The Suffix Tree Document Model Revisited

Sven Meyer zu Eißen Benno Stein Martin Potthast

Bauhaus University Weimar, University of Paderborn

Searching the Web: Today

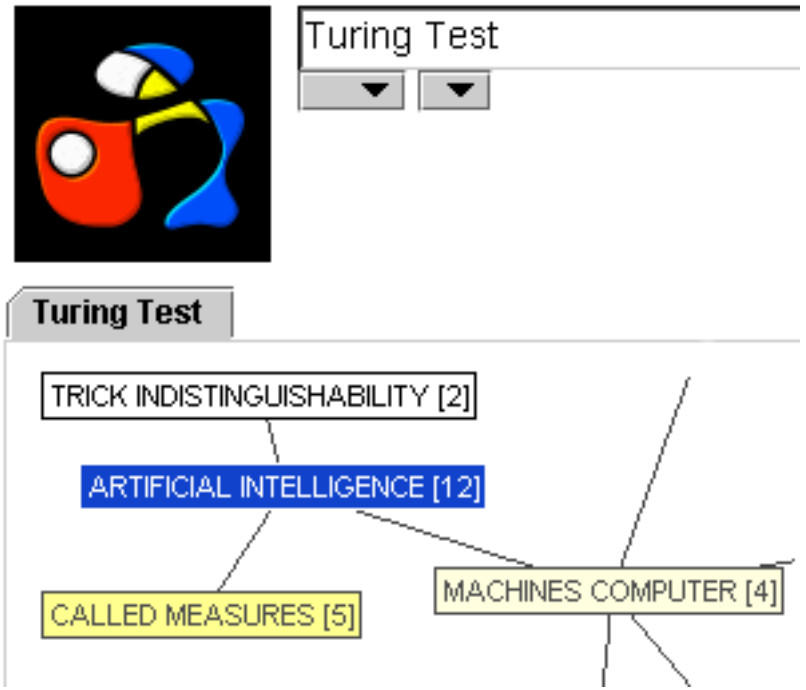
The problem:

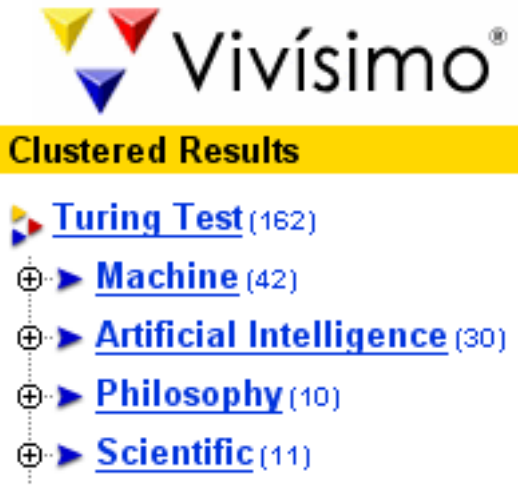
- Web search engines deliver very large result lists.
- Only a small subset is interesting for a user.
- Too many document snippets have to be read.

Searching the Web: Tomorrow

A solution:

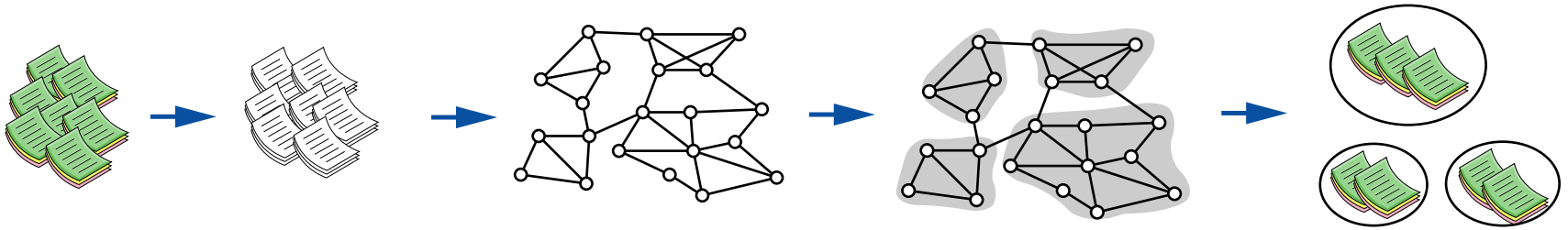
- Generate document categories.
- Assign short topic labels to the found categories.
- Let the user browse categories instead of snippets.





Automatic Category Formation

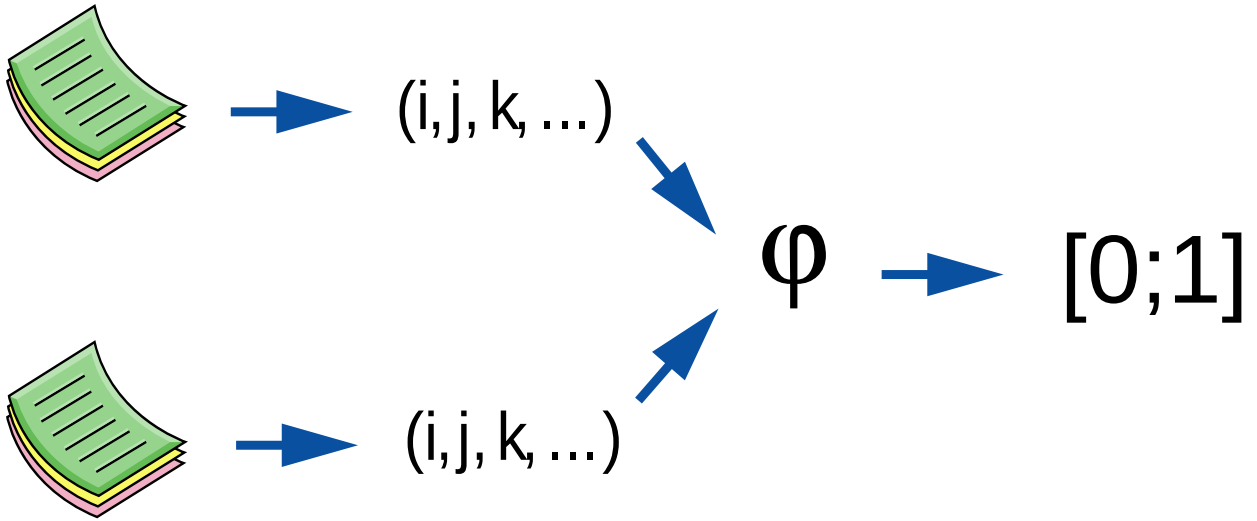
1. Find a document model.
2. Generate a similarity graph based on the document model.
3. Cluster the graph.
4. Assign labels to the document clusters.



Similarity Computation

Document Models:

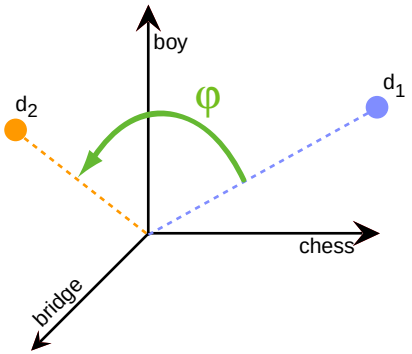
- Algorithm computes document models d.
- Function φ maps two document models d_1, d_2 to [0;1].



(φ is normalized, reflexive, symmetric)

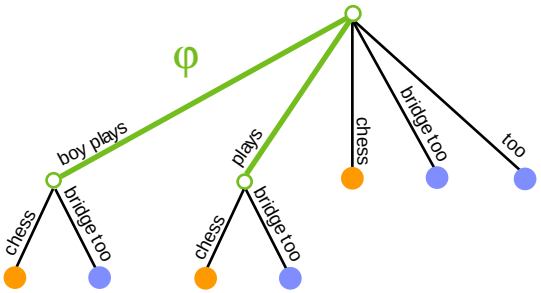
Research Question

Vector Space Model



VS.

Suffix Tree Model

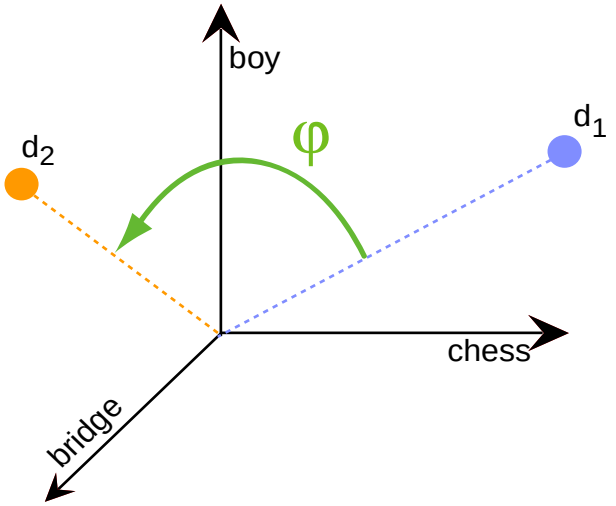


- How can suffix trees be used as document model?
- How does similarity computation work in the suffix tree model?
- Which of the document models is more powerful w.r.t. category formation?

The Vector Space Model

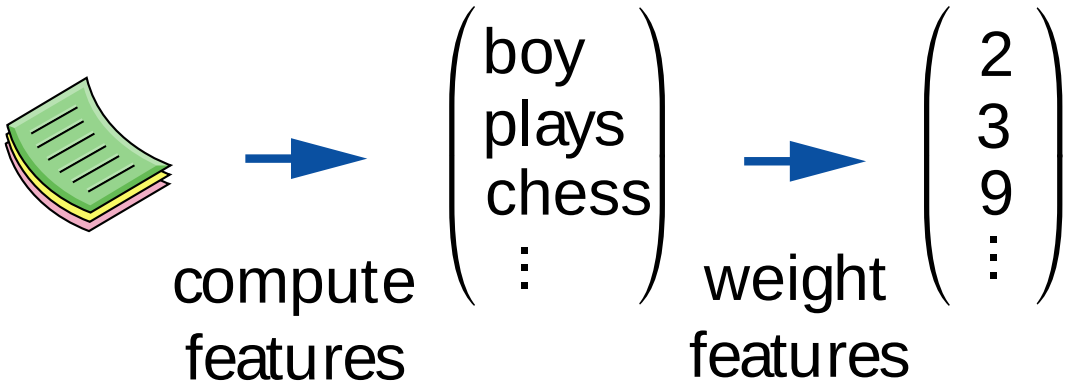
- Documents are represented as feature vectors.
- Two vectors are equal if they point in the same direction.
- Similarity is measured by the cosine of the angle between two vectors.

$$\varphi_{cos} = \frac{\langle \boldsymbol{d}_1, \boldsymbol{d}_2 \rangle}{\| \boldsymbol{d}_1 \| \cdot \| \boldsymbol{d}_2 \|}$$



Features for the Vector Space Model

1. Term concept (granularity of term units)
   and Term weighting schemes (importance measure for term units)
   - term frequency and inverse document frequency ($tf \cdot idf$)

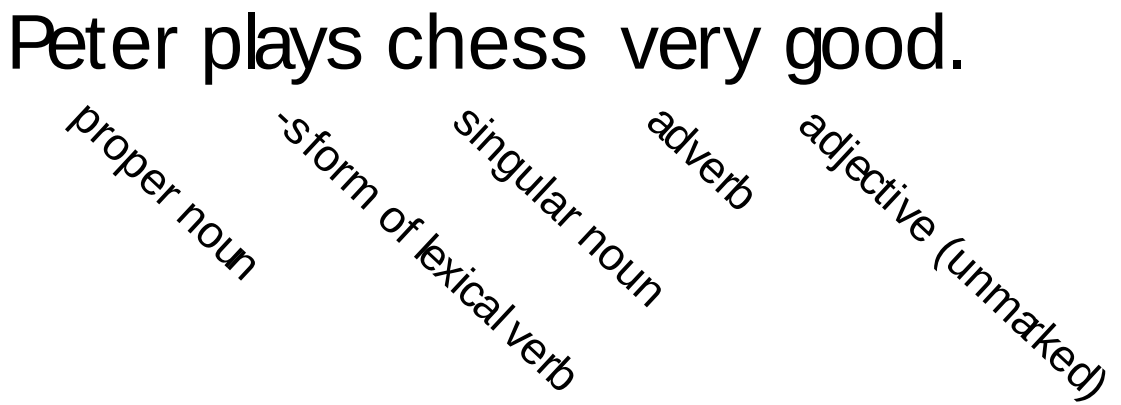


Features for the Vector Space Model

2. Text statistics
   (punctuation statistics, avg. sentence length,...)

3. Presentation-related features
   (headlines, captions,...)

4. Linguistic features
   - syntactic group analysis
   - part-of-speech analysis



Feature Selection for Category Formation

1. Term concept: single terms/stems.
   Term weighting scheme: $tf \cdot idf$.
2. No text statistics.
3. No presentation-related features.
4. No linguistic features.

Features from categories 2.-4. are employed in genre analysis.

[Finn and Kushmerick 03, Meyer zu Eissen and Stein 04]

➜ Observation: We rely on terms, but we do not exploit term order information.

Why bother?

“Concept hypothesis”:

- Term compositions describe more than the sum of their terms (like “computational intelligence”; or names like “George W. Bush”, “New York Yankees”, ...)
- Documents that share concepts are more similar than documents that share terms.
- But this assumption is modeled insufficiently using the vector space model.

Questions:

How can term order information be incorporated into the document model/similarity computation?

Does term order preservation have a measurable effect on category formation?

The Suffix Tree Document Model

Suffix:
The ith suffix of a document $d = w_1 ... w_n$ is the substring of d that starts with term w_i.

The Suffix Tree Document Model

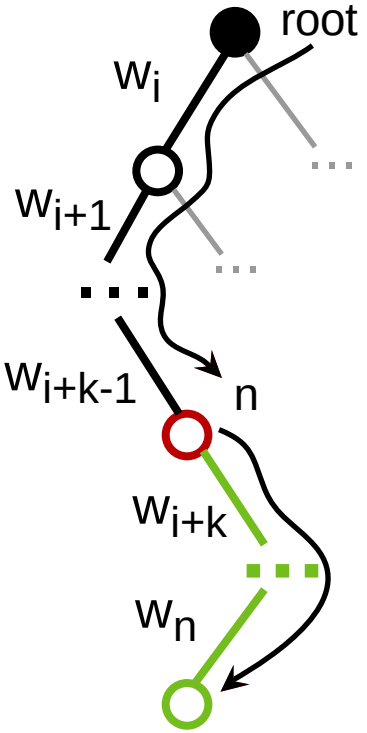
Suffix:
The ith suffix of a document $d = w_1 ... w_n$ is the substring of d that starts with term w_i.

Suffix Tree:
A suffix tree of d is a labeled tree that contains each suffix of d along a path that starts at the root and whose edges are labeled with the respective terms.

The Suffix Tree Document Model

Suffix:
The ith suffix of a document $d = w_1 ... w_n$ is the substring of d that starts with term w_i.

Suffix Tree:
A suffix tree of d is a labeled tree that contains each suffix of d along a path that starts at the root and whose edges are labeled with the respective terms.



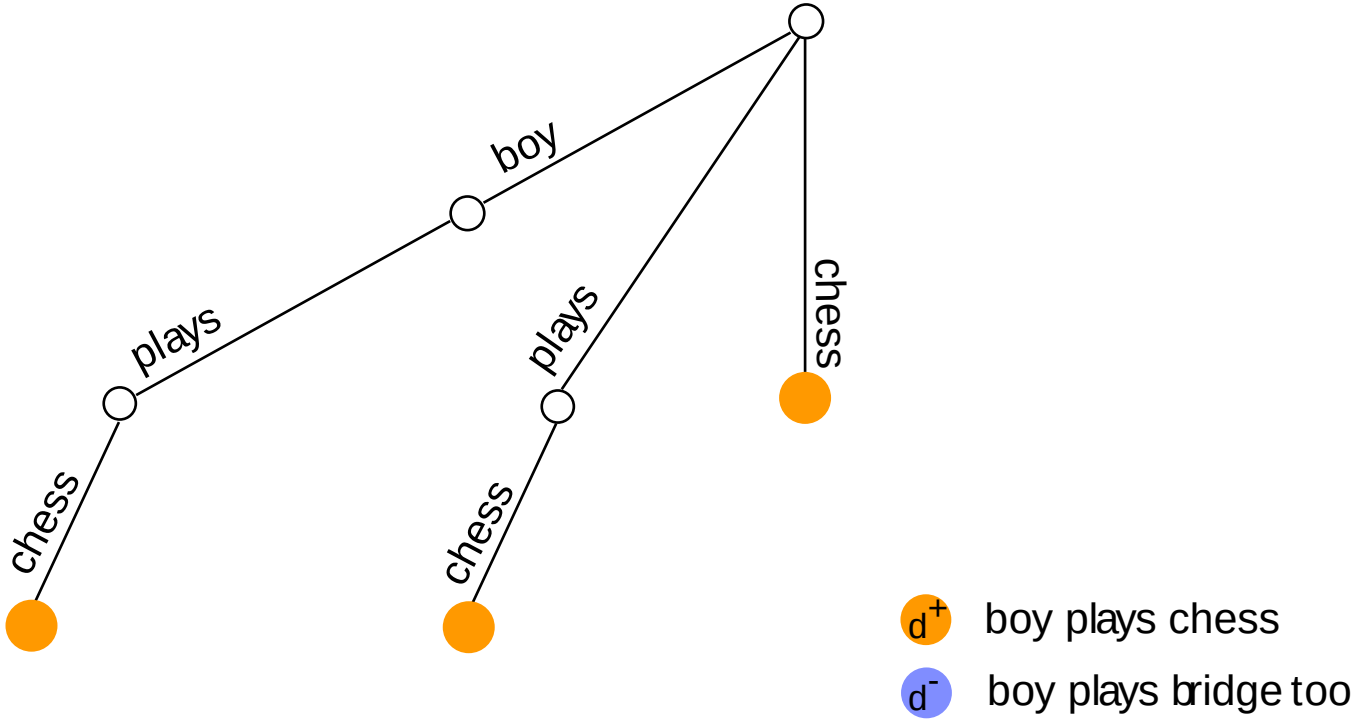
Insertion of the ith suffix of d:

Find node n in depth k with the properties:

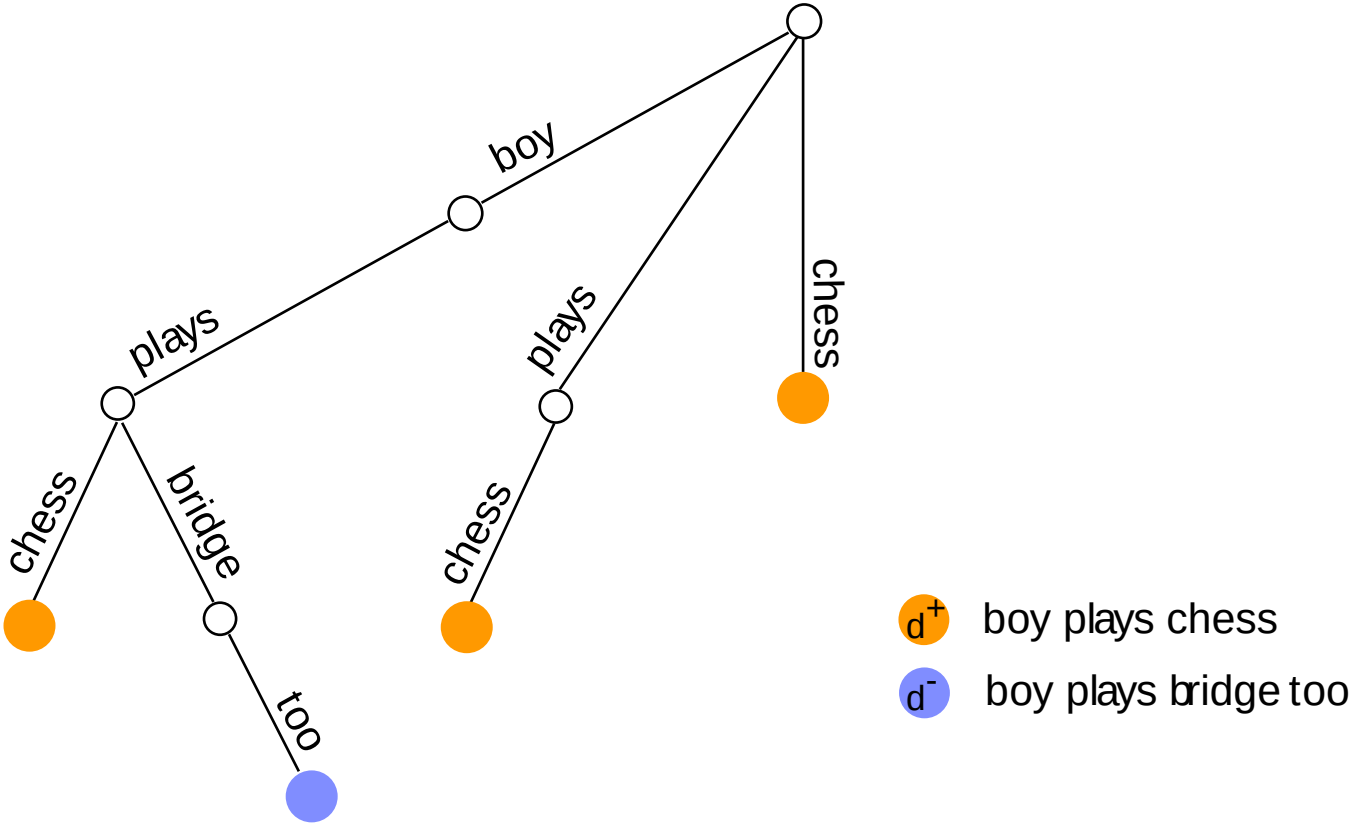
- The edge labels on the path to n correspond to $w_i ... w_{i+k-1}$.
- No outgoing edge is labeled with w_{i+k}.

Add a path to n corresponding to $w_{i+k} ... w_n$.

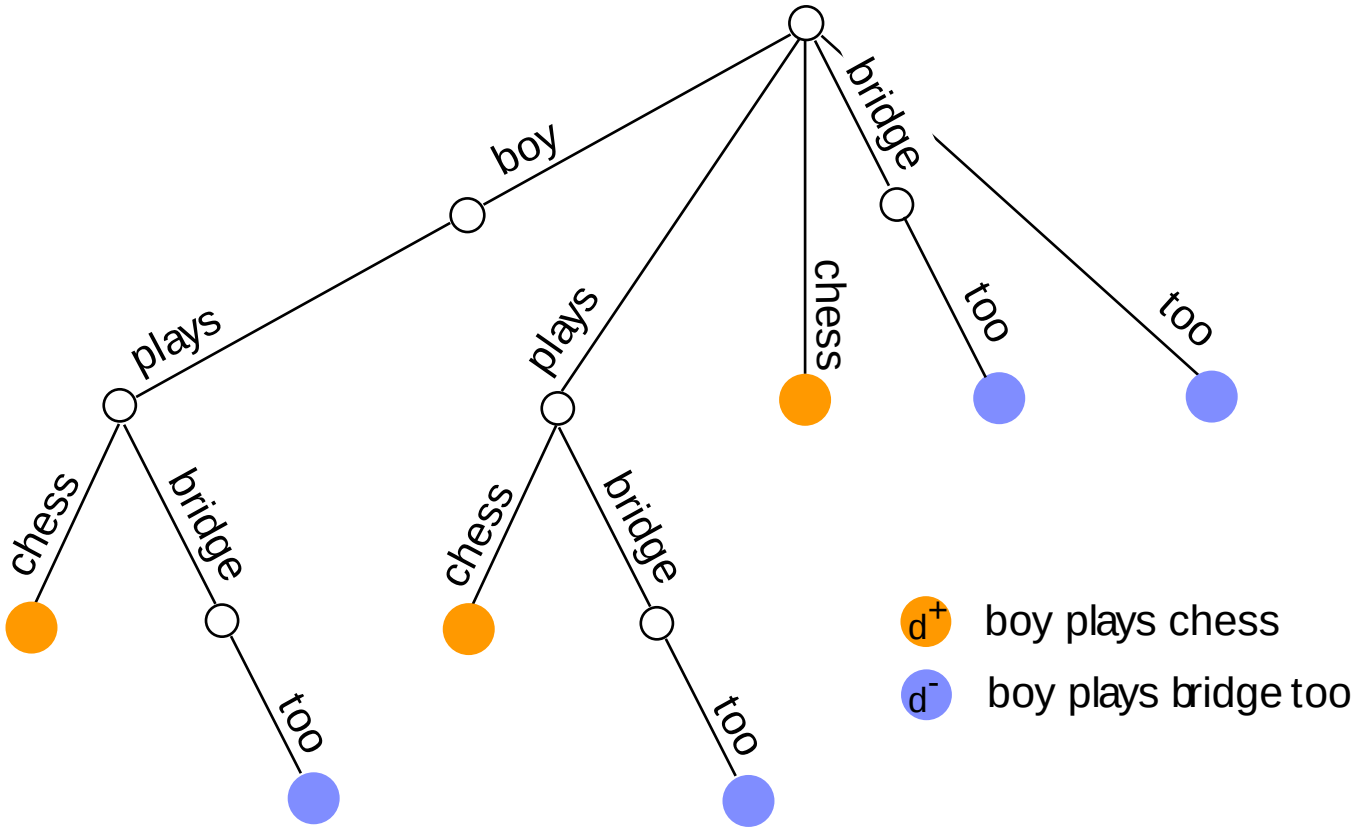
Suffix Tree Construction 1



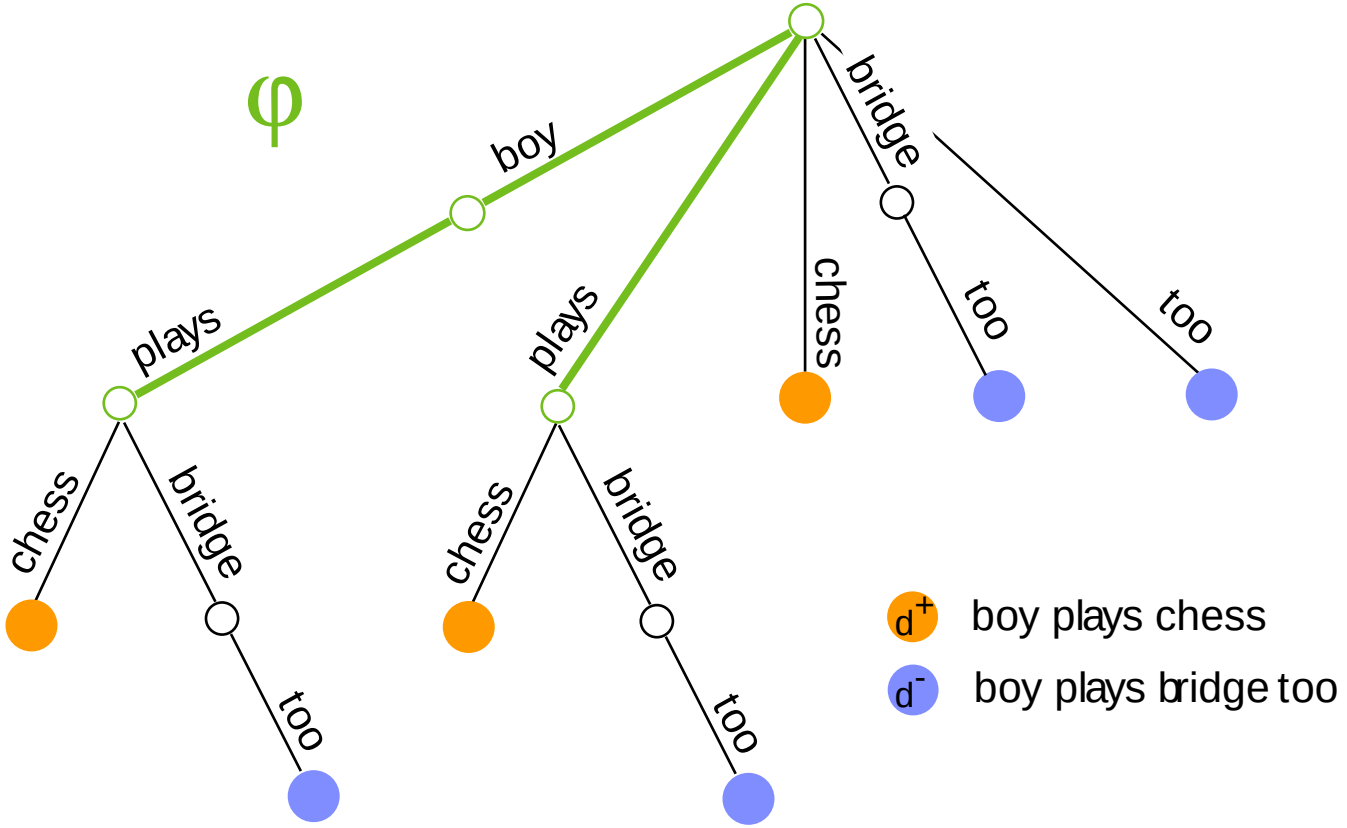
Suffix Tree Construction 2



Suffix Tree Construction 3



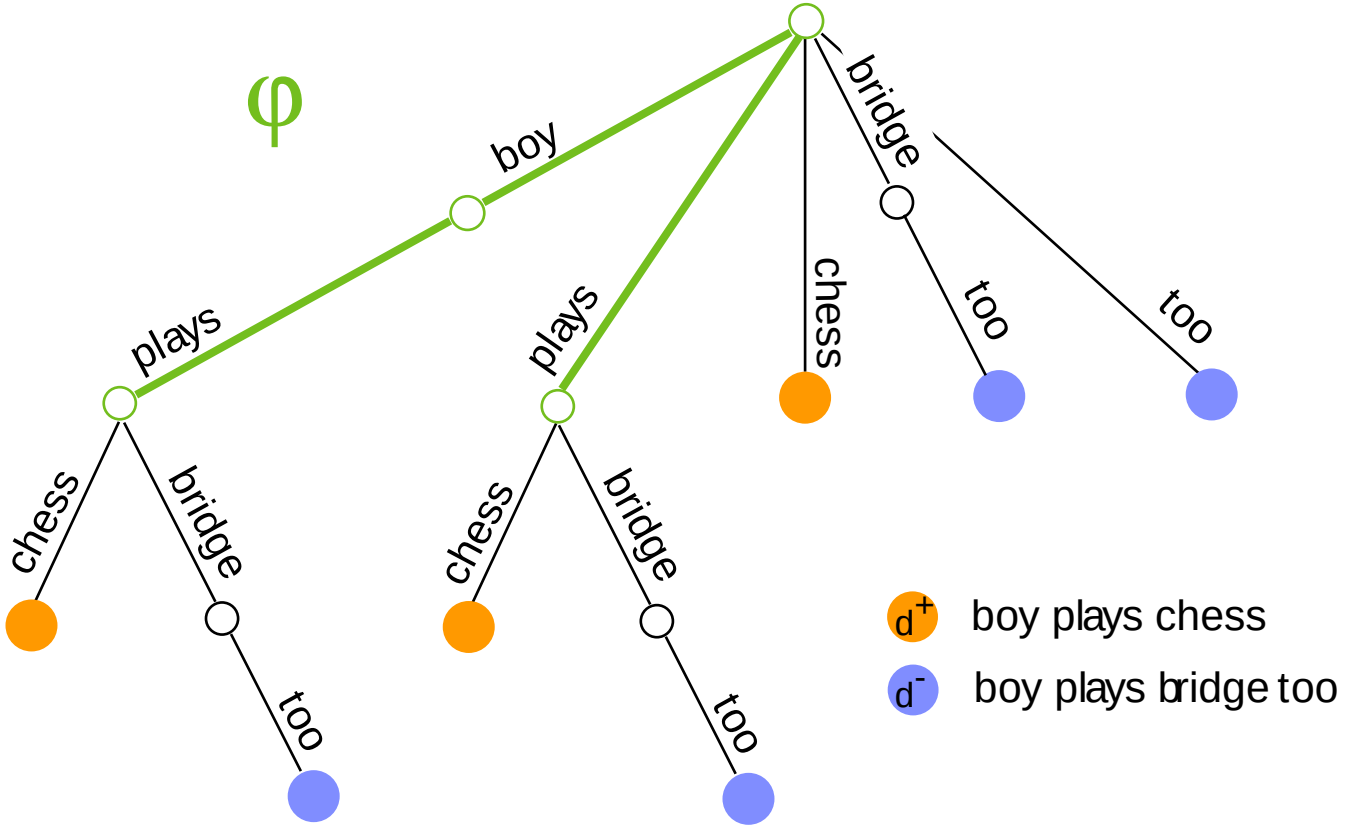
Graph-based Similarity Computation



$$\varphi_{ST} = \frac{|E^+ \cap E^-|}{|E^+ \cup E^-|}$$

$$\varphi_{HYB} = \lambda \cdot \varphi_{ST} + (1 - \lambda) \cdot \varphi_{\cos} \qquad \lambda \in [0; 1]$$

Graph-based Similarity Computation 2



$$\varphi_{STF} = \frac{1}{|E|}\Sigma_{e\in E}\frac{\min\{n^+(e),n^-(e)\}}{\max\{n^+(e),n^-(e)\}}$$

$$\varphi_{STFIDF} = \frac{1}{|E|}\Sigma_{e\in E}\frac{\min\{n^+(e),n^-(e)\}}{\max\{n^+(e),n^-(e)\}} \cdot idf(e)$$

Experimental Setup

- 6 pre-categorized data sets form the basis of our experiments.
- The data sets (DS) are drawn from The Reuters Corpus Volume 1.
- Each set contains between 300 and 800 documents, originally sorted in 3 - 6 categories, respectively.
- Each category contains between 50 and 300 documents, respectively.

- We applied the clustering algorithms MajorClust and Group Average Link.

- Clustering performance is measured with the F-Measure.
- The F-Measure quantifies the match of a clustering against a given (optimal) categorization.
  (no congruence: F=0; perfect congruence: F=1).

Quantitative Evaluation

Categorization with MajorClust:

	DS1	DS2	DS3	DS4	DS5	DS6	average
$\varphi_{\cos}$	0.80	0.60	0.62	0.67	0.66	0.49	0.64
φ_{ST}	0.55	0.46	0.61	0.38	0.45	0.55	0.50
φ_{STF}	0.82	0.70	0.70	0.68	0.76	0.55	0.70
φ_{STFIDF}	0.60	0.60	0.71	0.64	0.78	0.62	0.65
φ_{HYB}	0.84	0.83	0.72	0.74	0.93	0.64	0.78
Improvement in %	5%	38%	16%	10%	40%	31%	22%

Categorization with Group Average Link:

	DS1	DS2	DS3	DS4	DS5	DS6	average
$\varphi_{\cos}$	0.82	0.63	0.69	0.55	0.78	0.51	0.64
φ_{ST}	0.55	0.40	0.61	0.33	0.40	0.55	0.47
φ_{STF}	0.83	0.64	0.71	0.57	0.85	0.63	0.71
φ_{STFIDF}	0.84	0.72	0.71	0.64	0.80	0.60	0.72
φ_{HYB}	0.84	0.74	0.74	0.66	0.92	0.70	0.77
Improvement in %	2%	18%	7%	20%	18%	37%	17%

Experiment Results

Categorization with MajorClust:

	DS1	DS2	DS3	DS4	DS5	DS6	average
$\varphi_{\cos}$	0.80	0.60	0.62	0.67	0.66	0.49	0.64
φ_{ST}	0.55	0.46	0.61	0.38	0.45	0.55	0.50
φ_{STF}	0.82	0.70	0.70	0.68	0.76	0.55	0.70
φ_{STFIDF}	0.60	0.60	0.71	0.64	0.78	0.62	0.65
φ_{HYB}	0.84	0.83	0.72	0.74	0.93	0.64	0.78
Improvement in %	5%	38%	16%	10%	40%	31%	22%

Categorization with Group Average Link:

	DS1	DS2	DS3	DS4	DS5	DS6	average
$\varphi_{\cos}$	0.82	0.63	0.69	0.55	0.78	0.51	0.64
φ_{ST}	0.55	0.40	0.61	0.33	0.40	0.55	0.47
φ_{STF}	0.83	0.64	0.71	0.57	0.85	0.63	0.71
φ_{STFIDF}	0.84	0.72	0.71	0.64	0.80	0.60	0.72
φ_{HYB}	0.84	0.74	0.74	0.66	0.92	0.70	0.77
Improvement in %	2%	18%	7%	20%	18%	37%	17%

Conclusion

- Does term order preservation have a measurable effect on automatic category formation?

  We achieved performance improvements of up to 40% with the new similarity measures based on the suffix tree model and 20% on average.

- Which document model/similarity measure is more powerful w.r.t. automatic category formation?

  φ_{STF} *outperforms* $\varphi_{\cos}$*. The hybrid measure* φ_{HYB} *performs best.*

Thank you! Questions?