Machine Learning 2.11: Natural Language Processing

Tom S. F. Haines
T.S.F.Haines@bath.ac.uk

UNIVERSITY OF BATH

NLP

- Natural language processing:

  Understanding human communication
- Not just text, e.g. speech recognition
- Hard! Long way from solved

NLP

- Natural language processing:

  Understanding human communication
- Not just text, e.g. speech recognition
- Hard! Long way from solved

- Enormous – can only give a flavour
- Two kinds:
  - Rule based – No ML
  - Statistical – ML based

NLP

- Natural language processing:

  Understanding human communication

- Not just text, e.g. speech recognition
- Hard! Long way from solved

- Enormous – can only give a flavour
- Two kinds:
  - Rule based – No ML
  - Statistical – ML based

- Focusing on statistical (unsurprisingly)
- Little bit of rule based – systems usually use both

UNIVERSITY OF BATH

Challenges

- Variable input size:
  - "*The alien mothership is in orbit here! If we hit that bullseye, the rest of the dominoes will fall like a house of cards! Checkmate!*" – 25 words
  - "*Stop exploding you cowards!*" – 4 words

Challenges

- Variable input size:
  - "*The alien mothership is in orbit here! If we hit that bullseye, the rest of the dominoes will fall like a house of cards! Checkmate!*" – 25 words
  - "*Stop exploding you cowards!*" – 4 words

- Sensitive: Small changes can have large effects
  - "*Let's eat, Jack.*" vs "*Let's eat Jack!*" (comma)
  - "*Dog bites man.*" vs "*Man bites dog.*" (word order)
  - "*A car leaves its shed.*" vs "*A tree shed its leaves.*" (same word, different meaning)
  - "*I hit the man with a stick.*" (who is holding the stick?)

Challenges

- Variable input size:
  - "*The alien mothership is in orbit here! If we hit that bullseye, the rest of the dominoes will fall like a house of cards! Checkmate!*" – 25 words
  - "*Stop exploding you cowards!*" – 4 words

- Sensitive: Small changes can have large effects
  - "*Let's eat, Jack.*" vs "*Let's eat Jack!*" (comma)
  - "*Dog bites man.*" vs "*Man bites dog.*" (word order)
  - "*A car leaves its shed.*" vs "*A tree shed its leaves.*" (same word, different meaning)
  - "*I hit the man with a stick.*" (who is holding the stick?)

- Redundant: Many ways to say same thing
  - "*The same thing can be said in many different ways*" (longer)
  - "*There are a plurality of methods for communicating an identical concept*" (every word changed)
  - Yoda: "*To say same thing many ways*" (can still understand)

Challenges

- Variable input size:
  - "*The alien mothership is in orbit here! If we hit that bullseye, the rest of the dominoes will fall like a house of cards! Checkmate!*" – 25 words
  - "*Stop exploding you cowards!*" – 4 words
- Sensitive: Small changes can have large effects
  - "*Let's eat, Jack.*" vs "*Let's eat Jack!*" (comma)
  - "*Dog bites man.*" vs "*Man bites dog.*" (word order)
  - "*A car leaves its shed.*" vs "*A tree shed its leaves.*" (same word, different meaning)
  - "*I hit the man with a stick.*" (who is holding the stick?)
- Redundant: Many ways to say same thing
  - "*The same thing can be said in many different ways*" (longer)
  - "*There are a plurality of methods for communicating an identical concept*" (every word changed)
  - Yoda: "*To say same thing many ways*" (can still understand)
- Layered: Meaning, subtext, emotion, word play, sarcasm, puns ...

Tokenisation

- Chopping arbitrary text into words/punctuation

$$\underbrace{\text{I}}_{\text{t[0]}}\ \underbrace{\text{am}}_{\text{t[1]}}\ \underbrace{\text{the}}_{\text{t[2]}}\ \underbrace{\text{man}}_{\text{t[3]}}\ \underbrace{\text{with}}_{\text{t[4]}}\ \underbrace{\text{no}}_{\text{t[5]}}\ \underbrace{\text{name}}_{\text{t[6]}}\ \underbrace{\text{.}}_{\text{t[7]}}\ \underbrace{\text{Zapp}}_{\text{t[8]}}\ \underbrace{\text{Brannigan}}_{\text{t[9]}}\ \underbrace{\text{,}}_{\text{t[10]}}\ \underbrace{\text{at}}_{\text{t[11]}}\ \underbrace{\text{your}}_{\text{t[12]}}\ \underbrace{\text{service}}_{\text{t[13]}}\ \underbrace{\text{.}}_{\text{t[14]}}$$

- May throw away punctuation (task dependent)
- Language dependent!

Tokenisation

- Chopping arbitrary text into words/punctuation

$$\underbrace{\text{I}}_{t[0]}\;\underbrace{\text{am}}_{t[1]}\;\underbrace{\text{the}}_{t[2]}\;\underbrace{\text{man}}_{t[3]}\;\underbrace{\text{with}}_{t[4]}\;\underbrace{\text{no}}_{t[5]}\;\underbrace{\text{name}}_{t[6]}\;\underbrace{\text{.}}_{t[7]}\;\underbrace{\text{Zapp}}_{t[8]}\;\underbrace{\text{Brannigan}}_{t[9]}\;\underbrace{\text{,}}_{t[10]}\;\underbrace{\text{at}}_{t[11]}\;\underbrace{\text{your}}_{t[12]}\;\underbrace{\text{service}}_{t[13]}\;\underbrace{\text{.}}_{t[14]}$$

- May throw away punctuation (task dependent)
- Language dependent!
- English: Split on space, separate punctuation, except. . .
  - *Dr. Williams' velociraptor will be released at 11:15 a.m.*
  - *ice box* vs *ice-box* vs *icebox*
  - *Forgottenspaces* and *Accide ntal spaces*
  - `#HashTags`, `:-)`, . . .
- Rules get complicated – best to use a library

UNIVERSITY OF BATH

Stemming

- Problem: Lots of words!
  (150–500K? Many ways to count. . .)
- Treat as independent $\implies$ Learn about each independently

Stemming

- Problem: Lots of words!
  (150–500K? Many ways to count...)
- Treat as independent $\implies$ Learn about each independently

- Stemming: Mapping words with same meaning to their *stem*
  (language and context dependent)
- Less to learn!

- Examples:
  - "*cat*", "*cats*", "*kitten*", "*kittens*"
  - "*like*", "*likes*", "*liked*", "*likely*", "*liking*"
  - "*can't*", "*can not*"
  - "*I.O.U.*", "*I owe you*"

  (later steps may still need original)

UNIVERSITY OF BATH

Porter stemmer

- Many, many stemmers
- Popular for English: *Porter stemmer* + lookup table

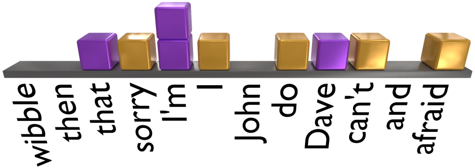
- Rules based; steps (simplified a little):
  1. Remove plurals, –*ed* and –*ing*
  2. Lookup table of suffixes, e.g. –*ational* to –*ate*,
     as in "*transformational*" to "*transformate*"
  3. Second lookup table of suffixes, e.g. removes –*ative*,
     as in "*appreciative*" to "*appreci*"
  4. Removes suffixes that are not needed (complex rules), e.g. –*ate*,
     as in "*transformate*" to "*transform*" (two steps)
  5. Final cleanup of tailing *e* and *ll*,
     as in "*appreciate*" to "*appreci*" (not real root, but consistent)

- Website with paper (1979) and code: `https://tartarus.org/martin/PorterStemmer/`

Bag of words

- Already mentioned (lecture 1)
- Ignore token order!

"*Im sorry, Dave. Im afraid I cant do that.*" $\Rightarrow$



- Sparse histogram (density estimate)
- Stupid, but works for some problems...

Topic models

- Input: Set of documents
- Output:
  - Set of topics (e.g. sport, politics)
  - Words associated with each topic
  - Topics of each document

Topic models

- Input: Set of documents
- Output:
  - Set of topics (e.g. sport, politics)
  - Words associated with each topic
  - Topics of each document

- Unsupervised – kind of *clustering*
  (Per-document mixture model with shared (tied) components)
- Topics subject to human interpretation

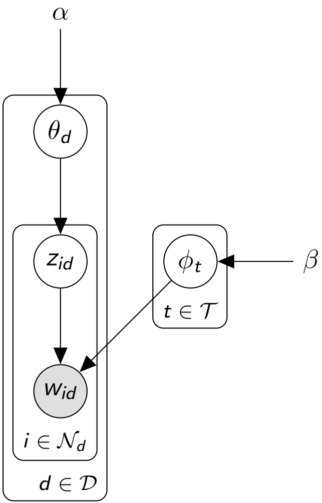
Concept

- Documents contain words (order ignored – bag of words)
- Documents have topics, e.g. politics, education, sport…
- Each word is associated with (drawn from) a topic
- Topics are shared between many documents

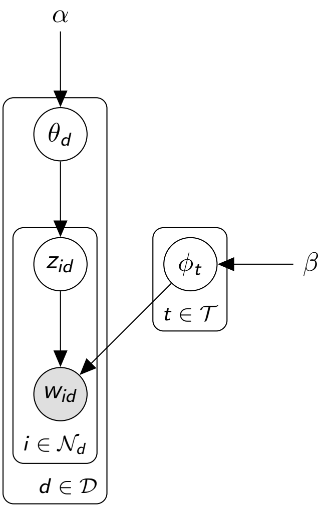
Concept

- Documents contain words (order ignored – bag of words)
- Documents have topics, e.g. politics, education, sport...
- Each word is associated with (drawn from) a topic
- Topics are shared between many documents
- First topic model: Latent semantic analysis (LSA)
  (one topic per document)
  (in ML1, lecture 13; as a recommender system)
- Most well known: Latent Dirichlet allocation (LDA)
  (weighted mixture of topics per document)

Latent Dirichlet allocation



Latent Dirichlet allocation



- α – Hyperparameter, indexed by topic, $t \in \mathcal{T}$
- β – Hyperparameter, indexed by word, $w \in \mathcal{W}$

- $\theta_d \sim \texttt{Dirichlet}(\alpha)$ – RV over topics in document $d \in \mathcal{D}$
- $\phi_t \sim \texttt{Dirichlet}(\beta)$ – RV over words, $w \in \mathcal{W}$ in topic $t \in \mathcal{T}$

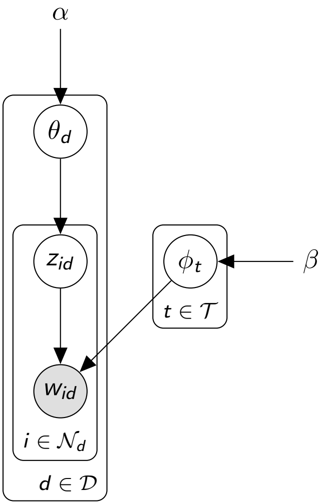
- $z_{id} \sim \texttt{Cat}(\theta_d)$ – Which topic word $i \in \mathcal{N}_d$ of document $d \in \mathcal{D}$ belongs to
- $w_{id} \sim \texttt{Cat}(\phi_{z_{id}})$ – Observed word, $w_{id} \in \mathcal{W}$

Inference

- Two choices:
  - Gibbs sampling
  - (Mean) field variational

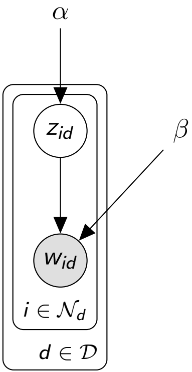
- Not going to explain: In *Bayesian machine learning*
- Will give Gibbs sampling equation
- Both collapse the model first...

Collapsing



- Three latent variables to infer: θ, ϕ and z

Collapsing



- Three latent variables to infer: θ, ϕ and z
- Integrate out: θ and ϕ
- z only – faster!

UNIVERSITY OF BATH

Gibbs sampling

- Gibbs sampling:
  Repeat many times: Resample each unknown in model (z_{id}), keeping all others fixed

$$P(z_{id} = t | \{z, w\}_{/z_{id}}, \alpha, \beta) \propto \underbrace{\frac{\beta_{w_{id}} + \sigma(w_{id}, \cdot, t)}{\sum_{v \in \mathcal{W}} \beta_v + \sigma(\cdot, \cdot, t)}}_{\int P(w|z,\phi)P(\phi|\beta)d\phi} \underbrace{\frac{\alpha_t + \sigma(\cdot, d, t)}{\sum_{s \in \mathcal{T}} \alpha_s + \sigma(\cdot, d, \cdot)}}_{\int P(z|\theta)P(\theta|\alpha)d\theta}$$

where

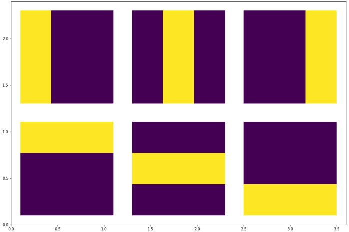
- $\sigma(w, d, t) =$ how many times word w in document d has been assigned to topic t
  with $\cdot$ to indicate summing out
- $\alpha =$ hyperparameter of Dirichlet prior over topic distributions (vector indexed by topic)
- $\beta =$ hyperparameter of Dirichlet prior over word distributions (vector indexed by word)

(z_{id} being resampled must be excluded from counts)

Visual results

Consider 3×3 images as documents, where pixels are words!

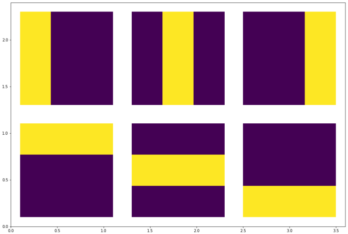
True topics:



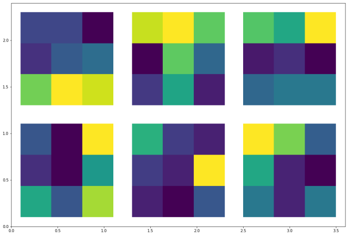
Visual results

Consider 3×3 images as documents, where pixels are words!

True topics:



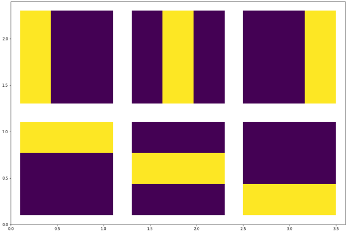
Documents:



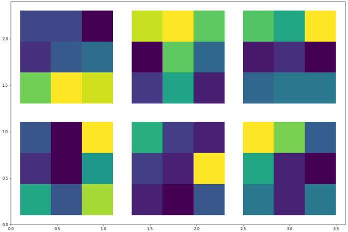
Visual results

Consider 3×3 images as documents, where pixels are words!

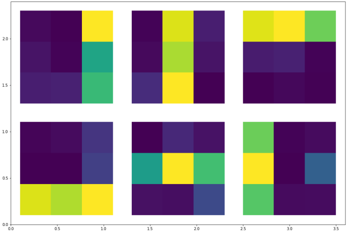
True topics:



Documents:



Estimated topics:



- Note how topic order is random

Textual results

Reuters news data set (from 1987!), 20 topics:

0	british churchill sale million major letters west
1	church government political country state people party
2	elvis king fans presley life concert young
3	yeltsin russian russia president kremlin moscow michael
4	pope vatican paul john surgery hospital pontiff
5	family funeral police miami versace cunanan city
6	simpson former years court president wife south
7	order mother successor election nuns church nirmala
8	charles prince diana royal king queen parker
9	film french france against bardot paris poster
10	germany german war nazi letter christian book
11	east peace prize award timor quebec belo
12	n't life show told very love television
13	years year time last church world people
14	mother teresa heart calcutta charity nun hospital
15	city salonika capital buddhist cultural vietnam byzantine
16	music tour opera singer israel people film
17	church catholic bernardin cardinal bishop wright death
18	harriman clinton u.s ambassador paris president churchill
19	city museum art exhibition century million churches

Term frequency – inverse document frequency

- Topic models treat all words as equal,
  e.g. “*logarithmic*” and “*is*” are equal (when discussing maths!)
- Solution:
  - Unimportant: Words that appear everywhere, e.g. “*is*”
  - Important: Rare words that are heavily used in current document, e.g. “*logarithmic*”
  - Also: Delete words that don’t appear often enough to learn from

UNIVERSITY OF BATH

Term frequency – inverse document frequency

- Topic models treat all words as equal,
  e.g. "*logarithmic*" and "*is*" are equal (when discussing maths!)
- Solution:
  - Unimportant: Words that appear everywhere, e.g. "*is*"
  - Important: Rare words that are heavily used in current document, e.g. "*logarithmic*"
  - Also: Delete words that don't appear often enough to learn from
- Term frequency – inverse document frequency:

$$\text{tf-idf}(w, d) = \frac{f_{w,d}}{f_d} \log\left(\frac{N}{d_w}\right)$$

where

- $f_{w,d}$ = number of times word w appears in document d
- f_d = number of words in document d
- N = number of documents in corpus
- d_w = number of documents containing word w

Word vectors

- Main weakness: Indicator vectors with flag for each word
- Independent words $\therefore$ learning about "*cat*" tells us nothing about "*lion*"

- ML is all about similarity – have none
- Need to *share statistical strength* between similar words

Word vectors

- Main weakness: Indicator vectors with flag for each word
- Independent words $\therefore$ learning about "*cat*" tells us nothing about "*lion*"

- ML is all about similarity – have none
- Need to *share statistical strength* between similar words

- What if we could embed words in a vector space?
  - Nearby = similar
  - Faraway = dissimilar

UNIVERSITY OF BATH

Distributional hypothesis

- Distributional hypothesis:

  Words that regularly occur together tend to have similar meanings

Distributional hypothesis

- Distributional hypothesis:

  Words that regularly occur together tend to have similar meanings

- Visible in ratios:

Equation	k = solid	k = gas	k = water	k = fashion
$P(k\|\text{ice})$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|\text{steam})$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$\frac{P(k\|\text{ice})}{P(k\|\text{steam})}$	8.9	8.5×10^{-2}	1.36	0.96

(context is 10 words either side of conditional word, corpus has 42 billion tokens)

  - 8.9: "*solid is related to ice but not steam*" (high value)
  - 8.5×10^{-2}: "*gas is related to steam but not ice*" (low value)
  - Around 1: Equally relevant (water) or not related (fashion)

GloVe I

- *GloVe* = **Glo**bal **Ve**ctors
- *word2vec* more popular, but *GloVe* is better and explainable

GloVe I

- *GloVe* = **Glo**bal **Ve**ctors
- *word2vec* more popular, but *GloVe* is better and explainable

- Imagine a function, F, that relates word vectors to probability ratios:

$$F(w_i, w_j, w_k) = \frac{P(k|i)}{P(k|j)}$$

(assuming symmetric context – see paper for asymmetric)

GloVe I

- *GloVe* = **Glo**bal **Ve**ctors
- *word2vec* more popular, but *GloVe* is better and explainable

- Imagine a function, F, that relates word vectors to probability ratios:

$$F(w_i, w_j, w_k) = \frac{P(k|i)}{P(k|j)}$$

(assuming symmetric context – see paper for asymmetric)

- Many choices of F – need to choose one. Aim for linear:

$$F((w_i - w_j)^T w_k) = \frac{P(k|i)}{P(k|j)}$$

GloVe I

- *GloVe* = **Glo**bal **Ve**ctors
- *word2vec* more popular, but *GloVe* is better and explainable

- Imagine a function, F, that relates word vectors to probability ratios:

$$F(w_i, w_j, w_k) = \frac{P(k|i)}{P(k|j)}$$

(assuming symmetric context – see paper for asymmetric)

- Many choices of F – need to choose one. Aim for linear:

$$F((w_i - w_j)^T w_k) = \frac{P(k|i)}{P(k|j)}$$

- Need symmetry, i.e. does right thing when swapping roles of i, j, and k

$$\text{symmetry} \implies F((w_i - w_j)^T w_k) = \frac{F(w_i^T w_k)}{F(w_j^T w_k)}$$

$$\therefore F(w_i^T w_k) = P(k|i)$$

GloVe II

- Only $F(\cdot) = \exp(\cdot)$ preserves symmetry:

$$F(w_i^T w_k) = P(k|i)$$

$$w_i^T w_k = \log P(k|i) = \log X_{ki} - \log X_{i\cdot}$$

where X_{ki} is the number of times word i is seen in the context of word k, $\cdot$ to sum out

GloVe II

- Only $F(\cdot) = \exp(\cdot)$ preserves symmetry:

$$F(w_i^T w_k) = P(k|i)$$

$$w_i^T w_k = \log P(k|i) = \log X_{ki} - \log X_{i\cdot}$$

where X_{ki} is the number of times word i is seen in the context of word k, $\cdot$ to sum out

- Symmetry hack: Replace $\log X_{i\cdot}$ with a bias term, b_i, and include bias for b_k as well

$$w_i^T w_k + b_i + b_k = \log X_{ki}$$

UNIVERSITY OF BATH

GloVe II

- Only $F(\cdot) = \exp(\cdot)$ preserves symmetry:

$$F(w_i^T w_k) = P(k|i)$$

$$w_i^T w_k = \log P(k|i) = \log X_{ki} - \log X_{i\cdot}$$

where X_{ki} is the number of times word i is seen in the context of word k, $\cdot$ to sum out

- Symmetry hack: Replace $\log X_{i\cdot}$ with a bias term, b_i, and include bias for b_k as well

$$w_i^T w_k + b_i + b_k = \log X_{ki}$$

- Optimise:

$$\text{argmin} \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij}) \left[w_i^T w_j + b_i + b_j - \log X_{ij} \right]^2$$

where $f(x)$ goes to zero as x does, to protect against $\log(0) = -\inf$
(they use $f(x) = \min\left\{ \left(\frac{x}{100}\right)^{0.75}, 1 \right\}$)

Solving

- Random initialisation then AdaGrad

- Vector length: 100 or 300
- Requires large corpus: 6 to 840 billion tokens!

- Slow to train – avoid!
- Get solution from `https://nlp.stanford.edu/projects/glove/`

Results: Distance

- Nearest neigbours to *frog*:
  1. *frogs*
  2. *toad*
  3. *litoria*
  4. *leptodactylidae*
  5. *rana*
  6. *lizard*
  7. *eleutherodactylus*

Litoria

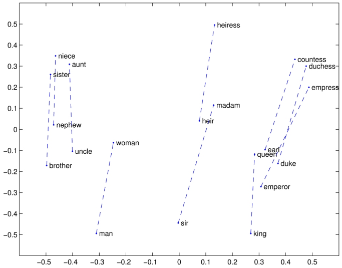
Leptodactylidae

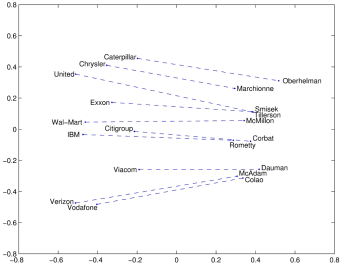
Rana (common frog)

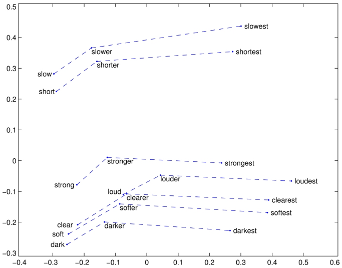
Eleutherodactylus

Results: Relationships

- Note how it was driven by three way relationships?
- Offsets often have meaning. . .







UNIVERSITY OF BATH

Sentiment analysis

- Already seen!
- Estimate how positive/negative text is
  e.g. to analyse peoples reaction to a politician
- Typically word based – take average for entire sentence

Sentiment analysis

- Already seen!
- Estimate how positive/negative text is
  e.g. to analyse peoples reaction to a politician
- Typically word based – take average for entire sentence
- Indicator vectors: Only have weights for known words
- Word vectors: Weight every word – ML interpolates from known

Example

- Positive/negative word lists of Minqing Hu and Bing Liu
  `http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html`
- Linear regression (reweighted for imbalance, matches linear assumption of model)

Positive (+1) word examples:

- knowledgeable
- brighten
- warmth
- lovably
- thank
- helping
- feisty
- sprightly

(total = 2006)

Negative (−1) word examples:

- antagonistic
- tepid
- malevolently
- rattle
- disingenuously
- ungovernable
- moronic
- invalid

(total = 4783)

Example output:

- sunrise = 0.690
- shoes = 0.409
- banker = 0.326
- lawyer = −0.040
- pirate = −0.452
- politician = −0.500
- snake = −0.694
- worm = −0.929

- Warning: Racist – weights assigned to names reflect biases of training corpus
- Training with neutral words (inc. names) helps

Part of speech

- Labelling words with role:

$$\underbrace{\text{They}}_{\texttt{PRP}}\ \underbrace{\text{say}}_{\texttt{VBP}}\ \underbrace{\text{the}}_{\texttt{DT}}\ \underbrace{\text{company}}_{\texttt{NN}}\ \underbrace{\text{has}}_{\texttt{VBZ}}\ \underbrace{\text{produced}}_{\texttt{VBN}}\ \underbrace{\text{some}}_{\texttt{DT}}\ \underbrace{\text{shoddy}}_{\texttt{JJ}}\ \underbrace{\text{work}}_{\texttt{NN}}\ \underbrace{\text{and}}_{\texttt{CC}}\ \underbrace{\text{charges}}_{\texttt{VBZ}}\ \underbrace{\text{too}}_{\texttt{RB}}\ \underbrace{\text{much}}_{\texttt{JJ}}\ \underbrace{\text{.}}_{\texttt{.}}$$

- Subset of POS labels:
  - `NN` = Noun
  - `VBN` = Verb, past participle
  - `JJ` = Adjective

(*Penn Treebank* labelling; there are others)

Part of speech

- Labelling words with role:

$$\underbrace{\text{They}}_{\texttt{PRP}}\ \underbrace{\text{say}}_{\texttt{VBP}}\ \underbrace{\text{the}}_{\texttt{DT}}\ \underbrace{\text{company}}_{\texttt{NN}}\ \underbrace{\text{has}}_{\texttt{VBZ}}\ \underbrace{\text{produced}}_{\texttt{VBN}}\ \underbrace{\text{some}}_{\texttt{DT}}\ \underbrace{\text{shoddy}}_{\texttt{JJ}}\ \underbrace{\text{work}}_{\texttt{NN}}\ \underbrace{\text{and}}_{\texttt{CC}}\ \underbrace{\text{charges}}_{\texttt{VBZ}}\ \underbrace{\text{too}}_{\texttt{RB}}\ \underbrace{\text{much}}_{\texttt{JJ}}\ \underbrace{\text{.}}_{\texttt{.}}$$

- Subset of POS labels:
  - `NN` = Noun
  - `VBN` = Verb, past participle
  - `JJ` = Adjective

(*Penn Treebank* labelling; there are others)

- "*work*": A **noun** above, but can also be a **verb**
- Context matters!

UNIVERSITY OF BATH

Penn Treebank labels

CC	Coordinating conjunction	NNP	Proper noun, singular	UH	Interjection
CD	Cardinal number	NNPS	Proper noun, plural	VB	Verb, base form
DT	Determiner	PDT	Predeterminer	VBD	Verb, past tense
EX	Existential there	POS	Possessive ending	VBG	Verb, gerund or present participle
FW	Foreign word	PRP	Personal pronoun	VBN	Verb, past participle
JJ	Adjective	PRP$	Possessive pronoun	VBP	Verb, non-3rd person singular present
JJR	Adjective, comparative	RB	Adverb	VBZ	Verb, 3rd person singular present
JJS	Adjective, superlative	RBR	Adverb, comparative	WDT	Wh-determiner
LS	List item marker	RBS	Adverb, superlative	WP	Wh-pronoun
MD	Modal	RP	Particle	WP$	Possessive wh-pronoun
NN	Noun, singular or mass	SYM	Symbol	WRB	Wh-adverb
NNS	Noun, plural	TO	to	IN	Preposition or subordinating conjunction

UNIVERSITY OF BATH

Inference

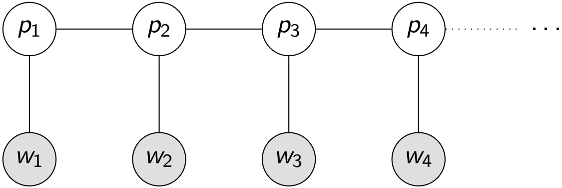
- Many algorithms. One approach: Train classifier on word vectors
- But no context...

UNIVERSITY OF BATH

Inference

- Many algorithms. One approach: Train **probabilistic** classifier on word vectors
- But no context...

- (conditional) Hidden Markov chain – learn POS transition matrix
  (solve with dynamic programming / forward-backwards / viterbi)



where

- w_i = word vector for token i
- p_i = part of speech tag for token i

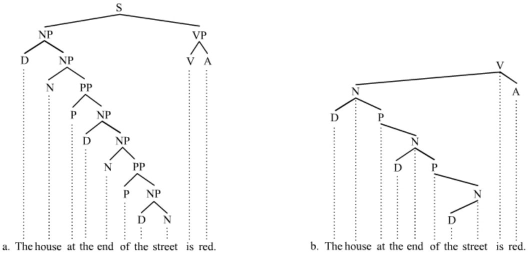
Parse trees

- Using *part of speech* a *parse tree* can be constructed
- Rule based (context-free grammars); fragile, e.g.

$$\texttt{S} \rightarrow \texttt{NPVP}$$

`S` – sentence `NP` – noun phrase `VP` – verb phrase

- Multiple kinds: Phrase structure, Dependency grammar



Constituency structure

Dependency structure

Named entity recognition

- Labelling words, again, but this time names:

$$\underbrace{\text{Mr.}}_{\texttt{B-per}}\ \underbrace{\text{Blobby}}_{\texttt{I-per}}\ \underbrace{\text{made}}_{\texttt{O}}\ \underbrace{\text{his}}_{\texttt{O}}\ \underbrace{\text{comments}}_{\texttt{O}}\ \underbrace{\text{to}}_{\texttt{O}}\ \underbrace{\text{the}}_{\texttt{O}}\ \underbrace{\text{British}}_{\texttt{B-org}}\ \underbrace{\text{Broadcasting}}_{\texttt{I-org}}\ \underbrace{\text{Corportation}}_{\texttt{I-org}}\ \underbrace{\text{Wednesday}}_{\texttt{B-tim}}$$

- Inside-outside-beginning 2 [IOB2] format: (there are others)
  - `O` – Outside, not a name
  - `B` – Beginning of a name
  - `I` – Inside of a name

  - `per` – Person
  - `org` – Organisation
  - `tim` – Time
  - `gpe` – Geo-political-entity
  - `loc` – Location
  - `fac` – Facilities

    ⋮

Inference

- Same as *part of speech*: Classifier + hidden Markov chain
- Provide POS as input
- Include capitalisation as a feature

Information extraction

- Extract facts/claims from text
- Major challenge of NLP

- Usually restricted domain, e.g. academic papers
- Unrestricted = *open information extraction*
- Mostly rule based

Information extraction

- Extract facts/claims from text
- Major challenge of NLP

- Usually restricted domain, e.g. academic papers
- Unrestricted = *open information extraction*
- Mostly rule based

- Output: (named entity 1, relationship, named entity 2)

UNIVERSITY OF BATH

Information extraction

- Extract facts/claims from text
- Major challenge of NLP

- Usually restricted domain, e.g. academic papers
- Unrestricted = *open information extraction*
- Mostly rule based

- Output: (named entity 1, relationship, named entity 2)

- Simple approach:
  - Find pairs of named entities (not crossing sentence boundary)
  - Search for relationship words between them
    (have type restrictions; makes many mistakes)
  - Use ML to filter results

Information extraction

- Extract facts/claims from text
- Major challenge of NLP

- Usually restricted domain, e.g. academic papers
- Unrestricted = *open information extraction*
- Mostly rule based

- Output: (named entity 1, relationship, named entity 2)

- Simple approach:
  - Find pairs of named entities (not crossing sentence boundary)
  - Search for relationship words between them
    (have type restrictions; makes many mistakes)
  - Use ML to filter results

- Typical failure: (insufficient context)
  "*Early scientists believed that the earth is the centre of the universe*"
  $\implies$ (*earth*, *centre*, *universe*)

Information extraction ideas

- Learn rules from examples
- Self-supervision by growing rule set from seed supervision
- Rules to transform sentences, simplifying them – meet in the middle

- Inadvertent data sets: e.g. Wikipedia fact boxes that are mirrored by text
- Capturing context, e.g.
  (*early scientists*, *believed*, (*earth*, *centre*, *universe*))

So much more

- Language identification
- Word sense disambiguation

- Semantic graphs
- Though vectors

- Text generation
- Text to speech

- Question answering
- Chat bots

- Machine translation
- Speech recognition (also, lip reading)

- Summarisation
- Text simplification

Summary

- NLP is large!
- Covered in some detail
  - Topic models
  - Word vectors
- + others!

Further reading

- Reasonable book on NLP (first 10 chapters are rule based however):
  "*An Introduction to Information Retrieval*",
  by Manning, Raghavan & Schütze (2008)
- Second LDA paper (much easier than first):
  "*Finding scientific topics*",
  by Griffiths & Steyvers (2004)
- Glove word vector paper – has great intuition:
  "*GloVe: Global Vectors for Word Representation*",
  by Pennington, Socher & Manning (2014)
- "*Survey on Open Information Extraction*",
  by Niklaus, Cetto, Freitas & Handschuh (2018)

Sources

- Leptodactylidae:
  Copyright Raul Maneyro, CC Attribution ShareAlike 2.5
  `https://commons.wikimedia.org/wiki/File:Leptodactylus_gracilis02.jpg`
- Rana:
  Copyright Richard Bartz & Munich Makro Freak CC Attribution-Share Alike 2.5 Generic
  `https://commons.wikimedia.org/wiki/File:Rana_temporaria.jpg`
- Glove graphs: Stolen from `https://nlp.stanford.edu/projects/glove/`