SchemaCMD:

An XML-based storage schema for the compilation of mixed-source CMD corpora

Cornelius Puschmann

Department of English Language and Linguistics

University of Düsseldorf, Germany

Short Abstract

This presentation will outline an XML schema for the segmentation and storage of data from Internet sources, specifically those which utilize so-called web feeds (often associated with the RSS protocol). It is based on the faceted classification scheme recently proposed by Susan Herring and aims to make data from diverse sources accessible and comparable in a single format.

Long Abstract

While the Internet has been a treasure trove for linguistic investigation ever since its inception, the systematic collection of data specifically for linguistic purposes has been partly hampered by the fact that textual data existing on the Web in HTML format is not tagged semantically but visually and structurally. Because tag information in HTML documents provides rendering instructions to web browsing clients but includes very little (useful) meta-data, documents retrieved from the Web are not as contextually rich as they could be, often omitting information on the author, the exact time of creation, etc.

This paper presents a simple, web-based corpus-management tool working with a faceted structuring schema based on the eXtensible Markup Language (or XML) for the storage and linguistic analysis of Internet sources that use so-called data feeds (or web feeds, often associated with the RSS and Atom protocols) for syndication. As the breadth of user-generated data, for example in blogs, wikis and services associated with social networking sites continually increases, such feeds are posited to become a standard method of content distribution - a channel that retains the meta-information that was previously omitted.

The specific investigative focus of SchemaCMD will be to highlight speaker variation as a significant factor in corpus linguistic analysis. Because content from web feeds can virtually always be directly linked to its respective author by merit of the feed meta data, such texts allow for a rich contrastive analysis of individual language production. This makes the compilation of large corpora that capture variation over time, variation between different texts produced by the same author and variation between texts written by different authors possible, which adds a new dimension to corpus linguistics in the area of computer-mediated discourse.

An example: the corporate web log corpus (CBC)

For my thesis work on The Corporate Blog as an Emerging Genre of Computer-Mediated Communication I decided to build a representative corpus of company blogs. The above-mentioned corpus tool that integrates the SchemaCMD classificatory scheme (following Herring, 2007) was built for this purpose, though it can be used for any corpus that relies on web feeds as a data source. The fact that feeds form the basis of the corpus makes an in-depth contrastive and diachronic investigation of corporate blogs as a text type possible. As data

taken from web feeds is already represented in XML, the storage of such information in a relational database (MySQL for my corpus) was merely a matter of processing and copying the data. Below is a typical feed entry, in this case taken from McDonald's Open for Discussion blog. The Magpie RSS library for PHP is used to fetch the entry as an array:

```
Array
(
    [title] => A Visit to McDonald's
    [link] => http://csr.blogs.mcdonalds.com/default.asp?item=256774
    [description] =>

Last week, Julia Hailes, a co-founder of SustainAbility, joined us in
Chicago at our Corporate Relations Conference.
Julia participated in a panel discussion [...]

    [comments] => http://csr.blogs.mcdonalds.com/default.asp?item=256774
    [pubdate] => Wed, 02 May 2007 10:11:15 EDT
    [author] => undisclosed@blogs.mcdonalds.com (csr)
    [guid] => http://csr.blogs.mcdonalds.com/default.asp?item=256774
)
```

There are a number of issues with and differences between different version of the RSS and Atom specifications that need to be considered before content from a feed can be retrieved. For example, the fields description (RSS) and summary (Atom) can both be used to include the full text of an entry or merely the first n words (50 words is one typical cut-off point). Feeds using the atom Atom 1.0 specifications usually encode the full text of entries in the appropriate content field instead, reserving the summary field for a summary in addition to the complete entry. Other possible sources of errors include date format conversion from RSS/Atom to MySQL and the large number of fields in both protocols which are optional, as they can't be relied on in all further processing steps. The data structure of the posts table inside the MySQL database thus closely mirrors the structure of the feed.

Table structure for cbdb.posts:

post_id	post_stats_wc
post_blog	post_stats_tc
post_title	post_stats_sc
post_link	post_stats_awl
post_fulltext	post_stats_asl
post_author	post_wordlisted
post_category	post_processed
post_comments	post_omitted
post_guid	post_realdate
post_pubdate	

The fields title, link, fulltext, author, category, comments, guid and pubdate are taken directly from the feed, while the other fields store statistical data and internal flags to indicate which processing steps have already been performed and which are still to follow. All entries in the posts table are relationally tied to their respective parent items in the blogs table via ID numbers.

When the blogs that are being tracked are checked for new content, the feed data provided by MagpieRSS is compared with the record inside the posts table. If the feed contains entries with guid values which do not occur in the table, it is assumed that they are new and they are copied to MySQL. Duplication of old items is avoided by checking guid values, or, in those

cases where guid is omitted in the feed, by checking title – pubdate combinations, as they can also be assumed to be unique.
Other tables are used to store derived statistical data and information that is relevant for the internal function of the tool. The following overview lists each table in MySQL, along with the current number of records and a description.

TABLE	RECORDS	DESCRIPTION
blogs	144	blog data
bnc	100	BNC list of 100 most frequent English words (for comparison)
categories	6	Manually created labels that can be applied to blogs (i.e. "product blog", "PR blog" etc)
collections	6	Manually created labels that can be used to created sub-corpora (e.g. "blogs", "press releases" etc)
gram2	165,789	2-grams by frequency(*)
gram3	280,338	3-grams by frequency(*)
gram4	318,056	4-grams by frequency(*)
gram5	325,720	5-grams by frequency(*)
pos	23,210	parts of speech frequencies by post
posts	23,210	post data
poststats	23,210	extended post statistics
tokens	5,920,583	tokens (1 row for each single word in the corpus; dependent on the types table)
types	210,393	types (unique strings) in the corpus; referenced by the tokens table
13 table(s)	7,290,765 (244.1 MiB)	

(04 May 2007)

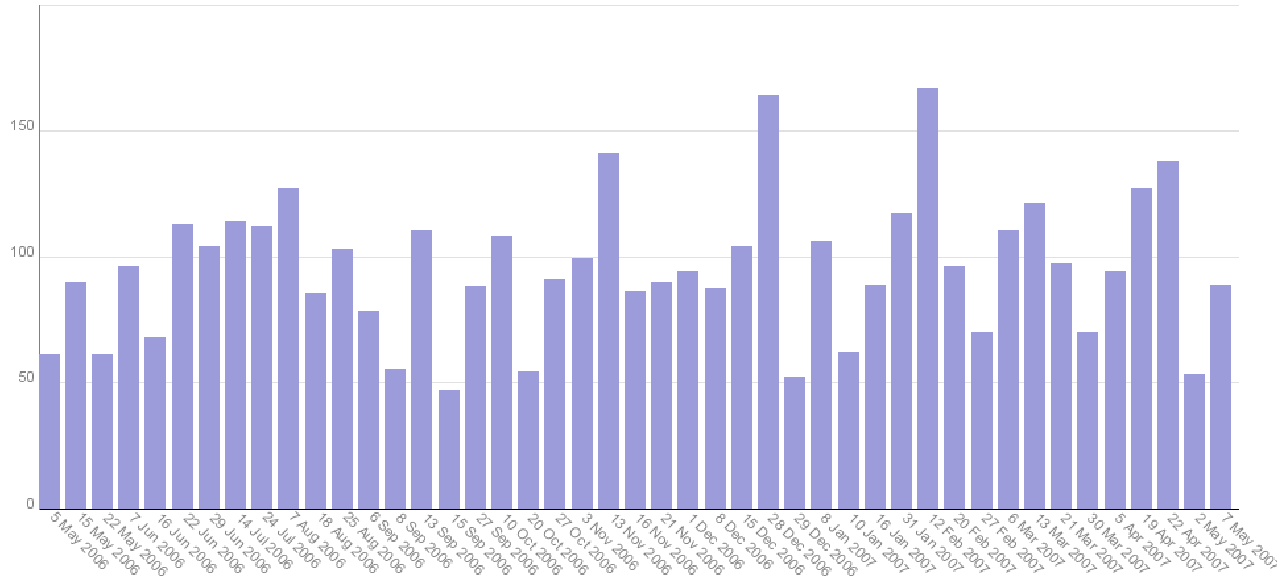
* while a function for counting n-grams (and pos-grams, because of the structure of the types table) has been implemented it is not currently in use due to the high computational cost associated with performing the necessary calculations for each entry

Once all new entries have been recorded, the individual items are wordlisted using TreeTagger. To do this as simply as possible, the main text of a post is written to a plain text file buffer (named `post.txt`) on the hard drive. TreeTagger is then called with the options `-token -no-unknown` and `-quiet` and the result of the tagging process is written to another buffer (`post-tagged.txt`). Finally, the tagged text is further processed with PHP and the result is written to the types and tokens tables, creating one row in the tokens table for every word and a new record in the types table if the string in question has not been previously recorded.
In a third step, extended statistical data is then computed and written to several different tables in the database. Virtually all of this information could also be computed on the fly, whenever the researcher needs it (e.g. the average word length for a given blog post), but it makes much more sense in terms of computational cost to make these calculations once and store and retrieve them as needed, especially in the case of cumulative computations where several results depend on one another. The following values are computed either upon indexing or later, when the researcher evaluates the data.

- word (token) count (WC)
- type count (TC)
- sentence count (SC)
- average word length (AWL)

- average sentence length (ASL)
- part-of-speech frequencies (POS chart)
- word frequencies (wordlist)

An interesting dimension in this context is that this data can be compared on multiple levels, i.e. for a single post, for all posts in a blog and for multiple blogs grouped together using a range of criteria. The extreme uniformity in how blog entries are segmented makes these comparisons feasible and allows for a detailed contrastive evaluation of the data.
The f-score (Heylighen & Dewaele, 2001) is a basic formality measure that is computed via the following frequency calculation: 0.5 * ((N + ADJ + PRP + DET) - (PN + V + ADV + ITJ) + 100).
The following table shows f-score variation over time for the abovementioned Open for Discussion blog.



This and other scalar measures can be used to track and evaluate the text production over time for a range of sources, which can be in turn compared contrastively inside the boundaries of an (assumed) functional genre. The advantage of such an approach is its greatly improved granularity: variation is no longer assessed *only* between genres, but inside of them and among individual datapoints as they are provided by a single source.
Implementing an annotation scheme such as SchemaCMD thus ultimately serves the purpose of creating a computationally exploitable link between a text and its producer that takes as many contextual variables as possible into account.

References

Herring, S.C. (2007). A Faceted Classification Scheme for Computer-Mediated Discourse. Language@Internet. http://www.languageatinternet.de/articles/761. Retrieved 2.5.2007.

Heylighen, F. and Dewaele, J.-M. (2002). Variation in the contextuality of language: An empirical measure. *Foundations of Science*, 7, 293-340.