PDFMeat: Managing Publications on the Semantic Desktop

David Aumüller, Erhard Rahm
University of Leipzig, Leipzig, Germany
{aumueller, rahm}@informatik.uni-leipzig.de

ABSTRACT

Researchers maintain bibliographies and extensive sets of PDF files of scholarly publications on their desktop. The lack of proper metadata of downloaded PDFs makes this task a tedious one. With PDFMeat we present a solution to automatically determine publication metadata for scholarly papers within the user's desktop environment and link the metadata to the files. PDFMeat effectively matches local full texts to an online repository. In an evaluation for more than 2.000 diverse PDF files it worked highly reliable and showed excellent accuracy of up to 98 percent. We demonstrate PDFMeat for different sets of papers, highlighting the semantic integration and use of the retrieved metadata within the file browser of the desktop environment.

Categories and Subject Descriptors

D.2 [**Software Engineering**]: Interoperability

General Terms

Design, Experimentation, Management, Measurement

1. INTRODUCTION

Researchers typically download and collect numerous PDF papers onto their desktop for reading and further reference. These full texts get lost between other files in the depth of the file system, often with similar and meaningless file names – resulting in low accessibility of valuable information. Organizing full text files by renaming and moving them remains tedious. Furthermore, the publication metadata that the researcher probably was presented with on the digital library website is in no way connected or attached to the downloaded file. Hence, this metadata is not available for further usage, e. g. for organizing the files as well as for easing the preparation of reference lists, for instance using BibTEX. With PDFMeat we present a tool to automatically determine the bibliographic metadata for PDFs of research publications. It establishes a link between the files and their metadata providing a tight integration of the metadata into the user's desktop environment, offering management of both files and metadata with little user effort.

The need to collect bibliographic metadata is caused by its absence in PDF files of research papers. Although the PDF specification allows embedding metadata into the header these metadata rarely correspond to the enclosed publication full text. Wikipedia lists many "reference management software" but their focus is mostly on maintaining bibliographies and not the automatic identification of bibliographic metadata from web sources. Some systems provide a limited form of metadata acquisition by a DOI-based metadata lookup. However, for many papers DOIs are either not available, hard to obtain, or cannot be resolved. Several studies focus on heuristic approaches to extract bibliographic attributes directly from PDF files [8, 11, 13], e. g. identifying the document title by font size. Such heuristics are error prone the more the publications vary in style, and cannot acquire information that is not present in the full text, as e. g. often the year and the venue (i. e. journal, or conference) are and remain missing. In [13], the thus extracted title is used to query a local bibliographic database snapshot and simply retrieve the first hit, i. e. no further matching is undertaken.

With PDFMeat we do not rely on such extraction heuristics or the availability of DOIs or need to setup a reference database. Instead, we utilize query-able repositories such as Google Scholar to retrieve and match bibliographic metadata. Our evaluation for computer science publications shows very high effectiveness and we do not know of any other tool offering such reliable metadata acquisition.

Bibliographic metadata obtained for PDF files of research publications empowers users in the following tasks:

- integrated management of both files and metadata
- semantic desktop search with true publication metadata
- ad-hoc availability of metadata for reference work
- provision of publications to community bibliographies

In this paper we provide a description of the PDFMeat approach that was first sketched in [1] as PDF Metadata Acquisition Tool. We present an evaluation of its matching effectiveness. Further, we describe how PDFMeat uses the retrieved metadata within the user's file manager rendering the desktop environment more semantic. That way, PDFMeat offers a light-weight and user-friendly solution to some challenges identified e. g. in personal information management [4, 7] and around the semantic desktop [3, 5].

2. MAPPING FULL TEXT TO METADATA

To determine bibliographic metadata for local full text files we exploit the available data and query capabilities of existing online repositories. Available data provided by Google Scholar includes typical publication metadata attributes, represented in HTML as well as in some user-specified structured formats, e. g. BibTEX. Like common web search engines, Google Scholar also provides links to the indexed full texts and allows searching the contents of the full texts. Our metadata acquisition approach utilizes this searchability of the full text, as PDFMeat queries Google Scholar using a kind of generated fingerprint which consists of a set of terms from the full text. The first step in constructing a query string from the full text is to convert the PDF to a plain text. We extract the text (using 'pdftotext') neglecting the layout except for new line characters. We consider the document head as the top part of the first page, up to keywords such as 'Abstract' or 'Introduction', or up to the end of the line containing the author e-mail addresses denoted by the 'at' sign. Failing to find such a landmark we just keep the whole first page as document head, thus not being dependent on heuristics.

We construct the query against the metadata repository out of sentences from the beginning of the full text. In only using whole sentences as base for query generation we avoid erroneous terms that may result e. g. from author name indices or other noise. From the selected sentences we remove stop words commonly ignored by web search engines as well as tokens that contain special characters. Problematic terms may result from the text conversion process due to character encoding or print layout, e. g. hyphenated words and ligatures. As these may be converted differently by Google Scholar's converter including them in the query may lead to non-matches in the metadata repository. We address these issues again in matching potential metadata entries to the full text in question. Further features that we extract in this process for supplying the user with more metadata attributes than offered and retrievable by the remote repository include the abstract, author affiliations by way of author e-mail addresses, and the DOI where available.

Querying search engines usually results in multiple hits. To identify the metadata entry corresponding to the full text in question, a matching of entries to the full text has to take place. Thus, we extract the publication metadata entities from the HTML result page and check which of the returned titles is contained as substring in the document head as defined above. In this step we are dependent on identical transliteration of special characters such as ligatures (ﬄ, ﬃ vs. ffl, ffi). For example, the returned metadata entry may contain the term "efficient" while the full text represents the same word using ligatures ("eﬃcient"). We also transcribe letters containing diacritics such as umlauts to their plain variant glyph having thus experienced better conformity to Google Scholar. Only if the title normalized that way is not contained in the analogously normalized document head we pursue a fuzzy substring matching, using edit distance as described in [14], thus also accepting matches that are a few characters off. For the identified matching metadata entry, PDFMeat then retrieves Google Scholar's BibTEX entry, enriched by the current citation count, which could be later updated using a cron job. Alternatively (or additionally for better quality due to manually curated metadata) we could further retrieve the corresponding entry from DBLP, ACM, or another suitable online repository.

3. ANNOTATION AND CATEGORIZATION

Generally, bibliographic metadata regarding the authors is limited to their names. The full text though usually contains the authors' affiliations, i. e. institution and location, as well as their email addresses. These are valuable annotations, e. g. for the organization of publications by country or institution. To acquire this data we do not try to extract the affiliations directly from the full text as affiliation strings are very heterogeneous and thus difficult to process. Instead, we merely extract the authors' email address hostnames via the 'at'-sign which usually denote their country, institution, and often department. Using our mailhost–institution web service [2] institution names and locations could be mapped.

Similar to the encyclopedic articles in Wikipedia, we also want to categorize publications. Various approaches exist, e. g. [9, 10], to tag content along the Wikipedia category system, e. g. by comparing features such as weighted occurrence frequencies of individual terms against a set-up local search index of Wikipedia articles. For PDFMeat we experiment with ad-hoc web-based approaches of matching papers to Wikipedia, e. g. by issuing a web search restricted on Wikipedia's category URLs. As keywords we extract most denoting terms from title and abstract using TF-IDF or determine compound nouns therein via POS tagging. The web search engine then returns links to Wikipedia category pages that correlate with the content in question. For categories new to our collection we extend our category graph by retrieving their broader category terms from Wikipedia. To not end up in a too large category graph, we prune those categories that are only used few times – the rationale being, that a user's or community's collection of publications usually not consists of totally unrelated works.

4. SEMANTIC DESKTOP INTEGRATION

PDFMeat maintains a local database for the retrieved metadata and establish the connection between metadata and file using an unique identifier for the local file. One candidate key is a hash/fingerprint value calculated of the file content. As this can be costly for large media files, some approaches only consider parts of the file. For our tool we use the 'inode' value of the local file as identifier, which is a numeric value distinct to each file of a device. The metadata for a file in question is thereafter located by its inode number in the metadata store. (We additionally store a hash value for globally identifying the PDF in a potential online PDFMeat store; as further alternative we consider the use of the 'xattr' extended file attributes for storing the retrieved metadata attribute values.) Also, PDFMeat offers the possibility to embed the metadata into the PDF file header, but this should not serve as first solution as users may be reluctant to allow file alterations. Both simple and extended PDF metadata specifications for embedding are supported, of which the XMP format can e. g. be imported by the popular JabRef bibliography manager. Such managers also allow to open PDF files linked in BibTEX databases, as produced by PDFMeat. As PDF files may be renamed and/or moved on the file system their links in the BibTEX file would be broken. Thus we implemented a syncing function to update the local file path as linked in BibTEX databases. To re-sync we need to crawl the file system or a user specified folder for PDFs and check whether the publication is known and still correctly referenced in the local metadata store and in the user specified BibTEX file.

To identify the according entry in a BibTEX file we maintain a special 'pdfmeat' attribute containing the inode number, restricting the approach only to entries that once were established by PDFMeat; for other cases the metadata has to be acquired to then try to find an according entry by way of title, author, and year attributes; failing that we store the new entry.

For increased usability of our metadata acquisition approach we provide a tight integration into the user's desktop environment. Dealing primarily with local publication full text files the most direct integration of PDFMeat is into the user's file manager. File managers provide a navigational view on the file system, typically with a tree representation of the directory structure or a list of bookmarks to directories in a sidebar on the left, and a list of files within the currently selected directory in the main window. In the tabular, column based view of this file list each list entry represents one file showing typical file attributes such as name, size, mime type, timestamps, etc. Our desktop integration implementation is exemplified using Nautilus, the default file browser in GNOME, which in turn is the default desktop environment of many Linux operating system distributions. (For non-Linux users we provide a web service to be able to use at least the metadata acquisition functionality, either by manually uploading a set of local PDF files or using a 'thin client' application that extracts and uploads the plain text from the first page(s) of the PDF to acquire the corresponding BibTEX metadata entry.) PDFMeat integrates the following functionality within the file manager:

- retrieve metadata for files via the context menu
- display metadata in additional sortable columns
- manage metadata within a property page tab
- re-organize full text files via metadata attributes

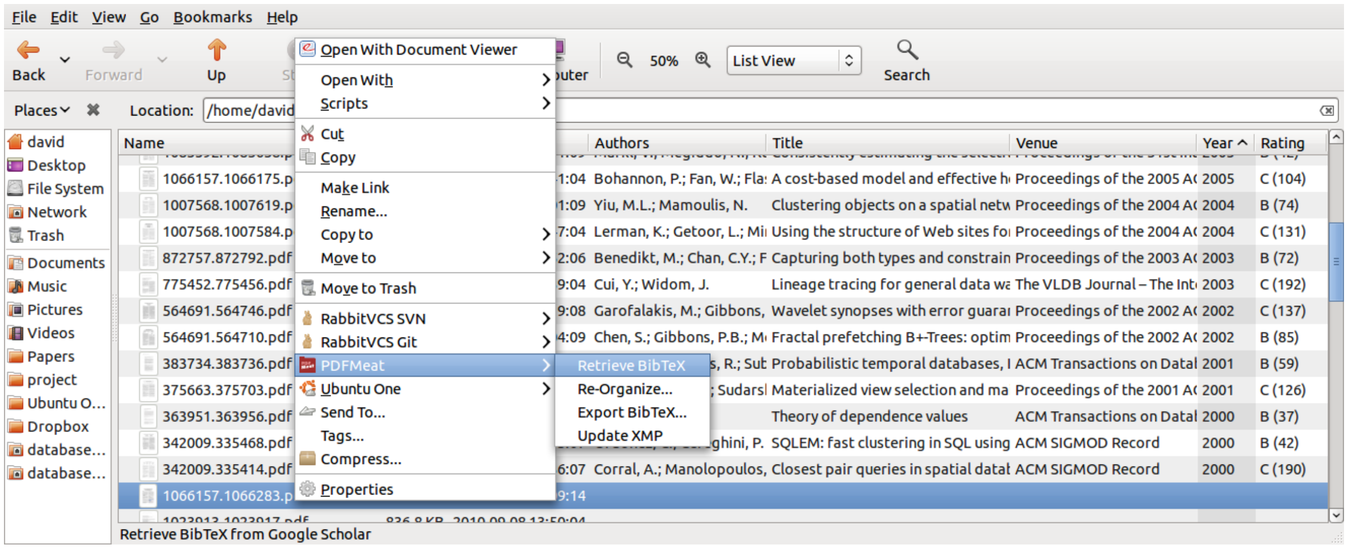
Figure 1 depicts the file manager listing a set of PDF files that are already handled and known by PDFMeat, i. e. BibTEX entries are downloaded, stored, and linked to the files, of which selected attributes (here: author list, title, venue, year, and rating/citations) are chosen by the user to be presented as extra columns in this view. Right-clicking on a PDF file shows the context menu which PDFMeat extends by a sub-menu to trigger the metadata acquisition process and other actions. For successful acquisitions, the metadata values instantaneously fill the according column values. A desktop notification informs unobtrusively about the result of the action. Further, PDFMeat extends the file property dialog, available via 'Properties', by an extra tab/page, showing the retrieved BibTEX entry, which can be edited by the user, i. e. it could also be supplied, curated, and extended manually.

Although PDFMeat currently uses an external database as metadata store, the linkage between full text files and metadata entries survives moving and renaming files – independently of the file manager and its integration. Also, a file re-organization function allows to automatically rename and move publication files according to user defined patterns, e. g. to move files into folders determined by some metadata attributes. Further semantic desktop integration and functionality provided by PDFMeat includes a desktop search plugin, allowing to search for publications on the desktop by their real metadata, as well as a standalone faceted search browser to have an overview of all one's publications on the desktop, but also for listing PDFs without yet retrieved metadata. A similar web application could also serve as community portal to a shared bibliography. Additionally, PDFMeat offers automatic addition of new publications by watching folder(s) for incoming PDF files that will be processed as soon as they are downloaded. That way, users do not need to trigger the metadata acquisition manually, incoming files can be re-organized instantaneously, and the entry appended to the BibTEX bibliography.

5. ENTITY MATCHING EVALUATION

As the general metadata attributes such as title, author(s), venue, and year are most important, also for further data integration tasks, we explicitly look at this matching process in more detail. For evaluating the quality of the mapping between local full text files and retrieved metadata entities we use a set of metadata entries and their linked full text files from the ACM digital library. Our test set resembles a random set of publications from out of ten years of four different venues, both conference proceedings and journal articles: VLDB, VLDBJ, SIGMOD, and PODS. Some of these PDF files did not convert to plain text – reasons can be that the text is represented as image only, enabled read restrictions, or unknown document encoding issues (the Linux 'file' command considered some of the pdftotext results as data instead of text). In our test set of 2.095 documents, the converted text of 89 files (4.2%) was unreadable for both human and machine, which we hence ignore in our further evaluation. Also, we focus on the accuracy, i. e. the share of input publications for which we could correctly determine the bibliographic metadata. To determine this measure, we need to take into account that publication titles may slightly vary between full text and bibliographic databases, e. g. one source may use colons where the other uses dashes to separate e. g. a prototype name from the rest of the title. Corresponding entities may vary in more than typos or different punctuation, though, as sometimes authors decided to use a slightly changed title to the one entered into the submission system when preparing the final print version. Thus, in judging true positives via the title attribute we allowed minor differences by computing the levenshtein edit distance for comparisons. To avoid a discussion of suitable similarity thresholds we manually labeled the share of correspondences having a similarity ratio below 1. The three evaluations in Figure 2a) give different values for accuracy, determining true positives via the title attribute only, both title and year attribute, as well as title attribute but year only if returned from the metadata repository. Overall, the evaluation shows good results with all values above 94%. The best accuracy of about 98% is of course achieved by only comparing the title. For establishing the data–metadata mapping we issued up to three queries per document. The majority of publications were identified using a single query only, as seen in Figure 2b), log. scale.

Our test set offers a high visibility of the documents on the web as Google was able to index their full texts. This accessibility of documents is not restricted to computer science literature or the selected test set. Nevertheless, successful results may vary for other sets of publications, depending on the coverage of documents in Google Scholar. Recent studies [6, 12] consider its coverage quite high for a large variety of domains and found improvements of the index over time. Nevertheless, PDFMeat could also use different bibliographic repositories. Querying a service without full text



Name		Authors	Title	Venue	Year	Rating
1066157.1066175.p	1:04	Bohannon, P.; Fan, W.; Fla	A cost-based model and effective h	Proceedings of the 2005 A	2005	C (104)
1007568.1007619.p	1:09	Yiu, M.L.; Mamoulis, N.	Clustering objects on a spatial netw	Proceedings of the 2004 A	2004	B (74)
1007568.1007584.p	7:04	Lerman, K.; Getoor, L.; Mi	Using the structure of Web sites fo	Proceedings of the 2004 A	2004	C (131)
872757.872792.pdf	2:06	Benedikt, M.; Chan, C.Y.; F	Capturing both types and constrain	Proceedings of the 2003 A	2003	B (72)
775452.775456.pdf	9:04	Cui, Y.; Widom, J.	Lineage tracing for general data wa	The VLDB Journal – The Int	2003	C (192)
564691.564746.pdf	9:08	Garofalakis, M.; Gibbons,	Wavelet synopses with error guara	Proceedings of the 2002 A	2002	C (137)
564691.564710.pdf	4:09	Chen, S.; Gibbons, P.B.; M	Fractal prefetching B+-Trees: optim	Proceedings of the 2002 A	2002	B (85)
383734.383736.pdf		s, R.; Sut	Probabilistic temporal databases, I	ACM Transactions on Datal	2001	B (59)
375663.375703.pdf		; Sudarsl	Materialized view selection and ma	Proceedings of the 2001 A	2001	C (126)
363951.363956.pdf			Theory of dependence values	ACM Transactions on Datal	2000	B (37)
342009.335468.pdf		ghini, P.	SQLEM: fast clustering in SQL using	ACM SIGMOD Record	2000	B (42)
342009.335414.pdf	6:07	Corral, A.; Manolopoulos,	Closest pair queries in spatial datal	ACM SIGMOD Record	2000	C (190)
1066157.1066283.p	9:14					

Retrieve BibTeX from Google Scholar

Figure 1: PDFMeat context menu and retrieved publication metadata values in provided file manager columns

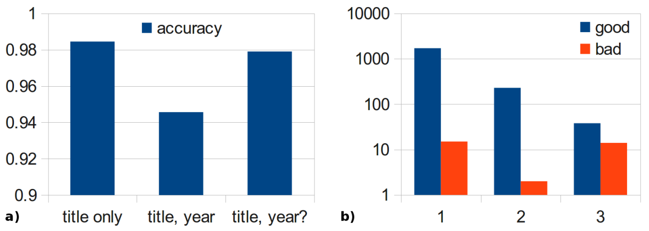


Figure 2: a) Retrieved metadata matching real data b) Queries needed to find correspondences (max. 3)

search, though, would require to know at least the most basic metadata, such as title and authors, calling for common extraction heuristics known to be errorprone.

6. DEMONSTRATION AND CONCLUSION

On-site attendees will be offered to try PDFMeat with files of their own choice. Selecting the PDFs in the file manager the user can trigger and inspect the metadata acquisition process (see the intermediate results such as text extraction, query generation, query results, entity matching, BibTEX retrieval) as well as affiliation and categorization annotation. Users may edit the metadata in the file property tab, rename and move the files manually or automatically in accordance to their metadata without losing the linked metadata. PDF files may be dropped into the watched folder for instant automatic results, e. g. having the downloaded paper at once available for citing in one's reference manager.

With PDFMeat we demonstrate a metadata acquisition tool for publications represented as PDF files. The tool incurs low to no effort for acquiring and managing metadata to one's local scholarly PDFs, and offers instant gratification by making the metadata available not only within the file explorer, but also as bibliographic database, and further tools, e. g. desktop search. PDFMeat does not intend to replace available reference management software but aims at providing a useful addition in acquiring metadata to files once downloaded and residing on the desktop that would be hardly accessible due to missing annotations. The overall approach of retrieving and semantically integrating metadata on the desktop is generic and applicable to other domains, e. g. to present title, year, rating, country, etc. to movie files.

7. REFERENCES

[1] D. Aumüller. Retrieving metadata for your local scholarly papers. In *German DB Conf. (BTW)*, 2009.

[2] D. Aumüller and E. Rahm. Web-based Affiliation Matching. In *ICIQ*, 2009.

[3] I. Cruz and H. Xiao. A layered framework supporting personal information integration and application design for the semantic desktop. *VLDB Journal*, 17(6), 2008.

[4] J.-P. Dittrich and M. A. V. Salles. iDM: a unified and versatile data model for personal dataspace management. VLDB, 2006.

[5] C. Duda, D. Kossmann, and C. Zhou. Predicate-based indexing for desktop search. *VLDB Journal*, 2010.

[6] M. Falagas et al. Comparison of pubmed, scopus, web of science, and google scholar: strengths and weaknesses. *The FASEB Journal*, 22(2):338, 2008.

[7] A. Halevy, A. Rajaraman, and J. Ordille. Data integration: the teenage years. VLDB, 2006.

[8] H. Han et al. Automatic document metadata extraction using support vector machines. In *JCDL*, 2003.

[9] J. Hu et al. Enhancing text clustering by leveraging wikipedia semantics. In *SIGIR*. ACM, 2008.

[10] X. Hu et al. Exploiting wikipedia as external knowledge for document clustering. In *SIGKDD*. ACM, 2009.

[11] T. Huynh and K. Hoang. GATE framework based meta data extraction from scientific papers. In *ICEMT*, 2010.

[12] J. Meier and T. Conkling. Google scholar's coverage of the engineering literature: an empirical study. *The Journal of Academic Librarianship*, 2008.

[13] E. Minack et al. Leveraging personal metadata for Desktop search: The Beagle++ system. *Web Semantics: Science, Services and Agents on the WWW*, 8(1), 2010.

[14] G. Navarro. A guided tour to approximate string matching. *CSUR*, 33(1):31–88, 2001.