Distributional approaches to semantic analysis

Diarmuid Ó Séaghdha

Natural Language and Information Processing Group
Computer Laboratory
University of Cambridge
do242@cam.ac.uk

HIT-MSRA Summer Workshop on Human Language Technology
August 2011
http://www.cl.cam.ac.uk/~do242/Teaching/HIT-MSRA-2011/


UNIVERSITY OF CAMBRIDGE

Your assignment (Part 2)

- We now have two sets of word pairs with similarity judgements: one in English and one in Chinese. They are available on the summer school website, along with a more detailed specification of the assignment.
- The task is to take some real corpora and build distributional models, exploring the space of design options that have been sketched in this course:
  - What is the effect of different context types (e.g., window context versus document context)?
  - What is the effect of different similarity and distance functions (e.g., cosine and Jensen-Shannon divergence)?
  - What is the effect of corpus size?
  - What is the effect of applying association measures such as PMI?
- If you do this assignment in full, you will have a good idea how to build a distributional model to help with the NLP tasks you care about!

Probabilistic semantics and topic modelling

What's in today's lecture:

A probabilistic interpretation of distributional semantics

Topic modelling

Selectional preferences

Word sense disambiguation

Probabilistic modelling

- Probabilistic modelling (or statistical modelling) explicitly recasts the problem of interest as one of inferring probability distributions over random variables and estimating properties of those distributions.
- From our perspective, it is very natural to think of a word's distributional profile over co-occurrences.
- Among the advantages of probabilistic modelling:
  - It provides interpretable models and predictions that are easy to combine.
  - The use of prior distributions make it (often) straightforward to regularise parameter learning and incorporate intuitions we have about the dataset.
- Here we will describe Latent Dirichlet Allocation, a probabilistic alternative to LSA that is *extremely* popular in lexical semantics these days (at least six papers at EMNLP-11 last week).

Words = distributions

- Instead of thinking of words as vectors in Euclidean space, we can associate them with probability distributions over the contexts in which they may appear.
- $P(c_j|w_i)$ = the conditional probability of seeing context c_j once we have seen word w_i.
- $P(w_i, c_j)$ = the joint probability of seeing word w_i co-occurring with context c_j.
- Discrete probability distributions belong to the space $\mathcal{M}_+^1$ of positive measures summing to 1 on some set C.
- A non-negative vector $\mathbf{w}_i$ normalised to sum to 1 gives the parameters $\boldsymbol{\theta}_i$ of a multinomial distribution $P(c|w_i; \boldsymbol{\theta}_i)$:

$$P(c_j|w_i; \boldsymbol{\theta}_i) = \theta_{w_i j}$$

Maximum likelihood estimation

- Given observations $\mathbf{f}_i$, the maximum likelihood estimate sets the distribution proportional to observed frequencies:

$$\hat{\boldsymbol{\theta}}_{i(ML)} = \arg\max_{\boldsymbol{\theta} \in \mathcal{M}^1_+} P(\mathbf{f}_i | w_i; \boldsymbol{\theta})$$

$$\hat{\theta}_{ij(ML)} = \frac{f_{ij}}{\sum_{j'} f_{ij'}}$$

- If a target-context pair (w_i, c_j) has not been observed in the corpus, the maximum likelihood estimate $\hat{\theta}_{ij(ML)} = 0$.
- Even if f_{ij} is non-zero, it will not be exactly proportional to the "true" $P(w_i, c_i)$ because of discretisation and finite-corpus effects (Evert, 2004).
- It is common in language modelling to smooth estimates of higher-order n-grams (Chen and Goodman, 1996) and these methods can also be applied to co-occurrence frequency estimates.

Priors and smoothing

- One probabilistic approach to smoothing is to put a prior distribution $P(\boldsymbol{\theta}; \boldsymbol{\alpha})$ on the parameters of the context distribution, reflecting our certainty or uncertainty about those parameters.
- The maximum a posteriori estimate of $\boldsymbol{\theta}_i$ is then:

$$\hat{\boldsymbol{\theta}}_{i(MAP)} = \arg\max_{\boldsymbol{\theta} \in \mathcal{M}_+^1} P(\mathbf{f}_i | w_i; \boldsymbol{\theta}) p(\boldsymbol{\theta}; \boldsymbol{\alpha})$$

- We have some choice in setting a prior distribution:
  - If we have no prior intuitions about the parameter values but do not want our model to decide too quickly from limited observations, we can set a *uniform* or *vague* prior.
  - If we do have an intuition about where the parameter values should lie, we can set a non-uniform prior.
  - Many likelihood distributions are associated with a specific prior distribution that makes computation much simpler; these are called *conjugate priors*.

Priors and smoothing

- The *conjugate* prior distribution for multinomial parameters $\boldsymbol{\theta}$ is the Dirichlet distribution:

$$p(\boldsymbol{\theta}; \boldsymbol{\alpha}) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta_i^{\alpha_i - 1}$$

  where $\Gamma(x)$ is the Gamma function.

- This choice of prior has nice mathematical properties, including the fact that $\hat{\theta}_{(MAP)}$ can be computed analytically:

$$\hat{\theta}_{ij(MAP)} = \frac{f_{ij} + \alpha_j - 1}{\sum'_j f_{ij'} + \alpha'_j - 1}$$

Priors and smoothing

- Instead of committing to a fixed value of $\boldsymbol{\theta}$, conjugacy allows us to "integrate it out" and derive a collapsed posterior:

$$p(\boldsymbol{\theta}) = \int P(\mathbf{f}_i | w_i; \boldsymbol{\theta}) p(\boldsymbol{\theta}; \boldsymbol{\alpha}) \, \mathrm{d}\boldsymbol{\theta}$$

- With Dirichlet prior and multinomial likelihood, the posterior has mean

$$\hat{\theta}_{ij(MEAN)} = \frac{f_{ij} + \alpha_j}{\sum_j' f_{ij'} + \alpha_j'}$$

- A simple approach is to set all α_j to 1, which has the effect of adding 1 to all observed frequencies. This common practice is known as "add-one" or "Laplace" smoothing.

Similarity and distance revisited I

- Euclidean distance is not the only (or the best) measure for comparing probability distributions; many suitable distance measures have been studied in information theory.
- Kullback-Leibler divergence:

$$dist_{KL}(P, Q) = \sum_j P(j) \log \frac{P(j)}{Q(j)} \tag{4}$$

  Asymmetric, not defined when $Q(j) = 0$
- α-skew divergence (Lee, 1999):

$$dist_{skew}(P, Q) = dist_{KL}(P, \alpha Q + (1 - \alpha)P)$$

  Asymmetric, always defined; Lee suggests $\alpha = 0.99$

Similarity and distance revisited II

- Jensen-Shannon divergence (Lin, 1991):

$$dist_{JSD}(P, Q) = dist_{KL}\left(P, \frac{P+Q}{2}\right) + dist_{KL}\left(Q, \frac{P+Q}{2}\right)$$

  Symmetric, always defined, $\sqrt{dist_{JSD}(P, Q)}$ is a metric

- As with Euclidean distance, it can be useful to derive measures of similarity from probabilistic distance measures; for example:

$$sim_{JSD}(P, Q) = \exp(-\beta * dist_{JSD}(P, Q))$$

  This similarity measure has been shown to be very effective in supervised learning applications of distributional information (Ó Séaghdha and Copestake, 2008).

Latent Dirichlet Allocation

- Blei et al. (2003) introduce Latent Dirichlet Allocation (LDA) as a probabilistic alternative to LSA.
- LDA is a latent variable model: each document is associated with a distribution over a set of Z latent variables and each latent variable is asssociated with a distribution over words.
- LDA is often referred to as a "topic model"; as the name suggests, its explicit goal is to discover topics in the data, where a topic is defined as a multinomial distribution over distributionally related words.
- One simple way of understanding what LDA does: it clusters words into topics and assigns each document a distribution over those topics.
- We can also view LDA as a method for dimensionality reduction from the full feature space to a $|Z|$-dimensional space.

Documents as mixtures of topics

Microsoft revenues hit a record as Xbox sales soar

The US technology giant Microsoft said its annual revenues hit a record of $69.94bn (£43.4bn).

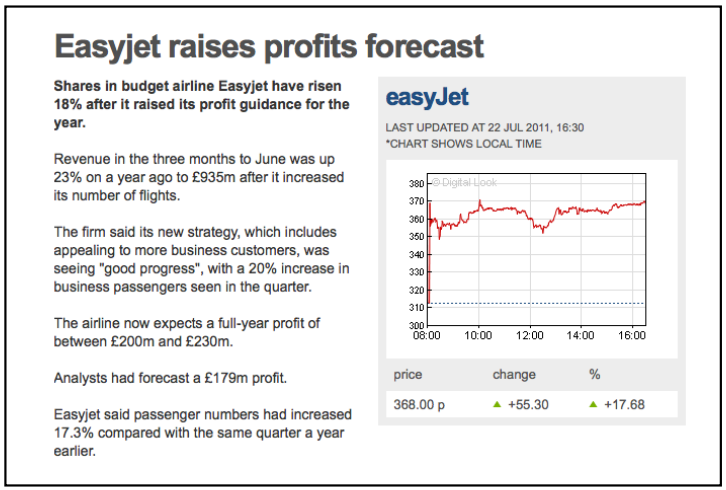
Sales of the company's Xbox 360 videogame console and its Office software helped fuel the growth.

Net income at the world's biggest software maker jumped 23% to 23.15bn for the year.

The figures, which beat forecasts, showed final quarter revenues reached a record high of $17.37bn, leading to profits of $5.87bn.

Microsoft's business division, which includes Office software - is its biggest seller

Topic 1 “business”	Topic 2 “technology”	Topic 3 “travel”
0.7	0.3	0



Easyjet raises profits forecast

Shares in budget airline Easyjet have risen 18% after it raised its profit guidance for the year.

Revenue in the three months to June was up 23% on a year ago to £935m after it increased its number of flights.

The firm said its new strategy, which includes appealing to more business customers, was seeing "good progress", with a 20% increase in business passengers seen in the quarter.

The airline now expects a full-year profit of between £200m and £230m.

Analysts had forecast a £179m profit.

Easyjet said passenger numbers had increased 17.3% compared with the same quarter a year earlier.

Topic 1 “business”	Topic 2 “technology”	Topic 3 “travel”
0.6	0	0.4

Documents as mixtures of topics

Mixture model assumption: each word in a document is associated with a single mixture component (i.e., topic).

Microsoft **revenues hit** a **record** as **Xbox** **sales soar**
Microsoft's business division which includes
Office software is its biggest seller . Sales of
the company's Xbox 360 videogame console and its
Office software helped fuel the growth . Net income
at the world's biggest software maker jumped 23% to
23.15bn for the year . The figures , which beat forecasts ,
showed final quarter revenues reached a record high of
$17.37bn , leading to profits of $5.87bn .

"business" "technology" "general"

LDA - the generative story

- LDA is a generative model of document content; it comes with a "story" about how a given corpus is produced:

for topic $z \in \{1 \ldots |Z|\}$ **do**
  $\Phi_z \sim Dirichlet(\beta)$
end for
for document $d \in \{1 \ldots |D|\}$ **do**
  $\theta_d \sim Dirichlet(\alpha)$
  for word $i \in d$ **do**
    $z_i \sim Multinomial(\theta_d)$
    $w_i \sim Multinomial(\Phi_{z_i})$
  end for
end for

LDA - the maths I

- The joint distribution of observed and hidden variables is:

$$P(D, \mathbf{z}, \mathbf{\Phi}, \boldsymbol{\theta}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{d \in D} p(\boldsymbol{\theta}_d; \boldsymbol{\alpha}) \prod_{i \in d} P(z_i; \boldsymbol{\theta}_d) P(w_i; \mathbf{\Phi}_z)$$

- Recall that we can integrate out the multinomial parameters $\mathbf{\Phi}$ and $\boldsymbol{\theta}$ due to Dirichlet-multinomial conjugacy. This means we average over all possible parameter values rather than committing to one particular value.
- The posterior distribution over topic assignments $\mathbf{z}$ is:

$$\begin{aligned} P(\mathbf{z}|D; \boldsymbol{\alpha}, \boldsymbol{\beta}) \propto & P(D|\mathbf{z}; \boldsymbol{\beta}) P(\mathbf{z}; \boldsymbol{\alpha}) \\ = & \int_{\mathbf{\Phi}} p(\mathbf{\Phi}; \boldsymbol{\beta}) \int_{\boldsymbol{\theta}} p(\boldsymbol{\theta}; \boldsymbol{\alpha}) \cdot \\ & \prod_{d \in D} \prod_{i \in d} P(w_i|z_i; \mathbf{\Phi}) P(z_i|d; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \mathrm{d}\mathbf{\Phi} \end{aligned}$$

LDA - the maths II

- Our learning problem is to evaluate this distribution and find quantities of interest such as the posterior mean. This turns out to be an intractable problem.
- For a full derivation of the LDA formulae, see Heinrich (2009).

LDA - parameter estimation

- Optimising the posterior distribution is an intractable problem; we must use approximate methods.
- Blei et al. (2003) propose using *variational inference*, whereby a function that approximates the posterior is optimised exactly.
- A second approach is to use Gibbs sampling (Griffiths and Steyvers, 2004), which is guaranteed to converge to the exact posterior but may take longer to do so.
- Asuncion et al. (2009) demonstrate that the choice of variational or sampling method does not directly affect model quality. The description here is based on Gibbs sampling.

Gibbs sampling for LDA I

- Gibbs sampling is a general Markov Chain Monte Carlo method for evaluating probability distributions that are difficult or impossible to evaluate analytically; see Resnik and Hardisty (2010) for an NLP-friendly introduction.
- The intuitive idea behind Gibbs sampling is to iterate through the dataset, updating one "small part" of the model at a time.
- Each update is non-deterministic: even from the same starting state two sampling runs will visit different states.
- Typically we run the sampler for a large number of iterations (e.g., 1000 passes through the corpus) and estimate the posterior using the final sampling state or an average over non-adjacent sampling states.

Gibbs sampling for LDA II

- For LDA, each sampling iteration updates the topic assignment z_i of each token w_i in the corpus in succession, fixing the assignments of all other tokens and using those assignments to compute the distribution over values of z_i:

$$P(z_i = z | w_i, \mathbf{z}^{-i}, \mathbf{w}^{-i}; \boldsymbol{\alpha}, \boldsymbol{\beta}) \propto \frac{f_{zd_i}^{-i} + \alpha_z}{f_d^{-i} + \sum_{z'} \alpha_{z'}} \frac{f_{zw_i}^{-i} + \beta_{w_i}}{f_z^{-i} + \sum_{w'} \beta_{w'}}$$

where we use the following notation:

$\mathbf{w}^{-i}$	all words other than the ith token
$\mathbf{z}^{-i}$	all topic assignments other than the ith token
f_{zd}	number of tokens in document d assigned to topic z
f_{zw}	number of tokens of type w in the corpus assigned to z
f_d	length in tokens of document d
f_z	number of tokens assigned to topic z

Gibbs sampling pseudocode – outer loop

Given documents D, topic vocabulary Z, no. of iterations ITS, hyperparameters α, β:

for $d = 1$ **to** $|D|$ **do**
  for all $w \in d$ **do**
    $TopicAssignments[z] = \text{RandomInt}(|Z|)$
  end for
end for
for $i = 1$ **to** $ITERATIONS$ **do**
  for $d = 1$ **to** $|D|$ **do**
    for all $w \in d$ **do**
      UpdateTopic(w,d,$TopicAssignments$)
    end for
  end for
end for

Gibbs sampling pseudocode – inner loop

To update the topic assignment for a single w:

DecrementCounts(w) {Subtract 1 from all counts related to w}
for $z = 1$ **to** $|Z|$ **do**
  $Score[z] =$ ScoreTopic(w,d,z)
  $Sum = Sum + Score[z]$
end for
$r =$ Random(Sum) {Random number between 0 and Sum}
$newZ =$ -1
while $r >= 0$ **do**
  $newZ = newZ + 1$
  $r = r$ - $Score[z]$
end while
$TopicAssignments[z] = newZ$
IncrementCounts(w,d,z)

Gibbs sampling example

Toy example: $|Z| = 2, \alpha = (1, 1), \beta = 0.1, |V| = 4$

Doc 1: a a b a b a a

Doc 2: c c d a c

Doc 3: a b b b

Doc 4: d

Gibbs sampling example

Toy example: $|Z| = 2, \alpha = (1, 1), \beta = 0.1, |V| = 4$

Doc 1:	a	a	b	a	b	a	a
	2	1	2	1	2	1	1
Doc 2:	c	c	d	a	c		
	1	2	2	1	2		
Doc 3:	a	b	b	b			
	2	2	1	2			
Doc 4:	d						
	1						

Random initialisation

Gibbs sampling example

Toy example: $|Z| = 2, \alpha = (1, 1), \beta = 0.1, |V| = 4$

Doc 1:	a	a	b	a	b	a	a
	???	1	2	1	2	1	1
Doc 2:	c	c	d	a	c		
	1	2	2	1	2		
Doc 3:	a	b	b	b			
	2	2	1	2			
Doc 4:	d						
	1						

$$P(z_1 = 1) \propto \frac{f_{z_1 d_1} + \alpha_1}{f_{d_1} + \sum_{z'} \alpha_{z'}} \frac{f_{z_1 w_a} + \beta_{w_a}}{f_{z_1} + \sum_{w'} \beta_{w'}}$$

$$P(z_1 = 2) \propto \frac{f_{z_2 d_1} + \alpha_2}{f_{d_1} + \sum_{z'} \alpha_{z'}} \frac{f_{z_2 w_a} + \beta_{w_a}}{f_{z_2} + \sum_{w'} \beta_{w'}}$$

Gibbs sampling example

Toy example: $|Z| = 2, \alpha = (1, 1), \beta = 0.1, |V| = 4$

Doc 1:	a	a	b	a	b	a	a
	???	1	2	1	2	1	1
Doc 2:	c	c	d	a	c		
	1	2	2	1	2		
Doc 3:	a	b	b	b			
	2	2	1	2			
Doc 4:	d						
	1						

$$P(z_1 = 1) \propto \left(\frac{4+1}{6+2}\right)\left(\frac{5+0.1}{8+0.4}\right)$$

$$P(z_1 = 2) \propto \left(\frac{2+1}{6+2}\right)\left(\frac{1+0.1}{8+0.4}\right)$$

Gibbs sampling example

Toy example: $|Z| = 2, \alpha = (1, 1), \beta = 0.1, |V| = 4$

Doc 1:	a	a	b	a	b	a	a
	???	1	2	1	2	1	1
Doc 2:	c	c	d	a	c		
	1	2	2	1	2		
Doc 3:	a	b	b	b			
	2	2	1	2			
Doc 4:	d						
	1						

$P(z_1 = 1) \propto 0.38 \qquad P(z_1 = 2) \propto 0.05$

Sample randomly in $(0, 0.43)$: 0.12

So z_1 is set to 1

Gibbs sampling example

Toy example: $|Z| = 2, \alpha = (1, 1), \beta = 0.1, |V| = 4$

Doc 1:	a	a	b	a	b	a	a
	1	???	2	1	2	1	1
Doc 2:	c	c	d	a	c		
	1	2	2	1	2		
Doc 3:	a	b	b	b			
	2	2	1	2			
Doc 4:	d						
	1						

And move on to the next token...

LDA - Interpreting the results

- The states visited by Gibbs sampler (or just the final state) can be used to estimate various properties of interest.
- Topic models are often evaluated from a language-modelling perpective according to the likelihood they attribute to a held-out document collection.
- However, Chang et al. (2009) show that held-out likelihood is not always a good predictor of semantic goodness.
- From a computational semantics perspective we are most interested in
  (a) Estimating the topic-word and document-topic distributions Φ and $\boldsymbol{\theta}$.
  (b) Evaluating the learned model in an application or through comparison to human judgements.

LDA - Interpreting the results

- The posterior mean of the topic distribution $\boldsymbol{\theta}_d$ for a document d is given by:

$$\hat{\theta}_{dz(MEAN)} = \frac{f_{dz} + \alpha_z}{f_d + \sum_z' \alpha_z'}$$

- The posterior mean of the word distribution $\boldsymbol{\Phi}_z$ for a topic z is given by:

$$\hat{\Phi}_{zw(MEAN)} = \frac{f_{zw} + \beta_w}{f_z + \sum_w' \beta_w'}$$

- Recall that the effect of the Dirichlet prior is to smooth the estimation of a multinomial distribution.

LDA - Without the maths

- You don't need to implement a sampler yourself in order to make use of LDA. There are many good software packages that will run LDA for you. My favourite is MALLET (`http://mallet.cs.umass.edu/`).
- The basic intuition to take home with you is that LDA discovers "topics" or clusters of co-occurring words and analyses each document in the corpus as a mixture of these clusters.
- Both the topics and the proportion of topics in each document are useful objects of study from a distributional perspective.

Example - biomedical corpus

- Most probable words for topics found by LDA in the OpenPMC corpus of biomedical scientific articles:

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
sequence	exposure	important	retinal	neurons
sequences	levels	number	lens	mice
genome	study	large	cells	receptor
genes	health	specific	retina	receptors
gene	data	studies	patients	pain
protein	environmental	result	expression	rats
species	risk	potential	corneal	synaptic
dna	effects	type	eye	brain
data	children	represent	rpe	nerve
proteins	studies	long	mutation	neuronal

Example - Twitter corpus

- Most probable words for topics found by LDA in a corpus of Twitter users in London during 2010:

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
world	lol	blog	love	baby
cup	haha	post	#xfactor	kids
england	good	updated	factor	family
#worldcup	dont	comment	big	children
football	yeah	published	cheryl	school
south	hey	entry	amazing	child
spain	love	blogs	show	parents
africa	hope	blogging	live	fun
game	gonna	posts	john	great
germany	time	posting	brother	toys

LDA as a model of distributional lexical semantics

- LDA is usually viewed as a model for modelling documents and how words typically co-occur in documents. In this sense we can view LDA as a distributional semantic model that learns from word-document co-occurrence, similar to LSA.
- Griffiths et al. (2007) demonstrate that standard LDA models perform well at predicting human judgements of word association and priming and outperform LSA on the multiple-choice synonymy task.
- Like LSA, we can also apply LDA to arbitrary kinds of co-occurrence data, e.g., the word-context data that we have used for building semantic vector space models. This makes LDA a general-purpose tool for learning about all aspects of distributional semantics.

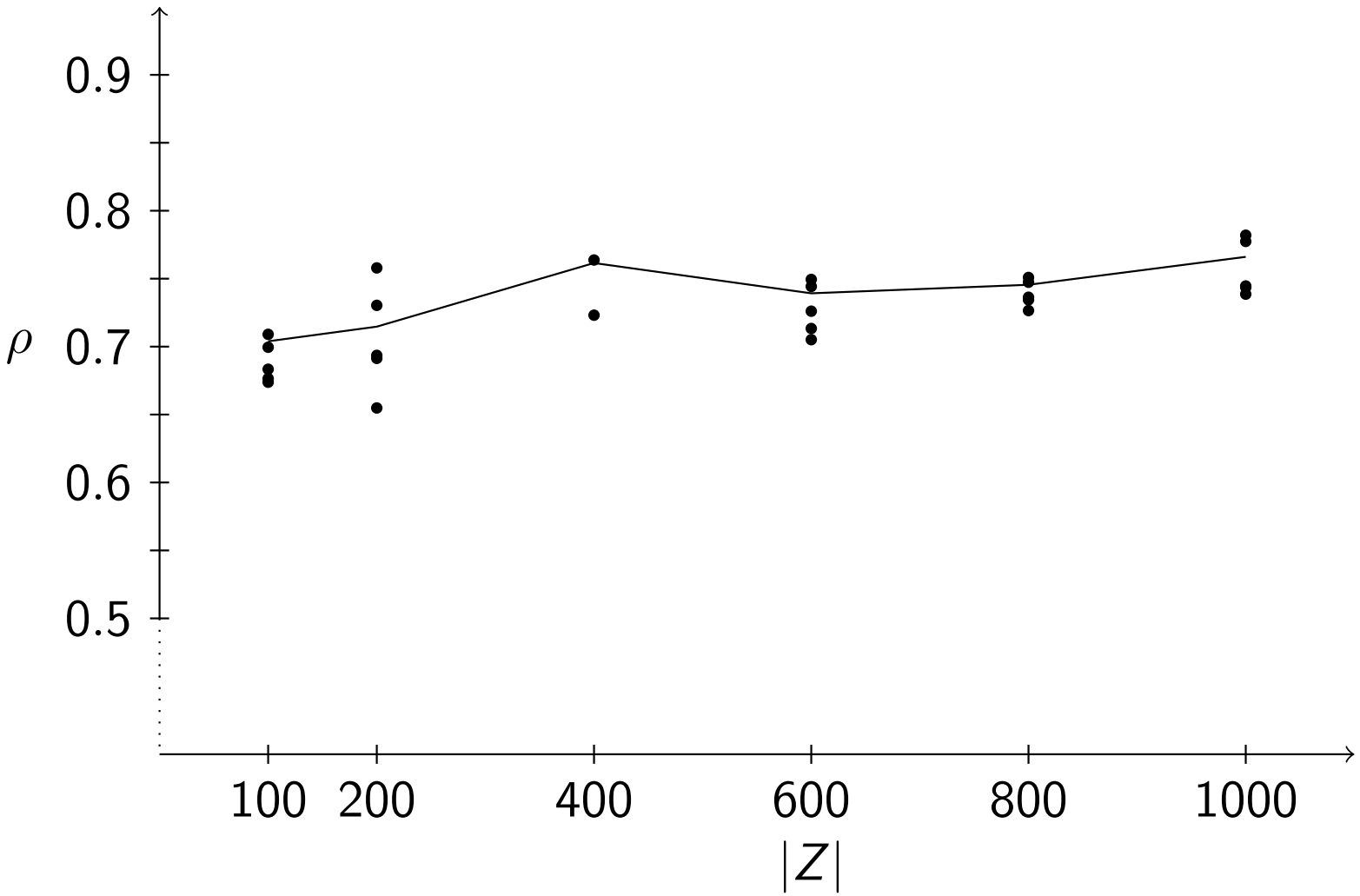
LDA for semantic similarity

- We revisit the Rubenstein and Goodenough (1965) dataset to test an LDA distributional similarity model.
- We use syntactic dependency co-occurrence data from the BNC, with which we previously attained $\rho = 0.70$.
- Our similarity measure is the Bhattacharyya similarity between the topic distributions associated with two words:

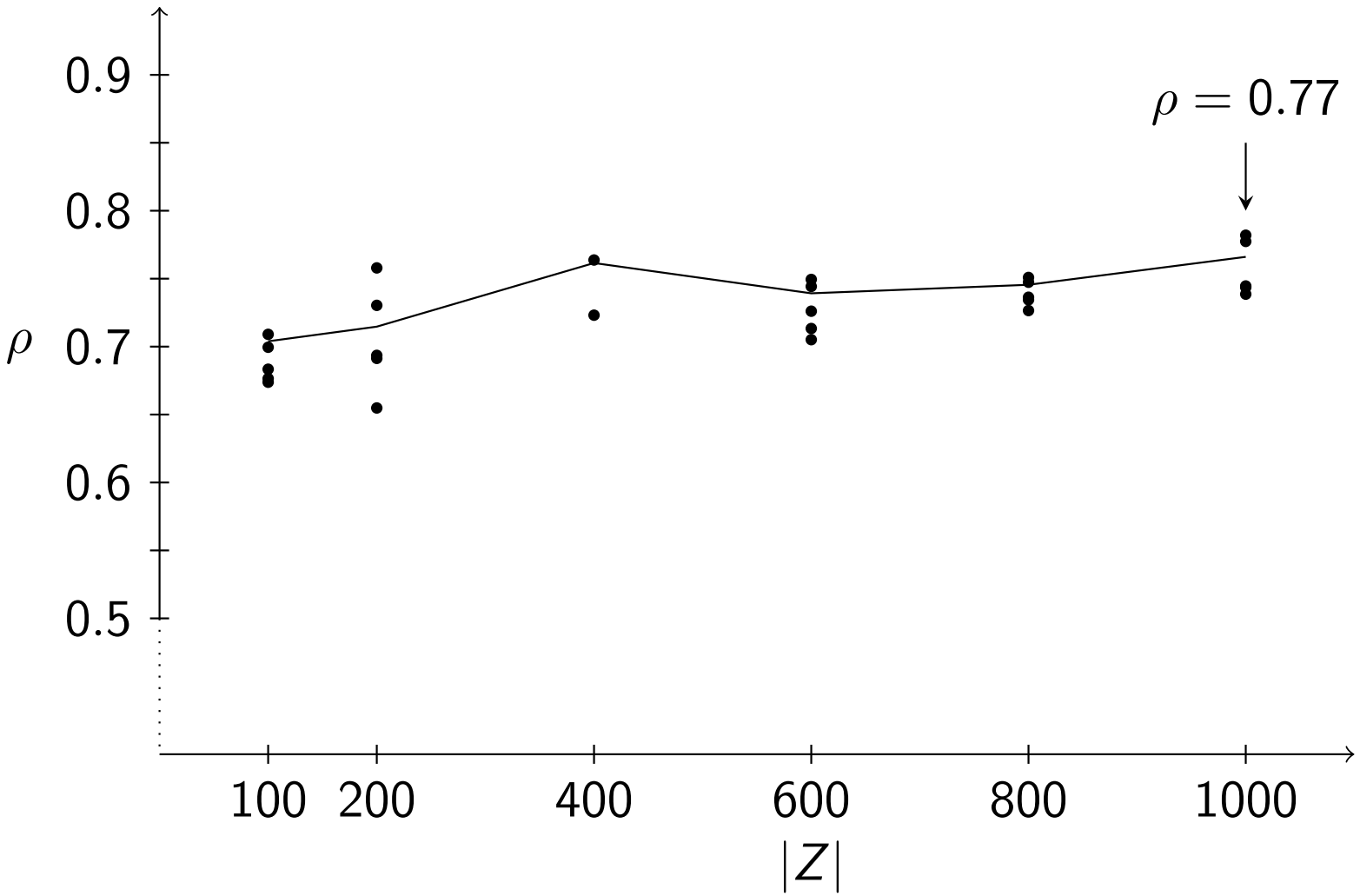
$$sim_{LDA}(w_1, w_2) = \sum_{z \in Z} \sqrt{P(z|w_1)P(z|w_2)}$$

- We investigate the effect of topic vocabulary size by learning models with different settings of $|Z|$. For each setting we run five independent simulations and average their five similarity predictions.
- This experiments was performed using the MALLET toolkit, which includes methods for automatically estimating optimal values for hyperparameters α and β.

LDA for semantic similarity



LDA for semantic similarity



Application: Selectional preference learning

- Part of our semantic knowledge relates to how predicates and entities typically interact in the world. For example, some kinds of entities (humans, animals) typically drink and some kinds of entities (liquids) are typically drunk. To refer to this knowledge we use the term *selectional preference*, e.g. "the selectional preferences of the verb *to drink*".
- We are most aware of selectional preferences when they are violated; often this is how we identify metaphors:

  My car just drinks gasoline.

- Even when we are unaware of it, selectional preference knowledge aids interpretation in many ways, e.g. coreference:

  After buying a bottle of Coca-Cola, John went down to the park and drank it.

A Web-based approach

- Keller and Lapata (2003) propose using co-occurrence counts returned by a search engine to estimate selectional preference strengths.
- For example, to compare the preference of the verb *to drink* for *gasoline*, *milk* or *compiler* as direct objects, we submit queries for

`drink|drinks|drank|drinking the|∅ milk`	15,000,000
`drink|drinks|drank|drinking the|∅ gasoline`	340,000
`drink|drinks|drank|drinking the|∅ gasoline`	132

- While the information returned by this search is very noisy, Keller and Lapata assume that the massive size of the (English) Web will guarantee reasonably accurate results.

A Web-based approach

Advantages

- Very simple to implement.
- We can use the largest corpus in the world (the Web).

Disadvantages

- Even with a massive Web corpus it can be difficult to distinguish rare but plausible predicate-argument combinations from implausible combinations.
- We will have poorer coverage over other languages or specialist domains.
- We cannot use this approach to learn selectional preferences for predicates that are not lexical items, e.g., IS-A relations or FrameNet predicates.
- It's not clear that we have learned anything truly "semantic".

Similarity-based smoothing

- Erk (2007) proposes a model of selectional preference that is based on similarity-based smoothing:

$$\text{Strength}(pred, w) = \sum_{w' \in Seen(pred)} \text{sim}(w, w') \, \text{weight}(pred, w')$$

  where $Seen(pred)$ is the set of words seen as arguments of *pred* in the corpus, sim is a similarity measure between words and weight is a weighting function (which may be a constant).
- Erk's model assumes a training set of seen arguments for each predicate, e.g., a parsed corpus, but the similarity model need not be trained on the same corpus. This modularity is useful when annotating predicate-argument examples is time-consuming, e.g., FrameNet annotation.

Topic models for selectional preferences

- Ó Séaghdha (2010) proposes using the topic modelling framework to learn selectional preferences.
- We wish to learn a distribution over latent variables for each predicate and a distribution over arguments for each latent variable:

$$P(w|pred) = \sum_z P(z|pred)P(w|z)$$

- Intuitively, we wish that the latent variables (topics) correspond to meaningful semantic classes that the topic distribution for a predicate captures the classes that typically fill its argument slot:
  - The subject slot of *drink* is most often filled by **humans** and **animals**.
  - The direct object slot of **drink** is most often filled by **beverages** and **liquids**.

Keller and Lapata (2003) plausibility data

- Human plausibility judgements for three classes of grammatical relation: verb-object, noun-noun modification and adjective-noun modification
- 30 predicates for each relation
- Each predicate matched with three arguments from the BNC (high/medium/low frequency) and three arguments that were not observed with that predicate:

Predicate	Seen		Unseen	
naughty	girl (h)	-0.0054	protocol	-0.3190
	dog (m)	-0.0645	regime	-0.1645
	lunch (l)	-0.6936	rival	0.0452

- Data collected from $>$ 20 subjects/item through Magnitude Estimation, log-transformed

Results - Seen Data

	Verb-object		Noun-noun		Adjective-noun	
	r	ρ	r	ρ	r	ρ
AltaVista (KL)	**.641**	–	**.700**	–	**.650**	–
Google (KL)	.624	–	.692	–	.641	–
BNC (RASP)	.620	**.614**	.544	.604	.543	**.622**
Padó et al.	.484	.490	.431	.503	.479	.570
LDA	.504	.541	.615	**.641**	.594	.558

- For frequent pairs, LDA (100 topics, BNC data) outperforms other selectional preference models (including others not shown here) but is not quite as good as querying the Web.

Results - Unseen Data

	Verb-object		Noun-noun		Adjective-noun	
	r	ρ	r	ρ	r	ρ
AltaVista (KL)	.551	–	.578	–	.480	–
Google (KL)	.520	–	.595	–	.473	–
BNC (RASP)	.196	.222	.114	.125	.135	.102
Padó et al.	.398	.430	.558	.533	.120	.138
LDA	**.558**	**.603**	**.636**	**.666**	.468	**.459**

- Where the data is sparser, LDA can outperform even Web queries.
- This shows that LDA can distinguish between *probability* and *plausibility*.

Induced semantic classes

BNC noun arguments, 4 predicate types (verb subject, verb object, adjective modifier, coordination), 600 topics:

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
attack	test	line	university	fund
raid	examination	axis	college	reserve
assault	check	section	school	eyebrow
campaign	testing	circle	polytechnic	revenue
operation	exam	path	institute	awareness
incident	scan	track	institution	conservation
bombing	assessment	arrow	library	alarm
offensive	sample	curve	hospital	finance

Looks quite good!

Induced semantic classes

BNC noun arguments, 4 predicate types (verb subject, verb object, adjective modifier, coordination), 600 topics:

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
attack	test	line	university	fund
raid	examination	axis	college	reserve
assault	check	section	school	eyebrow
campaign	testing	circle	polytechnic	revenue
operation	exam	path	institute	awareness
incident	scan	track	institution	conservation
bombing	assessment	arrow	library	alarm
offensive	sample	curve	hospital	finance

Looks quite good! But not perfect... Financial Non-financial
Both sets can be *raised*.

The word sense disambiguation problem

- Many words have multiple senses – they are *polysemous*.
- Some senses of the word *bank* with definitions and examples:

  (a) sloping land (especially the slope beside a body of water) *"they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"*

  (b) a financial institution that accepts deposits and channels the money into lending activities *"he cashed a check at the bank"; "that bank holds the mortgage on my home"*

  (c) the funds held by a gambling house or the dealer in some gambling games *"he tried to break the bank at Monte Carlo"*

  (d) a building in which the business of banking transacted *"the bank is on the corner of Nassau and Witherspoon"*

- If we want to extract the semantics of a text accurately, we must identify the correct sense of each polysemous word it contains. This task is known as *word sense disambiguation*. The most frequently used inventory of senses is WordNet (Fellbaum, 1998).

The Lesk method

- A simple method for disambiguating a word is based on a proposal by Lesk (1986) for comparing dictionary definitions.
- Given an inventory of senses and associated definitions, we can disambiguate a polysemous word by measuring the word overlap between its context and each sense.
- Given a sentence

  The slope of the river **bank** *passed by.*

  we can confidently assign sense (a) "sloping land (especially the slope beside a body of water)".
- However, exact word matching ignores the contribution of strongly related words; if we know that the context is discussing waterways then we can confidently assign sense (a) even without an exact match.

Topic models for WSD I

- Li et al. (2010) propose using LDA to smooth the Lesk word-matching approach.
- They train a standard topic model on a large Wikipedia corpus and use this model to compare texts to WordNet glosses.
- If information about the prior probability of senses is available (from a sense-annotated corpus), they propose the following method for choosing the most probable sense of a word w in a document d:

$$\hat{s}_w = \underset{s \in Senses(w)}{\arg\max} \; P(s) \sum_z P(z|d) P(z|Gloss(s))$$

Topic models for WSD II

- If no prior information about sense frequency is known, the following model can be used:

$$\hat{s}_w = \underset{s \in Senses(w)}{\arg\max} \ \text{Cosine}(\mathbf{f}_{d.}, \mathbf{f}_{Gloss(s).})$$

- Li et al.'s LDA method attains state-of-the-art performance for unsupervised word sense induction.

Extensions to LDA

- Focus of intense research in machine learning – most of these models have not been applied to semantics yet (opportunity!)
- Supervised topic modelling: Supervised LDA (Blei and McAuliffe, 2007), Dirichlet-multinomial regression (Mimno and McCallum, 2008), Labelled LDA (Ramage et al., 2009)
- Correlated topic models (Blei and Lafferty, 2007)
- Topic models with graph regularisation: Mei et al. (2008), Markov topic fields (Daumé III, 2009), Relational Topic Model (Chang and Blei, 2009)
- Non-parametric topic models: Hierarchical Dirichlet Processes (Teh et al., 2006)
- For even more, see David Mimno's bibliography at `http://www.cs.princeton.edu/~mimno/topics.html`

LDA and LSA - A comparison

- Both LDA and LSA have the same general goal: to find a description of co-occurrence data that uses far fewer components than the original feature space.
- The assumptions behind LSA are based on a geometric model of co-occurrence, while LDA is based on probabilistic assumptions.
- Which one works better in practice is often an empirical question.
- Griffiths et al. (2007) claim that the properties of LDA better reflect what we know about human cognition (asymmetry, triangle inequality).
- LDA, as a probabilistic model, has some advantages:
  - LDA components are often easier to interpret.
  - It is possible to add structure to LDA's generative model to account for a variety of factors such as word order, metadata, syntax, temporal information, geographical information, etc.
  - All quantities in the LDA model are normalised probabilities and may be easier to incorporate in a larger system.

Summary

- We have considered how distributional semantics can be expressed in a probabilistic modelling framework.
- We have seen that this probabilistic perspective leads to *Latent Dirichlet Allocation (LDA)* as an alternative to LSA.
- Two example applications of LDA were presented from the recent NLP research literature, where it gives state-of-the-art performance on important semantic tasks: selectional preference modelling and word sense induction.