Natural Language Processing for the Semantic Web

Isabelle Augenstein

Department of Computer Science, University of Sheffield, UK

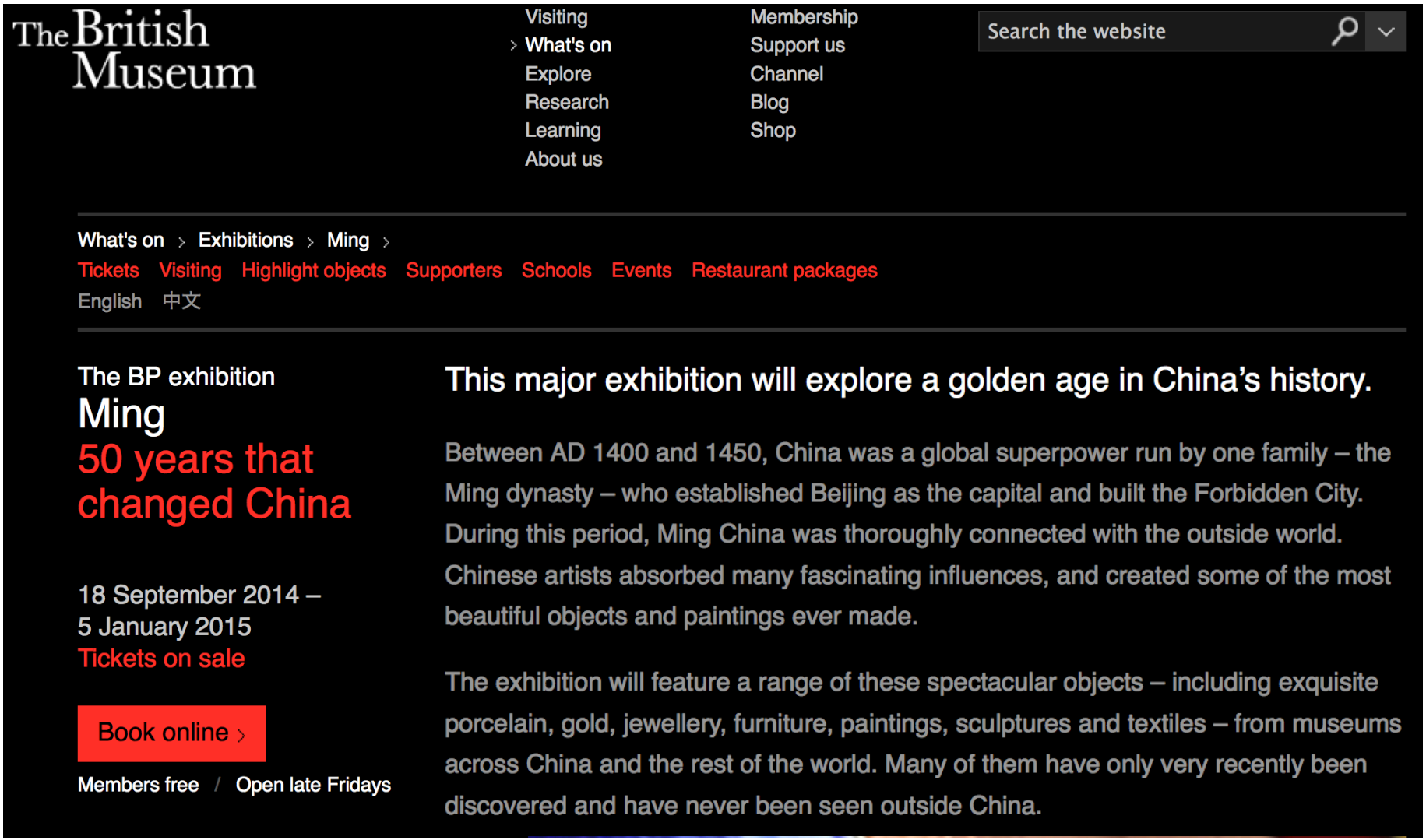
i.augenstein@sheffield.ac.uk

Linked Data for NLP Tutorial, ESWC Summer School 2014

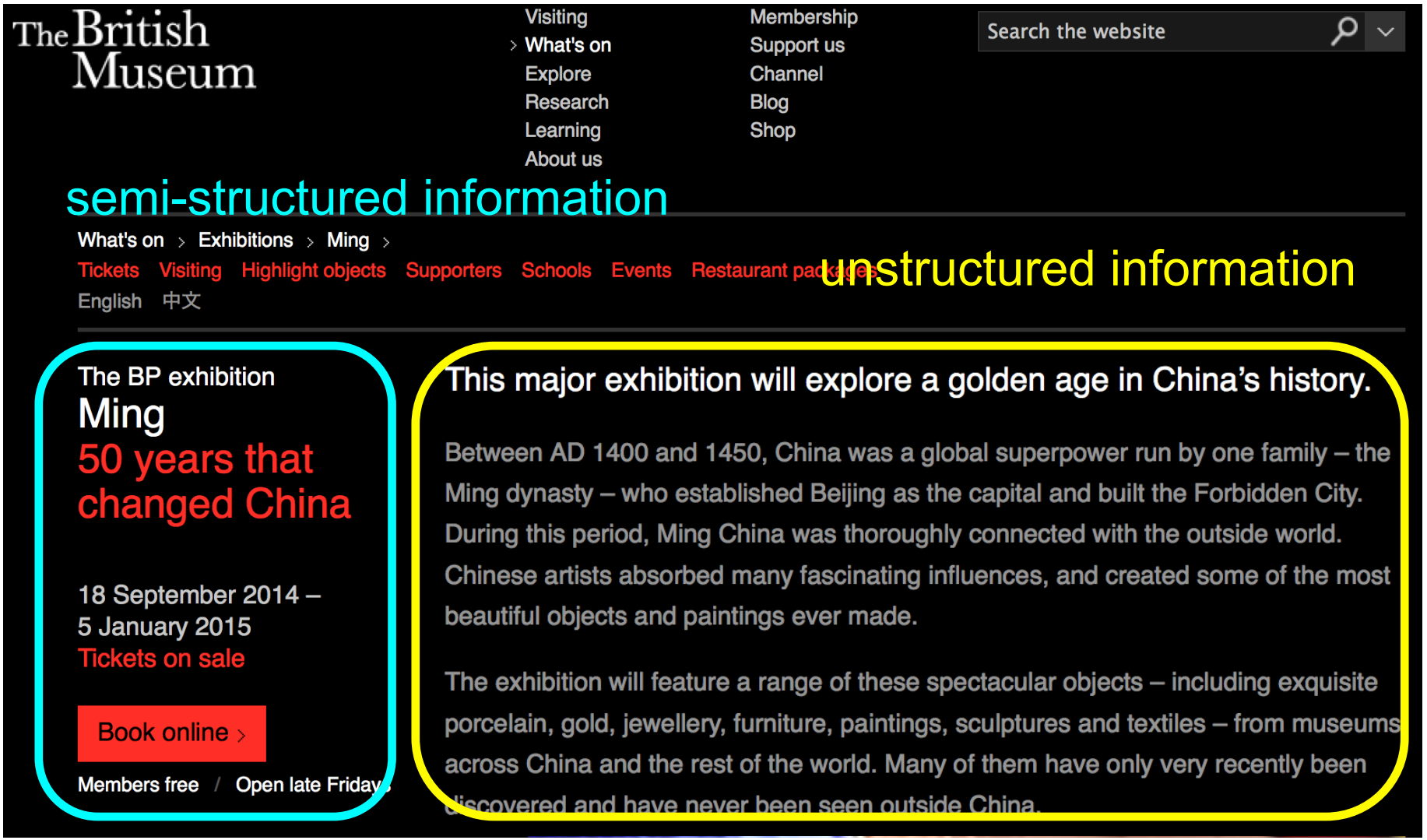
August 24, 2014

The University Of Sheffield.

Why Natural Language Processing?

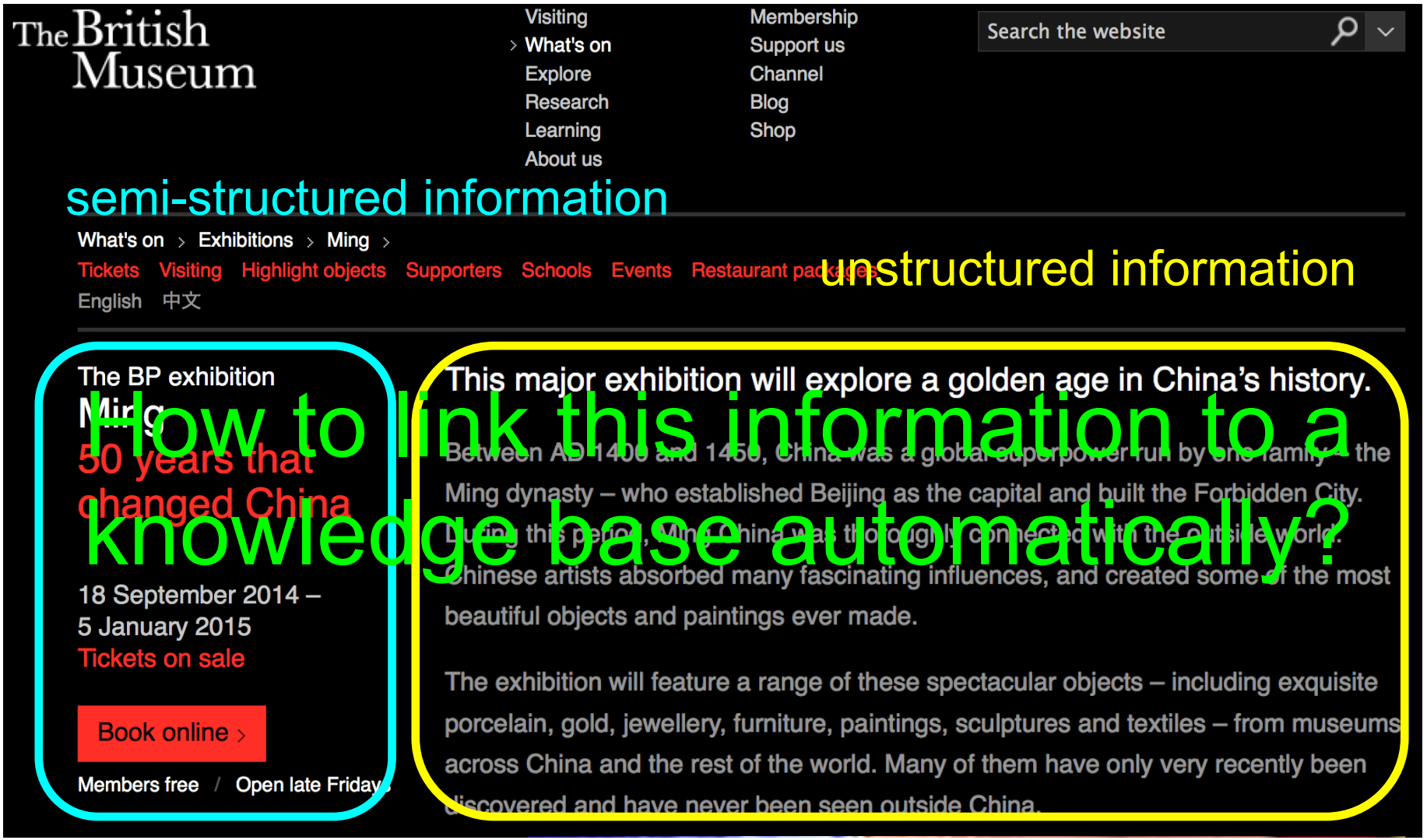


Why Natural Language Processing?

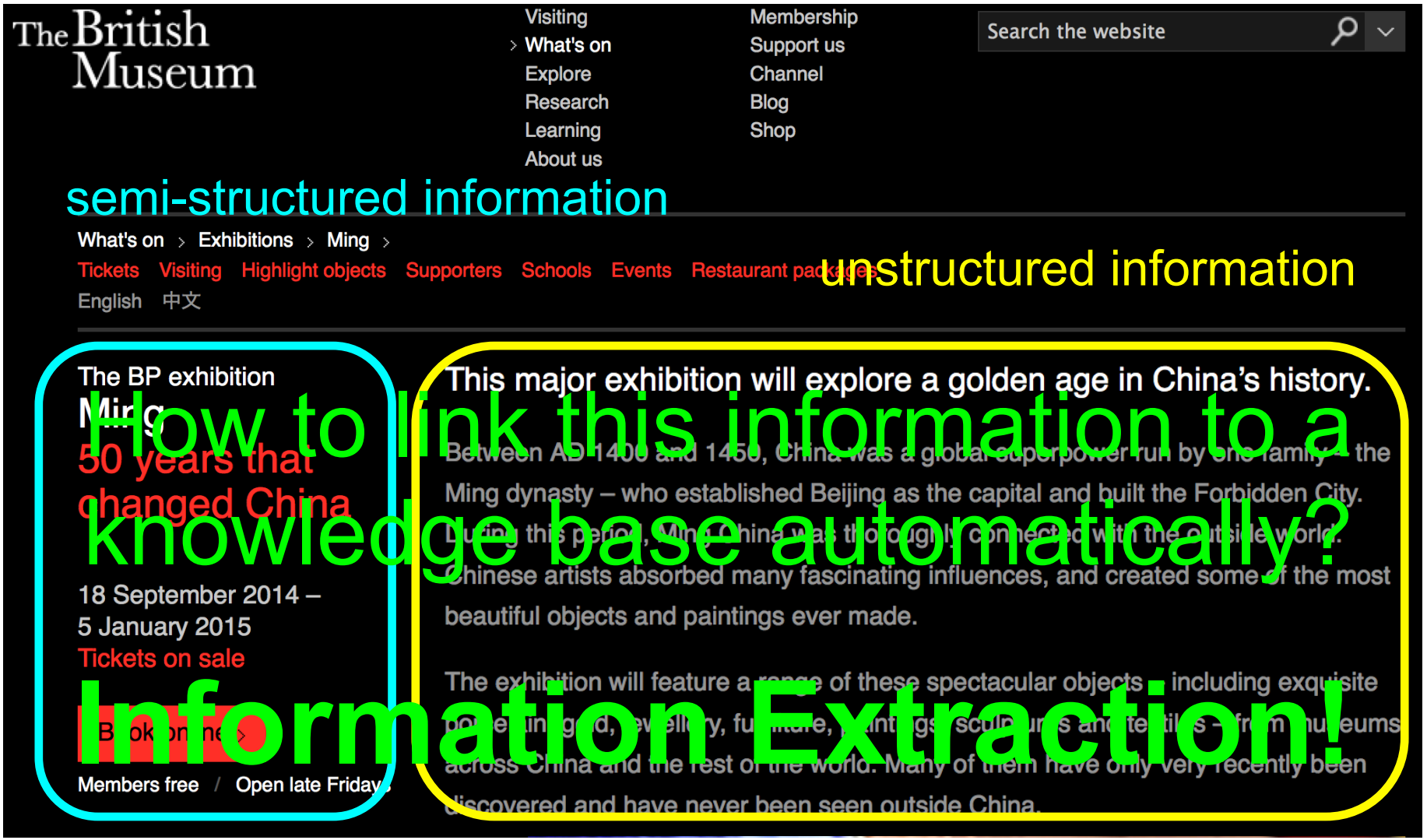


The University Of Sheffield.

Why Natural Language Processing?



Why Natural Language Processing?



Information Extraction

Between AD 1400 and 1450, China was a global superpower run by one family – the Ming dynasty – who established Beijing as the capital and built the Forbidden City.

Named Entities

Between AD 1400 and 1450, China was a global superpower run by one family – the Ming dynasty – who established Beijing as the capital and built the Forbidden City.

Named Entity Recognition

Named Entities

Between AD 1400 and 1450, China was a global superpower run by one family – the Ming dynasty – who established Beijing as the capital and built the Forbidden City.

Named Entity Recognition (NER)

Named Entity Classification (NEC):

China: /location/country
Ming dynasty: /royalty/royal_line
Beijing: /location/city
Forbidden City: /location/city

Named Entities

Between AD 1400 and 1450, China was a global superpower run by one family – the Ming dynasty – who established Beijing as the capital and built the Forbidden City.

Named Entity Recognition

Named Entity Classification:

China: /location/country
Ming dynasty: /royalty/royal_line
Beijing: /location/city
Forbidden City: /location/city

Named Entity Linking:

China: /m/0d05w3
Ming dynasty: /m/0bw_m
Beijing: /m/01914
Forbidden City: /m/0j0b2

The University Of Sheffield.

Named Entities: Definition

Named Entities: Proper nouns, which refer to real-life entities

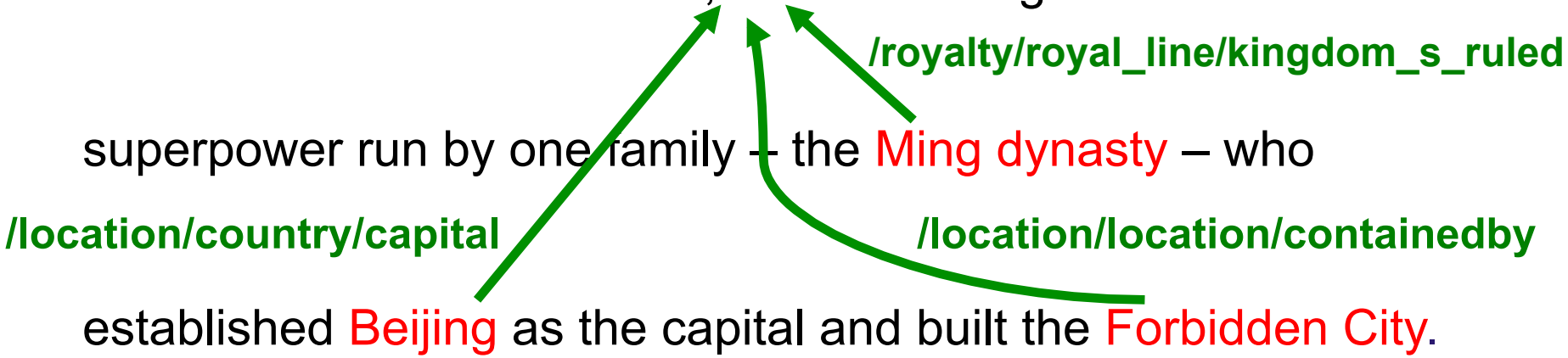
Named Entity Recognition: Detecting boundaries of named entities (NEs)

Named Entity Classification: Assigning classes to NEs, such as PERSON, LOCATION, ORGANISATION, or fine-grained classes such as ROYAL LINE

Named Entity Linking / Disambiguation: Linking NEs to concrete entries in knowledge base, example:

China -> LOCATION: Republic of China, country in East Asia
-> LOCATION: China proper, core region of China during Qing dynasty
-> LOCATION: China, Texas
-> PERSON: China, Brazilian footballer born in 1964
-> MUSIC: China, a 1979 album by Vangelis
-> …

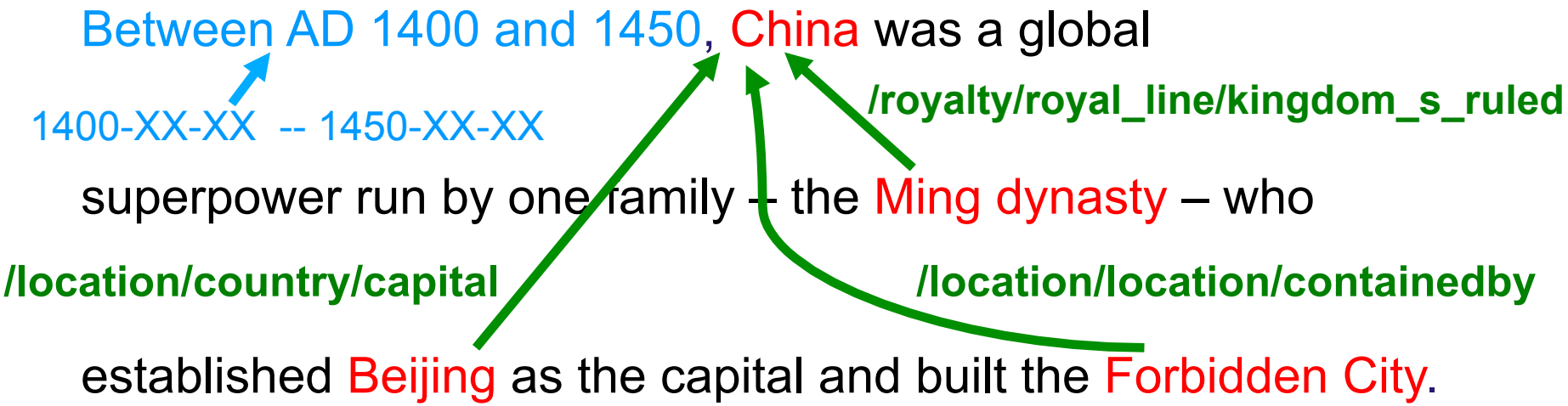
Relations



Named Entity Recognition

Relation Extraction

Relations and Time Expressions

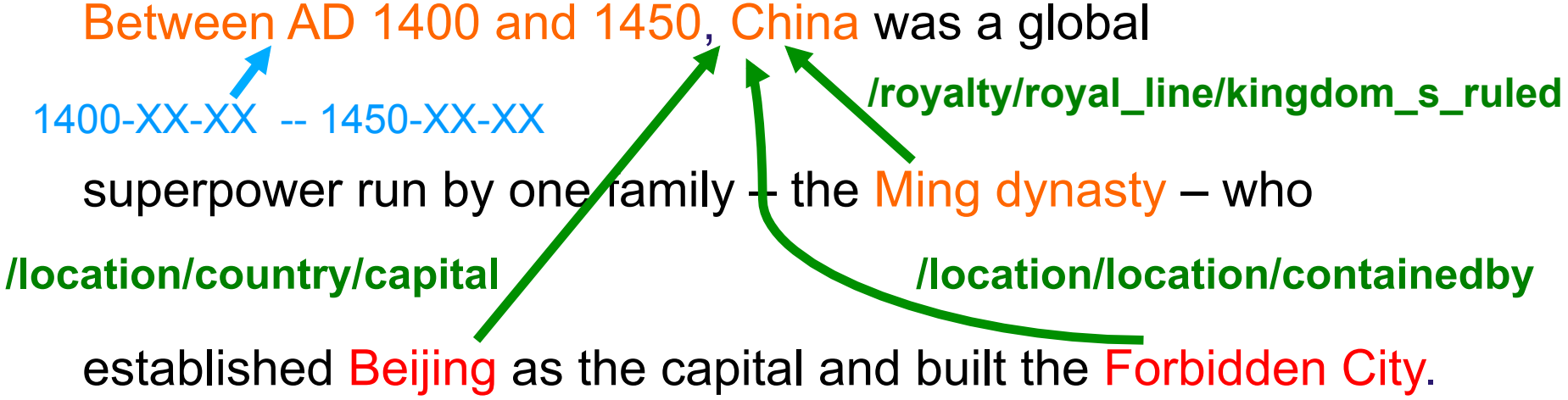


Named Entity Recognition

Relation Extraction

Temporal Extraction

Relations, Time Expressions and Events



Named Entity Recognition

Relation Extraction

Temporal Extraction

Event Extraction

Event: /royalty/royal_line: Ming dynasty

/royalty/royal_line/ruled_from: 1400-XX-XX
/royalty/royal_line/ruled_to: 1450-XX-XX
/royalty/royal_line/kingdom_s_ruled: China

Relations, Time Expressions and Events: Definition

Relations: Two or more entities which relate to one another in real life

Relation Extraction: Detecting relations between entities and assigning relation types to them, such as CAPITAL-OF

Temporal Extraction: Recognising and normalising time expressions: times (e.g. “3 in the afternoon”), dates (“tomorrow”), durations (“since yesterday”), and sets (e.g. “twice a month”)

Events: Real-life events that happened at some point in space and time, e.g. kingdom, assassination, exhibition

Event Extraction: Extracting events consisting of the name and type of event, time and location

Summary: Introduction

- Information extraction (IE) methods such as named entity recognition (NER), named entity classification (NEC), named entity linking, relation extraction (RE), temporal extraction, and event extraction can help to add markup to Web pages
- Information extraction approaches can serve two purposes:
  - Annotating every single mention of an entity, relation or event, e.g. to add markup to Web pages
  - Aggregating those mentions to populate knowledge bases, e.g. based on confidence values and majority voting

    China LOCATION 0.9
    China LOCATION 0.8
    China PERSON 0.4
    → China LOCATION

Information Extraction: Methods

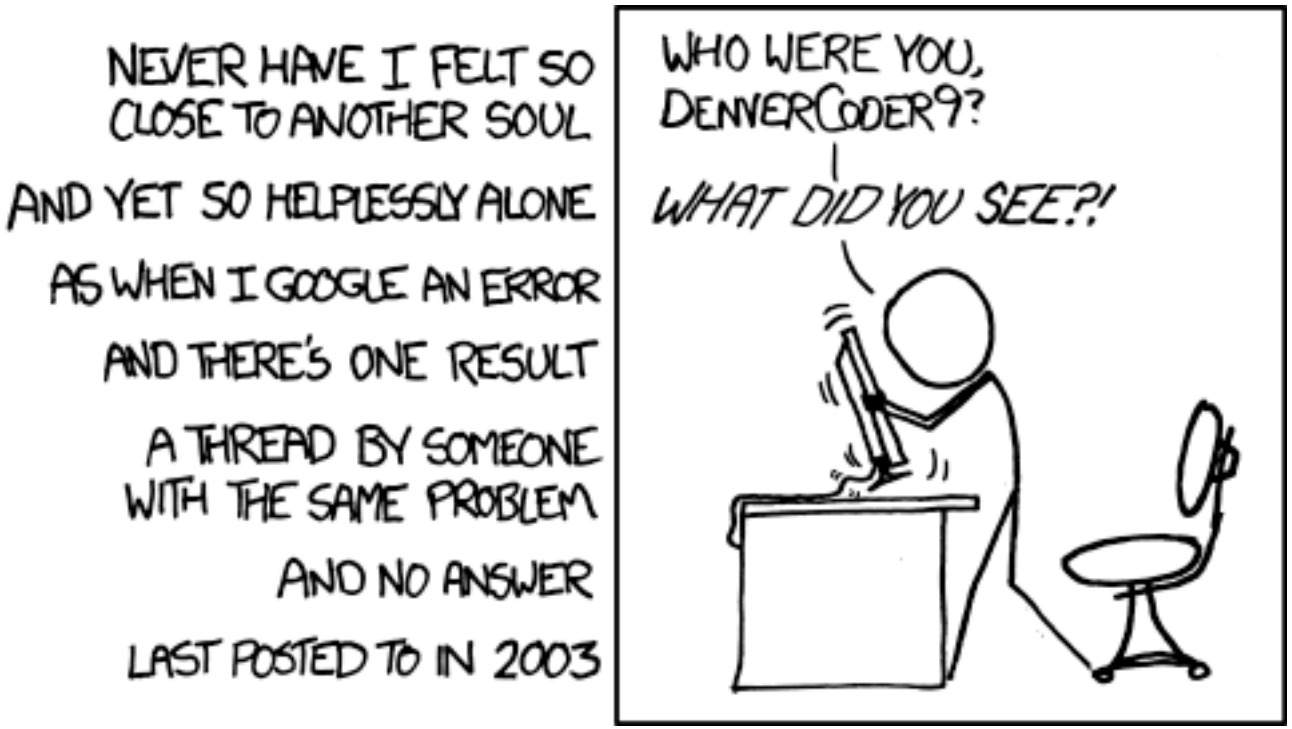
- Focus of the rest of the tutorial and hands-on session: Named entity recognition and classification (NERC)
- Possible methodologies
  - Rule-based approaches: write manual extraction rules
  - Machine learning based approaches
    - Supervised learning: manually annotate text, train machine learning model
    - Unsupervised learning: extract language patterns, cluster similar ones
    - Semi-supervised learning: start with a small number of language patterns, iteratively learn more (bootstrapping)
  - Gazetteer-based method: use existing list of named entities
  - Combination of the above

Information Extraction: Methods

Developing a NERC involves programming based around APIs..

Information Extraction: Methods

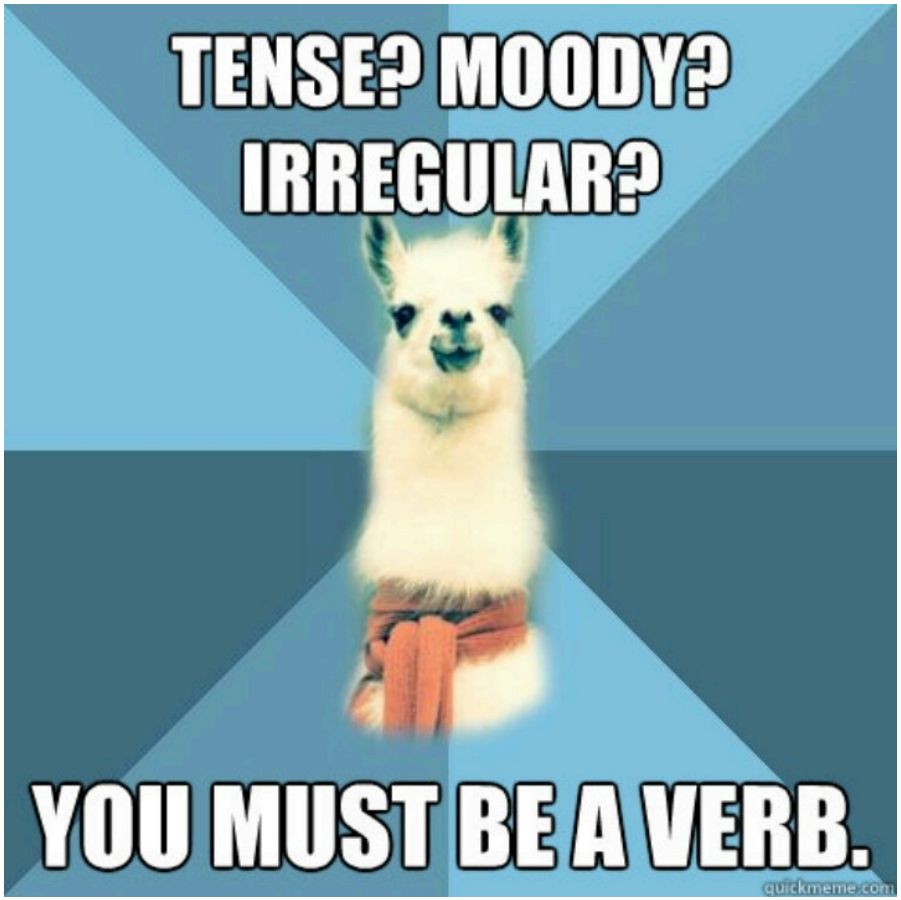
Developing a NERC involves programming based around APIs..



which can be frustrating at times

Information Extraction: Methods

and (at least basic) knowledge about linguistics



Background: Linguistics

Acoustic signal  -> /l/

↓ Phonology, phonetics

Phonemes *ʃ -> sh*

↓ Lexicon

Morphemes *wait + ed -> waited, cat -> cat*

↓ Morphology

Words *wait -> V, cat -> N*

↓ Syntax

Sentences *The (D) farmer (N) hit (V) the (D) donkey (N).* (NP: The (D) farmer (N); NP: the (D) donkey (N))

↓ Semantics, Discourse

Meaning *Every farmer who owns a donkey beats it.*
$\forall x \forall y$ (farmer(x) $\wedge$ donkey(y) $\wedge$ own(x, y) $\rightarrow$ beat(x, y))

Background: NLP Tasks

Acoustic signal -> /l/

↓ Phonology, phonetics

Phonemes

↓ Lexicon

Morphemes

↓ Morphology

Words

↓ Syntax

Sentences

↓ Semantics, Discourse

Meaning

Speech recognition

Lemmatisation or stemming, part of speech (POS) tagging

Chunking, parsing

Semantic and discourse analysis, anaphora resolution

Background: Linguistics

Acoustic signal  -> /l/

↓ Phonology, phonetics — *recognize speech – wreck a nice beach*
/ˈrek.əg.naɪz/ /spiːtʃ/ – /rek/ /ə/ /naɪs/ /biːtʃ/

Phonemes

↓ Lexicon

Morphemes

↓ Morphology

Words

↓ Syntax

Sentences

↓ Semantics, Discourse

Meaning

Background: Linguistics

Acoustic signal  -> /l/

↓ Phonology, phonetics — *recognize speech – wreck a nice beach* /ˈrek.əg.naɪz/ /spiːtʃ/ – /rek/ /ə/ /naɪs/ /biːtʃ/

Phonemes

↓ Lexicon — *She'd* /ʃiːd/

Morphemes

↓ Morphology

Words

↓ Syntax

Sentences

↓ Semantics, Discourse

Meaning

Background: Linguistics

Acoustic signal  -> /l/

↓ Phonology, phonetics — *recognize speech – wreck a nice beach* /ˈrek.əg.naɪz/ /spiːtʃ/ – /rek/ /ə/ /naɪs/ /biːtʃ/

Phonemes

↓ Lexicon — *She'd* /ʃiːd/ -> *she would, she had*

Morphemes

↓ Morphology

Words

↓ Syntax

Sentences

↓ Semantics, Discourse

Meaning

Background: Linguistics

Acoustic signal  -> /l/

↓ Phonology, phonetics — *recognize speech – wreck a nice beach* /ˈrek.əg.naɪz/ /spiːtʃ/ – /rek/ /ə/ /naɪs/ /biːtʃ/

Phonemes

↓ Lexicon — *She'd /ʃiːd/ -> she would, she had*

Morphemes

↓ Morphology — *Time flies like an arrow*

Words

↓ Syntax

Sentences

↓ Semantics, Discourse

Meaning

Background: Linguistics

Acoustic signal  -> /l/

↓ Phonology, phonetics — *recognize speech – wreck a nice beach* /ˈrek.əg.naɪz/ /spiːtʃ/ – /rek/ /ə/ /naɪs/ /biːtʃ/

Phonemes

↓ Lexicon — *She'd /ʃiːd/ -> she would, she had*

Morphemes

↓ Morphology — *Time flies(V/N) like(V/P) an arrow*

Words

↓ Syntax

Sentences

↓ Semantics, Discourse

Meaning

Background: Linguistics

Acoustic signal  -> /l/

↓ Phonology, phonetics — *recognize speech – wreck a nice beach* /ˈrek.əg.naɪz/ /spiːtʃ/ – /rek/ /ə/ /naɪs/ /biːtʃ/

Phonemes

↓ Lexicon — *She'd* /ʃiːd/ *-> she would, she had*

Morphemes

↓ Morphology — *Time flies(V/N) like(V/P) an arrow*

Words

↓ Syntax — *The woman saw the man with the binoculars*.

Sentences

↓ Semantics, Discourse

Meaning

Background: Linguistics

Acoustic signal  -> /l/

↓ Phonology, phonetics — *recognize speech – wreck a nice beach* /ˈrek.əg.naɪz/ /spiːtʃ/ – /rek/ /ə/ /naɪs/ /biːtʃ/

Phonemes

↓ Lexicon — *She'd /ʃiːd/ -> she would, she had*

Morphemes

↓ Morphology — *Time flies(V/N) like(V/P) an arrow*

Words

↓ Syntax — *The woman saw the man with the binoculars*. -> Who had the binoculars?

Sentences

↓ Semantics, Discourse

Meaning

Background: Linguistics

Acoustic signal  -> /l/

↓ Phonology, phonetics — *recognize speech – wreck a nice beach* /ˈrek.əg.naɪz/ /spiːtʃ/ – /rek/ /ə/ /naɪs/ /biːtʃ/

Phonemes

↓ Lexicon — *She'd /ʃiːd/ -> she would, she had*

Morphemes

↓ Morphology — *Time flies(V/N) like(V/P) an arrow*

Words

↓ Syntax — *The woman saw the man with the binoculars*. -> Who had the binoculars?

Sentences

↓ Semantics, Discourse — *Somewhere in Britain, some woman has a child every thirty seconds.*

Meaning

Background: Linguistics

Acoustic signal  -> /l/

↓ Phonology, phonetics — *recognize speech – wreck a nice beach* /ˈrek.əg.naɪz/ /spiːtʃ/ – /rek/ /ə/ /naɪs/ /biːtʃ/

Phonemes

↓ Lexicon — *She'd /ʃiːd/ -> she would, she had*

Morphemes

↓ Morphology — *Time flies(V/N) like(V/P) an arrow*

Words

↓ Syntax — *The woman saw the man with the binoculars*. -> Who had the binoculars?

Sentences

↓ Semantics, Discourse — *Somewhere in Britain, some woman has a child every thirty seconds.* -> Same woman or different women?

Meaning

Background: Linguistics

Acoustic signal  -> /l/

Ambiguities on every level

↓ Phonology, phonetics

Phonemes

recognize speech – wreck a nice beach
/ˈrek.əg.naɪz/ /spiːtʃ/ – /rek/ /ə/ /naɪs/ /biːtʃ/

↓ Lexicon

Morphemes

She'd /ʃiːd/ -> she would, she had

↓ Morphology

Words

Time flies(V/N) like(V/P) an arrow

↓ Syntax

Sentences

The woman saw the man with the binoculars. -> Who had the binoculars?

↓ Semantics, Discourse

Meaning

Somewhere in Britain, some woman has a child every thirty seconds.
-> Same woman or different women?

Background: Linguistics

Acoustic signal  -> /l/

Ambiguities on every level

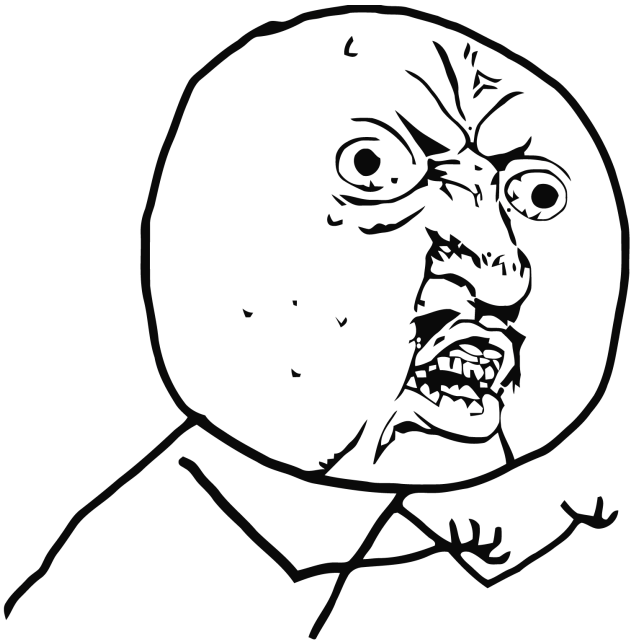
Phonemes

Morpheme

Words

Sentences

Meaning



vreck a nice beach
· /rek/ /ə/ /naɪs/ /biːtʃ/

uld, she had

/P) an arrow

man with the
ad the binoculars?

n, some woman
ty seconds.
different women?

Information Extraction

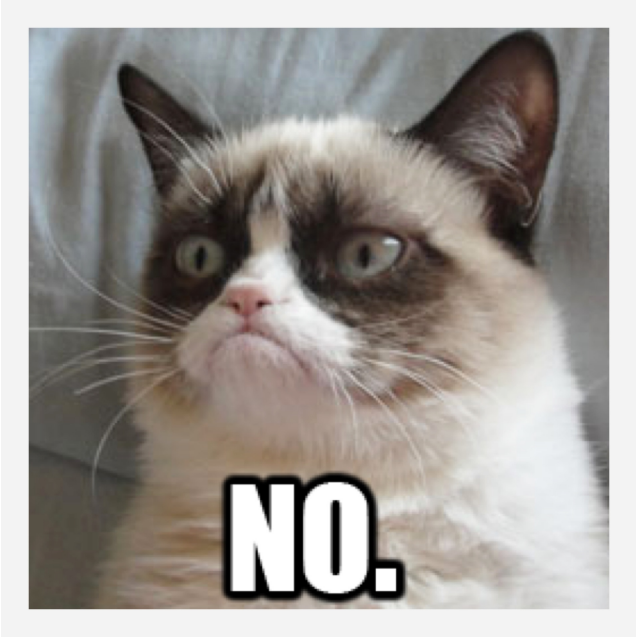
Language is ambiguous..

Can we still build named entity extractors that extract all entities from unseen text correctly?

Information Extraction

Language is ambiguous..

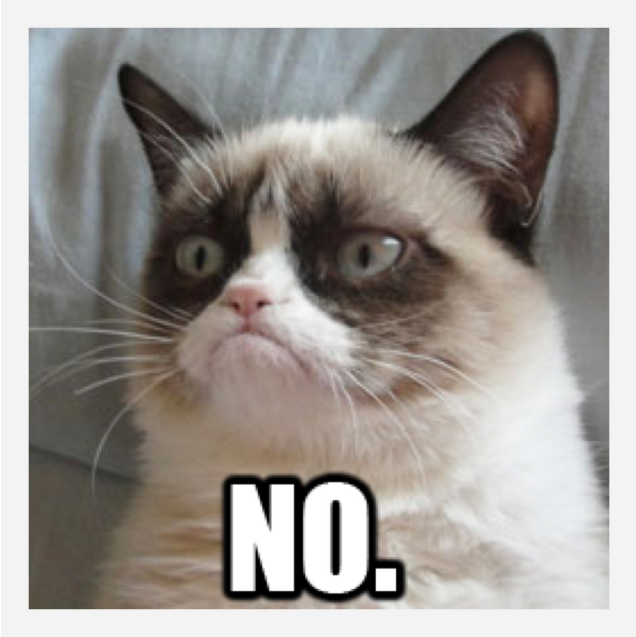
Can we still build named entity extractors that extract all entities from unseen text correctly?



Information Extraction

Language is ambiguous..

Can we still build named entity extractors that extract all entities from unseen text correctly?



However, we can try to extract most of them correctly

using linguistic cues and background knowledge!

NERC: Features

What can help to recognise and/or classify named entities?

- Words:
  - Words in window before and after mention
  - Sequences
  - Bags of words

Between AD 1400 and 1450, China was a global superpower

w: China w-1: , w-2: 1450 w+1: was w+2: a
seq[-]: 1450, seq[+]: was a
bow: China bow[-]: , bow[-]: 1450 bow[+]: was bow[+]: a

NERC: Features

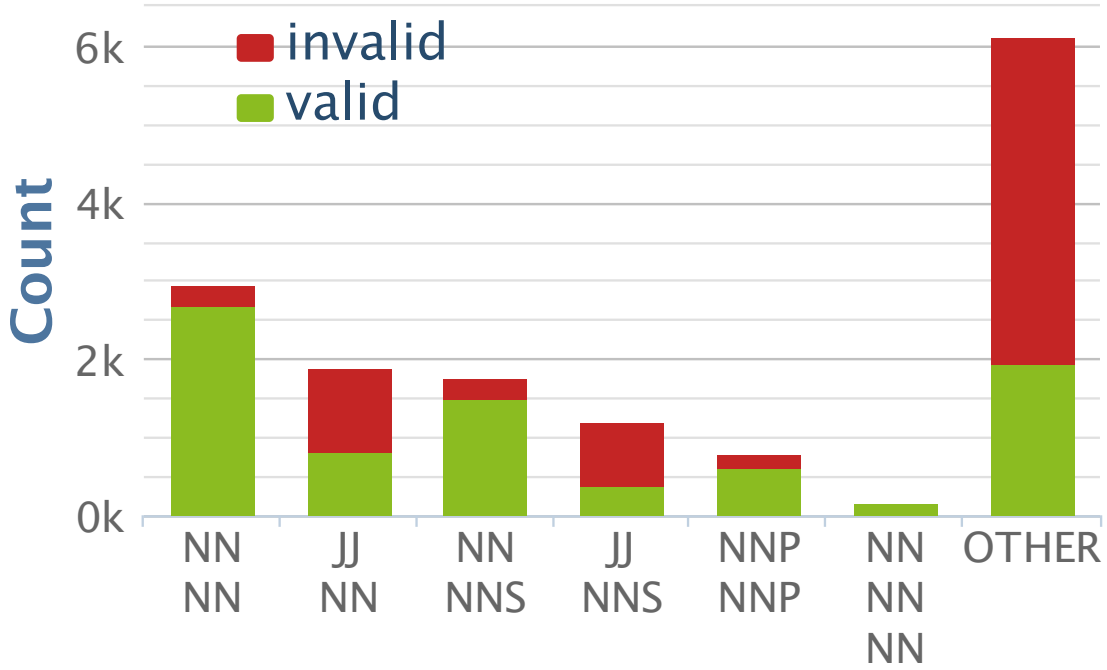
What can help to recognise and/or classify named entities?

- Morphology:
  - Capitalisation: is upper case (*China*), all upper case (*IBM*), mixed case (*eBay*)
  - Symbols: contains $, £, €, roman symbols (*IV*), ..
  - Contains period (*google.com*), apostrophe (*Mandy's*), hyphen (*speed-o-meter*), ampersand (*Fisher & Sons*)
  - Stem or Lemma (*cats -> cat*), prefix (*disadvantages -> dis*), suffix (*cats -> s*), interfix (*speed-o-meter -> o*)

NERC: Features

What can help to recognise and/or classify named entities?

- POS (part of speech) tags
  - Most named entities are nouns



Prokofyev (2014)

Morphology: Penn Treebank POS tags

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Morphology: Penn Treebank POS tags

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Adjectives (all start with *J*)

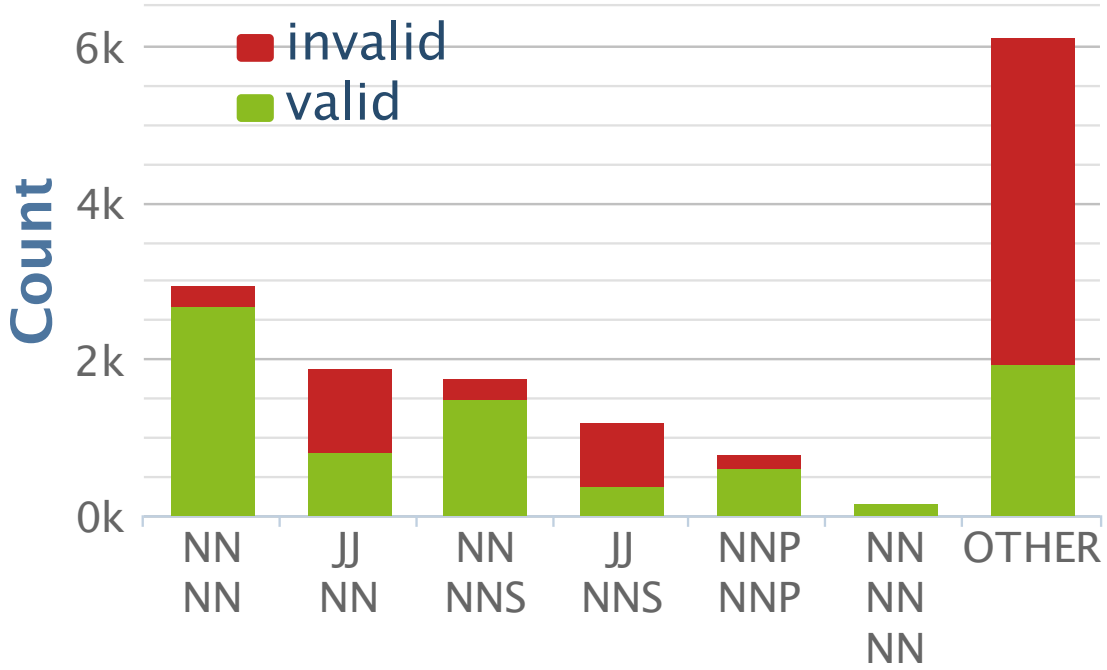
Nouns (all start with *N*)

Verbs (all start with *V*)

NERC: Features

What can help to recognise and/or classify named entities?

- POS (part of speech) tags
  - Most named entities are nouns



Prokofyev (2014)

NERC: Features

What can help to recognise and/or classify named entities?

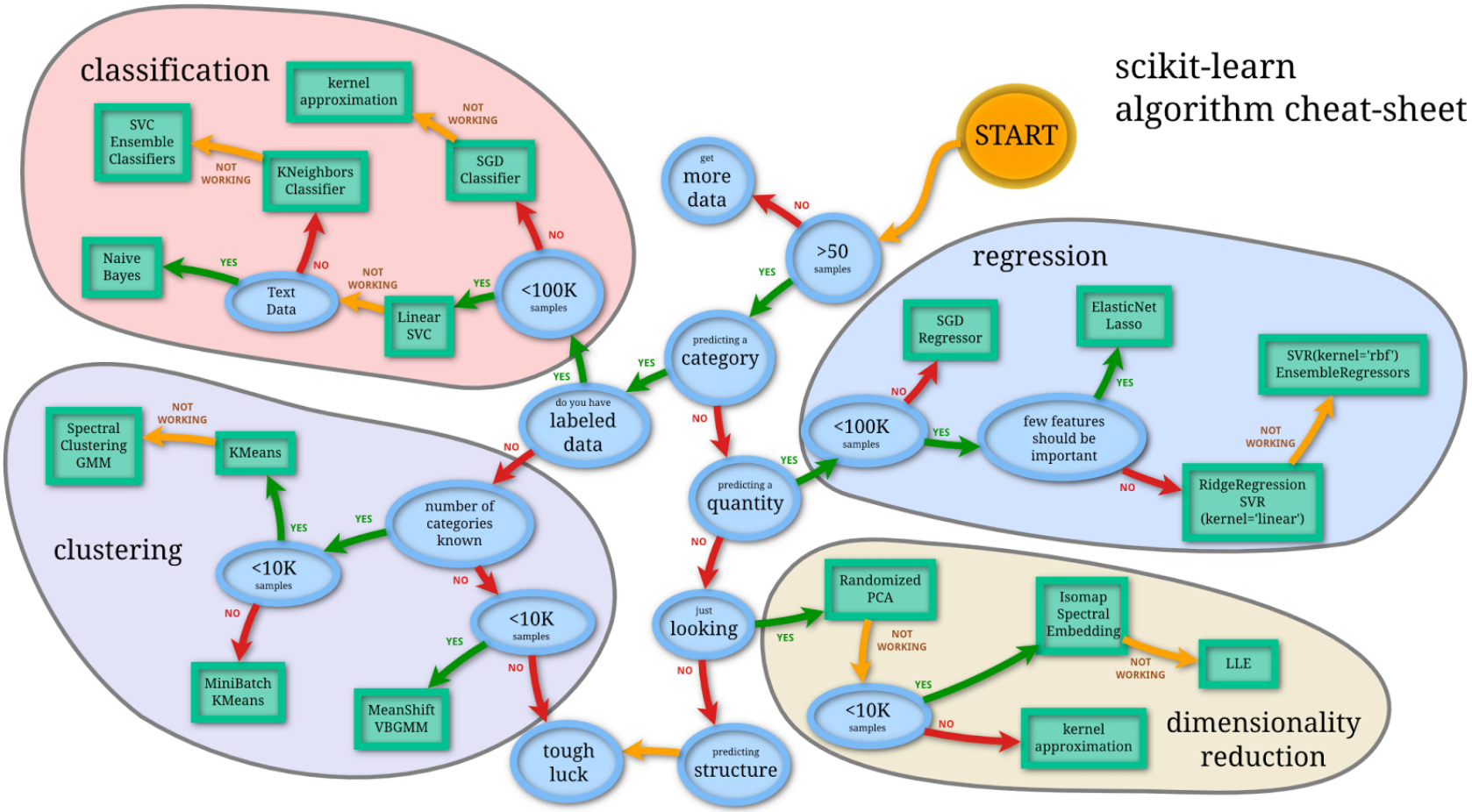
- Gazetteers
  - Retrieved from HTML lists or tables

Religion	Heaven worship, Taoism, Confucianism, Buddhism, Chinese folk religion, Islam

  - Using regular expressions patterns and search engines (e.g. "*Religions such as **"*)
  - Retrieved from knowledge bases

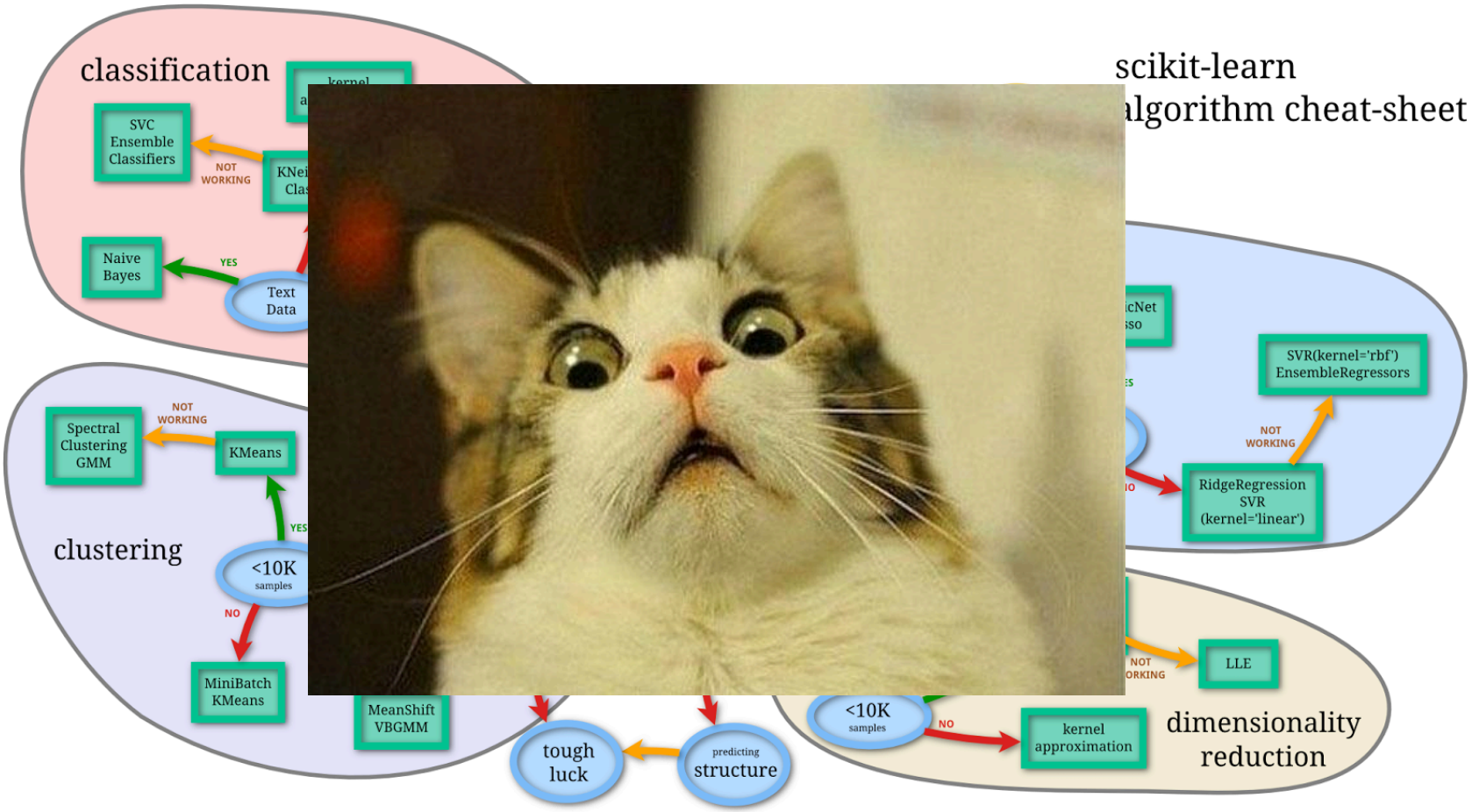
NERC: Training Models

Extensive choice of machine learning algorithms for training NERCs



NERC: Training Models

Extensive choice of machine learning algorithms for training NERCs



NERC: Training Models

- Unfortunately, there isn't enough time to explain machine learning algorithms in detail
- **CRFs** (conditional random fields) are one of the most widely used algorithms for NERC
  - Graphical models, view NERC as a sequence labelling task
  - Named entities consist of a beginning token (*B*), inside tokens (*I*), and outside tokens (*O*)
    China (*B-LOC*) built (*O*) the (*O*) Forbidden (*B-LOC*) City (*I-LOC*) . (*O*)
- For now, we will focus on **rule- and gazetteer-based** NERC
- It is fairly easy to write manual extraction rules for NEs, can achieve a high performance when combined with gazetteers
  - This can be done with the GATE software (general architecture for text engineering) and Jape rules
    -> **Hands-on session**

Outlook: NLP & ML Software

Natural Language Processing:

- GATE (general purpose architecture, includes other NLP and ML software as plugins)
- Stanford NLP (Java)
- OpenNLP (Java)
- NLTK (Python)

Machine Learning:

- scikit-learn (Python, rich documentation, highly recommended!)
- Mallet (Java)
- WEKA (Java)
- Alchemy (graphical models, Java)
- FACTORIE (graphical models, Scala)
- CRFSuite (efficient implementation of CRFs, Python)

Outlook: NLP & ML Software

Ready to use NERC software:

- ANNIE (rule-based, part of GATE)
- Wikifier (based on Wikipedia)
- FIGER (based on Wikipedia, fine-grained Freebase NE classes)

Almost ready to use NERC software:

- CRFSuite (already includes Python implementation for feature extraction, you just need to feed it with training data, which you can also download)

Ready to use RE software:

- ReVerb (Open IE, extracts patterns for any kind of relation)
- MultiR (Distant supervision, relation extractor trained on Freebase)

Web Content Extraction software:

- Boilerpipe (extract main text content from Web pages)
- Jsoup (traverse elements of Web pages individually, also allows to extract text)

Thank you for your attention!

Questions?