Supporting Case Acquisition and Labelling in the Context of Web Mining

Vojtěch Svátek, Martin Kavalec

Department of Information and Knowledge Engineering
and Laboratory of Intelligent Systems,
University of Economics, Prague, 13067 Praha 3, Czech Republic
email: {svatek|xkavm04}@vse.cz

Abstract. Case acquisition and labelling are important bottlenecks for predictive data mining. In the web context, a cascade of supporting techniques can be used, from general ones such as user interfaces, through filtering based on keyword frequency, to web-specific techniques exploiting public search engines. We show how a synergistic application of multiple techniques can be helpful in obtaining and pre-processing textual data, in particular for ILP-based web mining. The (two-fold) learning task itself consist in construction and disambiguation of categorisation rules, which are to process the results returned by web search engines.

1 Introduction

Most research efforts in the data mining community is concentrated on algorithms for *discovering* regularities in data. Comparably less attention is paid to the problems related to *acquisition* of this data, and their rendering to the form required by the learning algorithm. The characteristic feature of *supervised predictive learning* (which is topical for this paper) is the existence of a distinct goal (class) attribute, the values of which are assumed to be given, for each learning case. If the underlying reasoning task (classification) is novel in the sense that there are no historical data including the "real" class (such as the results of past loan contracts, in the credit-risk assignment task), then the actual mining process has to be preceded by a (usually) manual, dedicated process of class assignment, also denoted as *labelling*. Unfortunately, this process may present an important bottleneck in the overall data mining task. A specific form of the classification task is *document classification*, where, instead of explicit attributes, we can (and have to) deal with sequences of words and other symbols. If we refine the task even further, we can proceed to the classification of *web pages*. This leads us into an enormous space of documents, representing potential data usable for mining. However, the adequate documents are typically dispersed over many servers, and their properties are to a great extent unpredictable.

In this paper, we present several techniques that have been used to support data acquisition and labelling for a data mining task in the context of web search. In section 2 we present our view of the document categorisation problem, with respect to web documents, and in particular for the case of limited input data. In section 3, we describe the individual case acquisition and labelling techniques used as preprocessing within our web-mining tasks. In section 4, we show some preliminary results of the most complex mining task, obtained via relational (ILP) learning. Finally, we review some related work (section 5) and outline perspectives for the future (section 6).

2 Document Categorisation in the Web Context

Document categorisation is often understood as assignment of (possibly hierarchical) subject topics (such as "Computers", "Finance" or "Medicine") to documents. Subject topics are definitely useful for supporting web navigation; their utility for other tasks such as search and filtering is, however, spurious, as they are often strongly correlated with the actual user's query/profile. In the Vseved meta-search project (see [1] for more detail), we have proposed three "query-orthogonal" typologies, which can be applied on most WWW documents more-or-less independent of each other as well as of the subject topic. Most of this paper deals with *bibliographic categorisation* ("article", "bibliography", "pricelist" and the like), following the Dublin Core metadata system [7].

The input for document categorisation may be quite heterogeneous, ranging from explicit metadata or simple data such as URLs and page titles, to abstract concepts extracted from free text or images. In our work, we have concentrated on the situation when the amount of information is rather limited. This is typically the case for on-line *meta-search* systems, operating solely on a few data items they receive from the primary search engines in response to user queries, and on-line systems for support of *navigation*, which are, again, often constrained to using URLs when making judgements about the locations referenced by the page currently browsed. The importance of the URL in document categorisation lays in the insight it may provide to the directory structure at the host server, beside the information content of the server domain as a whole. Recent studies [5] show that humans can make significant deductions about the content of URLs (in particular, of "longer" ones); most of these deductions can be modelled as simple heuristic rules and performed by computer.

In the current project, we have attempted to learn a rulebase relating web document types to terms from and structure of the URL, as well as other information returned by search engines – name, size, date and textual "snippet" of the page. For the learning task, we have used a fast and straightforward *frequency analysis* of terms and symbols from URLs, completed with structure-sensitive but costly *inductive logic programming* (ILP) using all information mentioned above. We have shown in [6] that a few dozens of pure URL-based rules obtained thanks to frequency analysis can more-or-less successfully assign some generic category to approx. 70-90% of pages retrieved by search engines; 30-60% of the assignments account for bibliographic categories. Future experiments will show the impact of the newly-introduced, ILP-based disambiguation on these figures.

3 Case Acquisition and Labelling

Various techniques have been used, in turn, to eventually obtain labelled data.

Collection of Generic URLs. In order to obtain a large set of generic URLs required for the initial frequency analysis, we have submitted extremely general (to say, "empty") *queries to search engines* – e.g. in the form `+domain:com` in the case of AltaVista. The URLs of the "hits" returned have been then parsed to their constituent parts and to individual terms, abstract (e.g. calendar) concepts have been deduced, and, for the resulting structured collections of terms and concepts, relative frequencies have been computed, while discounting multiple occurrences of the same term from the same server (for details see [6]). The lists of terms and concepts served as a base for manual formulation of category-recognition rules.

Supporting Identification of Ambiguous Terms. In the process of manual formulation of recognition rules, *search engines* have served again, namely to verify the reliability of key terms. Queries of the sort `+url:<term>` or `+url:<term> -host:<term>` have been posed, and the hits visually inspected (only the first 20–30 per query, which should suffice to identify significant deviations from the main, expected, meaning). Some very frequent terms, such as "art" (article, but also page about art), "bio" (biography, but also page about biology), "cat" (catalogue, but also page about cats) or "pub" (list of publications, publicity page, or page about restaurants) have been then submitted for disambiguation.

Collection of Input Data for the Disambiguation Task. To obtain (still unlabelled) cases for the ILP learning (disambiguation) task, *search engines* have been used as in the previous step. This time, however, a fully automatic process has been employed: a special program has called the search engine, extracted the hits from the output pages, and parsed them into their descriptive elements: URL, title, extracted text, size and date.

Frequency-Based Case Filtering. During the visual inspection of hits corresponding to ambiguous terms, we have observed that in addition to clearly identifiable positive and negative examples of the category in question (e.g., for the URL term "art", scientific and newspaper articles vs. pages dealing with art), some "problematic" ones have been obtained. The difficulty was usually related to one of the following: unexpected, *marginal semantic* of the term (e.g. art can also stand for Arthur, artificial or artillery...), *cumulation of semantics* (e.g. articles about art), or *complete ignorance* (e.g. pages in an uncommon language, or with the visible part unrelated to the term). Obviously, such examples would not be of much use in inductive learning of the target concept, and their manual labelling would be waste of time. In order to eliminate "problematic" cases without human intervention, we have experimented with a frequency-based filter. The filter is based on the assumption that an example may be useful for learning only if it has some property that can be found in a sufficient number of other examples. In

	"Clear"	"Problematic"	Total
Before filtering	54 (62%)	33 (38%)	87
After filtering	50 (72%)	19 (28%)	69

Table 1. Effect of frequency-based filtering

text mining[1], the example must contain a sufficiently frequent term. We can thus select a set of terms (and abstracted concepts) with frequency above a certain threshold and keep only those examples containing them. We have experimented with filtering on a small set of 87 examples, of which approx. 38% have been previously identified as "problematic" in one of the senses above. By means of the frequency filter, we have eliminated all examples but those containing at least two "frequent" terms. In this way, approx. 20% of examples have been rejected, and in the resulting subset, only 28% of examples were "problematic"; at the same time, we have lost only few "clear"examples (Tab. 1). Note that the overall cutoff of the volunteers' time gained by frequency filtering may be even higher than the plain difference in the count of examples, since a large proportion of "problematic" examples were actually difficult to evaluate even by human, and, in addition to their inutility in learning, they would make the user spend more time on them than on "clear" ones.

Interactive Semantic Indexing. The actual labelling (assignment of semantic indices) was done by 2-3 volunteers, using an interactive program, which displayed the *information about each page* (i.e. re-structured output of the search engine), in turn, offered a *menu of semantic indices*[2], *recorded* the answer, and enabled to *backtrack* to previous answers and change them, if necessary. The labelling results of different people have been compared, and only the cases with identical index obtained (the degree of concordance was usually rather high, which can be attributed to prior frequency filtering) have been converted to predicate representation and submitted to the inductive learner.

4 Preliminary Results of ILP-Based Mining

The ILP task has been performed on the following predicate representation. `sterm(Id, Term, Pos)` indicates the occurrence of term $Term$ in the snippet of example Id, on position Pos. Analogously, we use `tterm` for the page title, `dterm` for the directory part of URL, and `fterm` for the filename. `no_fterm(Id]) and no_dterm(Id)` mean that the URL doesn't contain a filename or directory part,

[1] It is an interesting question whether frequency-based filtering could also be used when labelling tabular data; in principle, there is no hindrance to that.

[2] Such as, for the "art" cases: *scientific/newspaper article*, *page about art*, *catalogue article (goods)*, *article of law*, *other*, *undecidable.*

respectively. `owner(Id) and no_owner(Id)` indicate whether the URL contains the owner indication (~⟨*user*⟩ at the beginning of the path) or not. `in_srv(Id) and not_in_srv(Id)` specify whether the term to be disambiguated occurs in the name of server or not. Finally, `nextto(Pos1, Pos2)` holds if $Pos2 = Pos1 + 1$. It is used to express the adjacency of terms.

As inductive learner, we are currently using *Aleph*[3]; For the above mentioned example of "scientific/newspaper article" semantic index, it has returned (depending on the settings) approx. 10–15 positive and a similar number of negative rules. Some interesting positive ones were e.g.:

```
[Rule 10] [Pos cover = 9 Neg cover = 0]
pos_example(A) :- sterm(A,by,B), fterm(A,art,C), no_owner(A).
[Rule 13] [Pos cover = 8 Neg cover = 0]
pos_example(A) :- fterm(A,'_num',B), fterm(A,art,C).
```

Rule no.10 probably covers some articles placed on specialised publishing servers, since on a personal homepage the author would probably name the file according to the topic of the article rather than by "art", and also would not explicitly state the authorship using "by". The `'_num'` symbol in rule no.13 is the abstraction of number, thus if the filename contains art and a number, it is probably an article. For negative examples, we got e.g.:

```
[Rule 2] [Pos cover = 31 Neg cover = 0]
neg_example(A) :- sterm(A,x,B).
[Rule 4] [Pos cover = 17 Neg cover = 0]
neg_example(A) :- no_fterm(A).
```

Rule no.2 clearly covers some artwork, since online galleries often state physical dimensions, in ASCII, as *width* x *height*. Rule no.4 states the obvious fact that articles are not stored in the directory index page (URL with no filename). We can see the "truly relational" predicate `nextto` has not been needed, since "propositional" predicates sufficed to discriminate between positive and negative examples, (e.g. in Rule 2 above, the "x" symbol alone has "substituted" the sequence "number-x-number").

5 Related Work

Using data output by search engines for inductive learning has been the topic in the MetaCrawler-STC project [8]. Unlike our project, the task consisted in *clustering* the hits with respect to subject topic rather than in classification to predefined (moreover, bibliographic) categories. In this sense, our work is rather similar to the AdEater project [3], which concentrated on the *binary task* of distinguishing between banner ads (which can be understood as a sort of

[3] An implementation of *Progol*, available at `http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.html`

bibliographic category) and other graphics on web pages: an interactive, graphic tool has been used to assist users in labelling examples as positive or negative, and a decision tree has been induced over terms from URLs of images plus some additional information. Further, Mitchell's group at CMU [2] attempts to use ILP in order to recognise the page type. The input for learning is, however, fulltext analysis (including HTML), which implies the use of a more complex predicate representation. In terms of using ILP for term disambiguation, our project is also akin to some natural language disambiguation projects [4].

6 Future Work

In the paper, we have presented several techniques supporting the acquisition and labelling of cases to be input to the learning process, in the specific context of mining web search results. Some of the techniques are likely to be reused for other web-mining tasks, in particular for information extraction from the page fulltexts. Future work should also concentrate on assessing the utility of the ILP approach. It provides a comfortable way to specify background knowledge and term containment in examples; it is however unclear whether its representational power is indispensable for the tasks like the one above.

The research on this topic has been partially supported by Grant no.VS96008 of the Czech Ministry of Education, "Laboratory of Intelligent Systems".

References

1. Berka P., Sochorová M., Svátek V., Šrámek D.: The VSEved System for Intelligent WWW Metasearch. In: (Rudas I. J., Madarasz L., eds.:) INES'99 – IEEE Intl. Conf. on Intelligent Engineering Systems, Stara Lesna 1999, 317-321.
2. Craven M., DiPasquo D., Freitag D., McCallum A., Mitchell T., Nigam K., Slattery S.: Learning to Extract Symbolic Knowledge from the World Wide Web. In: Proc. of 15th AAAI, Madison, WI, 1998.
3. Kushmerick N.: Learning to remove Internet advertisements. In: 3rd Int. Conf. on Autonomous Agents, 1999.
4. Popelínský L., Pavelek T.: Mining Lemma Disambiguation Rules from Czech Corpora. In: PKDD'99 - Principles of Data Mining and Knowledge Discovery, Prague, 1999, 498-503.
5. Stanyer D., Procter R.: Human Factors and the WWW: Making sense of URLs. In: (Brewster S., Cawsey A., Cockton G., eds.:) Human-Computer Interaction – Interact'99 (Vol.2). The British Computer Society, 1999, 59-60.
6. Svátek V., Berka P.: URL as starting point for WWW document categorisation. In: (Mariani J., Harman D.:) RIAO'2000 – Content-Based Multimedia Information Access, CID, Paris, 2000, 1693-1702.
7. Weibel S., Kunze J., Lagoze C., Wolf M.: Dublin Core Metadata for Resource Discovery. IETF #2413. The Internet Society, September 1998.
8. Zamir O., Etzioni O.: Web Document Clustering: A Feasibility Demonstration. In: SIGIR'98, Melbourne, Australia.