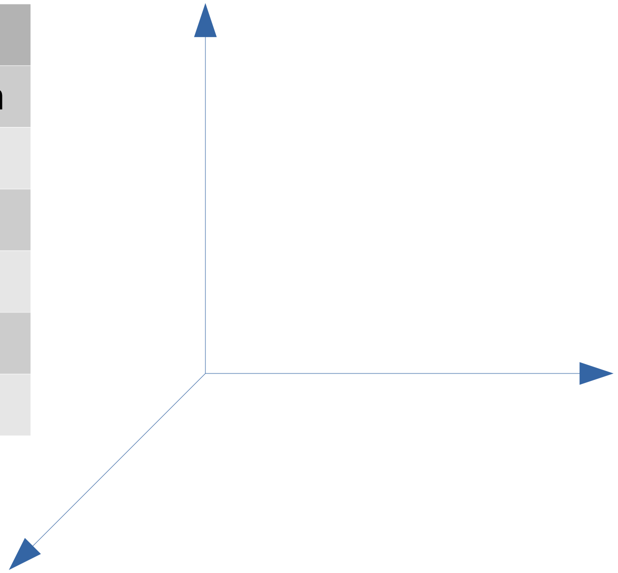
- Describes a feature space within which our examples can be represented

Learning from examples

Example sentence	Label	Features		
		smok*	cig*	negation
He smokes	T	1	0	0
Suffers from anhedonia	F	0	0	0
She does not smoke	F	1	0	1
20 cigarettes a day	T	0	1	0
Blood pressure 70/120	F	0	0	0

Learning from examples

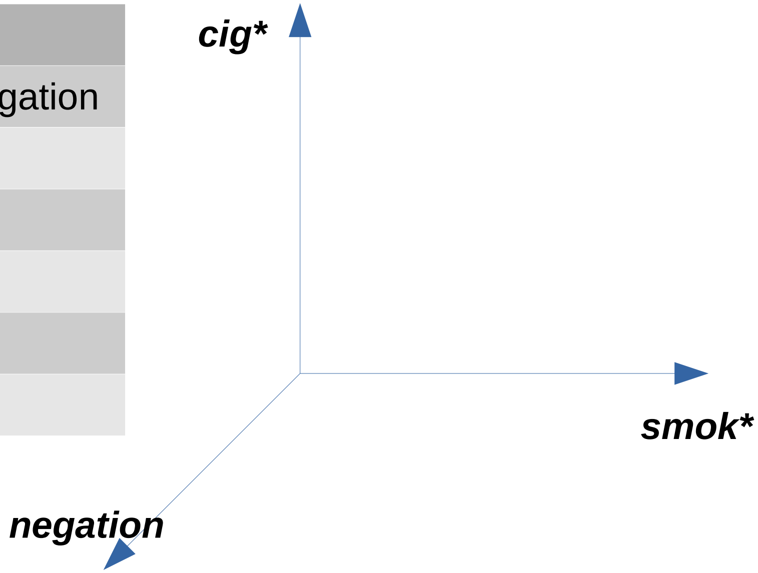
Example sentence	Label	Features		
		smok*	cig*	negation
He smokes	T	1	0	0
Suffers from anhedonia	F	0	0	0
She does not smoke	F	1	0	1
20 cigarettes a day	T	0	1	0
Blood pressure 70/120	F	0	0	0



- We model one dimension for each feature

Learning from examples

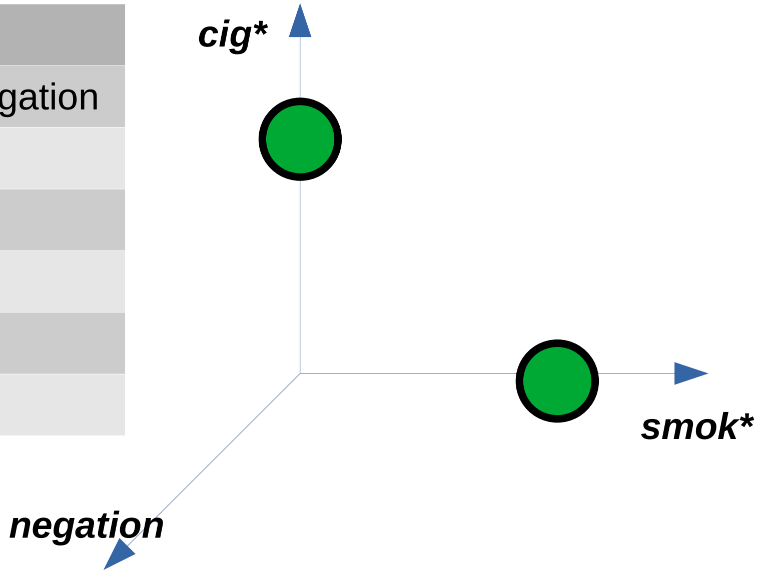
Example sentence	Label	Features		
		smok*	cig*	negation
He smokes	T	1	0	0
Suffers from anhedonia	F	0	0	0
She does not smoke	F	1	0	1
20 cigarettes a day	T	0	1	0
Blood pressure 70/120	F	0	0	0



- We model one dimension for each feature

Learning from examples

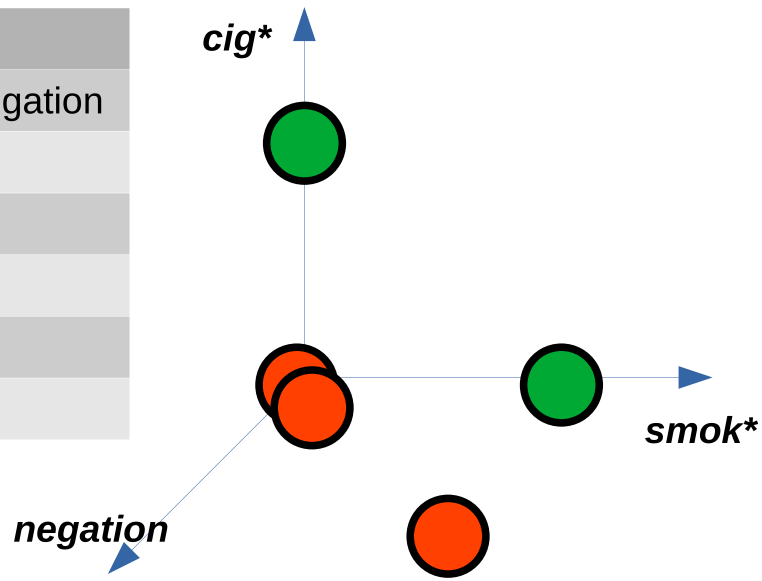
Example sentence	Label	Features		
		smok*	cig*	negation
He smokes	T	1	0	0
Suffers from anhedonia	F	0	0	0
She does not smoke	F	1	0	1
20 cigarettes a day	T	0	1	0
Blood pressure 70/120	F	0	0	0



- Our examples can be represented in this feature space

Learning from examples

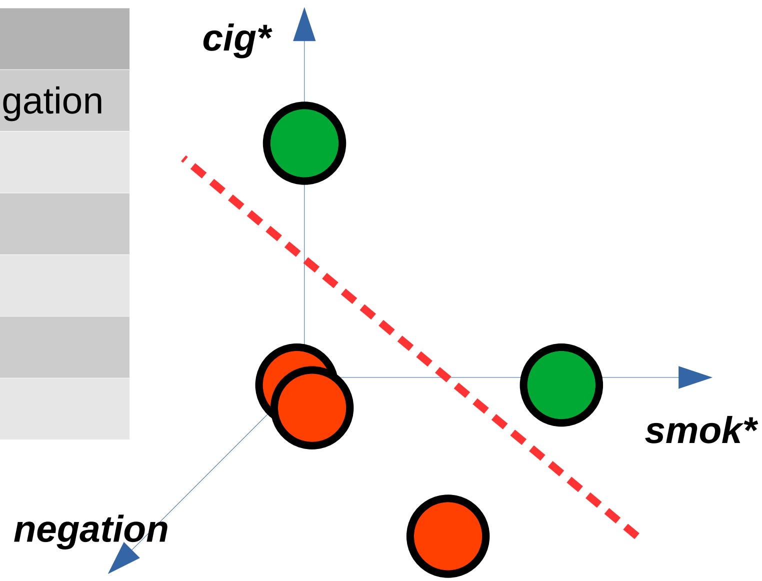
Example sentence	Label	Features		
		smok*	cig*	negation
He smokes	T	1	0	0
Suffers from anhedonia	F	0	0	0
She does not smoke	F	1	0	1
20 cigarettes a day	T	0	1	0
Blood pressure 70/120	F	0	0	0



- Our examples can be represented in this feature space

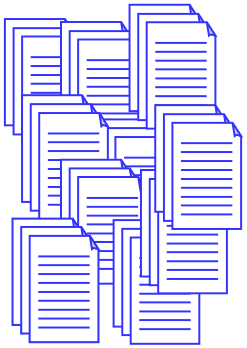
Learning from examples

Example sentence	Label	Features		
		smok*	cig*	negation
He smokes	T	1	0	0
Suffers from anhedonia	F	0	0	0
She does not smoke	F	1	0	1
20 cigarettes a day	T	0	1	0
Blood pressure 70/120	F	0	0	0



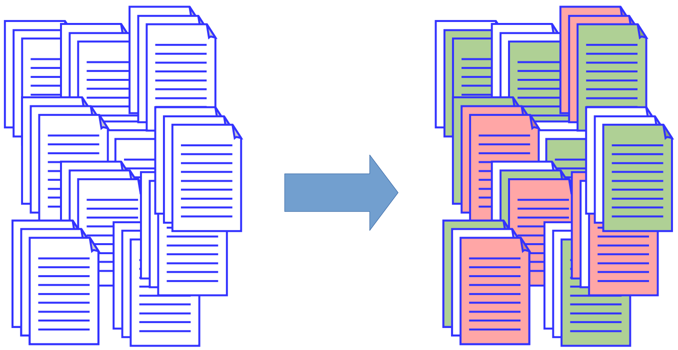
- Compute some line or plane separating our positive and negative examples
- This is *supervised classification*

Supervised classification



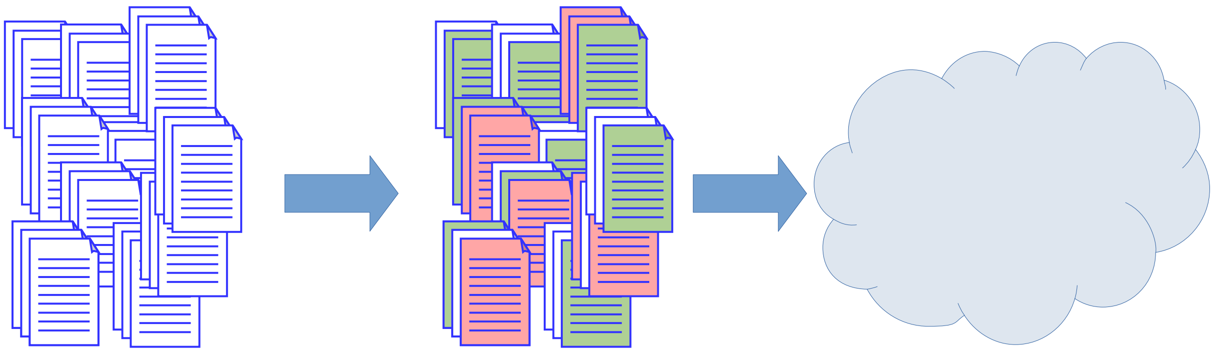
- Take a set of example texts.
- They might be sentences, whole documents, single words, or some other portion of text.
- This is our training corpus.

Supervised classification



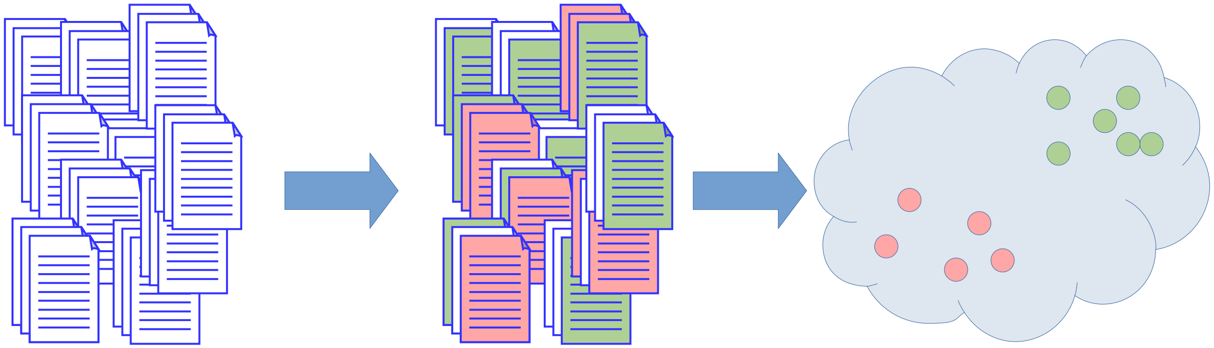
- Label each example, with the classes in our problem.
- Labelling will often be done by human.
- We might be lucky enough to have some existing labelled data, e.g. radiology reports with a code for tumour class attached..

Supervised classification



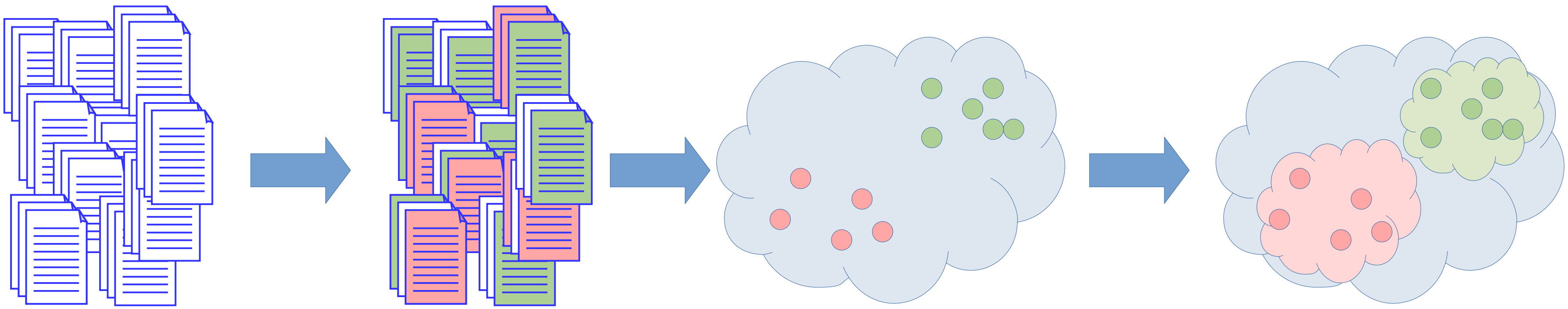
- Select features to represent our texts.
- These might be the presence of words, POS tags, distances between words, word sequences (ngrams), presence of word groups, sentence lengths, etc.
- We may use numeric representations of words as features, computed in a separate step. In the state of the art,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

Supervised classification



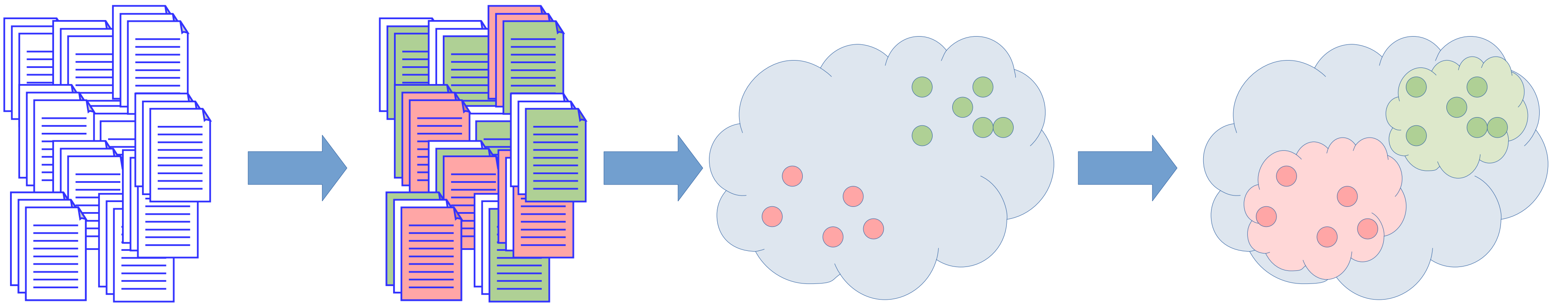
- Represent the texts in this feature space.

Supervised classification



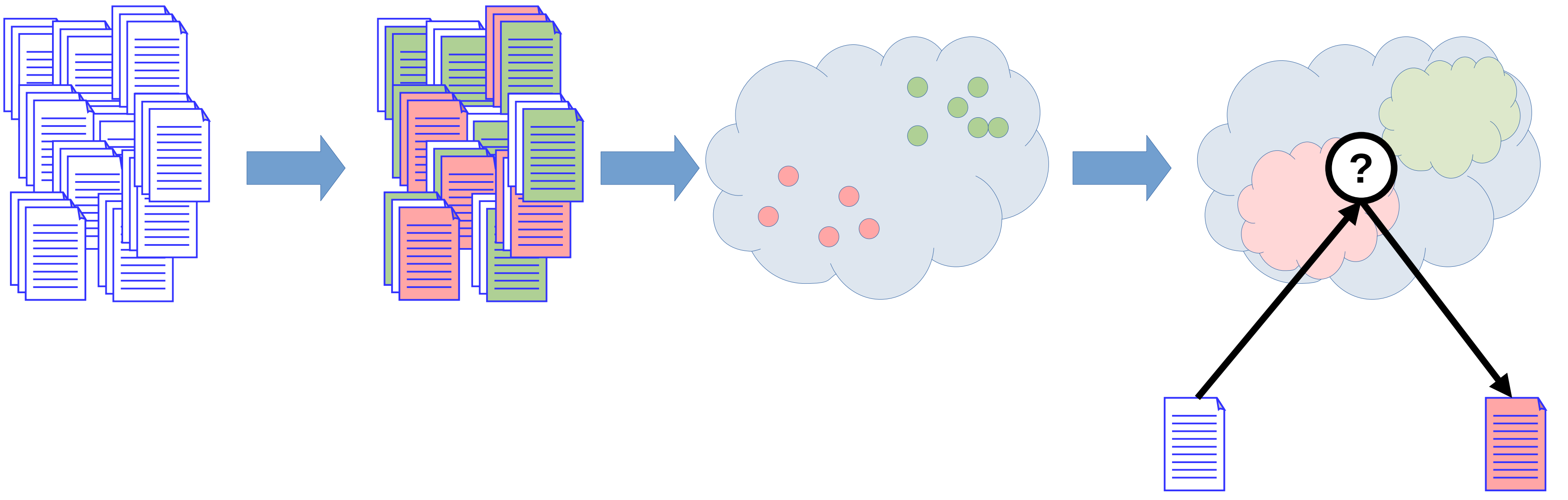
- Compute some separator between classes.
- This will involve measures of distance between points.
- It might also involve methods for projecting multiple dimensions in to different spaces in which they are separable (kernels).

Supervised classification



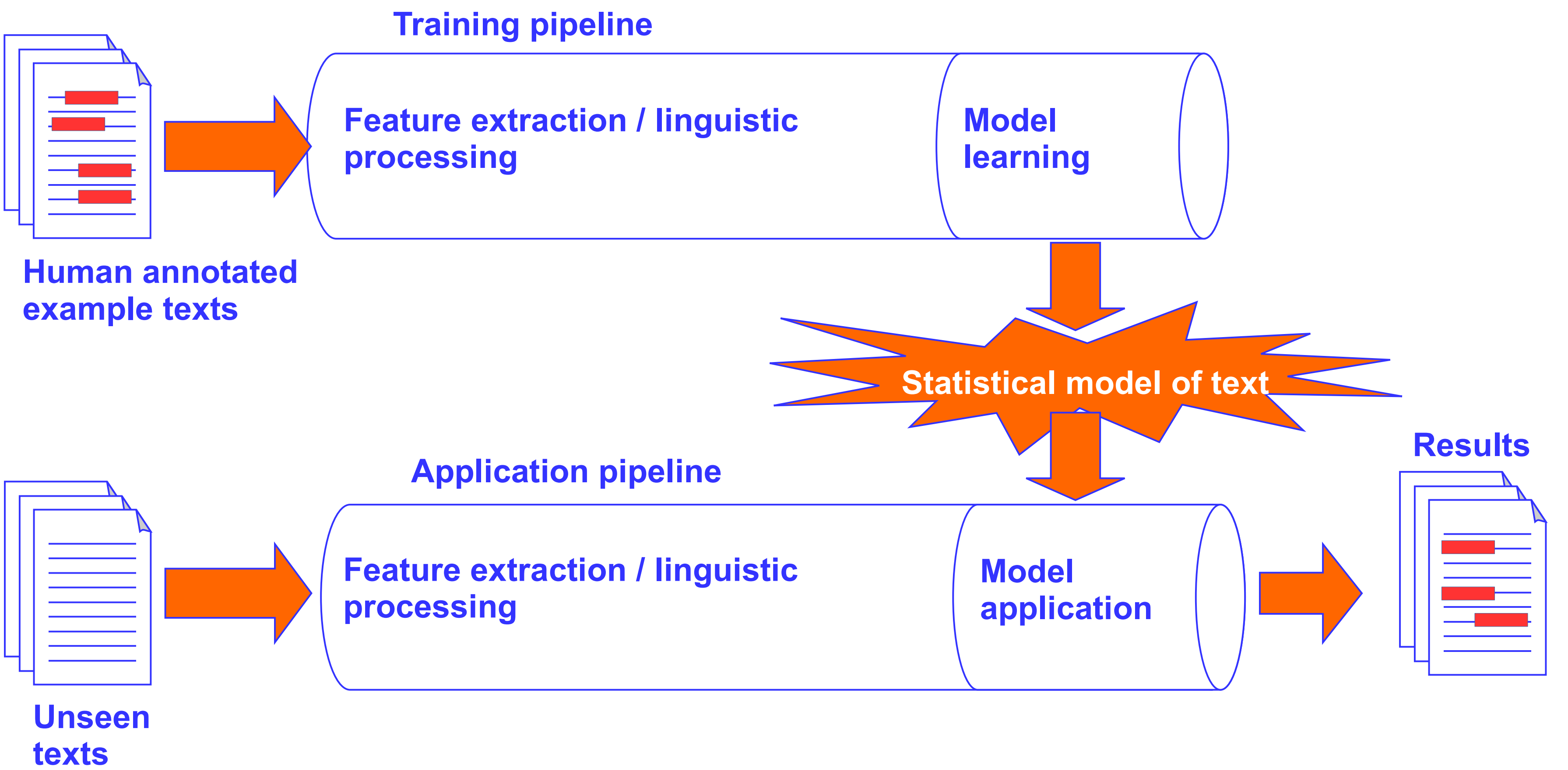
- Commonly used classification algorithms in NLP:
  - SVM (very popular)
  - CRF
  - Naive Bayes
  - KNN
  - Random Forest
  - State of the art: neural nets, e.g. CNNs, LSTMs

Supervised classification

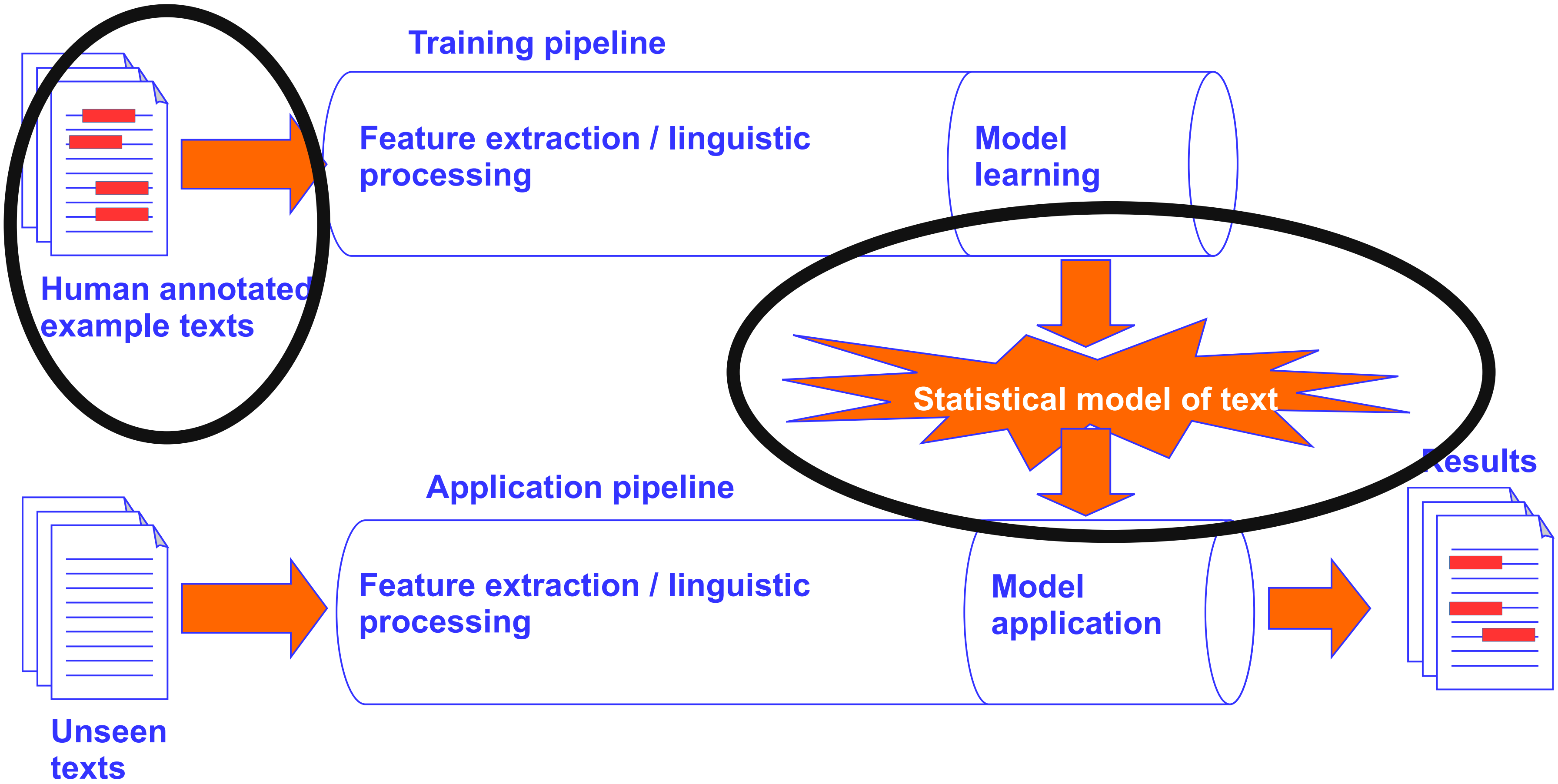


- Classify / label new, previously unseen examples by representing them in the same feature space.

Supervised classification - summary



Supervised classification - summary

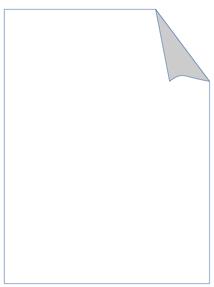


Supervised classification - problems

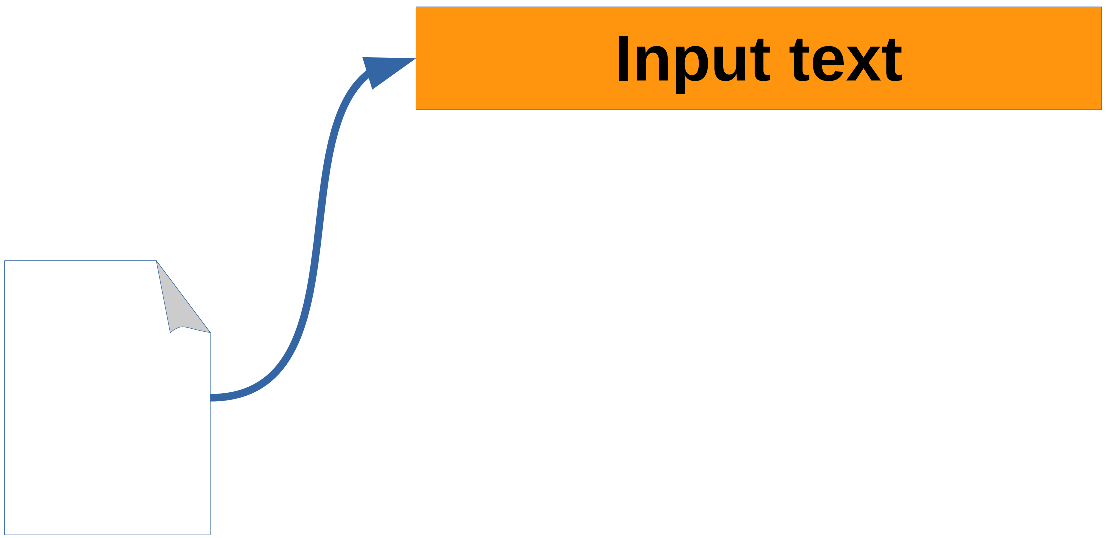
- The quality of our model depends on:
  - Number of labelled examples
  - Number of feature dimensions
- A large number of dimensions can lead to the curse of dimensionality
  - Our training data is sparse within the space
  - It is hard to find commonalities
  - Results are unreliable
- A small number of examples can lead to overfitting
  - There is insufficient variation in our data
  - It is not representative
  - Our model fits the training data very well, but is unreliable when confronted with the greater variation in real data

NLP as a processing pipeline

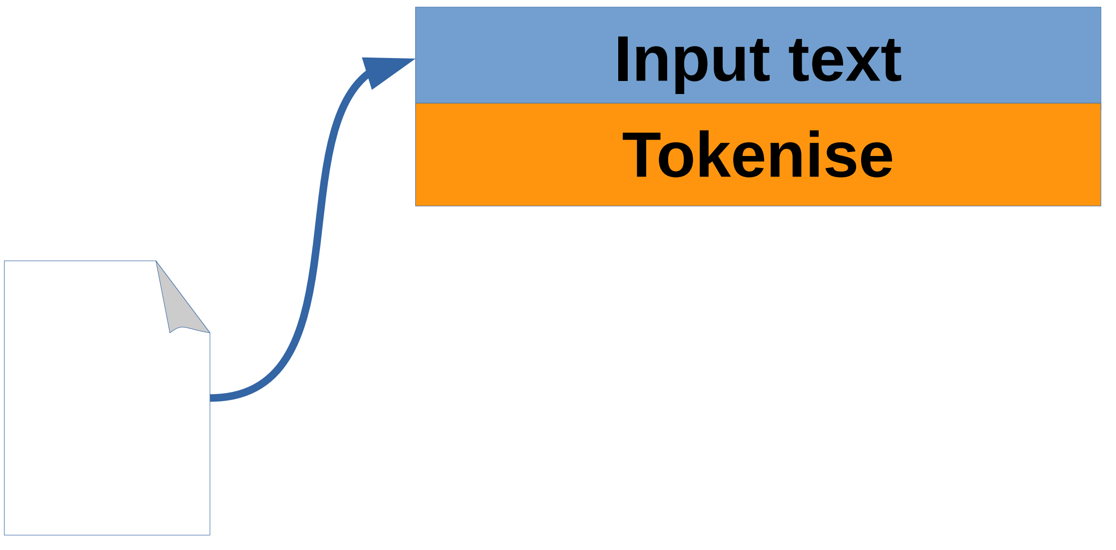
NLP as a pipeline of processing steps



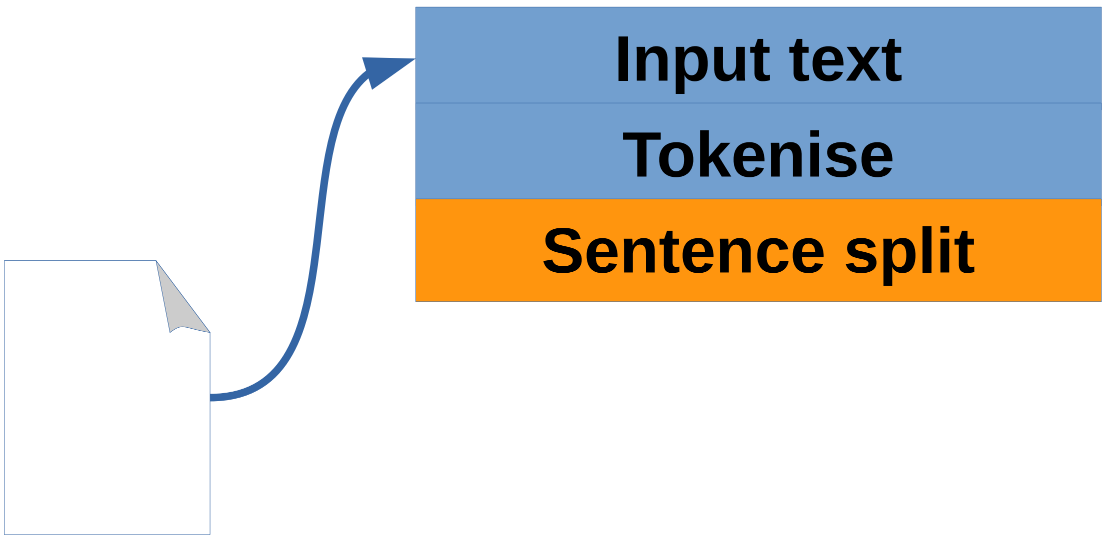
NLP as a pipeline of processing steps



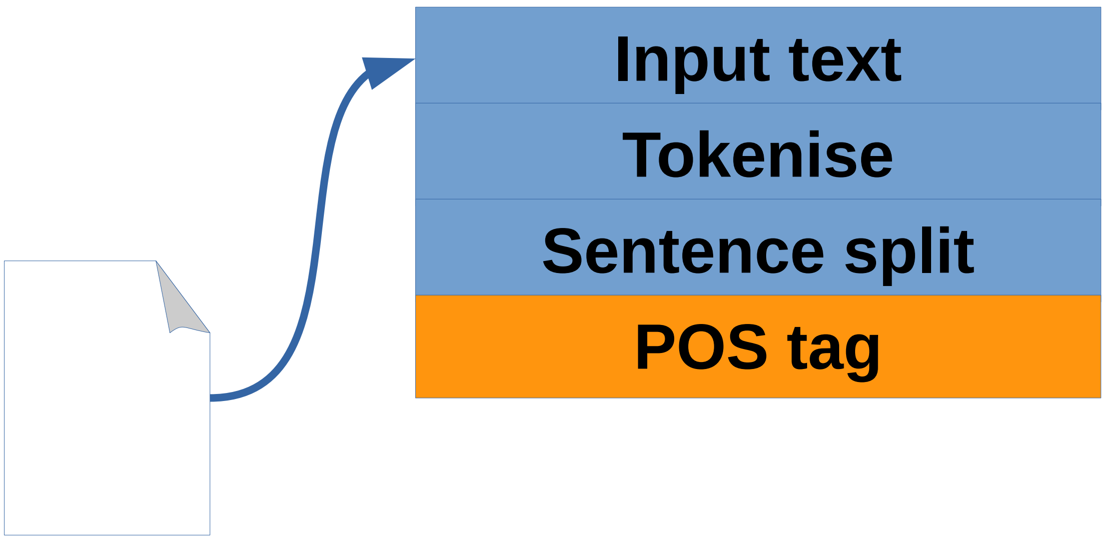
NLP as a pipeline of processing steps



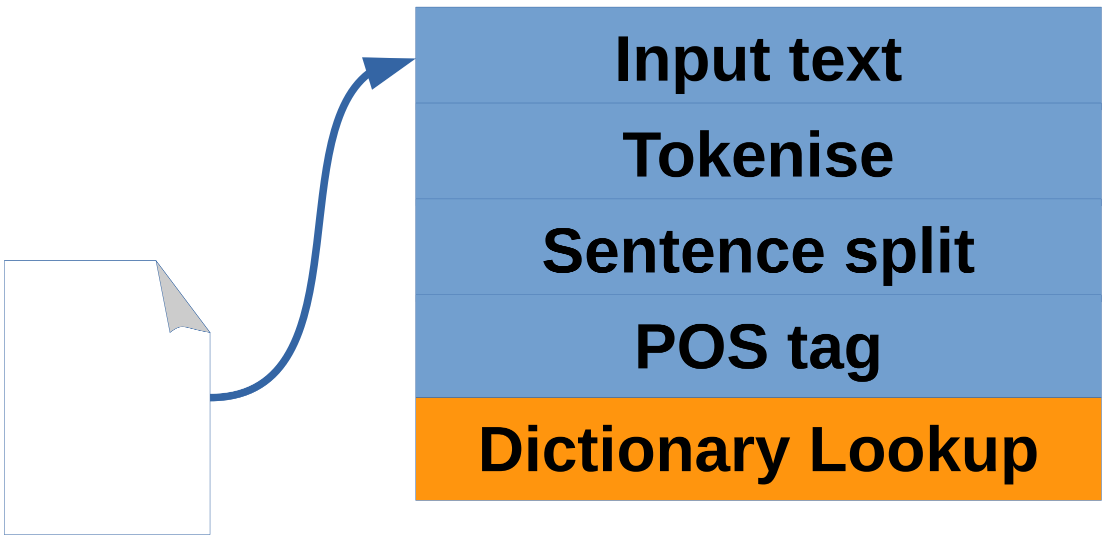
NLP as a pipeline of processing steps



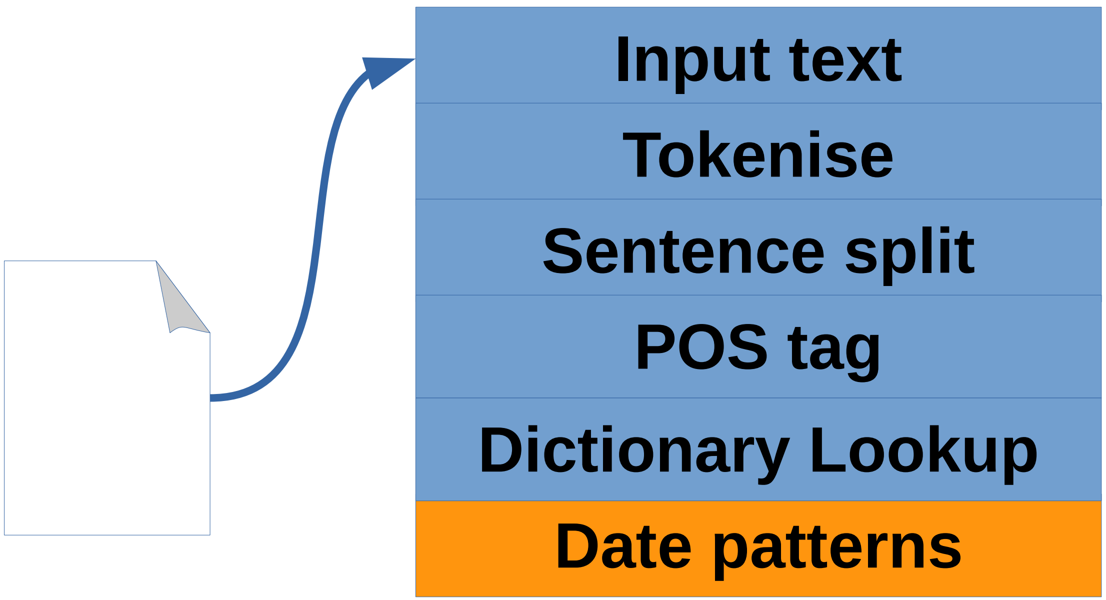
NLP as a pipeline of processing steps



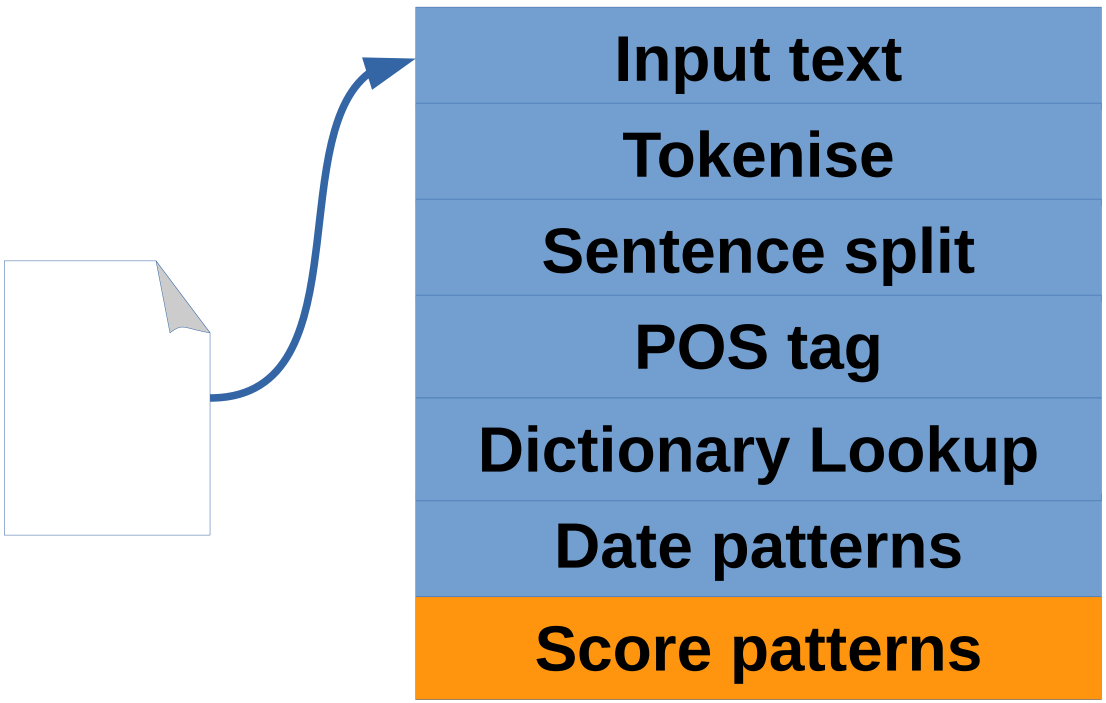
NLP as a pipeline of processing steps



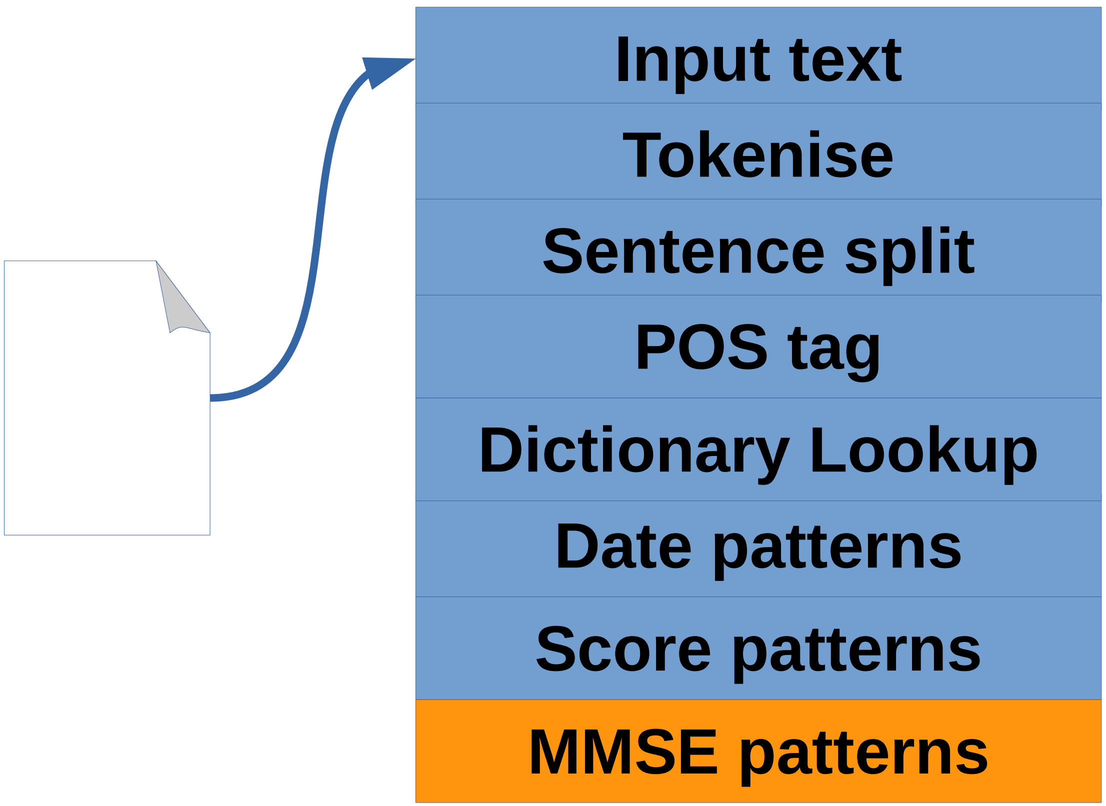
NLP as a pipeline of processing steps



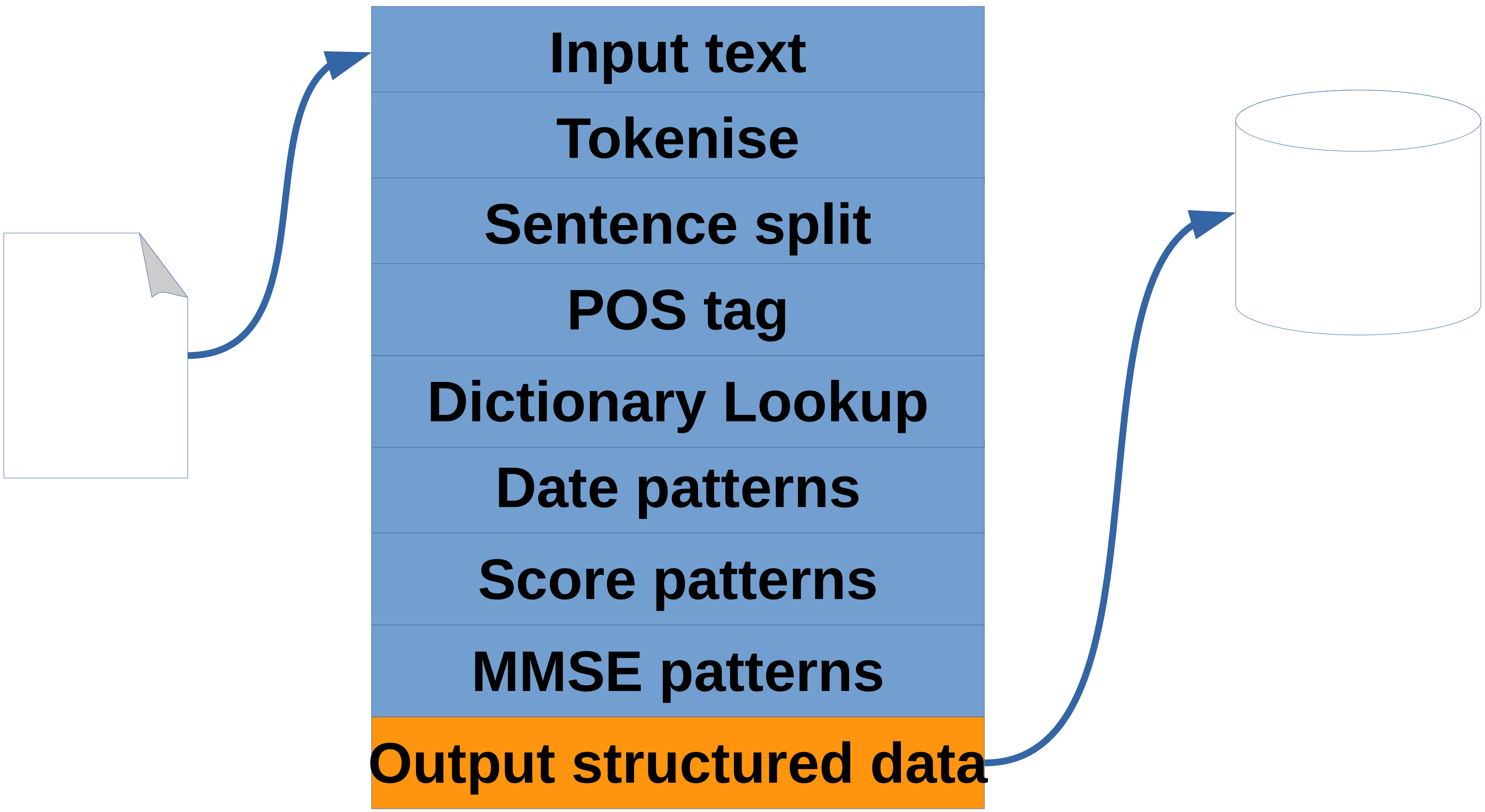
NLP as a pipeline of processing steps



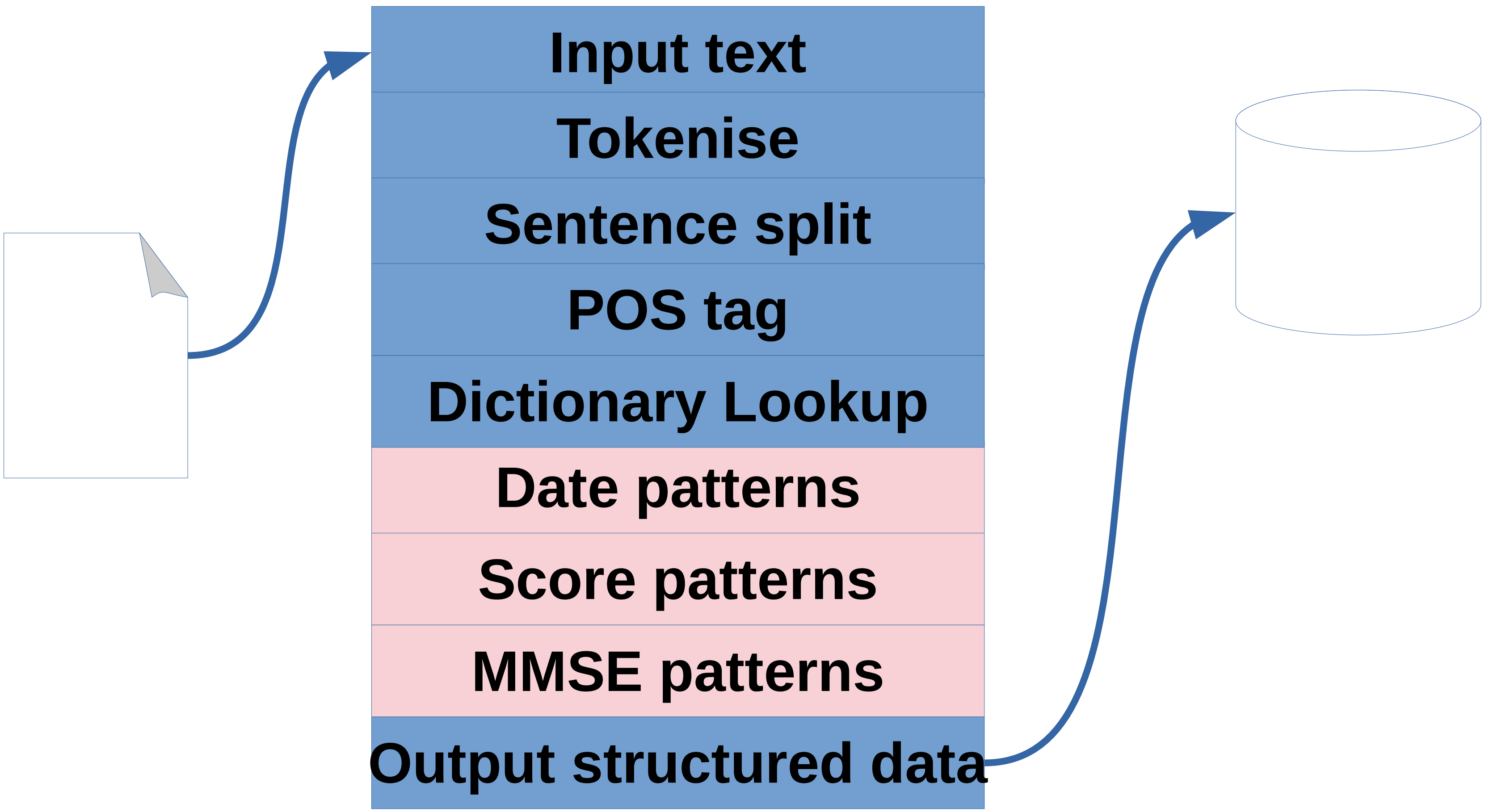
NLP as a pipeline of processing steps



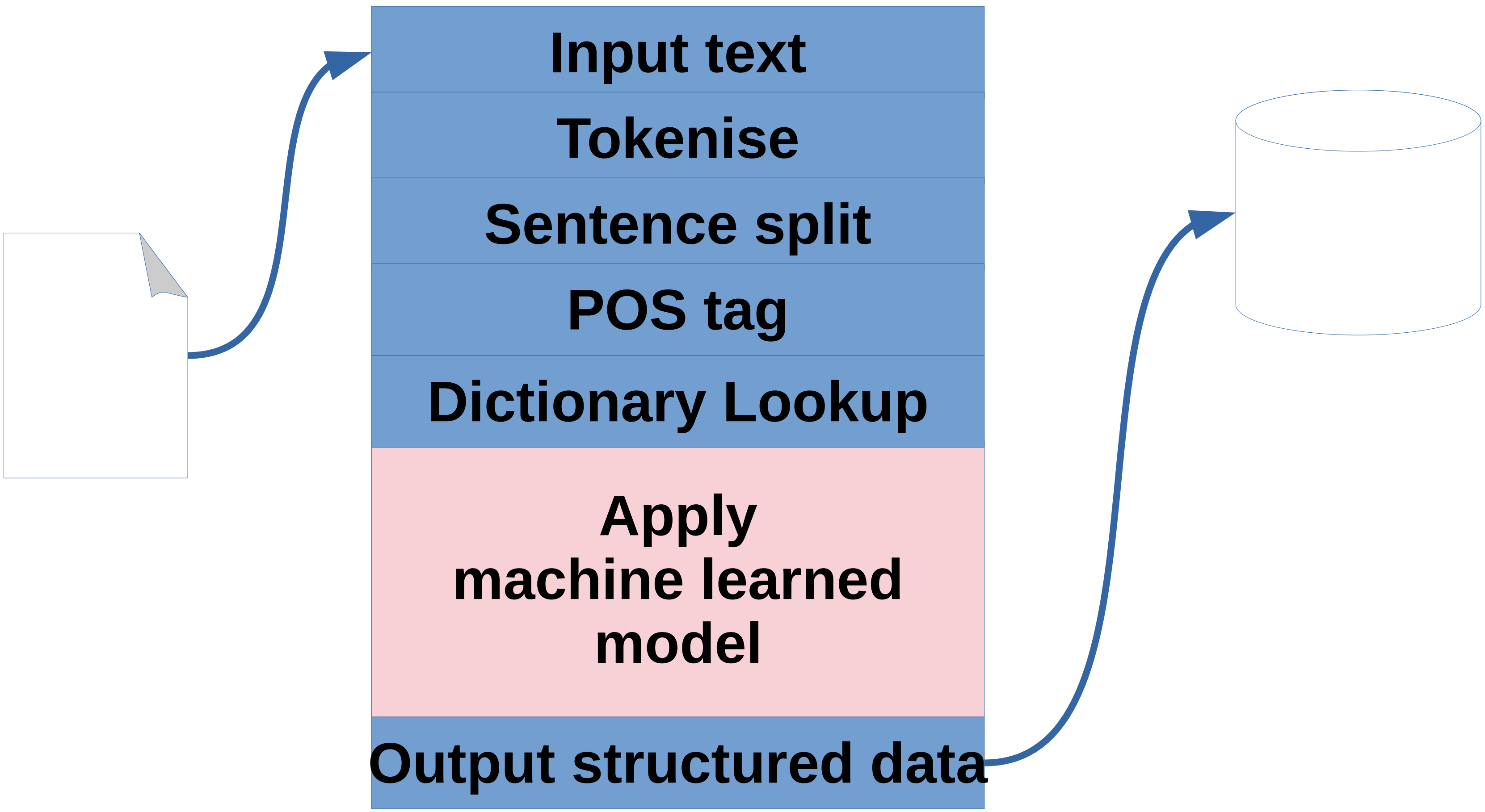
NLP as a pipeline of processing steps



NLP as a pipeline of processing steps



NLP as a pipeline of processing steps



NIHR | Maudsley Biomedical Research Centre

Comparing rule based NLP and machine learning

A pragmatic, hybrid approach

- Often, rule based methods and empirical / machine learning are combined
- We might use approximate dictionary lookup to create features or filter candidates for a machine learning application
- Or, we might use a machine learning based POS tagger in a rule based application
- The problem might suggest an obvious approach
  - e.g. repetitious text such as numeric scores or measurements might suggest pattern matching
  - e.g. lots of cheaply labelled data might suggest a supervised learning solution

Skills and trade offs – rule based NLP

- Need domain expertise to provide examples
- Small numbers examples can be augmented with human intuition
- Need skilled language engineer to craft the rules
- Development can be very time consuming
- Can give good accuracy for simple problems
- Hard to generalise and move to slightly different domains
- Transparent – rules can be explained to other people

Skills and trade offs – empirical / machine learning

- Need domain expertise to provide examples – often large numbers
- Developers do not need language engineering skills
- Feature engineering – may require experimentation
- Off the shelf / semi-automated software?
- Easier to generalise final models
- Accuracy depends on complexity of target, amount and quality of training data
- Not always transparent – models difficult to interpret and explain

NIHR | Maudsley Biomedical Research Centre

Thank you.
Any questions?

angus.roberts@kcl.ac.uk