The University of Melbourne

Department of Computing and Information Systems

COMP90042

Web Search and Text Analysis

June 2017

Identical examination papers: None

Exam duration: Two hours

Reading time: Fifteen minutes

Length: This paper has 6 pages including this cover page.

Authorised materials: None

Calculators: Not permitted

Instructions to invigilators: Students may not remove any part of the examination paper from the examination room. Students should be supplied with the exam paper and a script book, and with additional script books on request.

Instructions to students: This exam is worth a total of 50 marks and counts for 50% of your final grade. Please answer all questions in the script book provided, starting each question (but not sub-question) on a new page. Please write your student ID in the space below and also on the front of each script book you use. When you are finished, place the exam paper inside the front cover of the script book.

Library: This paper is to be held in the Baillieu Library.

Student id:

Examiner's use only:

Q1	*Q2*	*Q3*	*Q4*	*Q5*	*Q6*	*Q7*	*Q8*	*Q9*	*Q10*

Continued overleaf . . .

COMP90042 Web Search and Text Analysis
Final Exam

Semester 1, 2017

Total marks: 50

Students must attempt all questions

Section A: Short Answer Questions [10 marks]

Answer each of the questions in this section as briefly as possible. Expect to answer each sub-question in no more than two or three sentences.

Question 1: General Concepts [4 marks]

a) What is a key difference between "lemmatisation" and "stemming"? [1 mark]

> Lemmatisation generalises (simplifies) word types to reach a canonical dictionary form; whereas stemming also simplifies the type but can arrive at invalid word forms. (0.5 each)

b) State the goal that guides the training of a"logistic regression" classifier (as well as related models like Word2Vec). [1 mark]

> The both aim to maximise the likelihood of the training data, equivalently minimising cross-entropy.

c) Name one advantage associated with "dimensionality reduction" for representing text. [1 mark]

> - provide a more robust and generalisable representation of text; fewer dimensions means easier to use elsewhere in models / similarity functions etc
> - faster down-stream processing
> - learn hidden information from text, such as a topical word inventory, and synomymy and other lexical relations.

d) Describe the "IOB" tagging method used in information extraction, and explain why it is a useful technique. [1 mark]

> It's important as it allows for multi-word tagging ('chunking') to be represented as sequence labelling, allowing the use of a variety of off-the-shelf models (0.5). It works by tagging all words outside of a chunk as O, word starting a chunk as B-X where X is the chunk label, and subsequent words in the chunk as I-X (0.5).

Question 2: Information Retrieval [3 marks]

a) What is the intuition behind the use of "tf" and "idf" factors in ranked retrieval? [1 mark]

> tf: more common terms in a document are more important to that document (0.5); idf: global importance of a term related to its rarity (or informativeness). (0.5)

b) Give an example of a query that will result in a poor (very long) runtime of the standard vector-space model querying algorithm, and explain why this is the case. [1 mark]

> All terms are very common, e.g., "trump olympics paris" (0.5). All the postings lists will be very long, and thus there will be many candidate documents to evaluate (0.5).

Continued overleaf ...

c) The *page rank* and *hubs and authorities* (HITS) methods exploit the link structure of the web. What is the common intuition behind these methods? [1 mark]

Information on the importance of pages is *conferred* through incoming or outgoing edges in the network (1). Roughly, the network structure conveys the popularity of a page.

Question 3: Discourse [3 marks]

a) How is the problem of "discourse segmentation" usually framed to convert it into a standard classification task? [1 mark]

Converted into a binary classification task deciding whether or not to insert a boundary between each possible boundary location (usually between sentences).

b) Give an example of a "discourse marker" and an "RST discourse relation" that it indicates. [1 mark]

Some examples: "and": list, sequence
"but": contrast/consession
"because": cause, motivation, purpose
"in other words": restatement
"for example": evidence
"in conclusion": summary

Full list of RST relations: Antithesis, Background, Concession, Enablement, Evidence, Justify, Motivation, Preparation, Restatement, Summary, Circumstance, Condition, Elaboration, Evaluation, Interpretation, Means, Non-volitional Cause ,Non-volitional Result, Otherwise, Purpose, Solutionhood, Unconditional, Unless, Volitional Cause, Volitional Result, Conjunction, Contrast, Disjunction, Joint, List, Multinuclear Restatement, Sequence

c) State one property which always holds between an "anaphor" and its "antecedent", and another which often holds. [1 mark].

0.5 each for: Antecedent must agree with anaphor in number/gender; Antecedent should be in close proximity/grammatically salient/satisfy verb selectional preferences

Continued overleaf ...

Section B: Method Questions [14 marks]

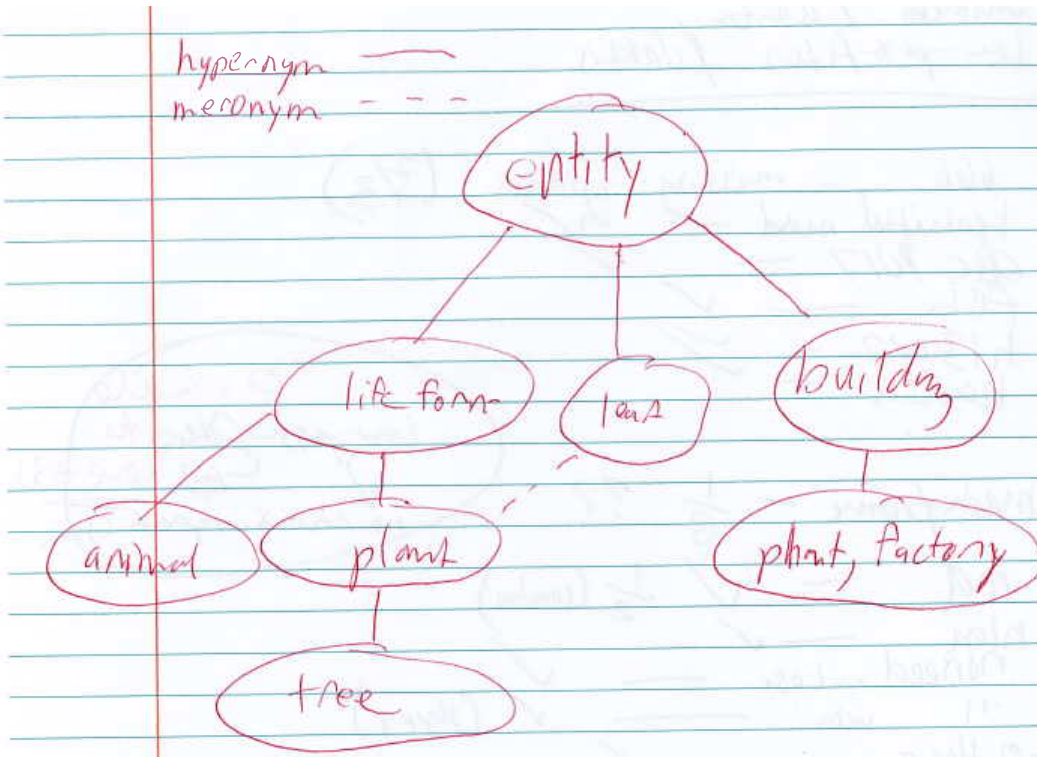
In this section you are asked to demonstrate your conceptual understanding of the methods that we have studied in this subject.

Question 4: Lexical Semantics [5 marks]

a) What is the term which refers to two unrelated senses of the same word type? [1 mark]

Homonym

b) Build a WordNet-style graph of lexical relations using the following lemmas: *plant*, *entity*, *leaf*, *lifeform*, *building*, *tree*, *factory*, *animal*. Circle lemmas to indicate synsets. Hypernym and meronym relations should be distinguished. [2 marks]



Lose 0.5 for every major error in the graph, but show leniency for subtle interpretations, e.g., conflating "plant, lifeform" or putting leaf under lifeform (although lose 0.5 if not showing any meronym in graph.)

c) Propose an unsupervised algorithm for word sense disambiguation which makes use of graphs of lexical relations. Use the graph you just created to provide an example of how it would work (you can add nodes if needed). [2 marks]

1 point for discussing an algorithm with compares the words in the context around the ambigious word in the text with the words whose synsets are nearby in the WordNet graph. 0.5 for a sensible choice of neighborhood, should limit using a distance measure or some such (should include sibilings in graph, e.g. animal). 0.5 for an example using "plant"

Question 5: Dependency parsing [6 marks]

a) Describe what it means for two words to be in a "dependency" relation, and provide an example. [1 mark]

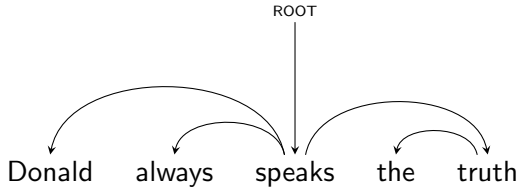
Continued overleaf . . .

The two words form part of a syntactic structure, where one modifies the syntactic behaviour and/or semantics of the other (0.5). E.g., fast ← car, where "fast" modifies car to add the attribute of being fast (0.5)

b) Show the dependency parse for the sentence

Donald always speaks the truth.

You do not need to provide edge labels. [1 mark]



Also valid for arrows to be directed in the opposite way (must be consistent with next answer.)

c) Show a sequence of parsing steps using a "transition-based parser" that will produce this dependency parse. [2 marks]

(a) SHIFT (Donald)
(b) SHIFT (always)
(c) SHIFT (speaks)
(d) LEFTARC (always ← speaks)
(e) LEFTARC (Donald ← speaks)
(f) SHIFT (the)
(g) SHIFT (truth)
(h) LEFTARC (the ← truth)
(i) RIGHTARC (speaks → truth)

d) Some dependency trees are "non-projective". Explain what this means, and then explain why the "transition-based parser" algorithm can only create projective trees. [2 marks]

Projective trees can be drawn on a plane without crossing edges, while non-projective ones cannot. More simply, the child nodes of a token do not form a contiguous sequence of the sentence, but have other tokens interspersed. (1)

Only projective trees can be created as you cannot 'look back' into the stack, rather arcs (LEFTARC and RIGHTARC) can only apply to the top two items on the stack, after which the dependent is removed. This means that the dependent cannot participate in any other dependency edges, e.g., to other parts of the sentence, which is needed to produce a non-projective tree. (1)

Question 6: Markov Models [3 marks]

a) State the probabilistic formulation of an "n-gram language model", and use this to explain why they are a type of "Markov model". [1 mark]

$p(w_i|w_1, w_2, \ldots, w_{i-1}) = p(w_i|w_{i-n}, w_{i-n+1}, \ldots, w_{i-1})$ where w are the words in a sentence, indexed by position (0.5). This is a Markov model as the context required is fixed to the previous $n-1$ words. (0.5)

b) Show how a bigram language model can be implemented as a *probabilistic context free grammar (PCFG)*. State what productions are needed, and their corresponding weights. [1 mark]

Continued overleaf ...

$S \to \#X_\#$	$[1]$	
$X_a \to bX_b$	$[p(a\|b)]$	For all word pairs a, b
$X_a \to \#$	$[p(\#\|a)]$	For all words a

where # denotes for start or end of sentence. Note that the start of sentence probabilities, $p(a|\#)$, are handled by the top two rules when used together. Students may chose to separate off this case, e.g., (almost) equivalently,

$S \to \#aX_a$	$[p(a\|\#)]$	For all words a
$X_a \to bX_b$	$[p(a\|b)]$	For all word pairs a, b
$X_a \to \#$	$[p(\#\|a)]$	For all words a

Rubric: -0.5 minor error, like missing base cases or being overly vague.

c) The "page rank" algorithm is a form of Markov chain. State the probabilistic formulation of page rank, and state how the parameters are defined using the hyperlink structure. [1 mark]

$$p(s_i|s_{i-1}) = (1-\alpha)\frac{A_{s_i,s_{i-1}}}{\sum_s A_{s,s_{i-1}}} + \alpha\frac{1}{N}$$

where s_i is the state at time i, A the adjacency matrix, N the number of nodes in the graph and α the teleport probability.

Continued overleaf ...

Section C: Algorithmic Questions [18 marks]

In this section you are asked to demonstrate your understanding of the methods that we have studied in this subject, in being able to perform algorithmic calculations.

Question 7: Document Search [7 marks]

Consider the following "term-document matrix", where each cell shows the frequency of a given term in a document:

DocId	alternative	fact	lie	truth	trump
doc_1	1	1	0	0	2
doc_2	0	2	0	3	0
doc_3	1	1	3	1	3
doc_4	0	1	0	1	1

We will be using the following query, alternative fact lie, and the vector-space model of retrieval.

a) Illustrate the "inverted index" for this document collection, showing both the "postings lists" and "document frequencies". In this part, use raw term frequencies. [1 mark]

term	df	postings
alternative	2	(1,1); (3,1)
fact	4	(1,1); (2;2); (3,1); (4,1)
lie	1	(3,3)
truth	3	(2;3); (3,1); (4,1)
trump	3	(1,2); (3,3); (4,1)

b) Compute the "IDF" term for each of the query terms. Use the standard logarithmic formulation of IDF, with base 2 logarithms (logarithm table provided below). [1 mark]

alternative: $idf = \log_2 \frac{N}{df} = \log_2 \frac{4}{2} = 1$
fact: $idf = \log_2 \frac{N}{df} = \log_2 \frac{4}{4} = 0$
lie: $idf = \log_2 \frac{N}{df} = \log_2 \frac{4}{1} = 2$

Also valid to use $idf = log_2(1 + \frac{N}{df})$, with the following results: 1.6, 1, 2.3.

c) Using your answers to the above, illustrate the progress of the querying algorithm in the TF*IDF vector space model for the given query. Show the state of the accumulator after each step, and the final document ranking. To keep things simple, do not perform vector length normalisation. [1 mark]

- See accumulator, $a = [0, 0, 0, 0]$
- Find postings for *alternative* – (1,1); (3,1) – and compute $idf = 1$
- Update accumulator $a[1] += 1 \times 1; a[3] = 1 \times 1$
- Accumulator now $a = [1, 0, 1, 0]$
- Find postings for *fact* – (1,1); (3,1) – and compute $idf = 0$; can SKIP
- Find postings for *lie* – (3,3) – and compute $idf = 2$
- Update accumulator $a[3] += 3 \times 2$
- Accumulator now $a = [1, 0, 7, 0]$
- Final document ranking doc_3; doc_1 and docs 2 and 4 tied in last place

Note similar ranking achieved using other version of idf above; although the last two are no longer tied.

Continued overleaf ...

Full marks = correct ranking; Otherwise deduct -0.5 for each error obvious from working. No working and wrong answer means 0 marks. Also fine to process query in different order, e.g., from smallest df to largest.

d) Assuming the final ranking was

doc_4, doc_2, doc_3, doc_1

and we have the following manual "relevance judgements (qrels)"

$\text{doc}_1 : 1$, $\text{doc}_2 : 1$, $\text{doc}_3 : 0$, $\text{doc}_4 : 0$

where 1 means the document relevant is relevant to the query, and 0 means irrelevant. Compute the "average precision" and "F_1 score". Show your working. [2 marks]

$$
\begin{aligned}
AP &= \frac{2}{4} \times (r_1 p@1 + r_2 p@2 + r_3 p@3 + r_4 p@4) \\
&= \frac{1}{2} \times (1/2 + 1/2) = \frac{1}{2} \\
P &= \frac{2}{4} \qquad R = \frac{2}{2} \\
F1 &= \frac{2PR}{P+R} \\
&= \frac{2 \times \frac{1}{2} \times 1}{\frac{1}{2} + 1} = \frac{2}{3}
\end{aligned}
$$

0.5 each for AP and F1

e) Show the results of "posting list compression", with the "variable byte compression" and "delta" method applied to the posting list:

5, 69, 70, 326, 329, 16714 .

Report the resulting byte sequence, using the integer value for each byte (e.g., 65 corresponds to 01000001). [2 marks]

First compute the delta gaps: 5, 64, 1, 256, 3, 16385

Now compress each into following byte sequence:

(a) 5 (less than 128, use raw)

(b) 64

(c) 1

(d) 128; 1 (256 = 10000000, split to 0000000; 1, encode first byte with leading 1 bit; second without)

(e) 3

(f) 129; 128; 1 ($126385 = 2^{14} + 1$; split into 1 (+128); 0 (+128); 1)

0.5 for gaps; 0.5 for simple cases (5, 64 etc); 1 mark for multi-byte items. Lose 0.5 for errors in arranging bits in higher order bytes, but no penalty for reasonable encoding of LSB-first vs MSB-first.

The following values may be useful:

Continued overleaf ...

x	0	1	2	3	4	5	6	7	8	9
$\log_2 x$	-	0.0	1.0	1.6	2.0	2.3	2.6	2.8	3.0	3.2
2^x	1	2	4	8	16	32	64	128	256	512
x	10	11	12	13	14	15	16	17	18	19
$\log_2 x$	3.3	3.5	3.6	3.7	3.8	3.9	4.0	4.1	4.2	4.2
2^x	1024	2048	4096	8192	16384	32768	65536	131072	262144	524288

Question 8: Machine Translation [5 marks]

This question is about word based models of machine translation. Consider the following English-Spanish parallel corpus, comprising three sentence pairs:

purple haze	grey door	grey haze
neblina púrpura	puerta gris	neblina gris

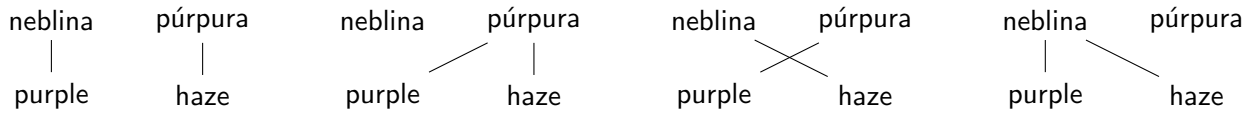
You will be showing how IBM model 1 is trained from parallel texts. This model defines the probability of a sentence translation as

$$P(\mathbf{f}, \mathbf{a}|\mathbf{e}) = \frac{\epsilon}{(1+I)^J} \prod_{j=1}^{J} t(f_j|e_{a_j})$$

where I and J denote the lengths of the English and foreign sentences, respectively, i.e., $I = |\mathbf{e}|$ and $J = |\mathbf{f}| = |\mathbf{a}|$, and $t(f|e)$ are the parameters of the model which describe the translation from an English word e to a foreign word f.

a) Starting with the parameters, $t(f|e)$ set to uniform probabilities, show the "expected counts" over word alignments the training corpus (the 'E' step in EM). You should exclude NULL alignments. [2 marks]

Pictorial answer: The probabilities are uniform over all configurations. I.e.,



(Must show all 4 configurations.)
Each of the four configurations has equal probability, because $P(\mathbf{f}, \mathbf{a}|\mathbf{e})$ has same value, as the words are ignored due to $t(f|e)$ being identical for every pair of words. This means the four configurations have a normalised probability of 0.25 each (can show the prob according to the formula, although must normalise). The same applies to the other 2 bilingual sentences. The counts of each word pair are then the 0.25 times the count of aligned pairs in these configurations.

Direct answer: state the expected counts

	neblina	púrpura	puerta	gris
purple	$\frac{1}{2}$	$\frac{1}{2}$	0	0
haze	1	$\frac{1}{2}$	0	$\frac{1}{2}$
grey	$\frac{1}{2}$	0	$\frac{1}{2}$	1
door	0	0	$\frac{1}{2}$	$\frac{1}{2}$

For full marks, values must be fractional reflecting the expectations (e.g., 0.5, 1) not just counts of configurations (2 and 4).

b) Now use your expected counts to derive updated values for the parameters $t(f|e)$ (the 'M' step in EM). For this question, just show the conditional probabilities for translating $e = grey$ and $e = door$ into Spanish. [2 marks]

Continued overleaf ...

Using the Spanish vocabulary order *1=neblina, 2=púrpura, 3=puerta, 4=gris*:

$$\begin{aligned} p(f|\text{grey}) &\propto [.25+.25; 0; .25+.25; (.25+.25)+(.25+.25)] \\ &= [0.25; 0; 0.25; 0.5] \quad \textit{normalised} \\ p(f|\text{door}) &\propto [0; 0; 0.25+0.25, 0.25+0.25] \\ &= [0; 0; 0.5; 0.5] \end{aligned}$$

0.5 each. Each 0.25 item in the above reflects a unique alignment configuration in which the word pair appears.

c) Repeating the above E and M steps will lead the EM algorithm to learn the correct alignments for this data (puerta = door, gris = grey, etc.) Explain what property of the data is being exploited to arrive at this solution, despite starting with uniform parameters? [1 mark]

The property is the asymmetry of bilingual word pair coocurrences, e.g., gris-grey appear more together than either word does with other words (0.5). As term pairs are "explained" by such translations, the ambiguity over the alignment of other word pairs can be resolved in later iterations of the algorithm (0.5).

Question 9: Grammars and Parsing [6 marks]

This question is about using analyzing syntax. Consider the following ambiguous sentence:

```
Hit the man with the ball.
```

a) Describe the syntactic ambiguity in this sentence [1 mark]

It's a PP attachment problem. Is the hitting to be done using a ball, or does the ball clarify which man is being referred to (e.g. the one holding the ball.)

b) Write a set of linguistically-plausible CFG productions that can represent and structurally differentiate the two interpretations. [2 marks]

S → VP
VP → V NP
VP → V NP PP
NP → NP PP
NP → D N
PP → P NP
V → hit
D → the
N → man
N → ball
P → with

Looking for (0.5 each):

- POS
- notion of constituency
- semi-plausible grammar, generally coherent
- generates both readings

c) Do an Earley chart parse of the sentence using your grammar. You should include the full chart, which will include the edges for both possible interpretations. [3 marks]

 Continued overleaf . . .

Looking for:

- basic chart
- rule structure
- progress of "dot" over rules
- rules added logically
- all edges present
- both parses included

Continued overleaf . . .

0

γ → · S [0] I (optional)
S → · VP [0] P
VP → · V NP [0] P
VP → · V NP PP [0] P

1

V → hit · [0] S
VP → V · NP [0] C
VP → V · NP PP [0] C
NP → · NP PP [1] P
NP → · D N [1] P

2

D → the · [1] S
NP → D · N [1] C

man
3

N → man · [2] S
NP → D N · [1] C
VP → V NP · [0] C
VP → V NP · PP [0] C
NP → NP · PP [1] C
S → VP · [0] C
PP → · P NP [3] P

with
4

P → with · [3] S
PP → P · NP [3] C
NP → · NP PP [4] P
NP → · D N [4] P

the
5

D → the · [4] S
NP → D · N [4] C

ball
6

N → ball · [5] S
NP → D N · [4] C
PP → P NP · [3] C
NP → NP · PP [4] C
VP → V NP PP · [0] C
NP → NP PP · [1] C
PP → · P NP [6] P
S → VP · [0] C
VP → V NP · [0] C
VP → V NP · PP [0] C
NP → NP · PP [1] C
γ → S · [0] C (optional)

S → VP
VP → V NP
VP → V NP PP
NP → NP PP
NP → D N
PP → P NP
V → hit
D → the
N → man
N → ball
P → with

Continued overleaf . . .

Section D: Essay Question [8 marks]

Question 10: Essay [8 marks]

Choose one of the three topics below, and discuss it in detail. At a minimum, your essay should do the following:

a) Define the topic, and motivate why it is important.

b) Explain how it relates to various tasks discussed in class. You should aim to cover 2-3 different tasks.

c) Discuss how its application is similar and differs across the tasks, and analyze why this is the case.

Marks will be given for correctness, completeness and clarity. Expect to write about 1 page.

- **Smoothing** (also known as **Regularisation**).
- **Evaluation Methods and Datasets.**
- **Vector Space Models.**

Breakdown: (1) definition; (1) motivation; (2) tasks; (1) similar; (1) dissimilar; (1) analysis; (1) general coherence.

Smoothing:

- unseen and rare events (just unseen -0.5)
- move mass from high to low frequency items
- more generally making distribution closer to uniform (linking to LR regularisation)
- tasks can include:
  - ngram LM (and add-K, interp, backoff etc)
  - LMIR (dirichlet); but NOT VSM-IR, BM25 etc.
  - PageRank (teleport; a slight stretch, but it is regularising the adjacency distribution)
  - text classification (overfitting with L2 etc regulariser; naive Bayes)
  - HMM (handling rare classes)
- similar: make more uniform, deal with sparsity, avoid degenerate cases (zeros, disconnectedness)
- differences: application in PageRank; use of different orders in Ngram; applied at different stages and explicit changes to counts versus regularisation bias.
- efficiency and simplicity implications driven by algorithm

Evaluation:

- measure generalisation performance
- compare different methods, or tune a single method
- manual versus automatic evaluation, reusability of data
- applications can include:
  - LM: pplx, use of text corpora like Brown, BNC, PTB etc
  - IR: measure P, R, AP, MRR; use of TREC with explicit qrels
  - MT: BLEU, WER, BEER etc; use of bilingual texts like politics, newswire, subtitles
  - Parsing: didn't really cover evaluation; PTB, universal dependency

 Continued overleaf ...

  - classification: acc, P, R, F1; labelled corpora, e.g., sentiment, etc
  - NER, QA etc ...
- similar: use of P, R, accuracy much of the time;
- similar: means of splitting data into training/dev/test or corss-validation
- similar: "shared-task" competitions
- similar: reuse of same underlying corpora for several tasks, e.g., PTB
- differ: metrics tend to be very task specific, e.g., generation vs classificaiton
- differ: data annotated in different ways

VSM:

- represents documents or words as real-valued vectors
- provides a fixed dimensional dense representation, allowing use of vector methods (e.g., distance)
- more robust, e.g., through conflating synonyms
- simpler & faster down-stream processing
- applications can include:
  - IR: query-document similarity over TDM
  - LSA: doc-doc and word-word application over TDM; learn synonymy
  - word2vec/PMI: learning word relations through word-context matrix
- similar: use of TDM
- differ: use of word-context vs word-document matrix
- similar: learn vector factorisation of co-occurrences
- similar: unsupervised
- differ: framing of learning objective: matrix factorisation vs learning a classifier with factorised parameters
- also could talk about interpretability, efficiency concerns, eigenvalue problems

— End of Exam —