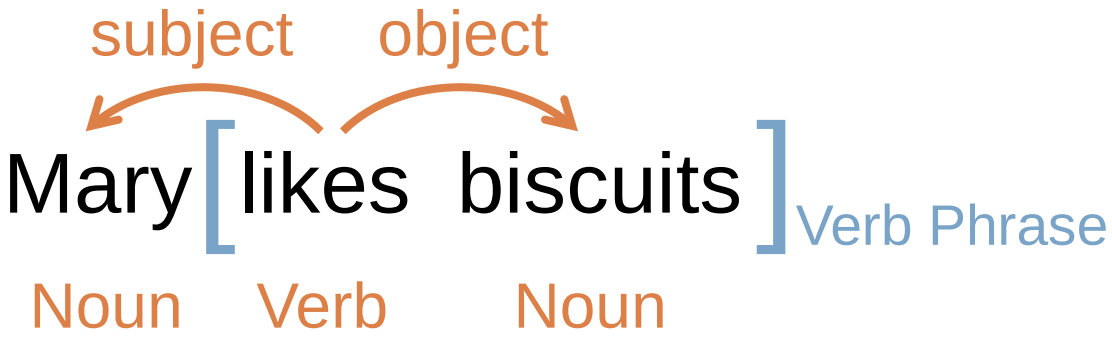
Using distributional semantics to improve parse ranking

Guy Emerson

Parsing


subject
object
Mary
likes
biscuits
Verb Phrase
Noun
Verb
Noun

Ambiguity in Natural Language

“She passed the port from the south”

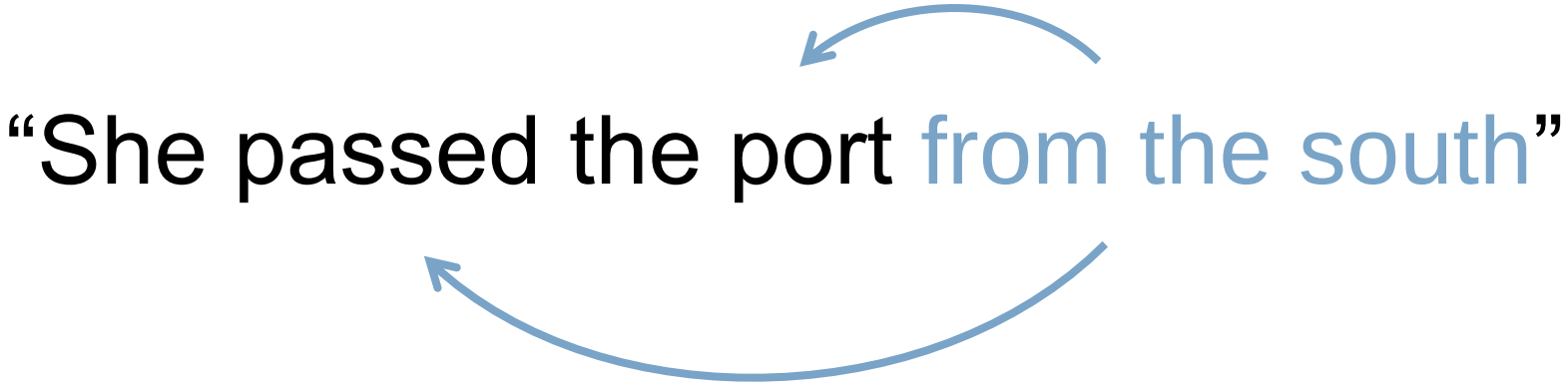
Ambiguity in Natural Language

fortified wine

"She passed the port from the south"

harbour

Ambiguity in Natural Language



Ambiguity in Natural Language

"A sheriff shot a dog with a rifle"

"A dog bit a sheriff with a moustache"

Parse Ranking

- A sheriff shot a dog [with a rifle]$_{PP}$
  - shot with a rifle
  - dog with a rifle
- A dog bit a sheriff [with a moustache]$_{PP}$
  - sheriff with a moustache
  - bit with a moustache

Association Score

$$score_r(x,y) = \frac{P_r(x,y)}{P(x)P(y)}$$

Maximum Entropy Parser

- Features f_i with weights λ_i
- Given a sentence s, and parse t,

$$P(t \mid s) = \frac{1}{Z} \exp \sum_{i=1}^{m} \lambda_i f_i(t)$$

PP-Attachment

- $w = (v, n_1, p, n_2)$
- Attachment site: V or N

$$\frac{P(N \mid w)}{P(V \mid w)} = \frac{P(N \mid p)P(v)P(n_1, n_2 \mid p, N)}{P(V \mid p)P(n_1)P(v, n_2 \mid p, V)}$$

PP-Attachment

- $w = (v, n_1, p, n_2)$
- Attachment site: V or N

$$\frac{P(N \mid w)}{P(V \mid w)} = \frac{P(N \mid p)score_{p,N}(n_1, n_2)}{P(V \mid p)score_{p,V}(v, n_2)}$$

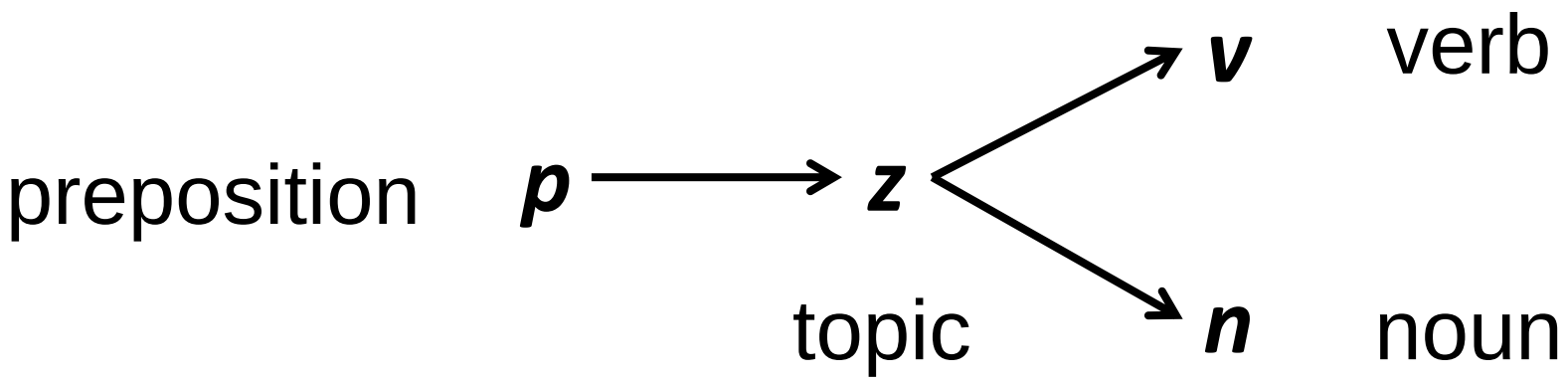
Latent Dirichlet Allocation

- Introduced by Blei, Ng, and Jordan (2003)
- Modified by Ó Śeaghdha (2010)
- Motivation: overcome data sparsity

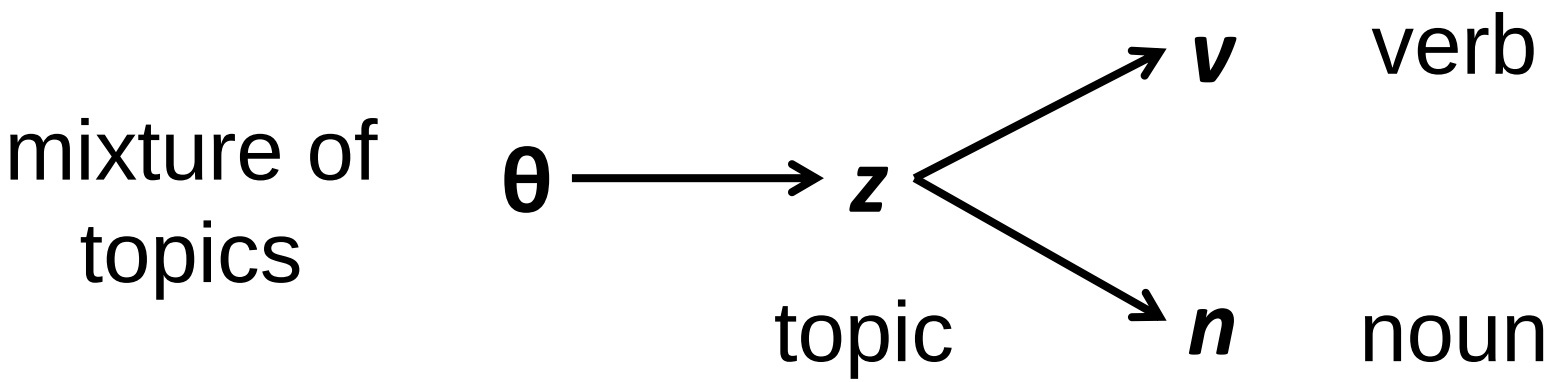
Latent Dirichlet Allocation

walk		road
run		street
drive	down	path
cycle		avenue
gallop		trail
saunter		boulevard

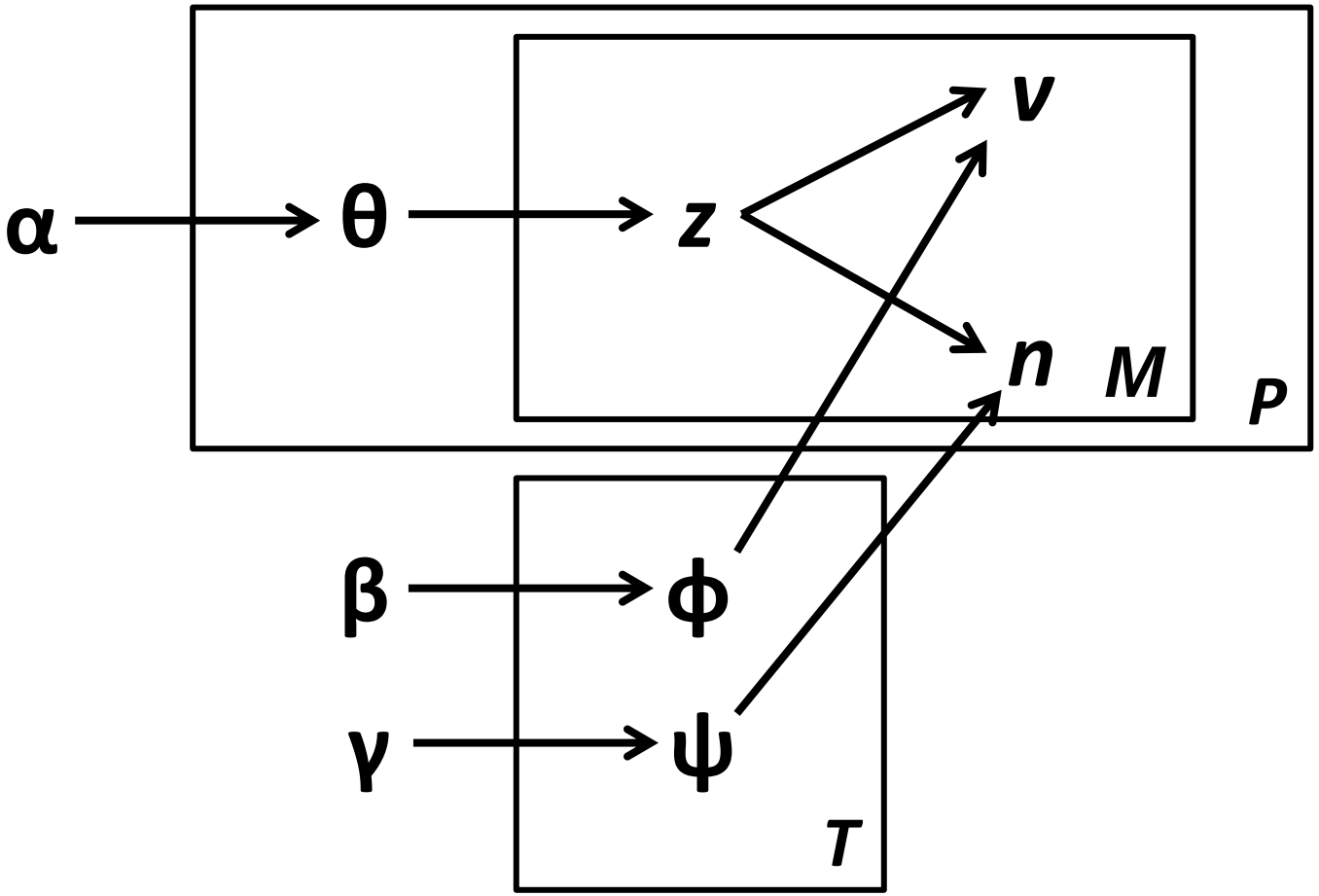
Generative Model



Generative Model



Generative Model



Training Data

- WikiWoods (Flickinger et al, 2010)
  - Snapshot of English Wikipedia (900m tokens)
  - Rich semantic and syntactic annotations
- Extracted all PPs, with attachment site
  - Not just ambiguous cases
- Considered nine prepositions (16m total tokens):
  - *as, at, by, for, from, in, on, to, with*

Example topic (*in*, *N*)

school	area
building	city
station	town
house	district
church	country
home	village
street	state
center	neighborhood
office	center
college	college

Example topic (*for*, *N*)

preparation	invasion
plan	war
time	attack
way	operation
force	battle
date	campaign
support	deployment
responsibility	election
point	landing
base	assault

Evaluation Data

- WeScience (Ytrestøl et al., 2009)
  - Manually treebanked portion of Wikipedia
  - Extracted ambiguous PPs – 2167 tokens
- Penn Treebank
  - Ratnaparkhi et al. (1994) – 1240 tokens

Baselines

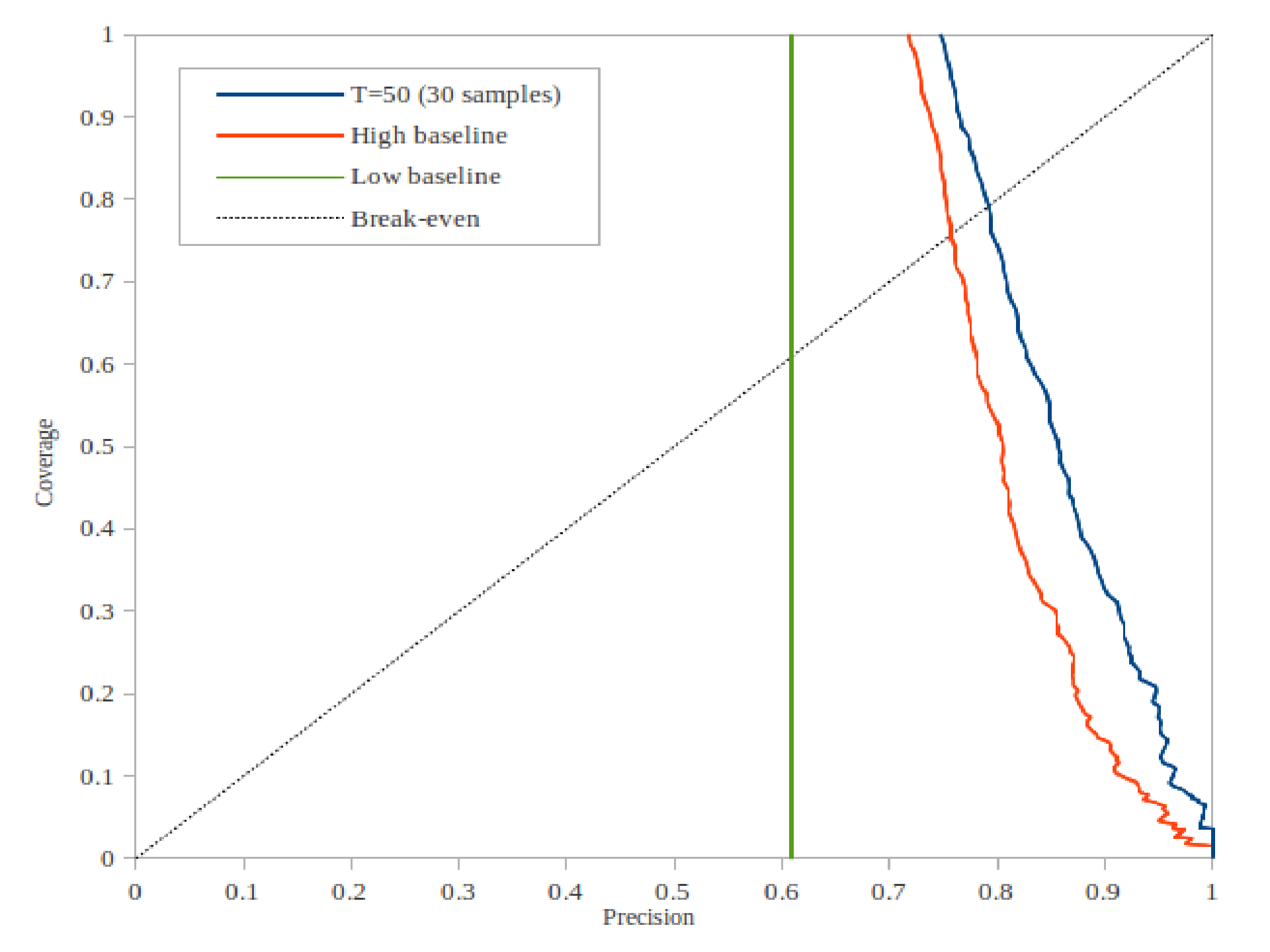
- High baseline
  - trigram frequencies (v, p, n_2), (n_1, p, n_2)
- Low baseline
  - preposition attachment frequencies

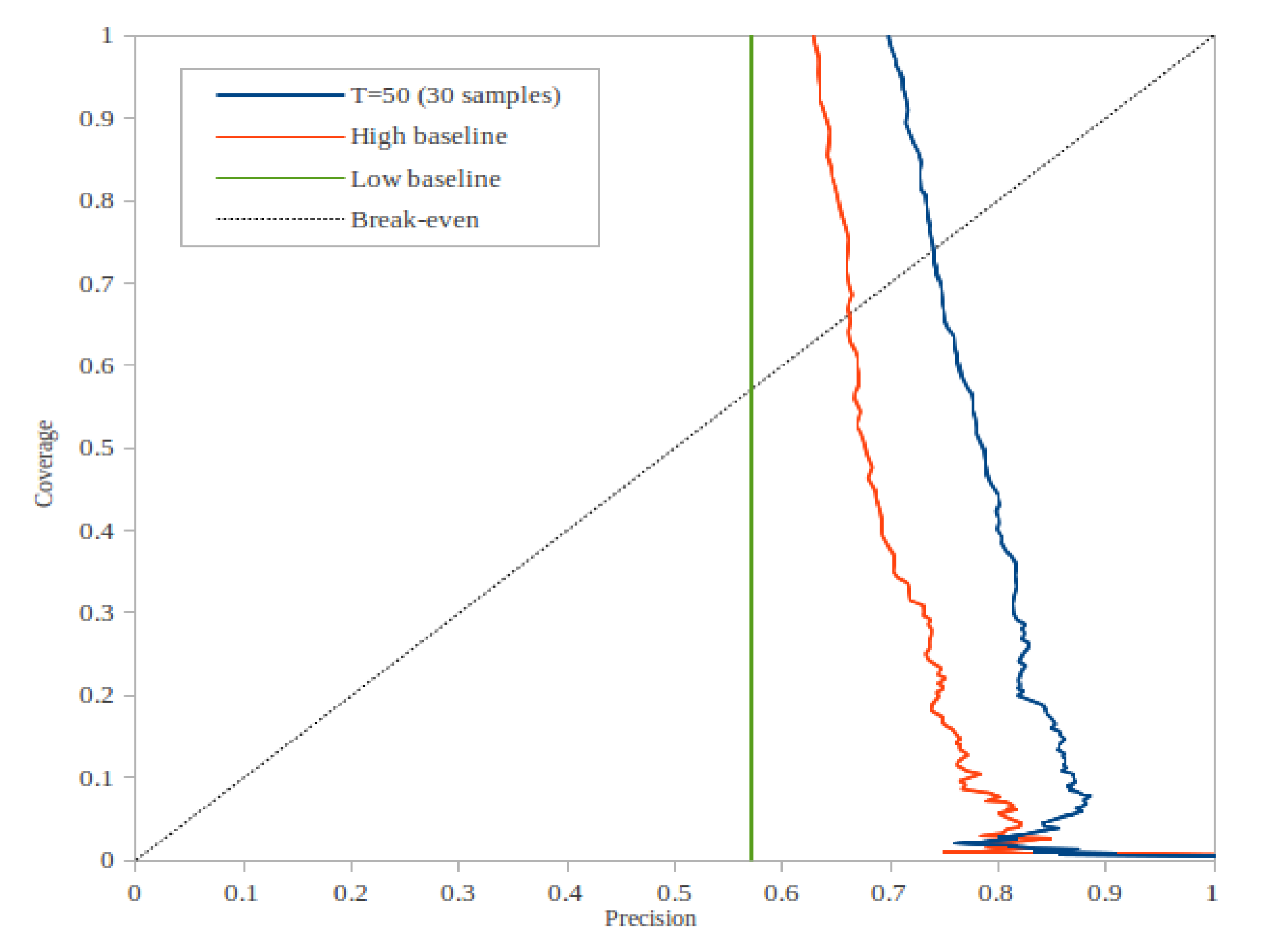
Results (WeScience)

Model	Accuracy
$T = 35$	0.744
$T = 50$	0.747
$T = 70$	0.738
$T = 300$	0.738
High baseline (trigram)	0.718
Low baseline (preposition)	0.609

Results (Penn Treebank)

Model	Accuracy
$T = 35$	0.701
$T = 50$	0.698
$T = 70$	0.700
$T = 300$	0.680
High baseline (trigram)	0.629
Low baseline (preposition)	0.571


T=50 (30 samples)
High baseline
Low baseline
Break-even
Coverage
Precision
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


T=50 (30 samples)
High baseline
Low baseline
Break-even
Coverage
Precision
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Conclusion

- Distributional information can be effectively used to rank parses
- LDA can overcome data sparsity
- Robust unsupervised training

Future Work

- Extend to other relations
- Integrate with MaxEnt parser

Notes to slides

1: The presentation has two halves: first, I present a general framework for incorporating distributional semantic information into a parser, and explain a method for overcoming data sparsity; second, I evaluate this method when applied to the specific problem of PP-attachment ambiguity.

2: I will focus on syntactic or semantic relations.

4: I will not consider lexical ambiguity, but it is essentially an independent problem.

7: In PP-attachment, lexical information is crucial for disambiguation – in these two sentences, exactly the same syntactic structures are being compared.

8: Given a relation r, we can define $P_r(x,y)$ to be the probability of observing the lexical items x and y in that relation. We can then define a plausibility score, as shown in the slide, by dividing by the background probabilities of observing those lexical items. This means that the score should be independent of the relative frequencies of those lexical items, and only depend on how closely they fit together in this relation.

9: To incorporate these scores into a maximum entropy parser, we can define a feature for each relation, whose value is the score (if the relation appears multiple times in parse, then the scores can simply be summed).

10: For PP-attachment, we can consider 4-tuples of the form (*verb, noun, preposition, noun*) such as (*bite, sheriff, with, moustache*). Given such a 4-tuple, we would like to decide if the PP attaches to the verb, or to the first noun. In general, there could be multiple attachment sites, but I consider this simpler case for ease of analysis.

The formula can be derived by applying Bayes' Theorem, and making two independence assumptions: if the PP attaches to the verb, the direct object is generated independently; and if the PP attaches to the noun, the verb is generated independently.

11: This formula can be rewritten in terms of the plausibility scores defined previously, where the preposition defines two relations (one for verbal attachment and one for nominal attachment). Hence, as long as these independence assumptions are valid, a maximum entropy parser equipped with such features should be able to calculate the correct answer.

12: To calculate these probabilities, we could directly use observed frequencies of triples of the form (*verb, preposition, noun*) and (*noun, preposition, noun*). However, such an approach will be limited by data sparsity. To overcome this, we can use a modified version of Latent Dirichlet Allocation (LDA).

13: To gain an intuition about how LDA will help, we can consider this cluster of words, where any of the verbs could plausibly combine with any of the nouns. However, while many of the common combinations may be observed, such as (*walk, down, street*) we may not observe a rarer combination such as (*gallop, down, boulevard*). LDA allows us to infer from other observations that these words can plausibly be combined.

[Aside: This is similar in spirit to using WordNet supertypes, which, as Francis mentioned, can improve performance.]

14: We define a generative process in two stages: from a preposition, we first generate a topic (such as the one seen on the previous slide); then from this topic, independently generate a verb and a noun (or, in the case of nominal attachment, two nouns).

15: A preposition is modelled as a mixture of topics.

16: The topic, verb, and noun are all generated every time we observe the relation. The topic mixture is generated once per preposition.

Each topic corresponds to two distributions, one over verbs and one over nouns. So, the overall process is to generate a topic, find the corresponding distributions, and generate a verb and noun from these.

Given a set of observed data, we would then like to infer these distributions, since they would allow us to make predictions about new data. However, to perform inference, we are forced to have assumptions about what kind of distributions to expect. These assumptions are made explicit in the three prior distributions α, β, γ. Intuitively, these give us some control over how clustered we expect the data to be. They are all Dirichlet distributions (hence the name LDA), but this is a mathematical convenience, and there is nothing linguistically meaningful to be said.

17: WikiWoods is a useful resource, both for its size, and for its rich annotations.

Note that unlike a discriminative model, the model can learn from all positive examples, not just ambiguous ones.

The prepositions were chosen because they are both common and have a roughly even split between nominal and verbal attachment in the corpus.

[Aside: since we are trying to improve parse ranking by using the output of a parser, we can see this as a method of self-training.]

18: The most likely nouns from a single topic are given. There is a clear semantic interpretation (something like a BUILDING in an AREA), but note that it does not quite correspond to an obvious supertype, as we have words such as *street*.

19: This topic also has a clear interpretation, but note that *election* has appeared along with the other warlike terms. I leave the reader to decide if this is a reasonable conclusion.

20: I used two corpora for evaluation: the first is WeScience, which is useful because it is in the same domain as WikiWoods (encyclopaedia text), and employs the same annotation conventions. The second is the Penn Treebank, which has been widely studied, so it allows comparison with other methods to disambiguate PP-attachment; however, it is in a different domain and relies on different annotation conventions, so we can expect performance to be lower.

21: The results can be compared against two baselines: using the observed trigrams directly, without LDA smoothing; and attaching based on the preposition alone, as in an unlexicalised model.

22: Using LDA smoothing provides about a 3 percentage point improvement over the unsmoothed model. Note also that the performance is fairly robust to changes in the number of topics, which means that the model could be easily trained without manual assistance.

23: On the Penn Treebank data, the overall performance is lower, but the improvement on the unsmoothed model is larger.

24: Since the model is probabilistic, it gives probabilities of attachment, not just a binary classification. We can choose to only accept the model's decision if the probability is over some threshold, and otherwise not commit to a decision. By increasing this threshold, we can hope to increase precision while losing coverage. This is shown graphically, for the WeScience dataset – the top of the graph corresponds to the figures given on the previous slide, where there is full coverage. As the threshold is increased, precision increases and coverage drops, bringing us to the bottom right of the graph. The difference between the red and blue lines shows the effect of the LDA smoothing.

25: The same method is applied to the Penn Treebank data.