Taxonomy Induction using Hierarchical Random Graphs

Trevor Fountain Mirella Lapata

Institute for Language, Cognition, and Computation
School of Informatics, The University of Edinburgh

Introduction

The semantic knowledge encoded in lexical taxonomies like WordNet is invaluable for countless tasks:

- question answering (Harabgiu et al., 2003)
- document classification (Hung et al., 2004)
- textural entailment (Geffet and Dagan, 2005)

Introduction

The semantic knowledge encoded in lexical taxonomies like WordNet is invaluable for countless tasks:

- question answering (Harabgiu et al., 2003)
- document classification (Hung et al., 2004)
- textural entailment (Geffet and Dagan, 2005)

...but difficult and expensive to create.

Introduction

Automatic taxonomy induction:

- **Term extraction** – finding the concepts to be taxonomised (Kozareva eta al., 2008; Navigli et al., 2011)
- **Term relation discovery** – learning semantic relations (e.g. IS-A between terms; Hearst, 1992; Berland and Charniak, 1999)
- **Taxonomy construction** – creating the taxonomy by organising a set of terms into an hierarchical structure (Kozareva and Hovy, 2010; Navigli et al., 2011)

Introduction

Traditional taxonomy construction methods:

- are often supervised or semi-supervised.
- generally operate directly on corpora.
- often impose a pre-determined structure on output
- build a single ‘correct’ taxonomy

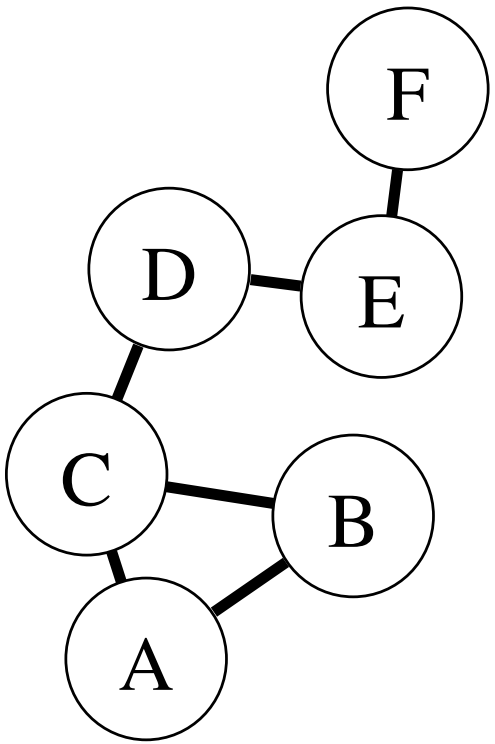
Hierarchical Random Graphs

Clauset et al.'s (2008) **Hierarchical Random Graph** provides a method for inferring hierarchical structure from networks.

- The algorithm is **unsupervised**.
- Operates on **networks** (i.e., an intermediate representation) rather than directly on corpora.
- Uses a **model averaging technique** to infer the depth and complexity of an hierarchy.
- Different from **hierarchical clustering**: does not seek a single representation for a given network.

Hierarchical Random Graphs: Input

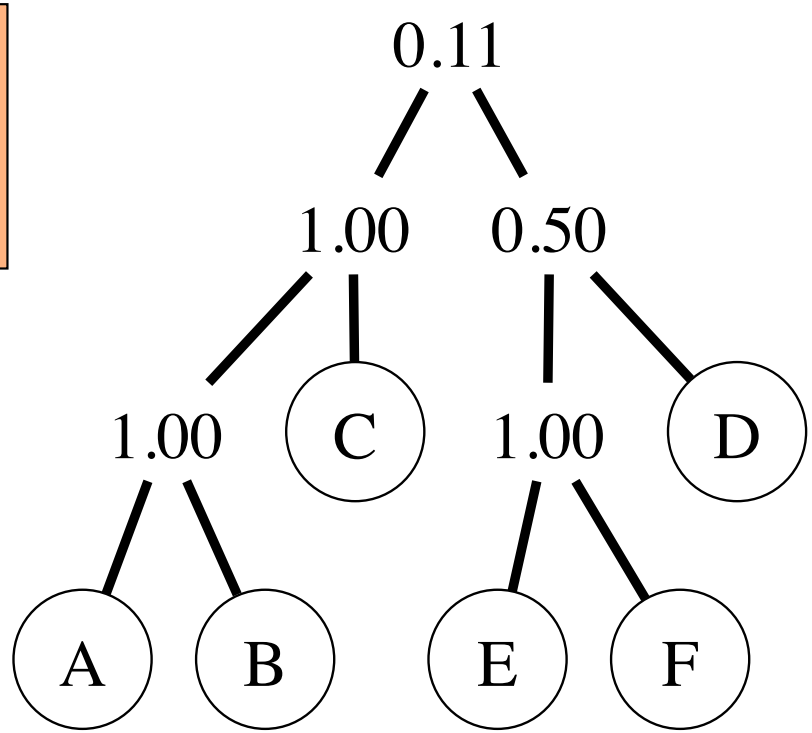
A **semantic network** is an undirected graph in which nodes represent terms and edges between nodes indicate a semantic relationship between pairs of terms.



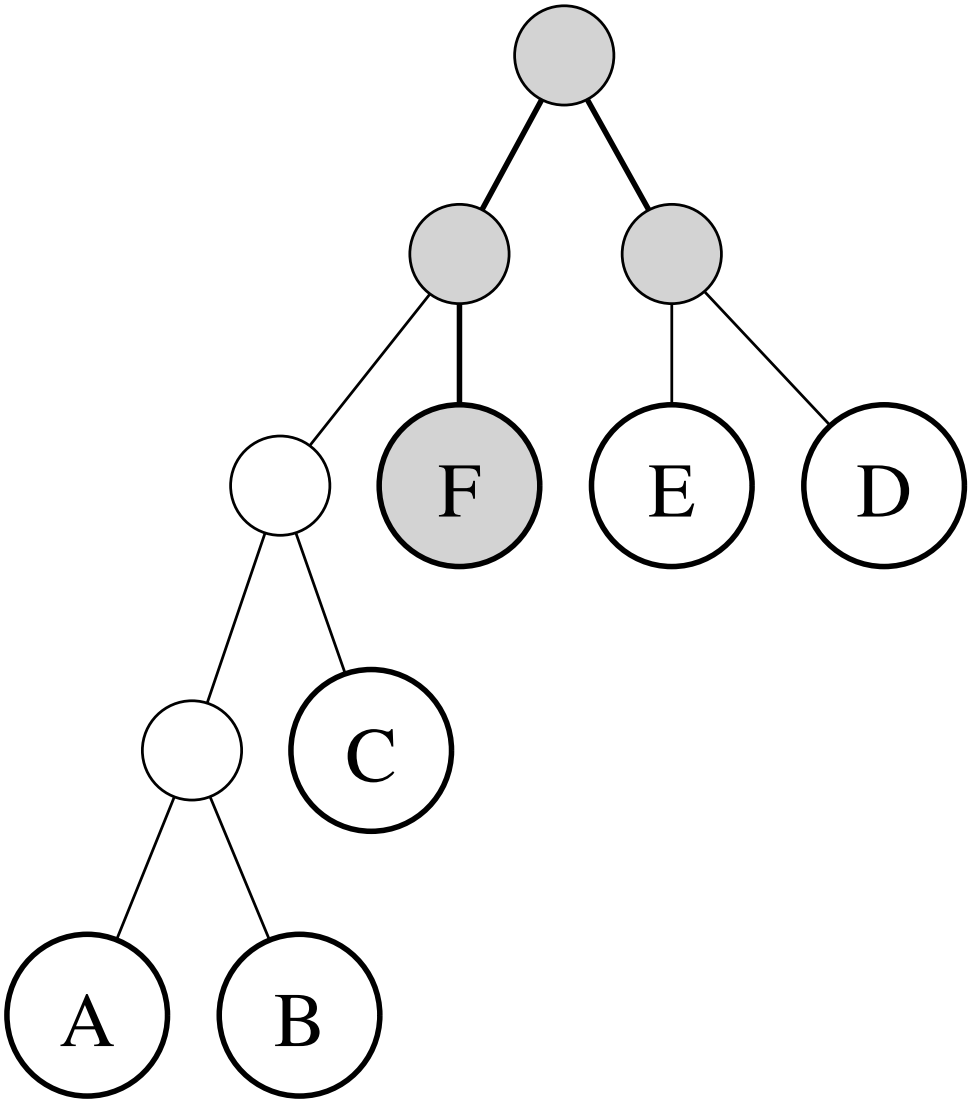
Hierarchical Random Graphs: Sampling

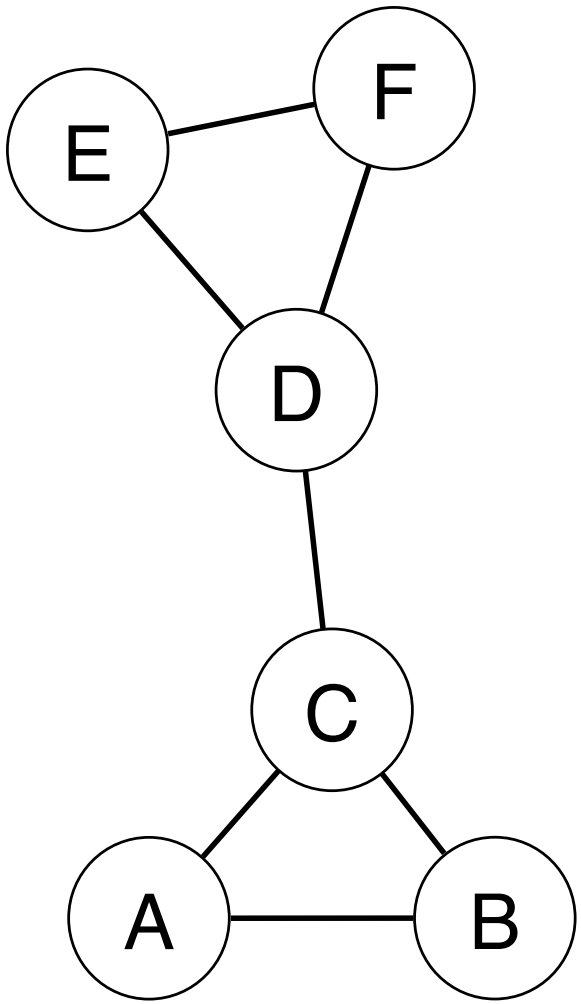
For sampling, construct a **binary tree** with leaves corresponding to nodes in the semantic network.

- Randomly construct initial binary tree D over nodes in input semantic network S
- Compute likelihood of D given S
- Resample using MCMC

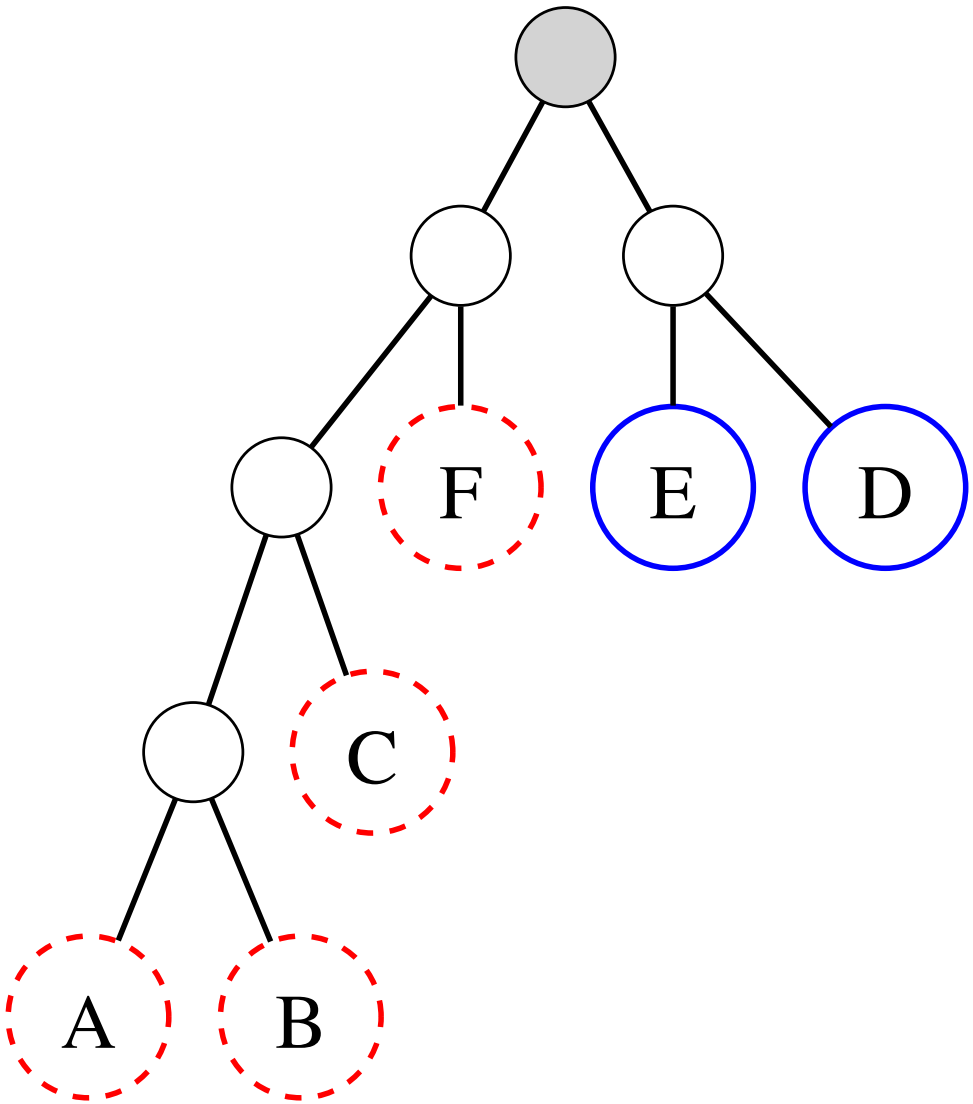


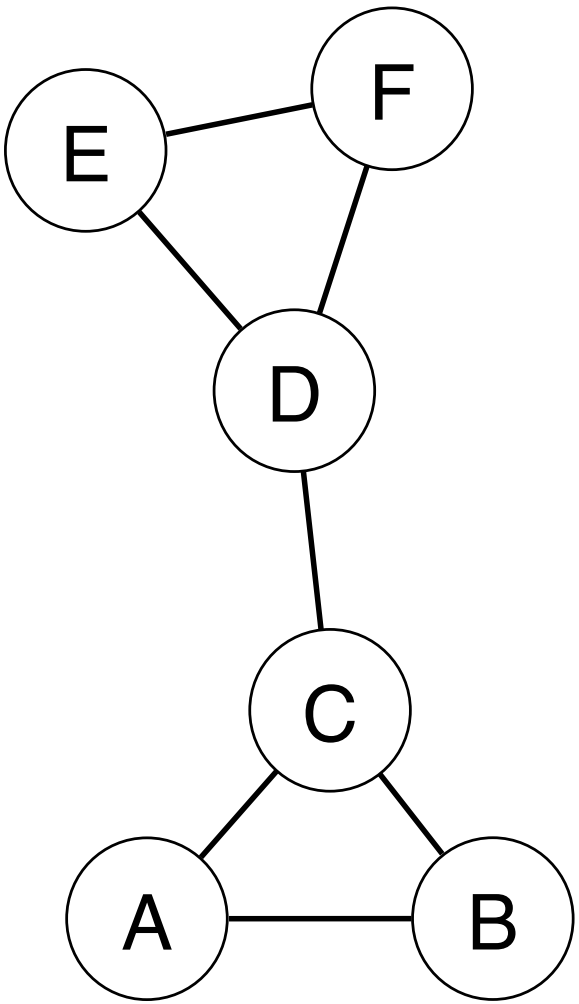
Hierarchical Random Graphs: Likelihood



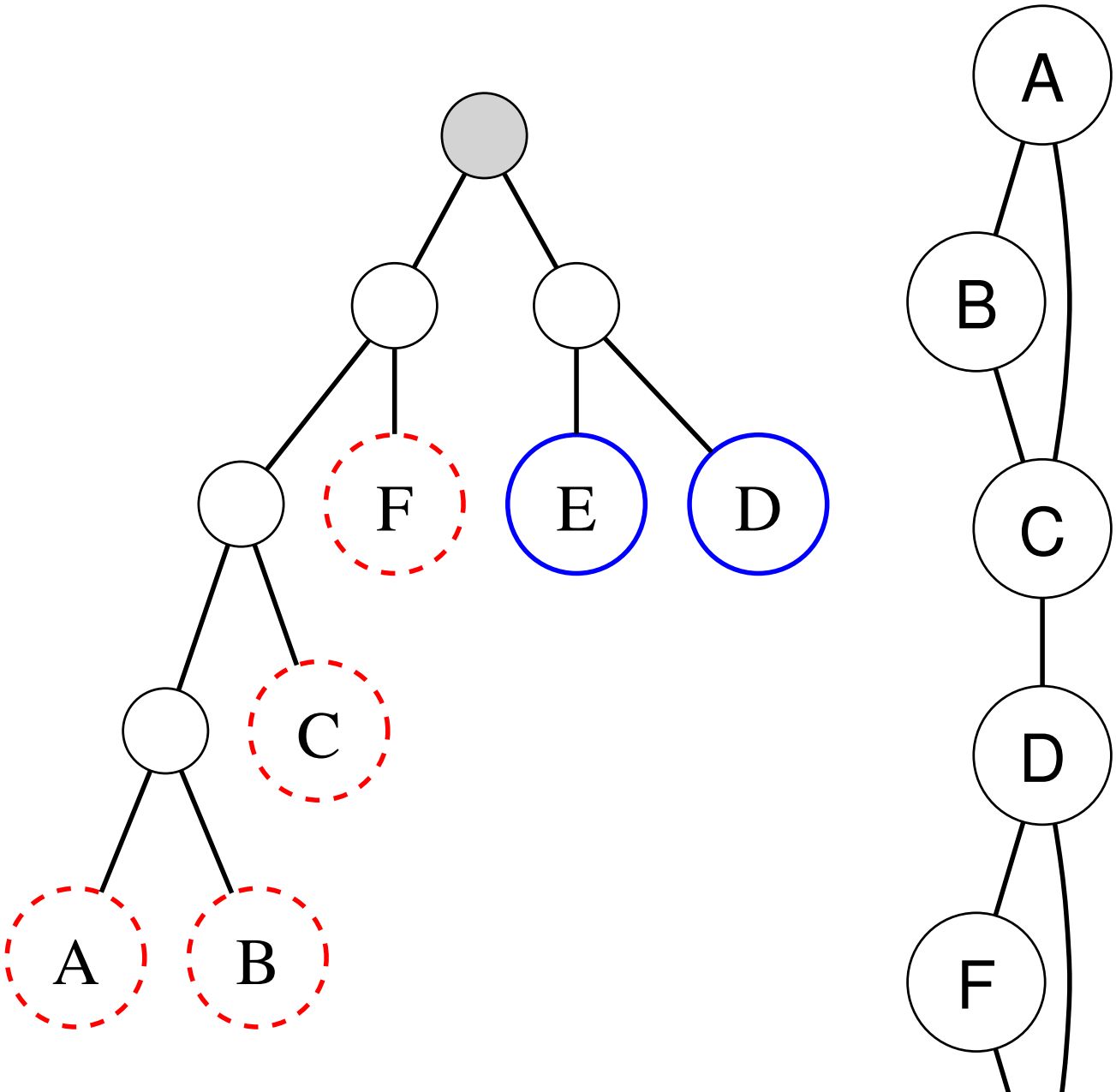


Hierarchical Random Graphs: Likelihood

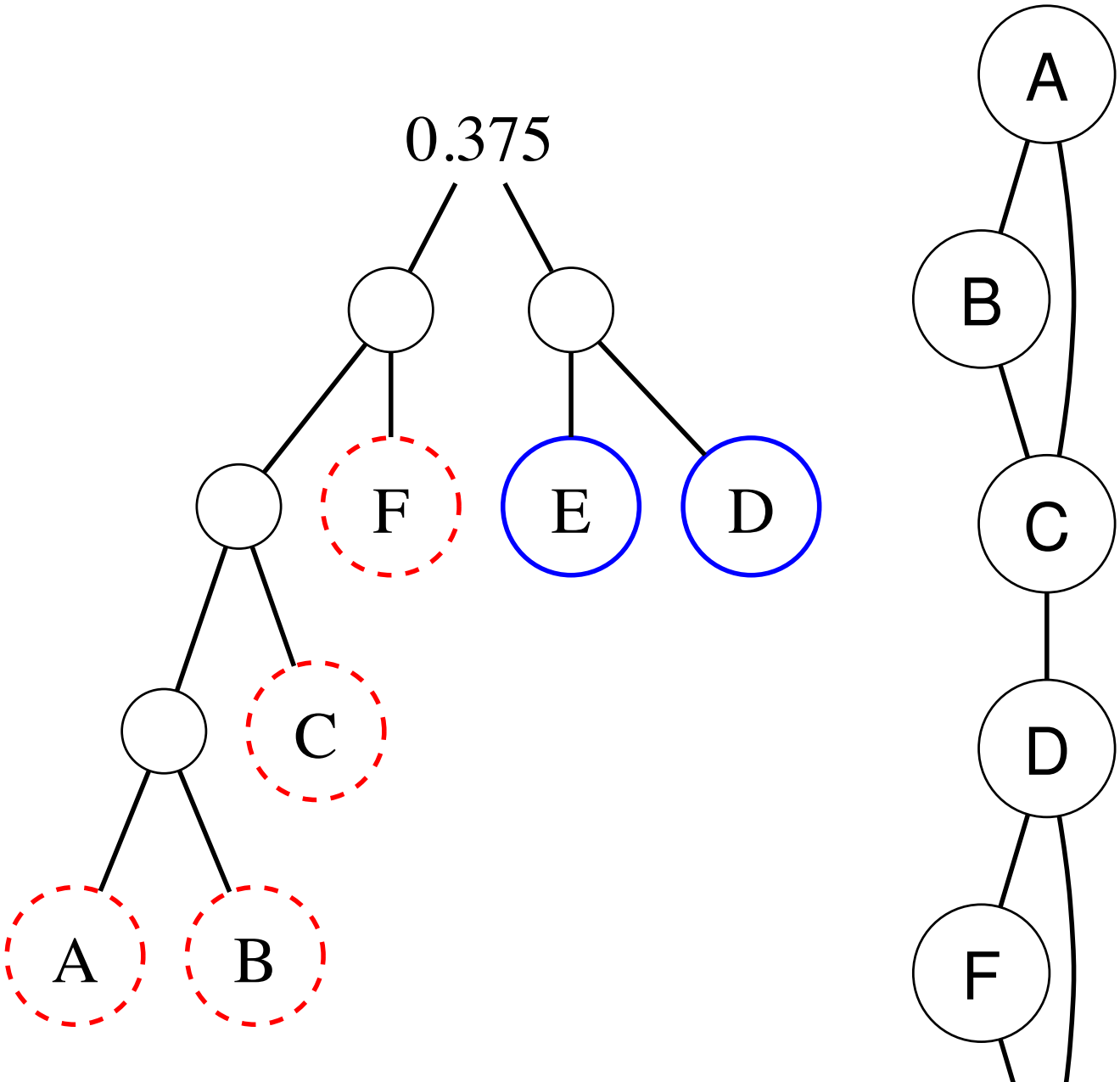




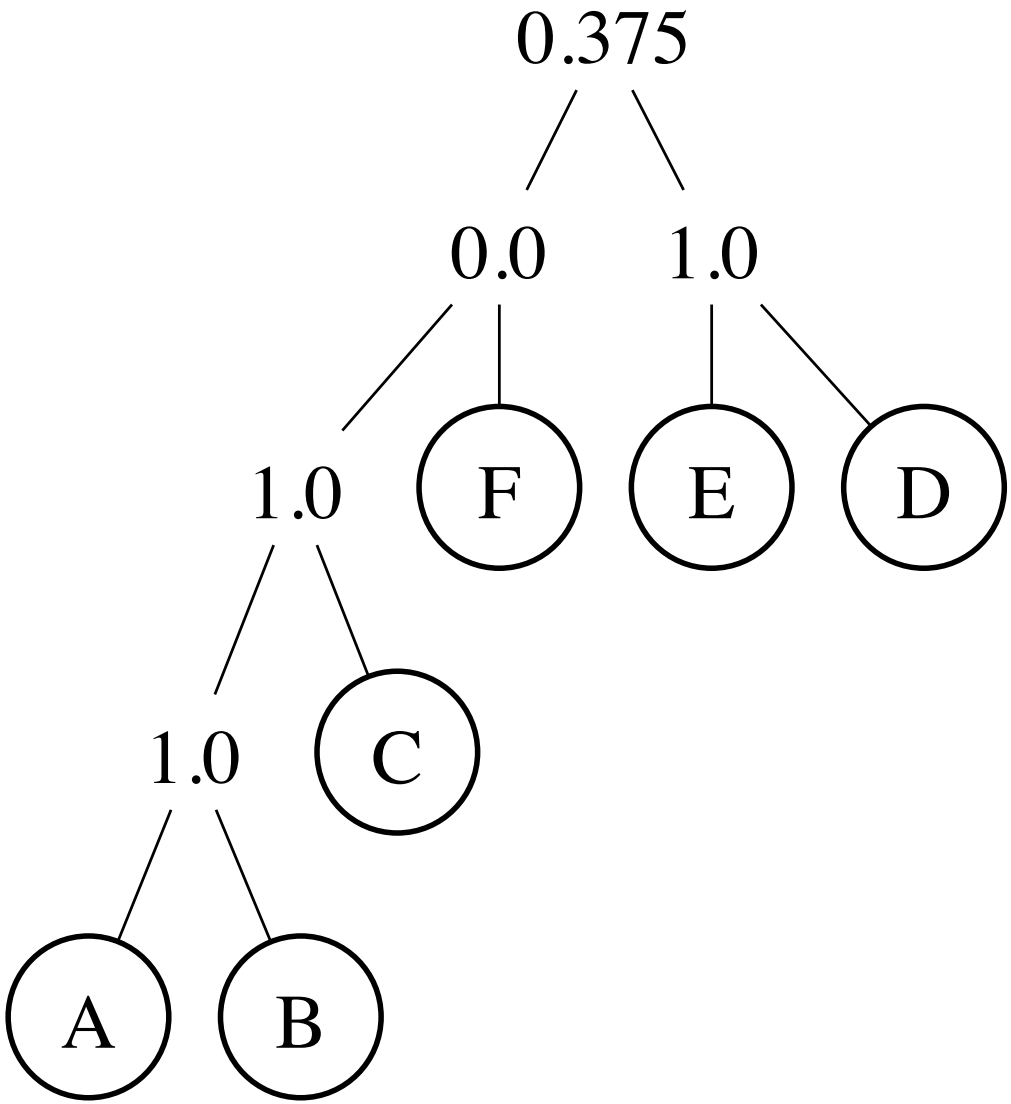
Hierarchical Random Graphs: Likelihood

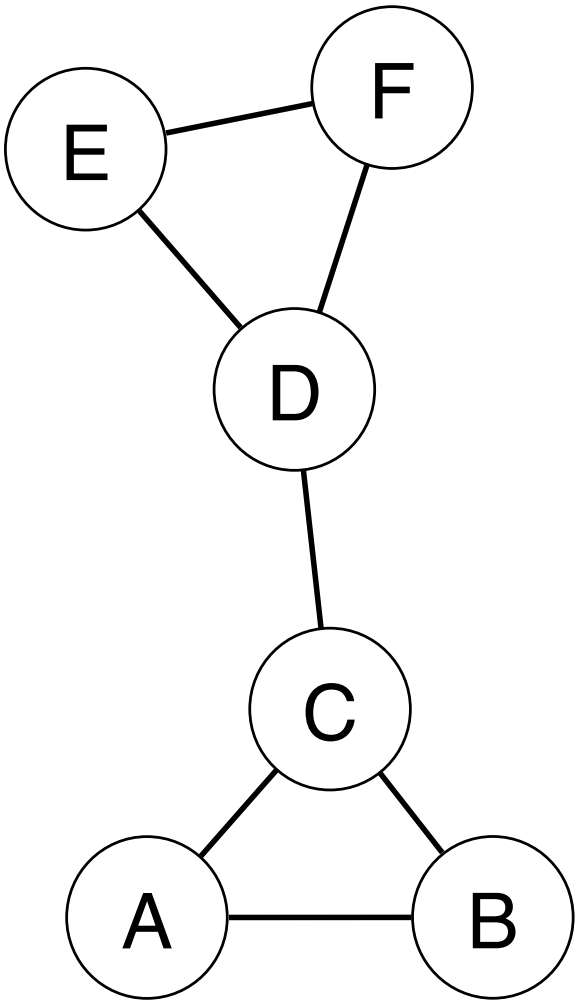


Hierarchical Random Graphs: Likelihood

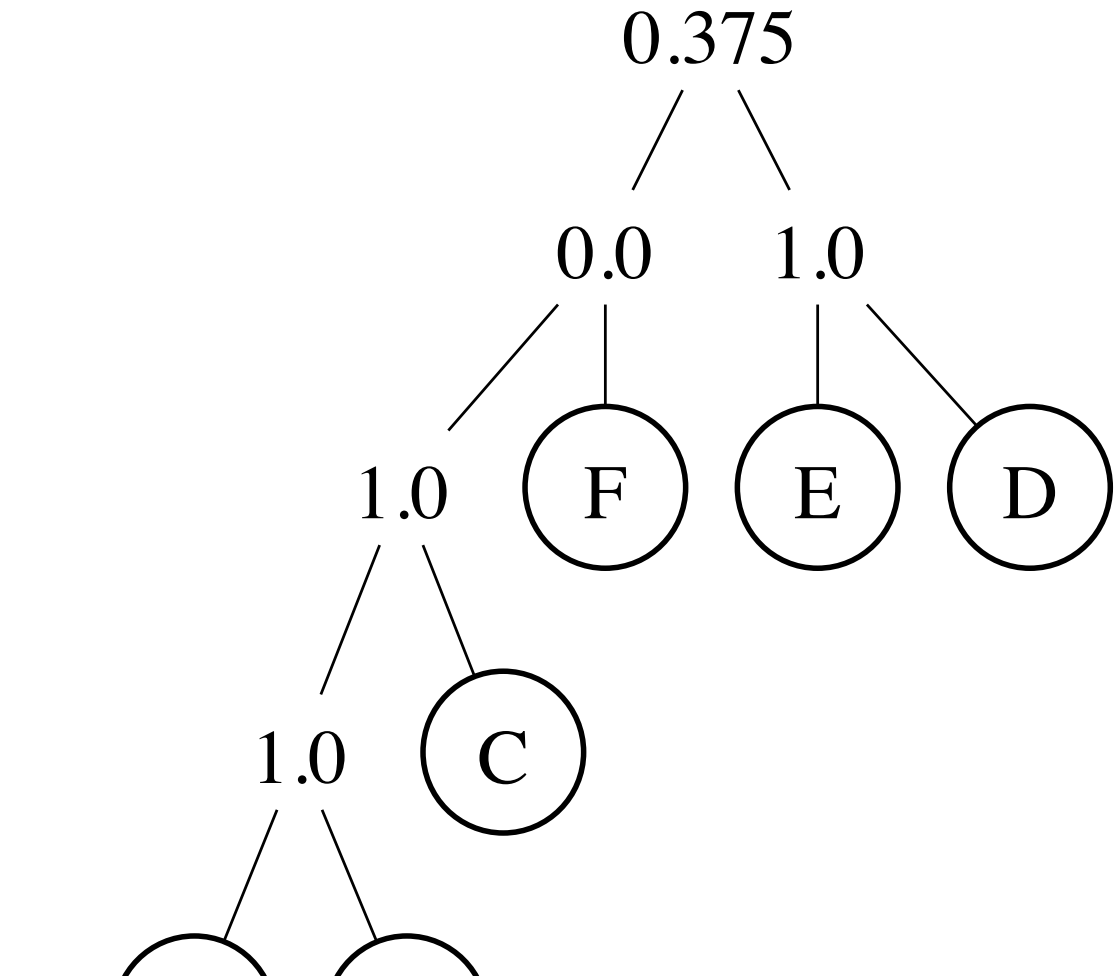


Hierarchical Random Graphs: Likelihood

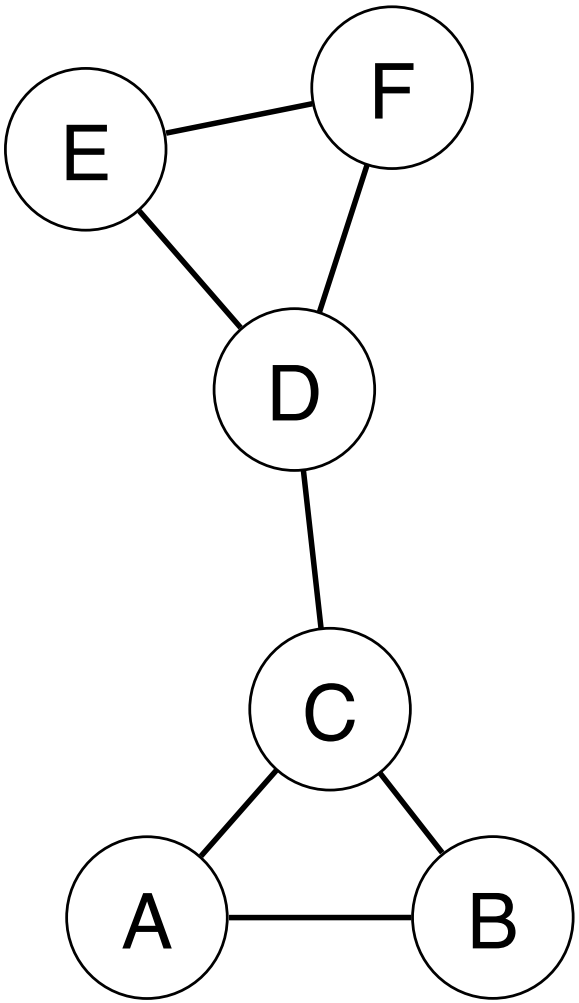




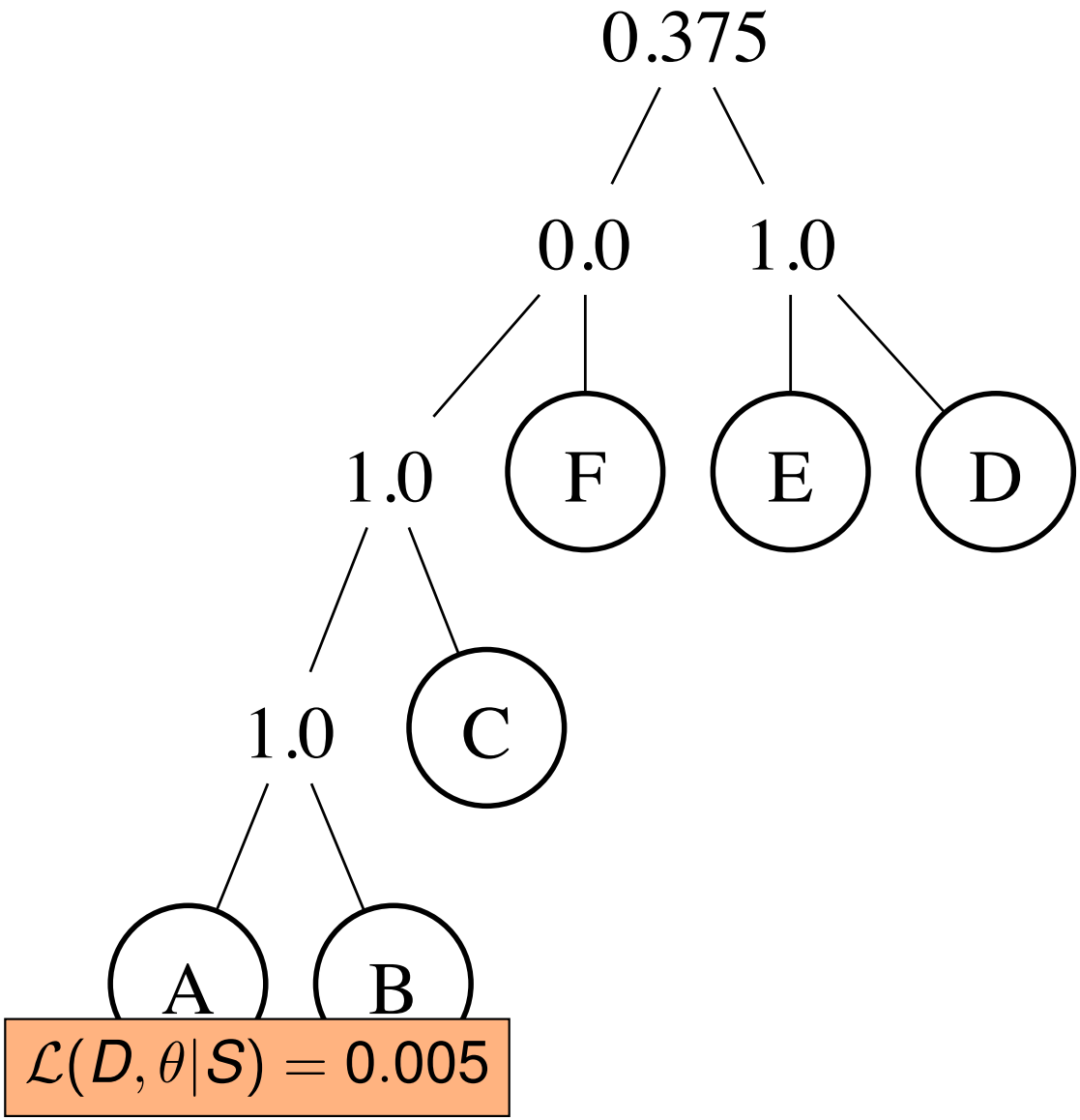
Hierarchical Random Graphs: Likelihood

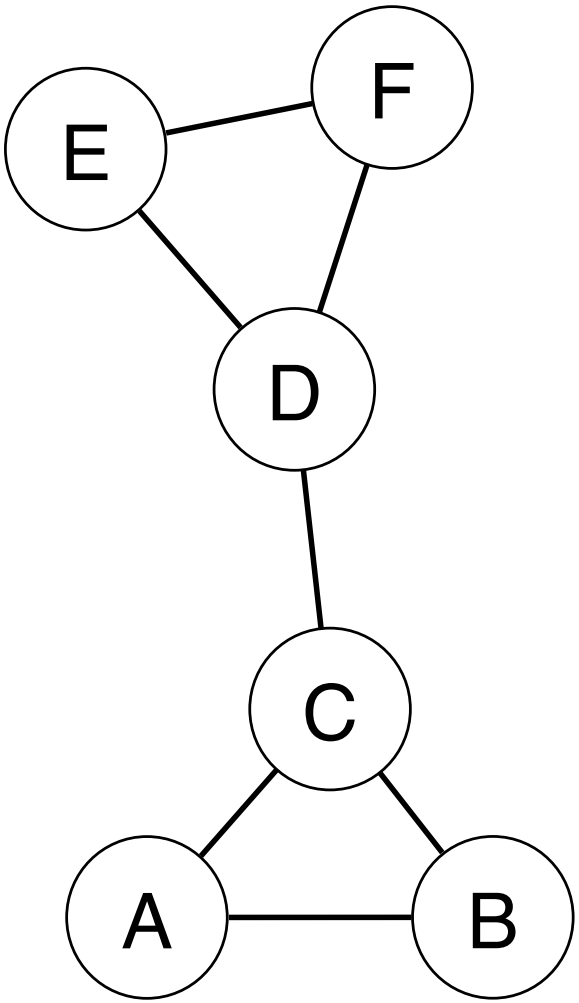


$$\mathcal{L}(D, \theta | S) = \prod_{i=1}^{n-1} (\theta_i)^{E_i} (1 - \theta_i)^{|L_i||R_i| - E_i}$$



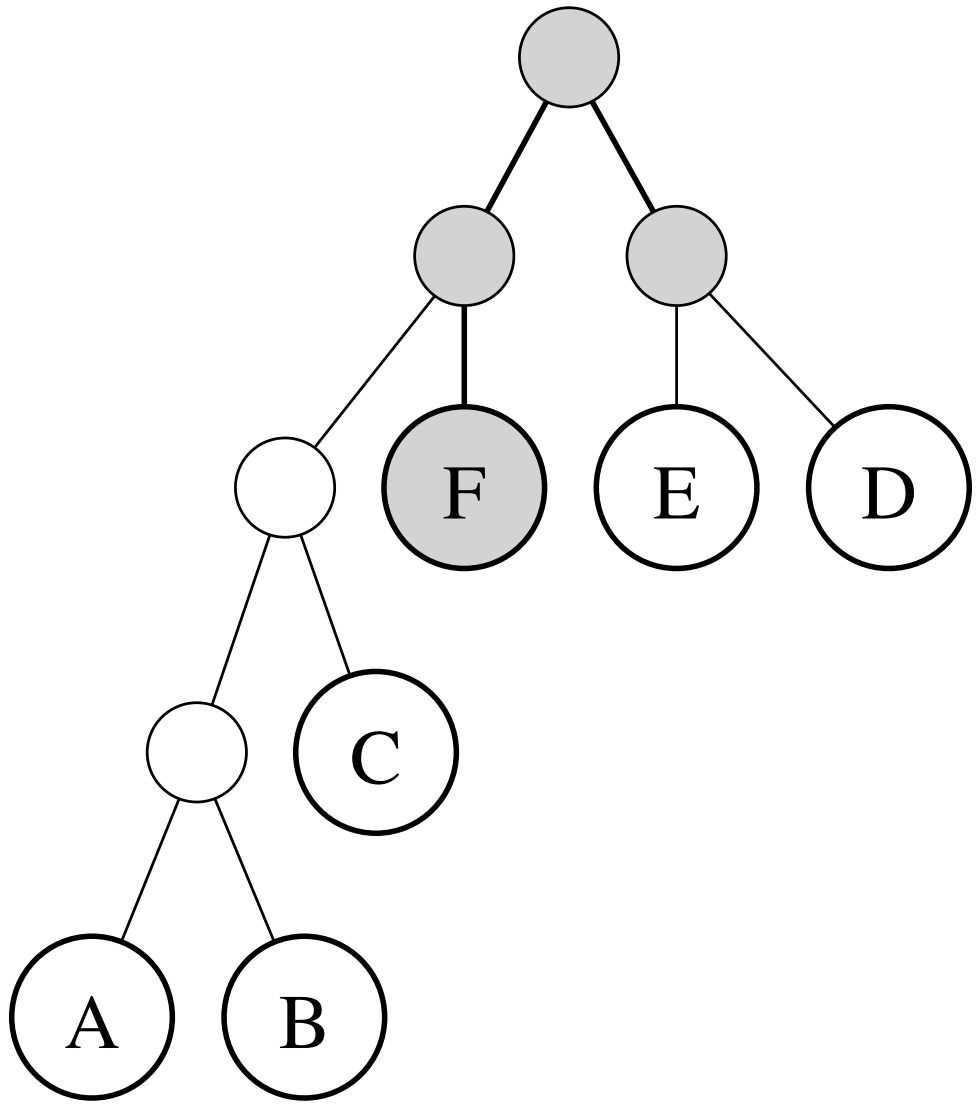
Hierarchical Random Graphs: Likelihood





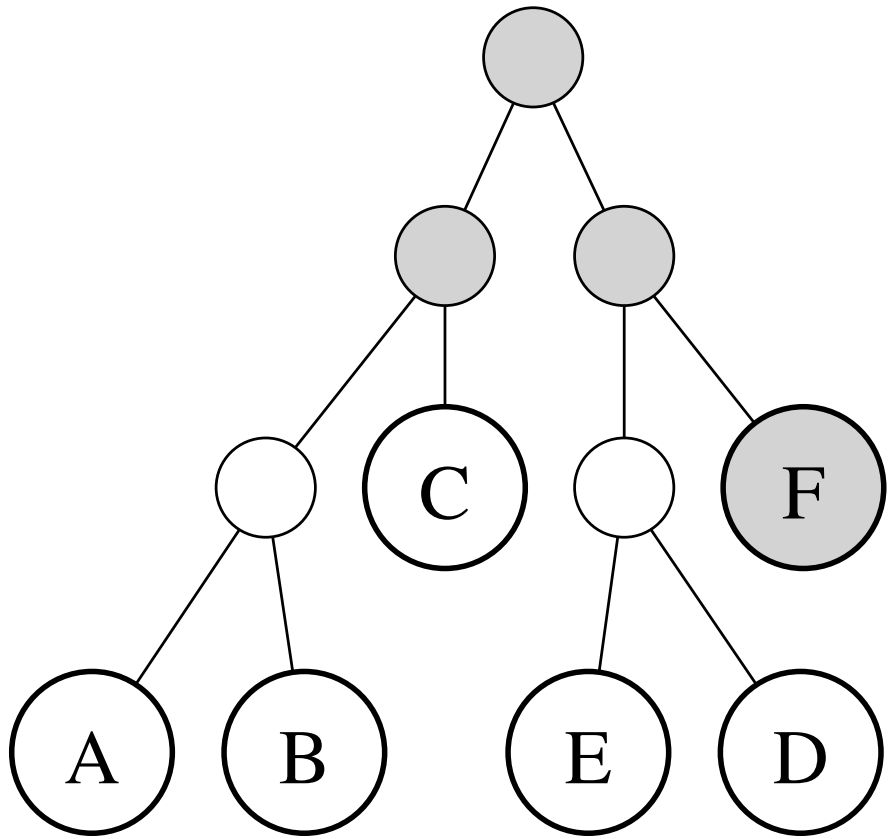
Hierarchical Random Graphs: Sampling

1. Select a random internal node



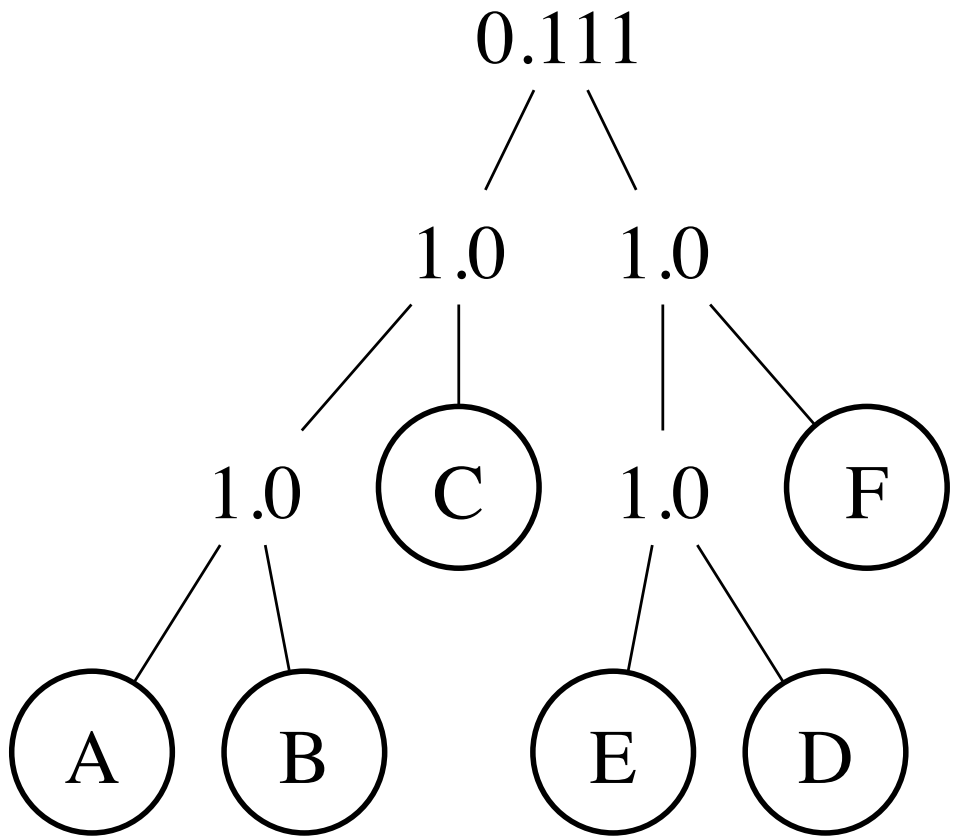
Hierarchical Random Graphs: Sampling

1. Select a random internal node
2. Permute



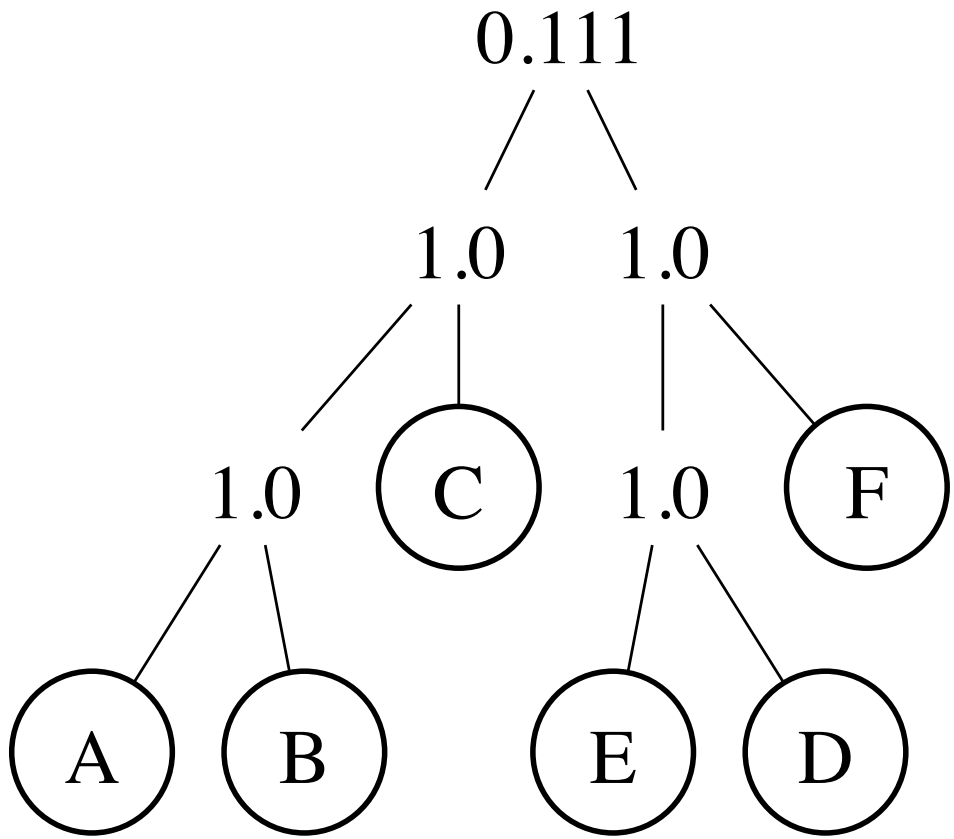
Hierarchical Random Graphs: Sampling

1. Select a random internal node
2. Permute
3. Re-compute θ parameters



Hierarchical Random Graphs: Sampling

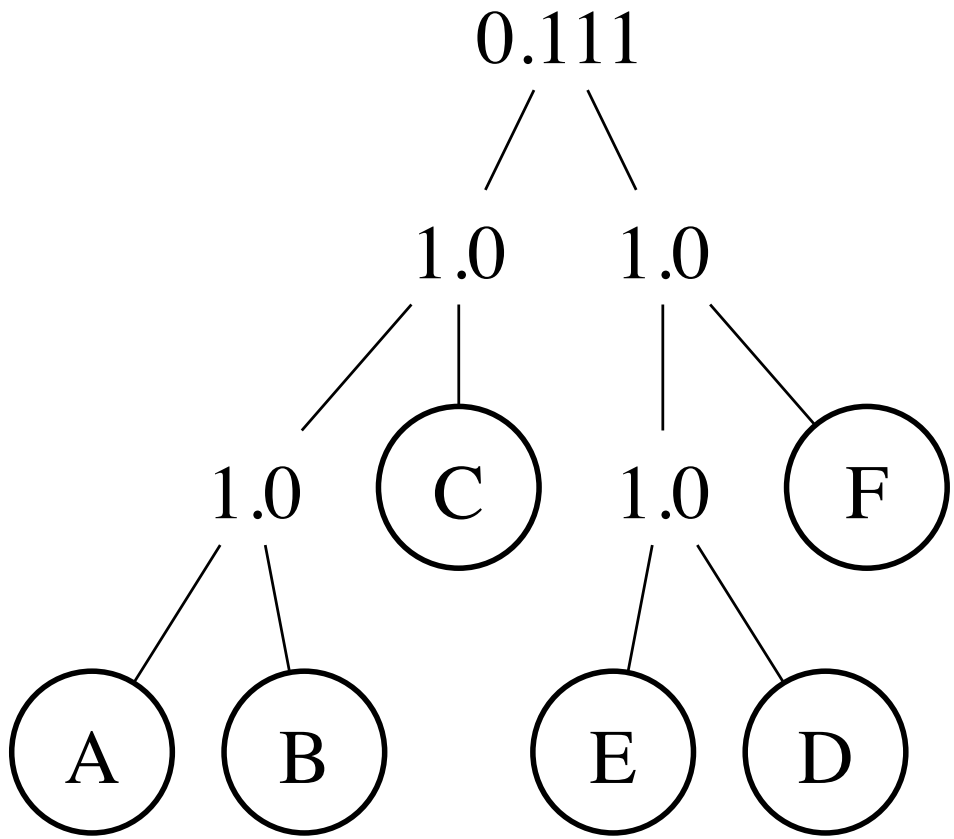
1. Select a random internal node
2. Permute
3. Re-compute θ parameters
4. Re-compute likelihood



$$\mathcal{L}(D, \theta | S) = 0.043$$

Hierarchical Random Graphs: Sampling

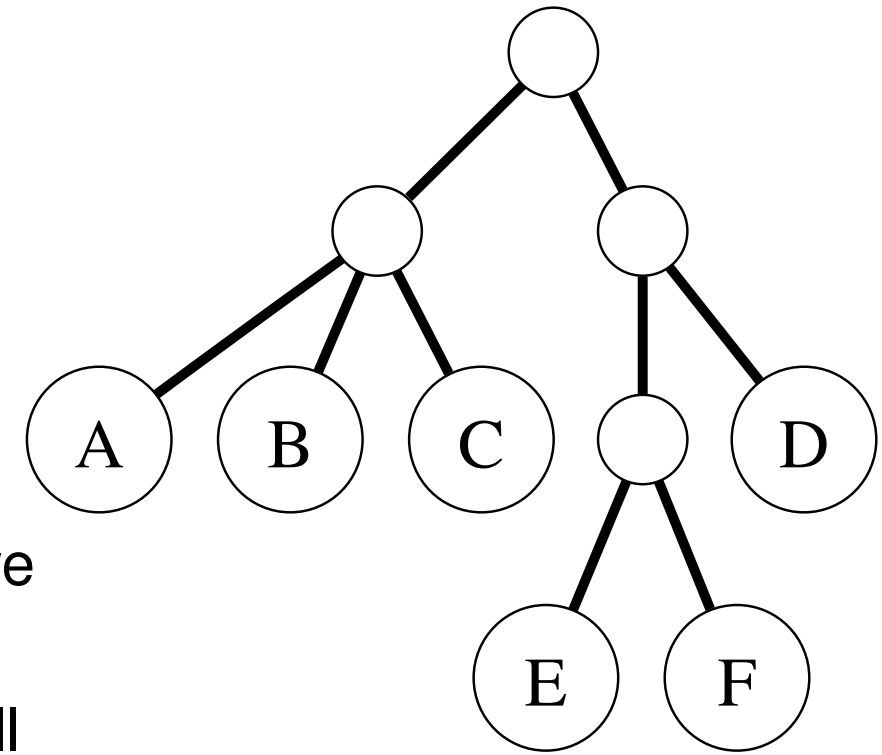
1. Select a random internal node
2. Permute
3. Re-compute θ parameters
4. Re-compute likelihood
5. Accept or reject transition and repeat



$$\mathcal{L}(D, \theta | S) = 0.043$$

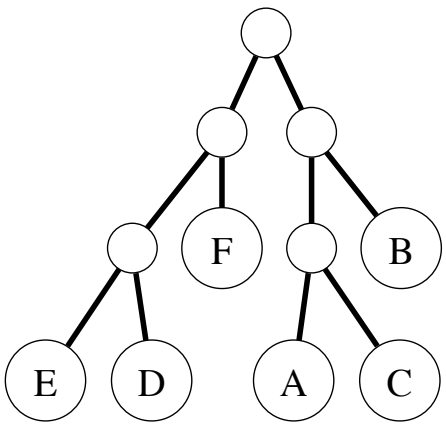
Hierarchical Random Graphs: Consensus

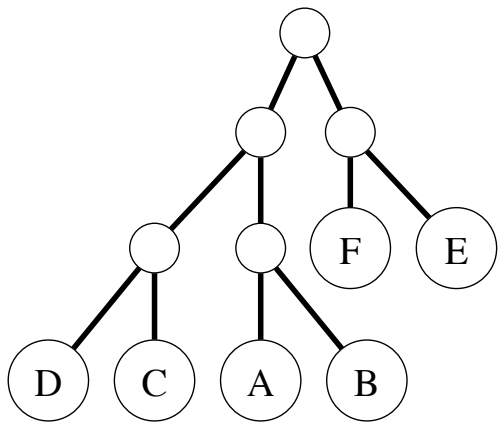
A **consensus hierarchy** is the output of model averaging (does not impose binary tree structure on the inferred taxonomy).

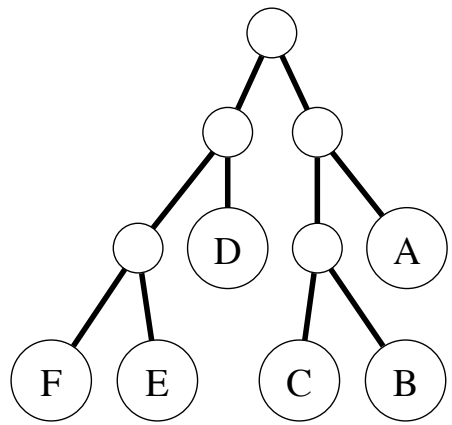


- After MCMC reaches consensus, we sample multiple trees
- Identify subtrees common across all
- These recombine into taxonomy in which each subtree appears in majority of sampled trees.

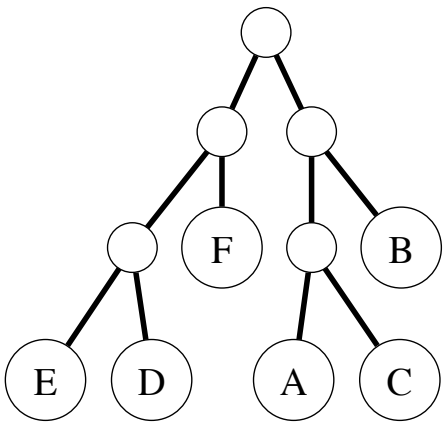
Hierarchical Random Graphs: Consensus



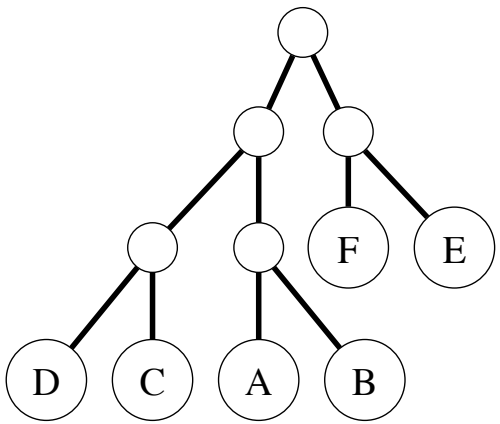




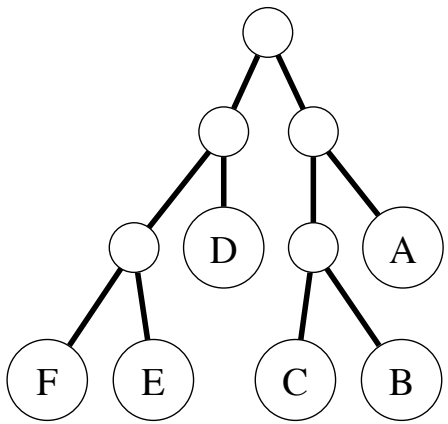
Hierarchical Random Graphs: Consensus



D E
D E F
A C
A B C
A B C D E F

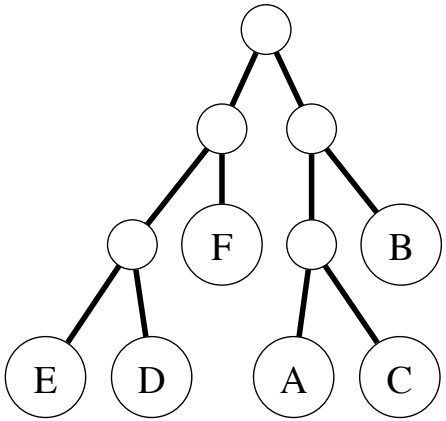


C D
A B
A B C D
E F
A B C D E F



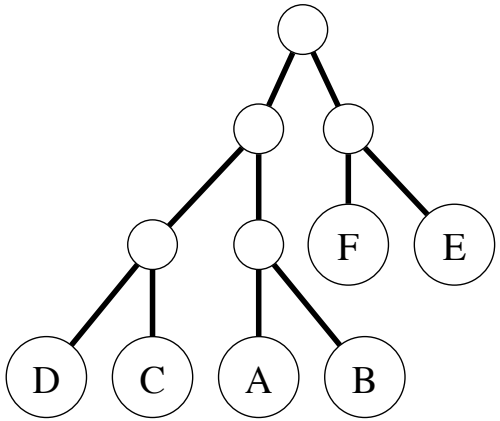
E F
D E F
B C
A B C
A B C D E F

Hierarchical Random Graphs: Consensus

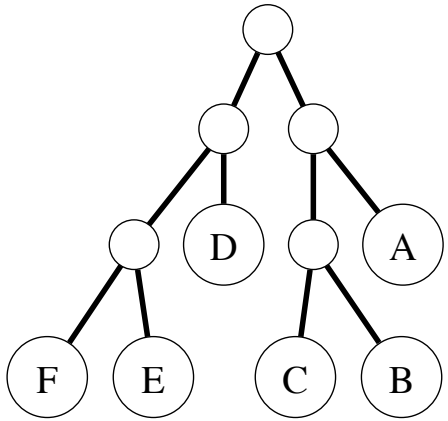


D E F

A B C
A B C D E F



E F
A B C D E F

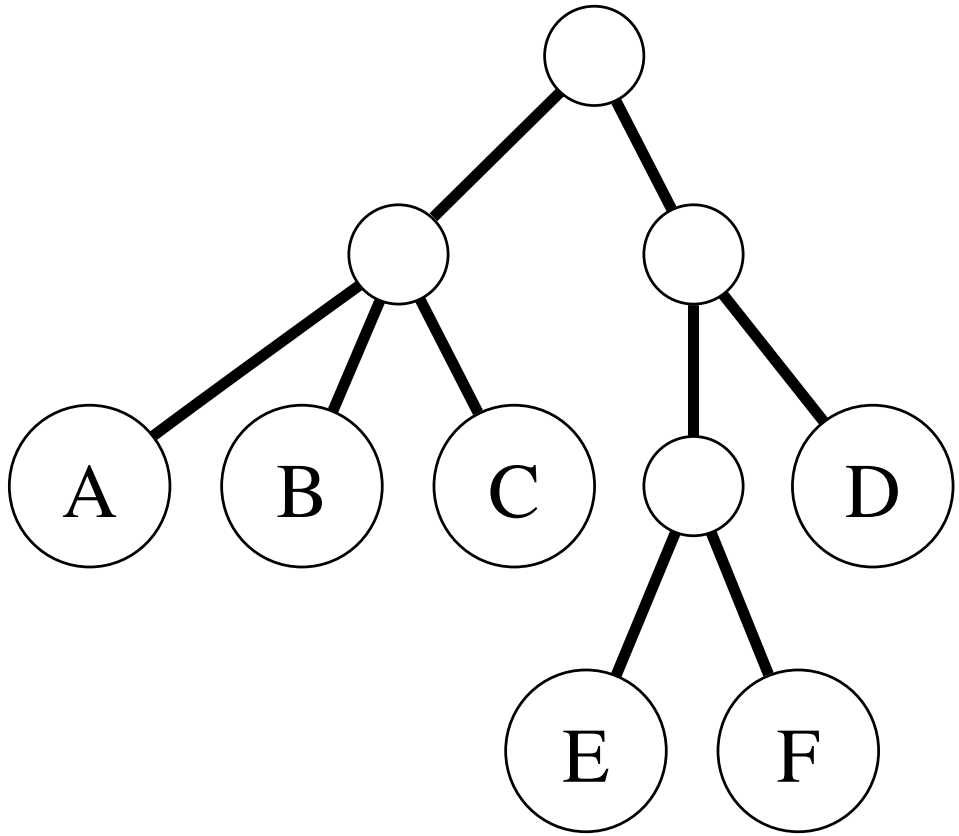


E F
D E F

A B C
A B C D E F

Hierarchical Random Graphs: Consensus

E F
D E F
A B C
A B C D E F



Evalution

Evaluate taxonomies using two measures:

- a **cluster-**based measure, F-score
- a **hierarchy-**based measure, tree-height correlation

Cluster Evaluation

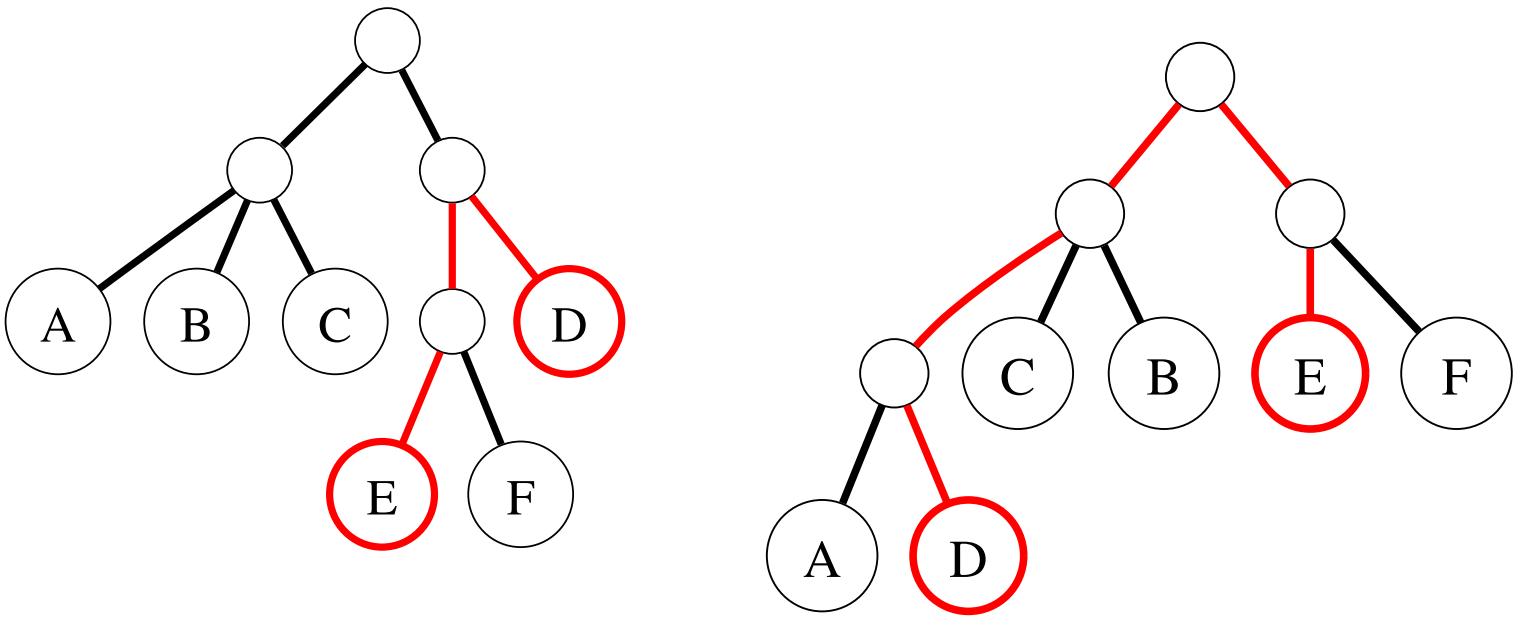
- Flatten output taxonomy into clusters and compare against a gold-standard clustering.
- Compute the cluster F-Score (Agirre and Soroa, 2007)
- For a candidate cluster c and the corresponding gold cluster g to which it is most similar:

$$\textit{Precision} = \frac{|c \cap g|}{|c|}$$

$$\textit{Recall} = \frac{|c \cap g|}{|g|}$$

Taxonomy Evaluation

Construct a plausible hierarchy and compare directly using a taxonomy similarity measure: **tree-height correlation**



Taxonomy Evaluation

Let $G = \{g_{0,1}, g_{0,2} \cdots g_{n,n-1}\}$ where $g_{a,b}$ is the walk distance between a and b in the gold taxonomy.

Let $C = \{c_{0,1}, c_{0,2} \cdots c_{n,n-1}\}$ where $c_{a,b}$ is the walk distance between a and b in the candidate taxonomy.

Compute the **tree-height correlation** as Spearman's ρ between G and C.

Evaluation Data

Gold-standard clusters:

- 541 words from McRae et al.'s (2005) feature norms
- 41 clusters from category naming study (Fountain and Lapata, 2010)

Gold-standard taxonomy:

- Construct taxonomy over 541 words using WordNet
- Find full hypernym path from each word to root (e.g. APPLE > PLANT STRUCTURE > NATURAL OBJECT > PHYSICAL OBJECT > ENTITY)
- Merge paths to form a full taxonomy
- Recursively remove single-child nodes

Baselines

Evaluate these two measures against three baselines:

- Chinese Whispers (Biemann, 2006), a flat clustering algorithm
- The Brown et al. (1992) agglomerative clustering algorithm
- Standard bottom-up agglomerative clustering (Sokal and Michener, 1958)

Experiment 1: Feature Norms

Evaluate performance using a high-quality semantic network.

- Represent terms in a vector space, with each dimension corresponding to a possible feature
- Compute cosine similarity between pairs of terms
- Construct a semantic network with terms as nodes, adding an edge between terms if similarity exceeds a threshold

Experiment 1: Feature Norms

. . .

apple: *is_edible*, *colour_red*, *is_sweet*, *is_natural*
apple: *colour_red*, *colour_green*, *is_edible*, *is_sweet*, *is_natural*
orange: *is_edible*, *colour_orange*, *is_sweet*
orange: *colour_orange*, *is_edible*, *is_bitter*
. . .

	is_edible	**colour_red**	· · ·	**is_sweet**
Apple	2	2	· · ·	2
Orange	2	0	· · ·	1

Experiment 1: Feature Norms

Method	F-score	Tree Correlation
HRG	**0.507**	**0.168**
CW	0.464	—
Agglo	0.352	0.137

Experiment 2: Corpora

How well does the HRG perform when provided a lower-quality network?

Idea: construct a semantic network as in Experiment 1, but use similarities derived from corpus counts rather than feature norms.

Experiment 2: Corpora

- Extract context windows of ± 5 for each term from a filtered version of the BNC
- Construct a vector representation for each term based on the frequency of co-occurring words
- Transform raw frequency counts using PMI
- Compute cosine similarity between terms and construct semantic network as before

Experiment 1: Corpora

. . .

pineapple natural says tried **apple** grapefruit low now whether grapefruit single rather others **orange** use range apple three orange another name grapefruit **apple** concerned study means grapefruit large deep top **orange** cross tangerine first europe lime grapefruit lines plate **orange** cut can use food

. . .

	context1	**context2**	· · ·	**contextN**
Apple	2	0	· · ·	1
Orange	3	1	· · ·	1

Experiment 2: Corpora

Method	F-score	Tree Correlation
HRG	**0.276**	0.104
CW	0.274	—
Brown	0.258	**0.124**
Agglo	0.122	0.077

Experiment 3: Small-World Graphs

The semantic network produced in Experiment 2 is *very* noisy; is there some way we can clean it up?

Idea: use a flat clustering to filter edges from the input graph and to impose a small-world structure.

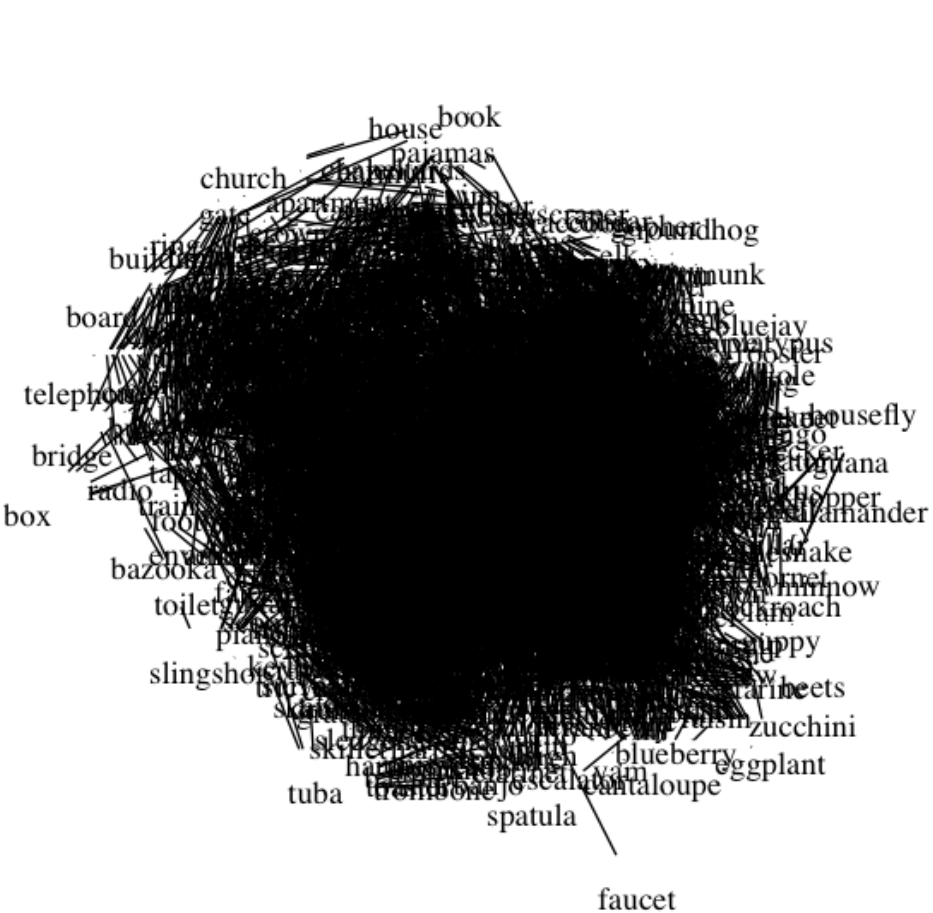
Experiment 3: Small-World Graphs

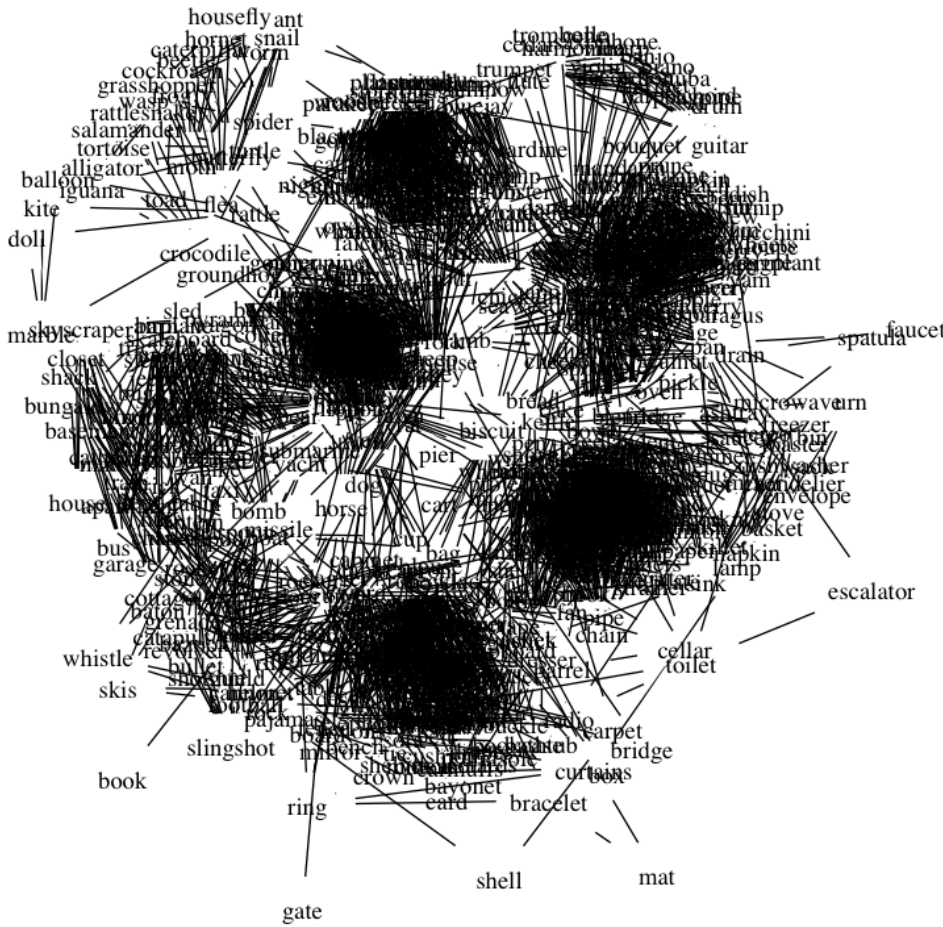
- Obtain a flat clustering using Chinese Whispers and Brown
- Use each clustering to re-weight the semantic network from Experiment 2

Formally,

- Let $W_{a,b} =$ the edge weight between terms A and B in the original graph
- Let $C_{a,b} =$ a binary value indicating whether A and B share a cluster
- Compute a new edge weight $\widehat{W}_{A,B} = (1 - s)W_{A,B} + sC_{A,B}$

Experiment 3: Small-World Graphs





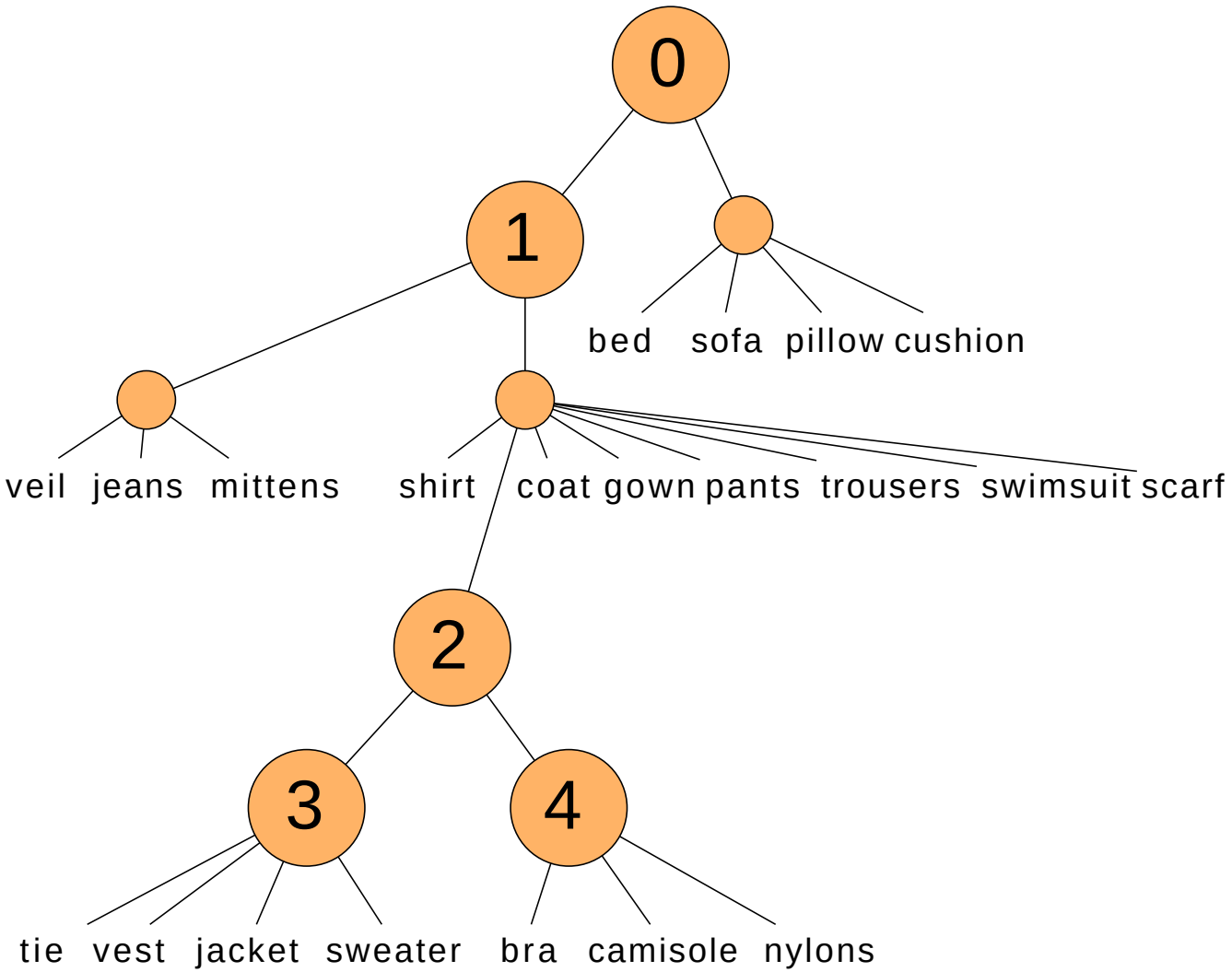
Experiment 3: Small-World Graphs

Method	F-score	Tree Correlation
HRG	0.276	0.104
CW	0.274	—
Brown	0.258	0.124

Experiment 3: Small-World Graphs

Method	F-score	Tree Correlation
HRG	0.276	0.104
CW	0.274	—
Brown	0.258	0.124
HRG + CW	**0.291**	0.161
HRG + Brown	0.255	**0.173**

Experiment 3: Small-World Graphs



Experiment 4: Human Upper Bound

Experiments 1 & 2 were evaluated against a single taxonomy, but rarely is there one correct taxonomy for a set of terms. How well does the HRG perform at capturing one of many *plausible* taxonomies?

Idea: collect taxonomies for a small set of terms from multiple human annotators.

Experiment 4: Human Upper Bound

An elicitation study for simple taxonomies:

- 50 participants in a Mechanical Turk experiment
- 12 manually chosen terms, at multiple levels of granularity
- Participants were asked to “organise the presented terms into a hierarchy”

Experiment 4: Human Upper Bound

An elicitation study for simple taxonomies:

- 50 participants in a Mechanical Turk experiment
- 12 manually chosen terms, at multiple levels of granularity
- Participants were asked to “organise the presented terms into a hierarchy”

Evaluation

- Evaluate the HRG and baselines against human taxonomies by computing the mean correlation between the model’s taxonomy and that of each participant.
- Compute *inter-annotator agreement*, the mean pairwise correlation between human-produced taxonomies.

Experiment 4: Human Upper Bound

Method	Tree Correlation
HRG	**0.412**
Brown	0.181
Agglo	0.274
Agreement	0.511

Conclusions

- Presented a novel method for inferring lexical taxonomies which:
  - is largely parameter-free
  - operates on abstract representations rather than corpora
  - uses a model averaging technique to avoid imposing bias
  - sensitive to quality and topology of input graph
- Evaluated model using pair of complementary measures
- Demonstrated humans can perform the task reliably and that the HRG approximates that performance

Data

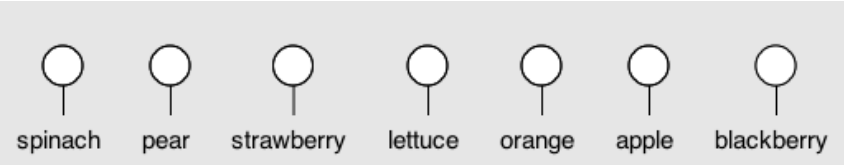
The datasets described in this talk, including the annotator-produced hierarches from Mechanical Turk, are available at:

`http://bit.ly/categorization`

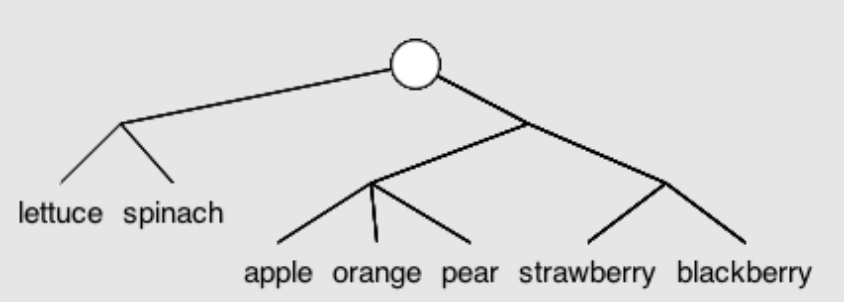
Mechanical Turk Task Description

In this task you'll be given a set of randomly-chosen words and asked to group them together into a hierarchy. Start by grouping words that belong to the same category, then group together categories that are most similar to one another.

For example, you might be given the following words:



and decide to group them into the following hierarchy:



Mechanical Turk Task Interface