Digital Libraries

Computational Literary Analysis,
Duplicate and Plagiarism Detection

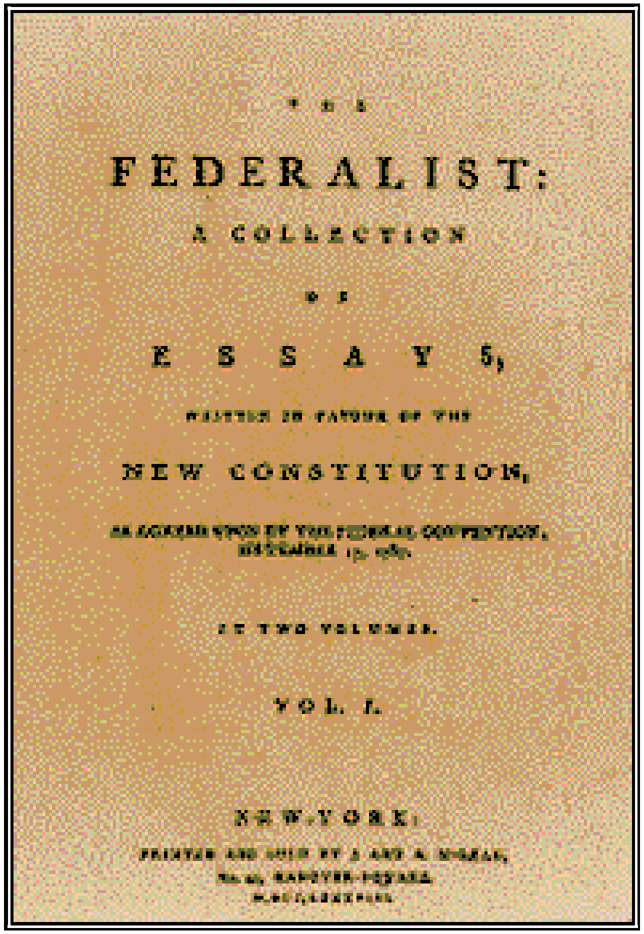
Week 9 Min-Yen KAN

Outline

- Literary Analysis
  - Authorship detection
  - Genre classification
- Duplicate Detection
  - Web pages
- Plagiarism Detection
  - In text
  - In programs

The Federalist papers

- A series of 85 papers written by Jay, Hamilton and Madison
- Intended to help persuade voters to ratify the US constitution


FEDERALIST:

A COLLECTION

ESSAYS,

NEW CONSTITUTION,

VOL. I.

NEW-YORK:

Disputed papers of the Federalist

- Most of the papers have attribution but the authorship of 12 papers are disputed
  - Either Hamilton or Madison
- Want to determine who wrote these papers
  - Also known as textual forensics

Hamilton

Madison

Wordprint and Stylistics

- Claim: Authors leave a unique *wordprint* in the documents which they author
- Claim: Authors also exhibit certain *stylistic patterns* in their publications

Feature Selection

- Content-specific features (Foster 90)
  - key words, special characters
- Style markers
  - Word- or character-based features
    - length of words, vocabulary richness
  - Function words (Mosteller & Wallace 64)
- Structural features
  - Email: Title or signature, paragraph separators (de Vel *et al.* 01)
  - Can generalize to HTML tags
  - To think about: artifact of authoring software?

Bayes Theorem on function words

- M & W examined the frequency of 100 function words

Frequency	Hamilton	Madison
0	.607	.368
1	.303	.368
2	.0758	.184

- Used Bayes' theorem and linear regression to find weights to fit for observed data

- Sample words:

as	do	has	is	no	or	than	this
at	down	have	it	not	our	that	to
be	even	her	its	now	shall	the	up

A Funeral Elegy and *Primary Colors*

"Give anonymous offenders enough verbal rope and column inches, and they will hang themselves for you, every time" – Donald Foster in *Author Unknown*

- *A Funeral Elegy*: Foster attributed this poem to W.S.
  - Initially rejected, but identified his anonymous reviewer
- Forster also attributed *Primary Colors* to Newsweek columnist Joe Klein
- Analyzes text mainly by hand

Foster's features

- Very large feature space, look for distinguishing features:
  - Topic words
  - Punctuation
  - Misused common words
  - Irregular spelling and grammar

- Some specific features (most compound):
  - Adverbs ending with "y": *talky*
  - Parenthetical connectives: … , *then*, …
  - Nouns ending with "mode", "style": *crisis mode*, *outdoor-stadium style*

Typology of English texts

- Biber (89) typed different genres of texts

- Five dimensions …
  1. Involved vs. informational production
  2. Narrative?
  3. Explicit vs. situation-dependent
  4. Persuasive?
  5. Abstract?

… targeting these genres

1. Intimate, interpersonal interactions
2. Face-to-face conversations
3. Scientific exposition
4. Imaginative narrative
5. General narrative exposition

Features used (*e.g.*, Dimension 1)

- Biber also gives a feature inventory for each dimension

THAT deletion
Contractions
BE as main verb
WH questions
1st person pronouns
2nd person pronouns
General hedges   +

Nouns   –
Word Length
Prepositions
Type/Token Ratio

35	Face to face conversations
30	
25	
20	Personal Letters Interviews
15	
10	
5	Prepared speeches
0	General fiction
-5	
-10	Editorials
-15	Academic prose; Press reportage Official Documents
-20	

Discriminant analysis for text genres

- Karlgren and Cutting (94)
  - Same text genre categories as Biber
  - Simple count and average metrics
  - Discriminant analysis (using SPSS software)
  - 64% precision over four categories

Some count features

- Adverb
- Character
- Long word (> 6 chars)
- Preposition
- 2nd person pronoun
- “Therefore”
- 1st person pronoun
- “Me”
- “I”
- Sentence

Other features

- Words per sentence
- Characters per word
- Characters per sentence
- Type / Token Ratio

Genre vs. Subject (Lee & Myaeng 02)

- Genre: style and purpose of text
- Subject: content of text

What about the interaction between the two?

Study found that certain genres overlap signficantly in subject vocabulary

- So, want to use terms that cover more subjects represented by a genre
- Do this by selecting terms that:
  1. Appear in a large ratio of documents belonging to the genre
  2. Appear evenly distributed among the subject classes that represent the genre
  3. Discriminate this genre from others

Putting the constraints together

Document Frequency Ratios
(coverage of term to genre or genre+subject)

$$DFR_g(t) = \frac{df_{g,t}}{df_g}$$

$$DFR_{g,s}(t) = \frac{df_{g,s,t}}{df_{g,s}}$$

What are some negative aspects of this approach?

Use these to define the weight

$$W_g(t) = DFR_g(t) * (1 - \sigma)$$

Where σ is a penalty ("deviation") factor for terms that are spread widely over different subjects

$$\sigma = \sqrt{\frac{\sum_{|S|} (DFR_g(t) - DFR_{g,s}(t))^2}{|S|}}$$

In summary…

Genre and authorship analysis relies on highly frequent evidence that is portable across document subjects.

Contrast with subject/text classification which looks for specific keywords as evidence.

References:

- Mosteller & Wallace (63) *Inference in an authorship problem*, J American Statistical Association 58(3)
- Karlgren & Cutting (94) *Recognizing Text Genres with Simple Metrics Using Discriminant Analysis, Proc. of COLING-94.*
- *de Vel, Anderson, Corney & Mohay (01) Mining Email Content for Author Identification Forensics,* SIGMOD Record
- Foster (00) *Author Unknown*. Owl Books PE1421 Fos
- Biber (89) *A typology of English texts,* Linguistics, 27(3)
- Lee and Myaeng (02)

To think about…

- The Mosteller-Wallace method examines function words while Foster’s method uses key words. What are the advantages and disadvantages of these two different methods?

- What are the implications of an application that would emulate the wordprint of another author?

- What are some of the potential effects of being able to undo anonymity?

Water Break

- See you in five minutes!

I will hold a short tutorial for HW #2 at the end of class today.

Copy detection

Duplicate detection characteristics

- Plagiarism
  - copies intentionally
  - may obfuscate
  - target and source relation
- Self-plagiarism*
  - copy from one's own work
  - Often to offer for background of work in incremental research
- (near) Clone/duplicate
  - same functionality in code / citation data
  - but in different modules by different developers
- Fragment
  - web page content generated by content manager
  - interferes with spiders' re-sampling rate

Signature method

1. Register signature of authority doc
2. Check a query doc against existing signature
3. Flag down very similar documents

Some design choices have to be made:

- How to compute a signature
- How to judge similarity between signatures

Effect of granularity

Divide the document into smaller chunks

document – no division
sentence
window of *n* words

- Large chunks
  - Lower probability of match, higher threshold
- Small chunks
  - Smaller number of unique chunks
  - Lower search complexity

Signature methods

For text documents

- Checksum
- Keywords
- *N*-gram (usually character) inventory
- Grammatical phrases

For source code

- Words, characters and lines
- Halstead profile

  (Ignores comments)
  - Operator histogram
    - e.g., frequency of each type sorted
  - Operand histogram

Distance calculations

Calculate distance between p_1, p_2

- VSM: L_1 distance $\Sigma_f|P_{f1}-P_{f2}|$
- VSM: L_2 Euclidean distance $(\Sigma_f|P_{f1}-P_{f2}|^2)^{1/2}$
- Weighted feature combinations
- For text features, can use edit distance
  - Calculate using dynamic programming

Detect and flag copies

- Assume top n% as possible plagiarisms
- Use a tuned similarity threshold
- Other way: do tuning on supervised set (learn weights for features: Bilenko and Mooney)

What are some problems with these approaches?

Subset problem

- Problem: If a document consists is just a subset of another document, standard VS model may show low similarity
  - Example: cosine $(D_1,D_2) = .61$
    D_1: <A, B, C>,
    D_2: <A, B, C, D, E, F, G, H>
- Shivakumar and Garcia-Molina (95): use only *close* words in VSM
  - Close = comparable frequency, defined by a tunable ε distance.

R-measure: amount repeated in other documents (Khmelev and Teahan)

- Normalized sum of lengths of all suffixes of the text repeated in other documents

$$R^2(T \mid T_1, \ldots T_m) = \frac{2}{l(l+1)} \sum_{i=1}^{l} Q(T[i..l] \mid T_1, \ldots T_m),$$

where $Q(S|T_1 \ldots T_n)$ = length of longest prefix of S repeated in any one document

  - Computed easily using suffix array data structure
  - More effective than simple longest common substring

R-measure example

T = cat_sat_on

T1 = the_cat_on_a_mat

T2 = the_cat_sat

$$\frac{2}{l(l+1)} \sum_{i=1}^{l} Q(T[i..l] \mid T_1, \ldots T_m),$$

Computer program plagiarism

- Use stylistic rules to compile fingerprint:
  - Commenting
  - Variable names
  - Formatting
  - Style (e.g., K&R)
- Use this along with program structure
  - Edit distance

What about hypertext structure in the web?

```
/************************************
* This function concatenates the first and
* second string into the third string.
*************************************
void strcat(char *string1, char *string2, char
     *string3)
{
 char *ptr1, *ptr2;
 ptr2 = string3;
/*
 * Copy first string
 */
for(ptr1=string1;*ptr1;ptr1++) {
*(ptr2++) = *ptr1;
}
```

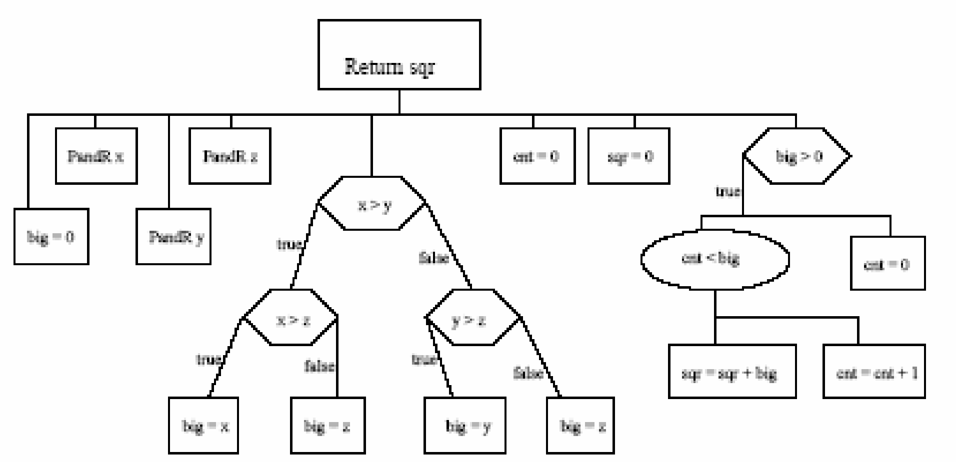
```
/*
 * concatenate s2 to s1 into s3.
 * Enough memory for s3 must already be
     allocated. No checks !!!!!!
 */
mysc(s1, s2, s3)
      char *s1, *s2, *s3;
{
  while (*s1)
    *s3++ = *s1++;

  while (*s2)
    *s3++ = *s2++;
}
```

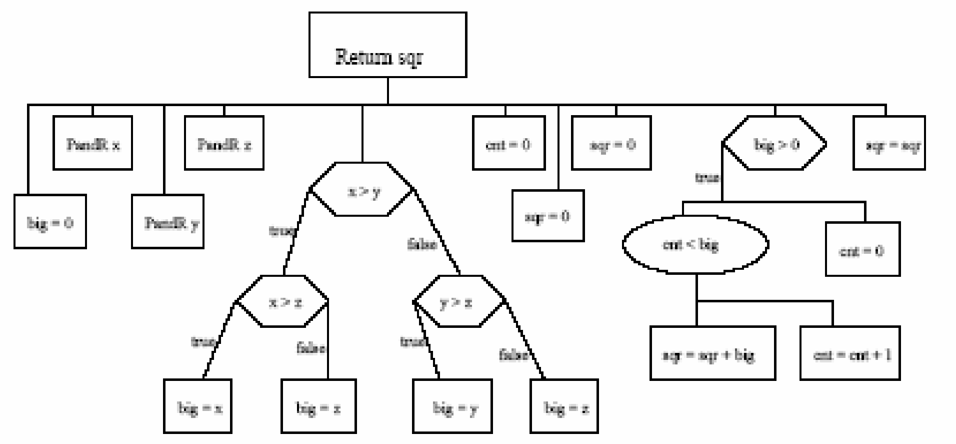
Design-based methods

- Idea: capture syntactic and semantic flow rather than token identity (for source code)

- Replace variable names with IDs correlated with symbol table and data type
- Decompose each *p* into regions of
  - sequential statements
  - conditionals
  - looping blocks – recurse on these
- Calculate similarity from root node downwards

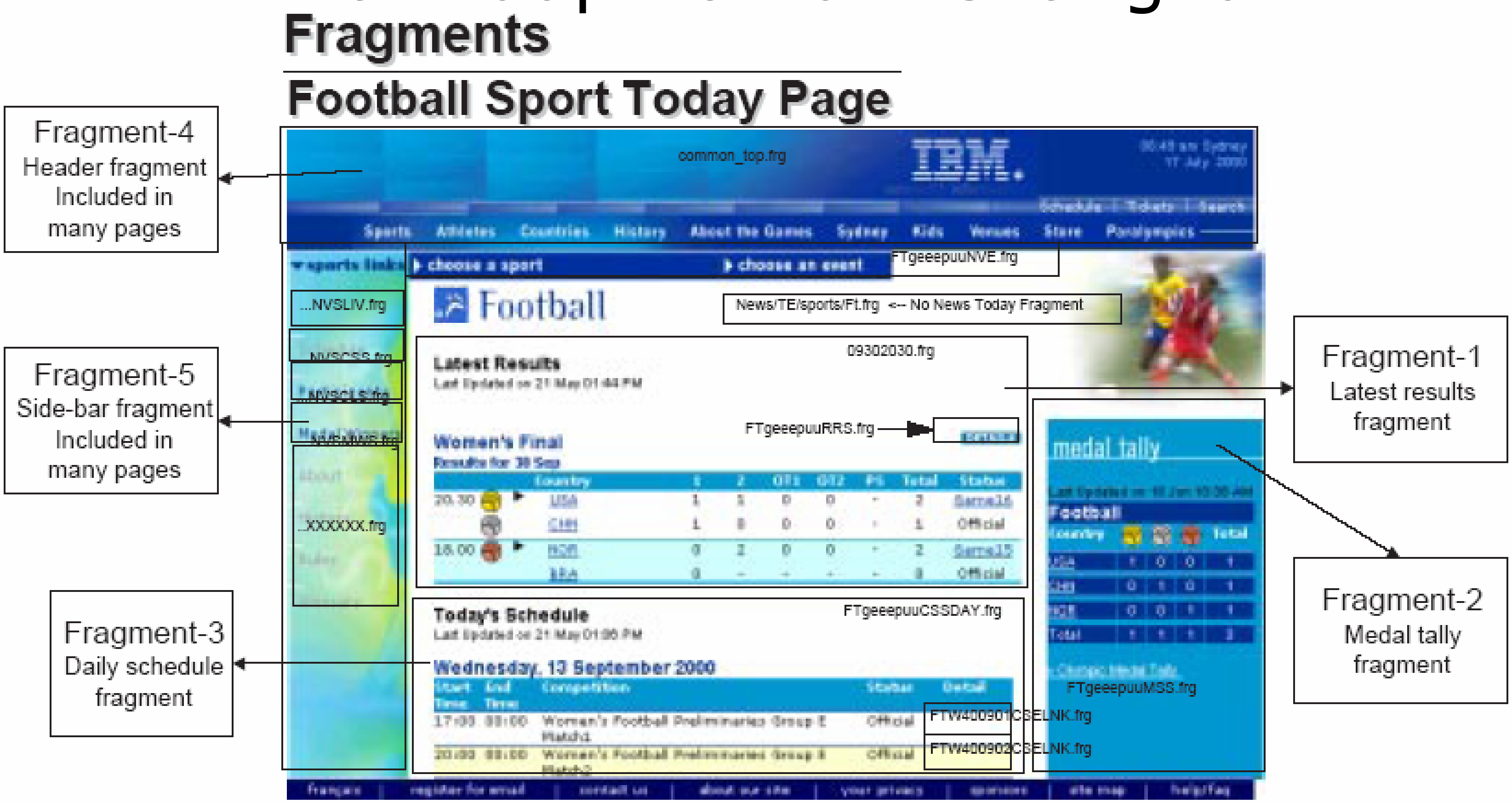
Recursive region coding





Fragments of a web page

Which are duplicated? Changed?



Defining fragments

- Base case: each web page is a fragment
- Inductive step: each part of a fragment is also a fragment if
  - Shared: it is shared among at least *n* other fragments (n > 1) and is not subsumed by a parent fragment
  - Different: it changes at a different rate than fragments containing it

Conclusion

- Signature-based methods common, design-based assumes domain knowledge.
  - The importance of granularity and ordering changes between domains
- Difficult to scale up
  - Most work only does pairwise comparison
  - Low complexity clustering may help as a first pass

References

- Belkouche *et al.* (04) *Plagiarism Detection in Software Designs,* ACM Southeast Conference
- Shivakumar & Garcia-Molina (95) *SCAM: A copy detection mechanism for digital documents*, Proc. of DL 95.
- Bilenko and Mooney (03) *Adaptive duplicate detection using learnable string similarity measures*, Proc. of KDD 03.
- Khmelev and Teahan (03) *A repetition based measure for verification of text collections and for text categorization*, Proc. SIGIR 03
- Ramaswamy et al. (04) *Automatic detection of fragments in dynamically generated web pages*, Proc. WWW 04.

To think about…

- How to free duplicate detection algorithms from needing to do pairwise comparisons?
- What size chunk would you use for signature based methods for images, music, video? Would you encode a structural dependency as well (ordering using edit distance) or not (bag of chunks using VSM) for these other media types?