Enhanced Hypertext Categorization using Hyperlinks

Soumen Chakrabarti, Byron Dom, Piotr Indyk

ACM SIGMOND, 1998, Seattle, WA

Presented by Yang Yu

Outline

- Challenges in hypertext categorization
- TAPER and its performance on text documents and hypertext documents
- The “Absorbing neighboring text” approach and its performance on IBM Patent Database
- The “Radius-one linkage enhanced analysis” approach and its performance on IBM Patent Database
- The The “Radius-one linkage enhanced analysis” approach and its performance on a sample of Yahoo! topic
- Comments on the paper

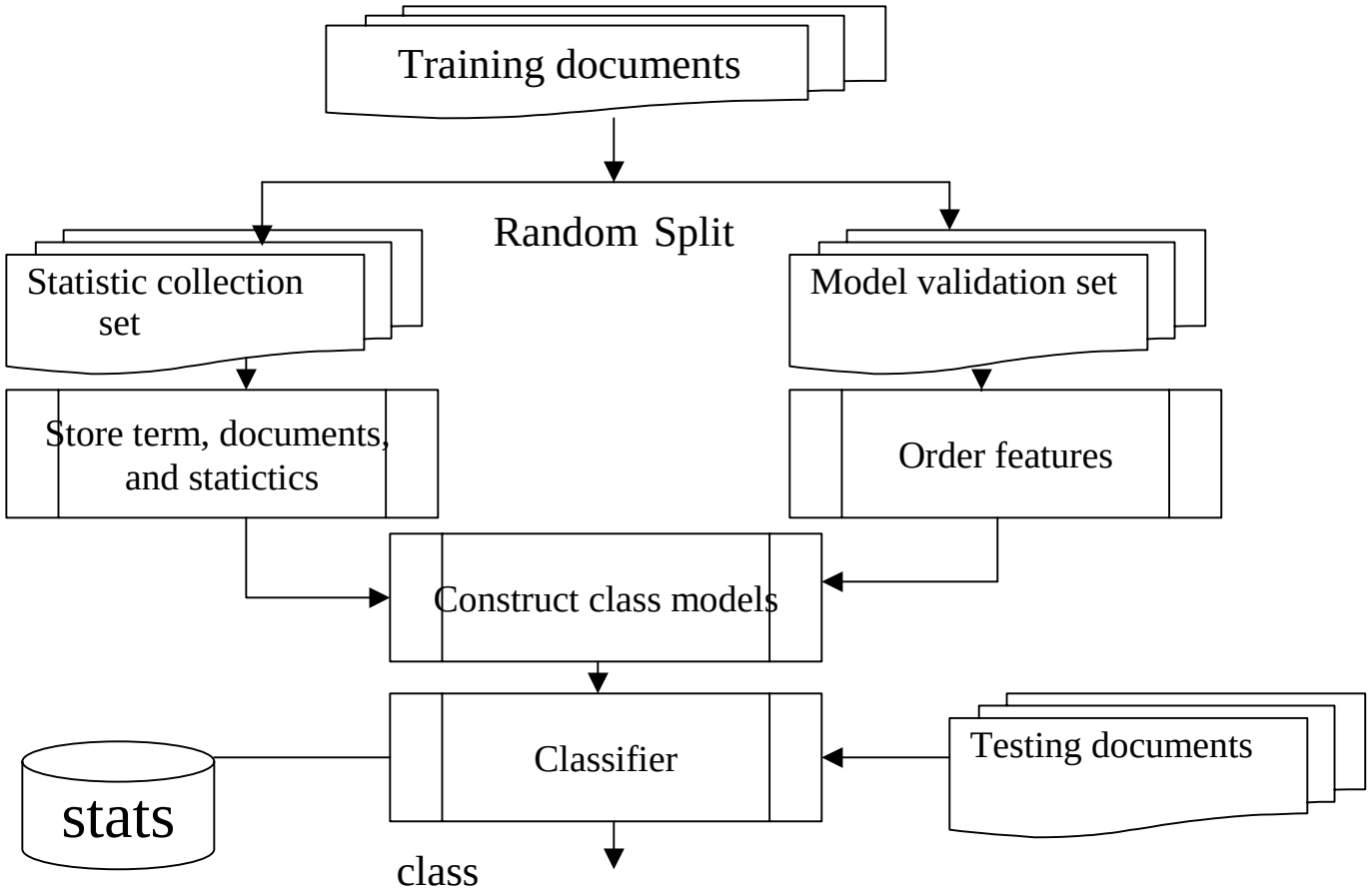
Challenges in Hypertext Categorization

- Hypertext documents' authorship is highly diverse
- Some web pages are simply lists of hyperlinks and contain no direct information themselves
- Links contain semantic information which will be lost when they are treated as simple text
- Links are noisy, some links lead to related documents, but others don't

Data Set for Evaluation

- IBM Patent server database
  - 3 first levels and 12 leaves. For each leaf, 630 documents are used for training, 300 for testing
- YAHOO topics
  - 13 top classes, 20,000 documents are used for the link locality analysis. 900 documents are used for the hyperlink only linkage enhanced analysis

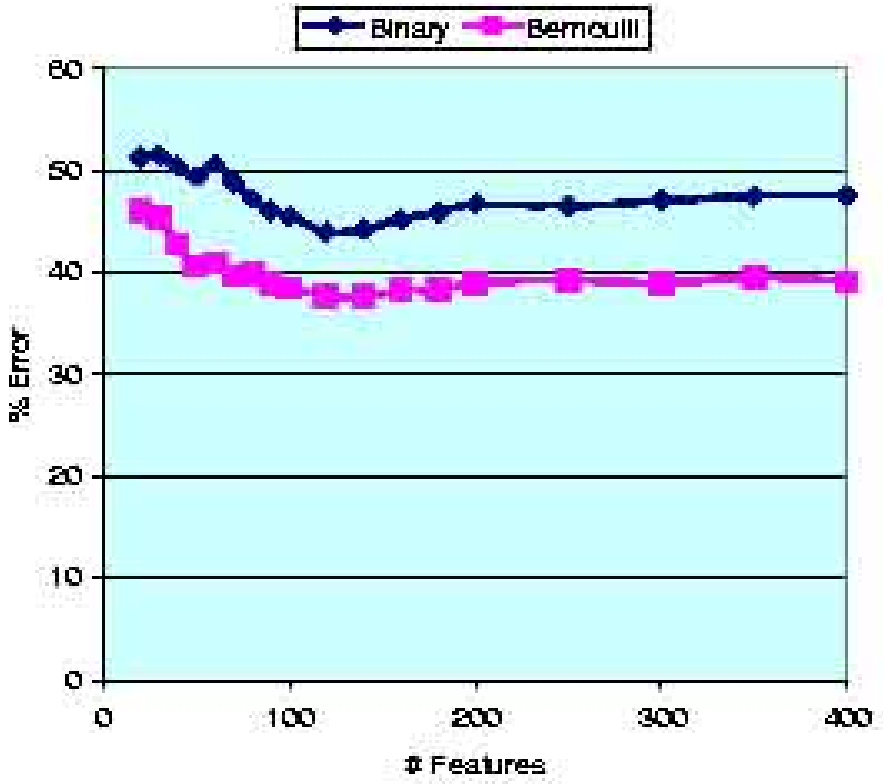
TAPER: Taxonomy and Path Enhanced Retrieval



Features of TAPER

- Training data are split into 2 parts. Some of them are used for feature selection, others are used for create the classifiers
- TAPER is a hierarchical categorizer, which maintains a topic tree and there is a classifier on each internal node
- Feature Selection: Terms are ordered by decreasing ability to separate the classes, then a prefix of the sorted list is picked which can give the best classification accuracy
- Class Models: Different ways a classifier uses to decide which child to choose. Bernoulli Model is generally better than the binary one

Feature Selection and Class Models



Results of TAPER

- The metric is error rate, which is the percentage of documents misclassified
- Reuters: Traditional text corpus
  - Pretty good, 13% error
- IBM Patent Database
  - Worse, 36% error
- Yahoo
  - Horrible, 68% error

Linkage Analysis

- Hypertext documents are not self-contained
- When training a classifier, link graph should also be part of the input
- When evaluating a document, the neighborhood of the document should be part of the input
- Let C be set of the classes, G be the link graph, T be the collection of text of the all the documents

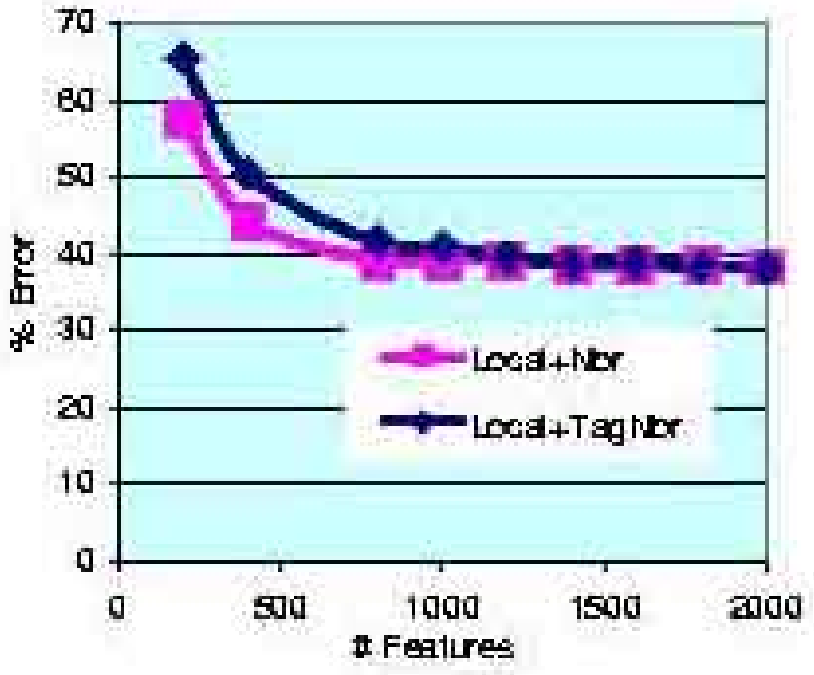
  The goal is to choose C such that $\Pr(C|G,T)$ is maximum

Absorbing Neighborhood Text

- Data set for evaluation: IBM Patent Database
- Options:
  - Local: Features of TAPER are terms of this document
  - Local+Nbr: Features of TAPER are terms from both the local document and its neighbors, including all the in-neighbors and out-neighbors
  - Local+TagNbr: Features are from the same documents as in Local+Nbr. But terms from neighbor text distinguished from local terms

Result of Absorbing neighborhood text

- Error Rate
  - Local: 36%
  - Local+Nbr: 38.3%
  - Local+TagNbr: 38.2%



Explanation of the Results

- Why does neighbor text do worse
  - Frequent cross boundary linkage between topics
- Why did not tagging help
  - Tagging split a term into many forms and make it rare
  - The heuristic of feature selection and learning of class models do poorly with many noisy seldom appearing features

The Completely Supervised Case of Radius-one Linkage Enhanced Analysis

- Assumption: All neighbor classes are known
- Class information from neighbors rather than their original text are used as features of TAPER
- The basic idea is still applying Bayesian Law:

  For document *Di*

  – Choose class *Ci* to maximize Pr(*Ci*|*Ni*), where *Ni* represents the collection of of all neighbor documents with known classes

  – Applying Bayesian law, the goal is turned into to maximize Pr(*Ni*|*Ci*)Pr(*Ci*)

Options of the above Approach

- Text: Only the text of the documents(IBM patent Database are used as features of TAPER
- Link: The class names of neighbor documents are the only features. Class names are paths in a topic hierarchy

  e.g. 29/X/Y/Z from [29] [Metal working] [X] …
- Prefix: All prefixes of paths are used as features
- Text + Prefix: Two copies of TAPER are run. One on local text, one on prefixes. The joint distribution is the product of their marginal distribution

Results of the above Approach

- Error Rate:
  - Text: 36%
  - Link: 34%
  - Prefix: 22.1%
  - Text+Prefix: 21%
- Conclusion
  - Much better performance
  - The major benefit is from extracting prefixes of links

The Partially Supervised Case of Radius-one Linkage Analysis

- In the real world, only some or none of the neighbor classes are known
- Neighbors whose classes are known: use the class labels as the sole feature
- Neighbors whose classes are not known: Using the relaxation labeling technique

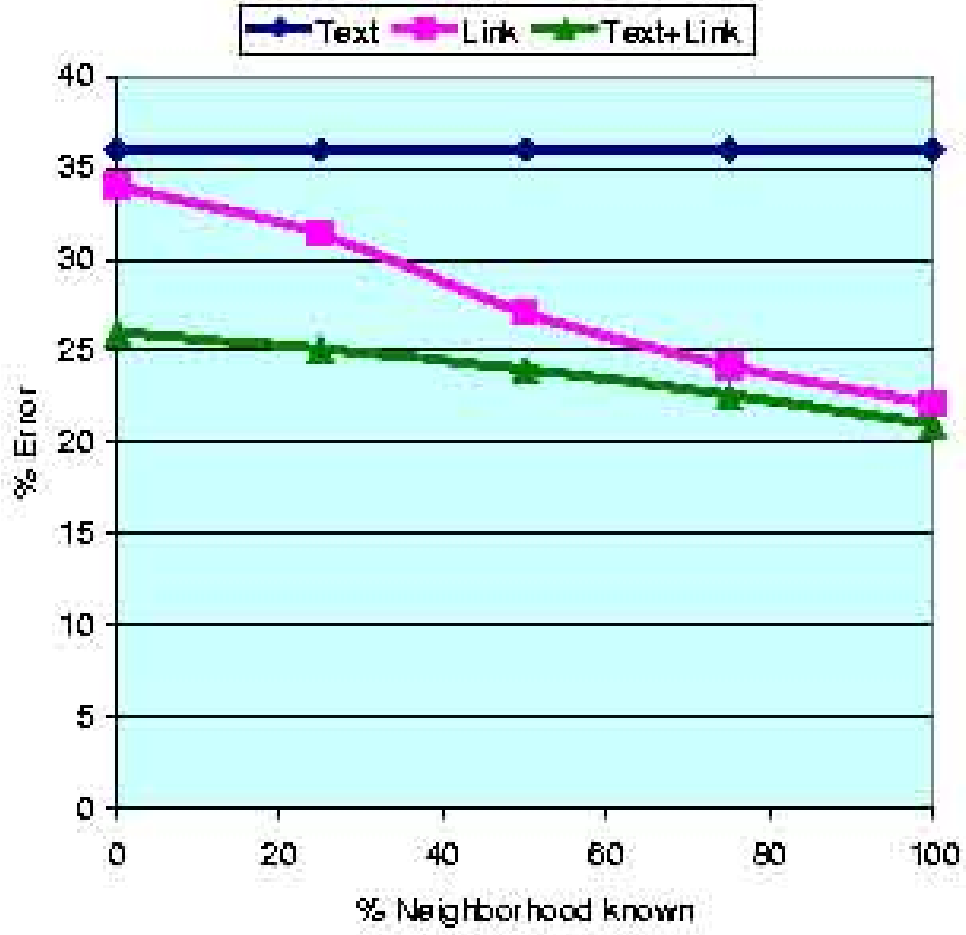
Relaxation Labeling

- Given a document d, construct the neighborhood graph around it
- Classify the neighbor document using their local text
- Iterate until convergence
  - Recompute the class for each document using both the local text and the class information of the neighbors
- The relaxation is guaranteed to converge to a consistent state provided it is initiated “close enough” to such a state

Options of the above approach

- Data Set for evaluation: IBM Patent Database
- Options:
  - Text: Only the text of the documents are used as features of TAPER
  - Link: Only the class information of neighbor documents are used as features
  - Text+Link: Two TAPERs are run on local text information and link information
  - Does Link here actually mean Prefix?

Results of the above Approach



Conclusion from the Results

- Adding link information improves accuracy
- Even when 0% neighbors have known classes, it is beneficial to add link information
- Text+Link always beats Link, but the margin is small when a large fraction of neighbors have known classes
- Text+Link is more stable than Link

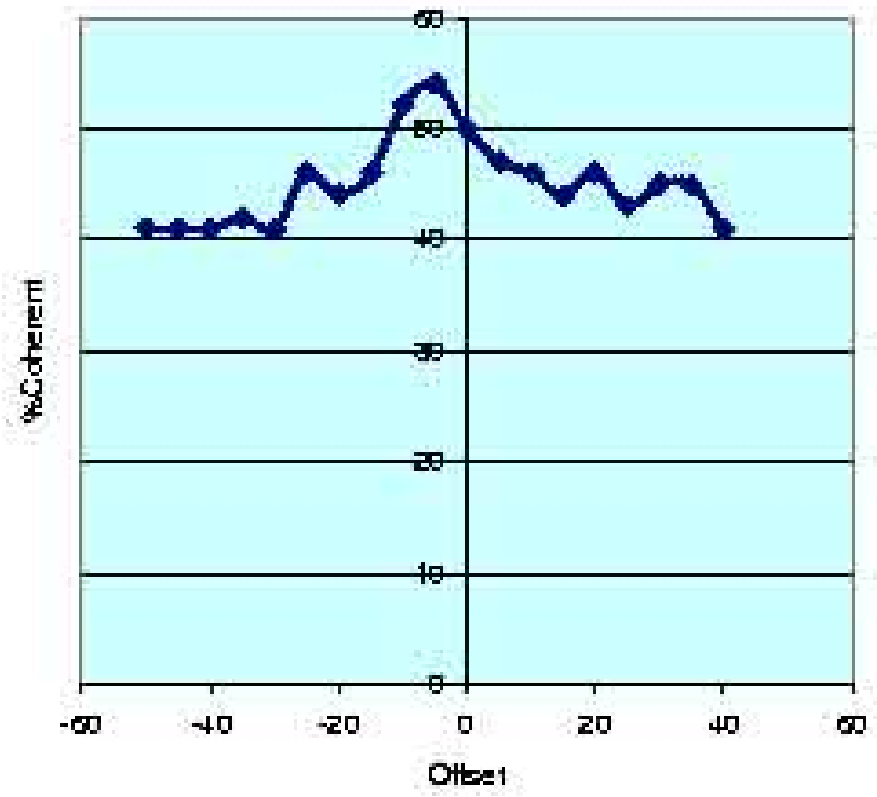
Problems with the Yahoo topics

- Yahoo! documents are more diverse than the Patent's
- The link graph of the Yahoo! documents are not complete
  - Only 28% have some out-links to some Yahoo! document
  - Only 19% have some in-links from some Yahoo! document
  - A larger fraction of documents have links to totally unrelated document
  - Co-Citation is popular in Yahoo! documents

Radius-two Linkage Analysis:Bridges

- Idea: Documents cited by many common documents are likely to be in the same topic
- A “Bridge” is a document that hint two or more other documents are in the same class
- There are II, IO, OO, OI bridges, IO bridges is more meaningful
- IO- Bridge: *B* is a IO-bridge for *D1* and *D2* iff there are links from *B* to both *D1* and *D2*

Are IO-Bridges useful?



How to get the graph

- For each page *D* in Yahoo!, consider all the pages *Di* that point to it
- Each page *Di* is regarded as a sorted list of out-links
- For each links *D'* in Di check whether the class of *D* and *D'* are the same, if so, they are called coherent
- For each offset *D*, calculate the percentage of coherent pairs for which *(Pos(D')- Pos(D))i = D* for some *Di*, *D/D* appears at *Pos(D)/Pos(D')* in the out-link list of *Di*

Comments on this graph

- Interesting things in the graph
  - The bridge is not pure, the non-coherent rate is always significant
  - Peak does not appear at offset 0
  - The curve is quite flat, yet the coherent rate around offset 0 is somewhat higher
- Questions about the graph
  - What is it not symmetric?
  - Why the coherence is not 100% at offset 0

Locality

- There are often several segments in bridges, the out-links in each segment point to documents in the same topic
- Closer links have larger tendency to point to documents in the same topic
- Trading coverage for accuracy

  A class *C* is treated as a feature of document *D* if there is a IO-bridge *B* which has 3 out links point to *D1 D D2* such that the classes of *D1* and *D2* are both *C* , and there are no out links between *D1* and *D2* point to a known class page

Options of the above Approach

- Data set for evaluation: A small subset of Yahoo! (about 900 documents, each of them is IO-bridged to some other Yahoo! pages)
- Text: Again, only the text of local documents are features
- IO-Bridge: For a given document *D*, all prefixes of the class paths of all the documents which are IO-bridged to *D* are treated as features of the document. (In testing, only prefixes from the training set is considered)
- IO-Bridge+Locality: Refer the previous slide

Results of the above Approach

- Error Rate:
  - Text: 68%
  - IO-Bridge: 25%
  - IO+locality: 21%
- Coverage:
  - Text: 100%
  - IO-Bridge: 75%
  - IO+locality: 62%

Comments of this paper

- First paper to combine textual / linkage features for hypertext categorization
- Good ideas (treating links as features, path prefixes)
- Inconsistent data set for different approaches
- Some results are unclear
- Some terms and formulas are unclear