Automated subject classification of textual Web pages, for browsing

Koraljka Golub
KnowLib, Department of Information Technology, Lund University
http://www.it.lth.se/knowlib/

Introduction

Purpose

explore to what degree automated subject classification based on a controlled vocabulary could be utilized in automated subject classification of textual Web pages
– in comparison to state-of-the-art approach to text categorization, SVM, which requires training documents
– the context of browsing
– user-based evaluation

Research questions

Is hierarchical browsing of Web pages being used today?
How does performance of SVM compare to performance of the string-matching algorithm on a test collection?
What can be done to improve performance of the string-matching algorithm on a test collection?
How does performance of SVM compare to performance of the string-matching algorithm, in the context of browsing of Web pages?

Approaches to classification

SVM
- machine-learning
- state-of-the-art algorithm for automated text classification
- requires training documents
- cca. 8000 training and testing documents per class in RCV1

String-matching on controlled vocabulary
- matching of strings in text to be classified against terms designating subject classes in a controlled vocabulary
- so far in research considered too simple to be good
- our assumption: a good controlled vocabulary provides enough for good classification performance
- doesn't require training documents

Practical info

doctoral project period: July 2003 – July 2007
thesis form: compilation of papers
– 2 different kinds of dissertations in Sweden
  1. monographs
     - almost exclusively in the humanities, theology and law
  2. compilations
     - 3-6 peer-reviewed papers published during the period of postgraduate training and a summary of the papers
     - 2/3 of all theses in Sweden

(From http://www.doktorandhandboken.nu/english/phd_studies.html#3_3)

Lacks in current research

evaluation challenge
- automated classification performance not really tested in the context of operational systems and users
- main problem: indexing inconsistency
- automated classification has not been tested in the context of browsing
**Introduction**

**Background**

**Methodology**

**Results so far**

**To do**

---

**Controlled vocabulary**

- Ei classification scheme
  - 6 main classes
  - decimally subdivided
  - up to 5 hierarchical levels
  - pre-existing intellectual mappings between terms in the Ei thesaurus to terms in the Ei classification scheme

**Table 1.** The number of different types of terms for 92 sub-classes from class 9

<table>
<thead>
<tr>
<th></th>
<th>All</th>
<th>HT</th>
<th>Ca</th>
<th>NT</th>
<th>Pf</th>
<th>RT</th>
<th>ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>8099</td>
<td>932</td>
<td>92</td>
<td>1423</td>
<td>1691</td>
<td>4378</td>
<td>1739</td>
</tr>
<tr>
<td>Avg. class</td>
<td>88</td>
<td>10</td>
<td>1</td>
<td>15</td>
<td>18</td>
<td>48</td>
<td>19</td>
</tr>
</tbody>
</table>

**Problem: test collection**

- test collection of Web pages
  - ca. 1000 Web pages
- the only one that then is for the chosen controlled vocabulary
- very small compared to standard test collections for automated classification (Reuters RCV1 has ca. 100 classes and 800,000 documents)
- decision:
  1) conduct performance evaluation of the SVM algorithm on scientific abstracts collection against intellectually assigned classes
  2) conduct performance evaluation of string-matching algorithm based on different parameters on the same scientific abstracts collection against intellectually assigned classes
  3) cred a collection of Web pages, classify them with each of the two algorithms and conduct user-based evaluation

**Methodology so far**

- log analysis of Renardus for user behaviour
- 92 classes selected from the class Engineering, General (class 9)
  - about 35,000 abstracts from Compendex
- the 1000 Web page collection
  - used for deriving weights for different parts of Web pages

**SVM vs. string-matching**

- on the abstracts collection, SVM outperforms string-matching
  - string-matching in its simplest form without weighting and cut-offs

**Table 2.** Experimental results comparing performance of the two approaches per class, and number of original terms per class in SM (Terms).

<table>
<thead>
<tr>
<th>Class</th>
<th>String matching (SM)</th>
<th>SVM learning (SVM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ei</td>
<td></td>
<td></td>
</tr>
<tr>
<td>445.1.1</td>
<td>0.34</td>
<td>0.54</td>
</tr>
<tr>
<td>445.1.2</td>
<td>0.35</td>
<td>0.56</td>
</tr>
<tr>
<td>445.1.3</td>
<td>0.34</td>
<td>0.54</td>
</tr>
<tr>
<td>445.1.4</td>
<td>0.35</td>
<td>0.55</td>
</tr>
<tr>
<td>445.1.5</td>
<td>0.35</td>
<td>0.55</td>
</tr>
<tr>
<td>445.1.6</td>
<td>0.34</td>
<td>0.54</td>
</tr>
</tbody>
</table>

**Browsing of Web pages**

- browsing of Web pages is heavily used when using classification schemes such as DDC, to which Ei is similar
- it is implied that DDC could serve as a good browsing structure, including terminology

**Improving string-matching**

- words from all elements of Web pages need to be taken, but it doesn’t really matter which weight you use
- the best results in F1 were 3% better than baseline:

\[
\text{Score} = 86 \cdot \text{ScoreTitle} + 5 \cdot \text{ScoreHeadings} + 6 \cdot \text{ScoreMetadata} + \text{ScoreMainText}.
\]
Currently working on

- improving string-matching
  - weights and cut-offs
  - NLP applied to string-matching
    a. single-word inflection
    b. single-word derivation
    c. multi-word morpho-syntactic analysis (c.1. change order, c.2. derivation and permutation, c.3. coordination, c.4. insertion)
    d. semantic variation (d.1. synonymy, d.2. hyperonymy, d.3. manual verification)

To do: user study

- create test collection
  - harvest Web pages from engineering
    - e.g. from Sina or EEVL.
  - classify them using the two algorithms
    - SVM's training documents are Compendex, since no Web pages classified into Ei

- purpose of the user study:
  1. classification accuracy – topical relevance
  2. browsing – being able to find the right class

Issues to discuss

- Any problems with research done so far?
- How to design the final user study?
  - suggestions
    - create a simple user interface
    - given a task such as "find documents on irrigation":
      - first, find the right class
      - second, in that class look at the documents and say if they are in the right place
    - how to make sure to evaluate only accuracy and browsing, and not user interface etc.
    - how many and what type of tasks for how many participants in the study