EuDML

The EUROPEAN DIGITAL MATHEMATICS LIBRARY

ICTPSP

ICT POLICY SUPPORT PROGRAMME

part of the Competitiveness and Innovation Framework Programme CIP

DELIVERABLE

Project Acronym: EuDML

Grant Agreement number: 250503

Project Title: The European Digital Mathematics Library

D8.1: Association Analyzer Implementation: State of the Art

Revision: 1 as of 27th November 2010

Authors:

Mark Lee	University of Birmingham, UB
Petr Sojka	Masaryk University, MU
Volker Sorge	University of Birmingham, UB
Josef Baker	University of Birmingham, UB
Wojtek Hury	University of Warsaw, ICM
Łukasz Bolikowski	University of Warsaw, ICM

Project co-funded by the European Comission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	✓
C	Confidential, only for members of the consortium and the Commission Services	

Revision History

Revision	Date	Author	Organisation	Description
0.1	November 5th	Mark Lee	UB	Draft before internal review.
0.2	November 11th	Petr Sojka	MU	gensim added, typos and bib corrected.
0.3	November 15th	Petr Sojka	MU	crossref section added, typos pointed by TB corrected.
0.4	November 18th	Mark Lee	UB	Typos and minor changes
0.5	November 23th	Petr Sojka	MU	Finalizing JR's improvements, cyrillic
1	November 27th	Petr Sojka	MU	Two typos fixed, new eudml.cls applied

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Contents

1 Introduction **2**

2 Citation Indexing **3**
2.1 Citation Parsing 4
2.1.1 ICM's Citation Parser 5
2.1.2 Affiliation Parsing 5
2.2 Reference Matching 6
2.2.1 CrossRef 6
2.2.2 ICM's Reference Matcher 6
2.2.3 Identity Discovery 7
2.3 Conflating Parsing and Reference Matching 7
2.3.1 The Matching Strategy 8
2.3.2 Evaluation 9
2.4 Summary 9

3 Document Classification and Clustering **9**
3.1 Pre-existing classification schemes 10
3.2 MU's Supervised Document Classification 11
3.3 Unsupervised Document Clustering 13
3.3.1 MU's Experiments with Latent Semantic Analysis 14
3.3.2 MU's Implementation of Streamed Clustering 15
3.3.3 Future Directions in Research 15
3.4 Summary of Document Classification and Clustering 16

4 Towards a Prototype **16**

5 Summary **17**

1 Introduction

Mathematical works always build on previous results, and are thus part of what can naturally be considered a network of literature. This was the case even in the past, long before the advent of electronic communications. But the digital infrastructures that have been built over the last decades now potentially make this network of knowledge widely and easily available.

Therefore, our goal in this Work Package is to put a given work into a rich context of related documents:

- works that are cited in the reference section of the given work;
- works that cite the given work;
- possible other parts of the given work;
- corrigenda to the given work;
- existing reviews of this work;
- works which deal with the same topic but which do not cite or are not cited by the given work.

Relevant works might be external to the current collection, and so we need to develop methods which are able to link to documents both within and outside of the collection. In many cases, relevant documents can be identified by the associated metadata of the current work. However, this metadata may be incomplete, incorrect or absent. In particular, in this work package we wish to develop techniques for *discovering* documents relevant to the current work which have not previously been identified as being so. Therefore, the main objective of the work package is to provide tools to identify referential and semantic links:

- between items in the content repositories;
- between items and external resources.

This report of the State of the Art will focus on two key technologies: Citation Indexing and Document Clustering. Citation Indexing concerns the automatic parsing and linking of citations to create a network of documents within the collection. This technology is well established in digital libraries and searchable archives such as CiteSeerX [12], Google Scholar [39, 7], general projects as *DRIVER*, and mathematical specific digital libraries such as NUMDAM, DML-CZ or referative databases *Zentralblatt MATH* (Zbl) and Mathematical Reviews (MR). Document Classification and Clustering are also established technologies within Information Retrieval (IR) but have not to date been widely used within digital libraries. In particular, there is very little previous work applying classification and clustering techniques to mathematical documents. However, initial research appears promising and we believe that the addition of these technologies will allow facilities beyond the current state of the art.

The rest of this report is structured as follows: Section 2 reviews previous work on citation indexing, first introducing the general problem and then focussing on recent work by ICM, MU and UJF/CMD on this problem. Section 3 will cover document classification and clustering techniques and report on some recent work by MU on applying both supervised and unsupervised techniques to the classification and clustering of mathematical documents within DML-CZ. Section 4 will outline steps towards building

a prototype suite of tools for association analysis and Section 5 will provide a summary of the state of the art and some conclusions.

2 Citation Indexing

Like all academic papers, mathematical documents extensively reference other works. There are three general instances: either the work cites earlier work on which it builds upon as an extension, elaboration or correction, the work cites earlier work since it applies a theorem or technique introduced by the earlier work or the work cites earlier work to locate the current work within a general subject area. We can use all three types of citation to discover associations between works by citation indexing and cross-referencing.

The use of citation indexing and the use of interlinking via citations to retrieve similar documents are not new ideas: Kessler in 1963 [29] introduced the general approach and Small in 1973 [49] used bibliographic coupling to measure document similarity. The use of citation indexing has also been used in document retrieval [3, 9] and has been used extensively to measure research impact [21].

However, the automatic extraction of citations from documents and their cross-referencing to documents within a digital library is a more recent innovation—first developed as part of the CiteSeer public search engine and digital library for scientific and academic papers [32]. Council et al. [13] further develop citation indexing in CiteSeerX to use Bayesian networks to combine citation information from several documents to construct a "canonical reference" for each given work in the CiteSeerX collection. More recently, Google Scholar has developed a ranking algorithm for document relevance which relies heavily on co-citation indexing [7].

Good bibliographic databases exist in the field of mathematics, and can potentially be used to identify a given mathematical work, enabling cross-referencing of items. However, to fully use such databases, we need to develop software which is able to cope with two problems. First, any given work may be cited in a number of different ways. Secondly, any citation may contain errors (which may be either human errors or may have been introduced by the digitization process, e.g. optical character recognition errors).

In the context of this work, the matching problem can be defined as follows:

Given a database of bibliographic items, and a bibliographic reference string, find database entries that describe the same work as the reference string.

At first sight, the above matching problem does not appear difficult since it seems simply to be a matter of aligning the various fields of the citation against the same fields found in either document metadata or another citation. However, reference strings usually have the following characteristics:

- they are not tagged (separated into fields);
- they are noisy, containing typing errors, optical recognition errors.;
- they are inaccurate, containing wrong volume numbers, wrong journal titles, wrong page numbers, wrong publication year;
- they are incomplete, lacking author's names, titles, or other bibliographic data;
- titles may be translated from a different original language;
- a reference string might be correct but expressed in a different format than expected;

- titles of journals, conference proceedings etc. may be abbreviated in unpredictable ways.

It is thus clear that when deciding whether a given database item matches a given citation string, we must use a combination of string metrics and heuristics. We can distinguish several steps needed for citation matching:

- citation extraction;
- citation parsing;
- reference matching.

Citation extraction is the problem of actually extracting citation information from a document. For digitally born documents with searchable free text, this is usually easy to achieve. However, it is non-trivial for digitised documents. In WP8 we assume that citation extraction is an issue for content providers and do not go into further detail in this report. Citation parsing involves identifying the various parts of the citation e.g. the author, title, date, place of publication. Finally, reference matching involves matching the citation either to documents in the collection or other citations in other documents. The following subsections go into greater detail on these last two steps.

2.1 Citation Parsing

Bibliographic references are often supplied as part of the metadata of a document. However, more often than not, metadata sources provide bibliographic references in the form of raw, untagged text. In order to build a citation network, one has to parse the raw texts of references into fragments such as: author, title, journal, volume, year, etc. For example, the following input text:

```
. Vinovsk, Czech. J. Phys. B 36, 625 (1986)
```

should be parsed as follows (here in the RIS exchange format):

```
TY  - JOUR
AU  - Vinovsk,.
JO  - Czech. J. Phys. B
VL  - 36
SP  - 625
PY  - 1986
```

Only then can a reference be matched against other documents in a collection. However, reference parsing is not a trivial task, for several reasons:

- There are dozens of established reference formats, and a great variety of formats "invented" by authors.
- Reference texts are "noisy" due to: misspellings, OCR errors and imperfect transformations from one format to another (the latter especially affects diacritics and formulae).
- Interpretation of a reference is sensitive to punctuation: a single comma changed to a colon may alter the meaning of the whole citation.

A number of approaches to reference parsing have been developed. Template matching using regular expressions is arguably one of the earliest techniques. It is actively used to this day, for example in the ParaCite project [41]. The BibPro tool [11] uses the

sequence alignment algorithm BLAST [2] to find the best-fitting reference template. Several machine learning approaches exist, based on: Hidden Markov Models [27], Conditional Random Fields [52], and other probabilistic models. Last but not least, there are efforts to combine the above techniques [25].

2.1.1 ICM's Citation Parser

A citation parser engine based on regular expressions has been implemented at ICM. One million references from the Elsevier collection were sampled and reference templates were inferenced. Validation revealed accuracy in the range of 90–95%, depending on reference set.

A major drawback of the regular-expressions-based solution is its limited maintainability. Each new template or feature has to be added with great care, a haphazard change may render a number of templates useless. Regular expressions are "rigid", as opposed to "flexible" probabilistic models, which generally (if reluctantly) accept unobserved transitions.

For that reason, the ICM team is currently developing another parsing engine, based on Conditional Random Fields. The GRMM library [51] has been evaluated and the initial results are promising. Still, a lot of work is required to understand the inner workings of the library and fine-tune the CRF model.

2.1.2 Affiliation Parsing

As in the case of references, metadata sources usually provide affiliations in the form of raw text. Both user interface and content analysis can benefit from more detailed information, namely separation into fragments such as: division, institution, street name, city, state, country, postal code, etc.

For example, when the following affiliation:

```
Interdisciplinary Centre for Mathematical
and Computational Modelling,
University of Warsaw, ul. Pawinskiego 5A,
02-106 Warsaw, Poland
```

is presented, the "University of Warsaw" text might be a link to a page on the University, while the "Interdisciplinare Centre [...]" text might be a link to a page on ICM. Furthermore, content analysis could infer that the following affiliation:

```
Interdyscyplinarne Centrum Modelowania
Matematycznego i Komputerowego
Uniwersytet Warszawski, ul. Pawinskiego 5A,
02-106 Warszawa, Polska
```

is equivalent to the former. However, in order to be able to perform such an inference, the two affiliations should be parsed and compared field-by-field.

Yu et al. [58] have created an affiliation parser in order to map the collaboration network of human genome epidemiology researchers. The ICM team will leverage their valuable knowledge when designing its own affiliation parser.

2.2 Reference Matching

Once a reference is parsed and documents in a collection indexed, the reference can be matched against the documents. This is a relatively straightforward step, hence only a small amount of published research focuses solely on the issue of reference matching [1]. Nevertheless, due to incomplete and noisy data at hand, the task is not entirely trivial. Several "tricks of the trade" improve the quality of matching. For example, fields such as title or journal name require "fuzzy" matching and therefore need to be indexed using custom indexing algorithms[1]. Furthermore, fields such as year are more reliable than volume and issue, which may sometimes be confused—either by human editors or reference parsers.

2.2.1 CrossRef

As for the best matching results it is essential to have *global* database of bibliographic items, a demand by publishers for global storage of bibliographic items has been fullfilled by commercial service named CrossRef [14].

CrossRef is an independent membership association, not-for-profit network founded and directed by publishers. CrossRef's mandate is to make reference linking throughout online scholarly literature efficient and reliable. CrossRef is also the official DOI link registration agency for scholarly and professional publications. DOI is a unique alphanumeric string assigned to a digital object—in this case, an electronic journal article or a book chapter. In the CrossRef system, each DOI is associated with a set of basic metadata and a URL pointer to the full text, so that it uniquely identifies the content item and provides a persistent link to its location on the internet.

CrossRef's citation-linking network today covers almost 45 millions of articles and other content items from thousands of scholarly and professional publishers.

CrossRef offers to its affiliates set of services, e.g. cited-by linking, crosscheck, and enhanced CrossRef Metadata Services (CMS), for a fee.[2]

2.2.2 ICM's Reference Matcher

ICM has implemented a Lucene-based reference matcher for the YADDA platform. The engine managed to match approximatively 6% of references within the Elsevier collection (i.e., Elsevier-to-Elsevier references). No further refinements to the algorithm are planned at this time.

MU uses the citation matcher described in [31] in DML-CZ's workflow. It checks the reference strings against MR and Zbl, after some string normalization. It is fine-tuned with some heuristics for OCRed data—if e.g. a single Cyrillic character appears within a word in Latin then several variants of the reference string are tried to increase the matching coverage.

---

1. In the nomenclature of Lucene [35], a popular indexing engine, this phase would read: "certain fields require custom `Analyzers`".
2. "CMS was developed to standardize how published content is crawled, indexed, and linked to on the Web." (Ed Pentz, Executive Director of CrossRef).

2.2.3 Identity Discovery

Typically, a document's metadata contains contributors to the document (authors, editors, etc.). It is either a single text field per contributor, or a set of text fields such as: forenames, surname, title. Also, although less often, a document's metadata lists its contributors' affiliations as well.

Given a collection of documents, we would like to know which *contributor names* refer to the same *person*. Similarly, we would like to know which *affiliations* refer to the same *institution*.

Several approaches to the problem were proposed, especially on the person identification side. Some researchers employed so-called author co-citation analysis, or ACA [38, 34], others proposed clustering techniques [26, 36], probabilistic approaches [54] and usage of finite-state graphs [19].

The above research results will be used to implement identity discovery for EuDML [53]. Ultimately, a semantic network will emerge with three main object types: document, person and institution. The network will store relations between these objects, expressing: references (document-to-document), contributions (document-to-person) and affiliations (person-to-institution).

Such a network will allow for better navigation in the user interface of EuDML. Citations are so few in mathematics that citation based bibliometrics are easily biased [4].

2.3 Conflating Parsing and Reference Matching

The approaches described above explicitly separate the steps of citation parsing and reference matching. However, it is possible to combine the two steps so that individual fields within the citation string are *discovered* by matching metadata already within the collection.

For example, UJF/CMD developed a citation extraction and indexing system [22] for use within the Zentralblatt MATH collection. The UJF/CMD citation extraction and matching algorithm does not attempt to perform citation parsing or citation field tagging prior to trying to find matching citations. Instead, citations are viewed simply as strings of characters with no attempt to parse a structure to the citation string. Goutorbe [22] argues that making no attempt to parse the citation string avoids several problems:

1. Even if citation parsing is successful, individual fields often remain coded in different ways and cannot be compared using exact comparison methods.
2. Errors in parsing can result in the complete failure of the matching process.
3. The parsing process is both costly and error prone.

For these reasons, the UJF/CMD approach relies on just a shallow analysis of the input string and relies on a number of string similarity evaluation methods and ad hoc heuristics which work reasonably well in practice in the context of mathematical databases.

For example, numerical data in the citation string appears to be highly effective in distinguishing documents. This is supported by the following figures generated by scanning part of the *Zentralblatt MATH* database and collecting author names, volume number, publication year and paging information for each entry.

Total number of journal articles: 413721

v = volume number
y = publication year
fp = first page number
lp = last page number
t = first (significant) title word
a = first author name (without initials)
Total number of different v|fp-lp strings: 376038 (90.89)
Total number of different t|fp-lp strings: 380623 (92.26)
Total number of different a|y|fp strings: 402594 (97.31)
Total number of different a|fp-lp strings: 406844 (98.33)
Total number of different a|v|fp strings: 410735 (99.28)
Total number of different a|y|fp-lp strings: 411959 (99.57)
Total number of different a|v|y|fp strings: 412350 (99.67)
Total number of different a|v|fp-lp strings: 412710 (99.76)
Total number of different a|v|y|fp-lp strings: 412889 (99.80)
(Adapted from [22])

What the above figures show is that numerical information in the citation string plus key words from either the author's names or title almost always distinguish any citation to a unique document reference. This suggests that an effective approach to matching citations is to extract all numerical information plus a small number of words and use these to identify possible matches.

2.3.1 The Matching Strategy

The algorithm works as follows:

1. Generate possible candidates
   A boolean query is constructed consisting of all numerical data found in the citation string plus the first few words of the string.
2. Rank candidates by evaluating their similarity to the reference to match
   For each candidate compute the *cosine similarity* with the given reference string, using n-grams vectors. This allows in particular for small mistakes and variations in spelling. n-grams are a set of n consecutive characters from the input string. After some experimentation, it appears that a value of $n = 3$ is a reasonable choice.
3. Output the first n candidates

At this point, the algorithm has a set of *structured* database items, and other metrics/heuristics may be used to further rank this set.

- approximate substring matching (similar to *agrep*) is used to check author names;
- the similarity of numbers is computed using the Dice coefficient;
- paging information is matched. Page numbers that differ only by one are considered equal;
- titles can potentially be discovered and compared using approximate substring matching.

In practice, there is often no need to perform all of the above before the set of potential matches consists of just one document.

2.3.2 Evaluation

Goutorbe [22] reports several evaluation results using different datasets.

1. Journal articles from the NUMDAM project: metadata is of good quality (reference strings are accurate). Depending on the journal coverage in *Zentralblatt MATH* (Zbl), 96% accuracy is achieved.
2. Bibliographic references cited by these same articles: metadata may be noisy because of optical recognition errors and inaccurate or incomplete because of authors' mistakes. They include every possible kind of reference (journal articles, books, theses, reports,...). The average rate of matches is 75% of the total number of bibliographic items, and may grow up to 85%, depending on the journal.
3. Bibliographic references from the *Journal of Differential Geometry* (project Euclid). The matching rate is 89%, including dubious or irrelevant matches (no checking was performed)

One interesting result is that because citation matching is led by numerical data, it is possible for citations to be matched even if they are expressed in different languages. However, the full extent of this has not been fully evaluated. These results demonstrate that good results for citation matching can be obtained by very shallow analysis of the citation string. The major impact in performance is due to the quality of the data and, in particular, with the digitisation process introducing errors into the citation data and therefore impacting on accuracy.

2.4 Summary

In summary, citation indexing and matching is an established method for developing associations between documents which is already implemented in other large scale digital libraries and document focussed search engines such as CiteSeerX and Google Scholar. Moreoever, evaluation of accuracy in this task is generally high enough for it to be a useful feature expected by users.

There is a wide variety of competing techniques and approaches used for this task. However, it is difficult to directly compare approaches described in the literature for the following reasons:

Different metrics Reported work uses a variety of evaluation metrics such as accuracy, precision and recall, and metrics which combine precision and recall such as weighted F measures. These metrics cannot often be directly compared.

Different Collections The constitution of the collection has an effect on accuracy. For example, a collection composed of digitally born documents will result in far better results than one composed of digitised text. In addition, different journals have different degrees of strictness in how closely authors must follow their style guidelines. Greater variety in how a citation can be legally expressed directly affects citation parsing and matching accuracy.

3 Document Classification and Clustering

In addition to grouping documents in terms of what other documents they cite, it is possible to group documents by their actual content, i.e. the raw text within the document—

words, formulae and expressions. Broadly we can distinguish two types of task: *supervised document classification* and *unsupervised document clustering*. Document clustering may be called *streamed* if document is assigned to cluster by just being seen once.

In supervised document classification, we aim to group documents according to pre-existing categories. These categories can be hierarchically structured as an ontology which represents the range of topics within the collection.

In unsupervised document clustering, no pre-existing classification scheme is used and instead documents are clustered based on their "semantic similarity". Unsupervised document clustering allows us to discover possible categories implicit in the collection.

Approaches to both tasks usually assume that any document is just a *bag of words* and therefore ignore any syntactic or structural information. However individual words may be pre-processed using either a stemmer or lemmatizer to reveal the stem or base form of the word. Most approaches also adopt a *vector space model* [46] where documents (and search queries) are represented as vectors of terms with, depending on the approach, terms representing individual words, characters or concepts. Depending on the task, these terms can be weighted. For instance, a common approach is to weight terms by their frequency within a particular document divided by their frequency across the entire collection (*tfidf*) (see [28]).

Central to both tasks is the assumption that term overlap is indicative of semantic similarity. Within a vector space model, the degree of overlap between two documents (or a document and a search query) can be measured by calculating the cosine of the angle between the two resulting vectors. When two documents are identical, their cosine will equal one, when they are orthogonal (i.e. share no common terms), their cosine will equal zero.

Precision and recall are the most commonly used metrics used in the evaluation of supervised document classification systems where precision is defined as the number of correctly classified documents returned divided by the total number of documents returned and recall is calculated as the number of correctly classified documents returned divided by the total number of documents in the collection. Both metrics can be combined into a F-measure with either precision or recall weighted depending on the particular application. However, the evaluation of unsupervised document clustering is more difficult since there is no gold standard collection to be measured against. Often the only appropriate metric is to see whether or not the clusters generated actually aid users in their search for relevant documents in practice.

In this section, we will first provide an overview of classification schemes already in use within mathematical document collections and then review work on supervised mathematical document classification and then unsupervised mathematical document clustering.

3.1 Pre-existing classification schemes

Mathematicians are used to classifying their papers. One of the first mathematical classification schemes appeared in the subject index for *Pure Mathematics* of 19 volumes of the *Catalogue of Scientific Papers 1800–1900* [45]. This attempt was continued but not completed by the *International Catalogue of Scientific Literature (1901–1914)*. About two

hundred classes were used. Headings in the *Jahrbuch* [40] may be considered as another classification scheme.

The Library of Congress classification system has 939 subheadings under the heading of QA: Mathematics. Another schemes used in many libraries around the world are the Dewey Decimal system and the Referativnyi Zhurnal System used in the Soviet Union. To add to this wide variety of schemes, we may mention systems used by NSF Mathematics Programs, by various encyclopaedia projects such as Wikipedia, or by the arXiv Preprint project [5]. However, the most commonly used classification system today is the Mathematics Subject Classification (MSC) scheme (`http://www.ams.org/msc/`), developed and supported jointly by reviewing databases *Zentralblatt MATH* (Zbl) and *Mathematical Reviews* (MR).

It is clear that no fixed classification scheme can survive longer time period, since new areas of mathematics appear every year. Mathematicians entered the new millennium with the MSC version 2000, migrating from MSC of 1991. MSC 2010 has already been prepared and published at `msc2010.org` recently, and is used by publishers. The primary and secondary keys of MSC 2000, requested today by most mathematical journals are used for indexing and categorizing a vast amount of new papers (100,000 new math items per year).

Editors of mathematical journals usually require the authors themselves to include the MSC codes in manuscripts submitted for publication. However, most retrodigitized papers published before the adoption of MSC are not classified yet. Some projects, e.g. Jahrbuch, use MSC 2000 even for the retroclassification of papers. Human classification needs significant resources of qualified mathematicians and reviewers. A similar situation is in the other retrodigitization projects such as NUMDAM [10] (`http://www.numdam.org`), or DML-CZ [50, 6] (`http://www.dml.cz`): classifying digitized papers with MSC 2000 manually is expensive.

As there are already many papers properly classified (by authors and reviewers) in recent publications, methods of machine learning may be used to train an automated classifier based on the full texts obtained by optical character recognition (OCR) from author- and/or reviewer-classified papers. This is a clear example of a supervised classification task.

3.2 MU's Supervised Document Classification

MU have performed a series of experiments in developing a supervised classification system for automatically attributing MSC classification codes to papers based on their textual content.

There are many modelling techniques available for this task. To design a classifier, we have to choose measurable features to be as discriminative as possible.

It is widely known that the design of the learning architecture is very important, as is preprocessing, learning methods and their parameters [42].

For the purpose of building an automated MSC classification system, MU chose the standard Vector Space Model (VSM) together with various statistical Machine Learning (ML) methods. In order to convert the text in the natural language to vectors of features, several pre-processing steps were required —for a more thorough explanation, see e.g. [42].

A detailed description of all ML methods and Information Retrieval (IR) notions is beyond the scope of this report; the reader is referred to the overviews [48, 56, 37] for exact definitions and notation used.

The setup of the experiments is such that MU ran a vast array of training attempts in multidimensional learning space of tokenizers, feature selectors, term weighting types, classifiers and learning methods' parameters:

tokenization and lemmatization: the first part of the preprocessing relates to how the text is split into tokens (words)—alphabetic, lowercase, Krovetz stemmer [30], lemmatization, bi-gram tokenization (collocations chosen by MI-score);

feature selectors: how to choose the tokens that discriminate best—χ^2, mutual information (MI-score) [57, 18, 17];

feature amount: how many features are needed to classify best—500, 2,000 or 20,000 features [18];

term weighting: how the features will be weighted (*tfidf* variants [47] or [37, Fig. 6.15]) and smart weights normalizations (*atc* (augmented term frequency), *bnn* and *nnn*) [33];

classifiers: Naïve Bayes (NB), k-Nearest Neighbours (kNN), Support Vector Machines (SVM), decision trees, Artificial Neural Nets (ANN), K-star algorithm, Hyperpipes;

threshold estimators: how to choose the threshold category of the classifier: *fixed* or *s-cut* strategy for threshold setting [55];

evaluation and confidence estimation: how results are measured and how the confidence is estimated in them—Receiver Operating Characteristic (ROC), Normalized Cross Entropy (NCE) [20].

To give an example, evaluating one particular combination might mean that the corpus is tokenized using an alphabetic tokenizer, the best 2,000 tokens (words aka features aka terms) chosen using χ^2 and weighted using an *atc* scheme.

In all experimental set ups, one part of the corpus was used for training the binary classifiers and the rest used to evaluate whether the predicted MSC equalled the expected MSC.

Each classifier was a binary classifier and responsible for recognising one category (MSC class). Given a full text on input, each classifier returned whether the document belonged to that category or not. Therefore, each article could be predicted to belong to any number of categories, including none or all.

Out of the seven classifiers listed above, only the first three were used in the final experiments. The other four were discarded on the ground of poor performance in preliminary experiments not reported here. That is to say that only Naïve Bayes (NB), k-Nearest Neighbours (kNN), Support Vector Machines (SVM) based classifiers were found to be accurate enough for further investigation.

In order to evaluate the quality of each learned classifier, an average of ten cross-validation runs were calculated and standard measures such as micro/macro F_1[3], accuracy, precision, recall, correlation coefficient, break-even point and their standard

3. The F_1 measure is the harmonic mean of precision and recall.

deviations [37, 42] calculated. All these results are then compared to see which 'points' in the parameter space performed best.

A full analysis of the results is beyond the scope of this report. However, a significant outcome was that a F_1 classification score of 80% is easily achievable using either Naïve Bayes (NB), k-Nearest Neighbours (kNN), or Support Vector Machines (SVM) based classifiers.

In addition, the best performing method of Support Vector Machines trained on a large number of features was, with some fine tuning, able to achieve an F_1 score of 89%. This result is highly encouraging and it shows that classifiers can be trained to a high degree of accuracy on this task.

One limitation however is that many of the methods used in these experiments are computationally expensive and therefore only appropriate for batch processing the collection at regular intervals rather than ad hoc document classification. One of the few exceptions seem to be new GenSim framework [44].

3.3 Unsupervised Document Clustering

Recall that one of the purposes of the automated MSC classification detailed above is to enable a similarity search. The idea being that, given MSC categories, the user may browse articles with similar MSCs and thus (hopefully) with similarly relevant content.

However, in the absence of MSC codes or similar metadata, it is possible to compute similarity measures directly based on the articles' content, with no reference to human-entered or human-revised metadata.

Since many texts come from OCR-based sources containing errors at the character level, fine grained linguistic analysis tools are ineffective but brute-force Information Retrieval approaches can be used.

Řehůřek and Sojka [43] describe experiments in computing paper similarities using *tfidf* [47] and Latent Semantic Analysis (LSA) [15] methods. Again, both use a Vector Space Model, first converting articles to vectors and then using the cosine of the angle between the two document vectors to assess their similarity [37]. The difference between them is that while *tfidf* works directly over tokens, LSA first extracts concepts, then projects the vectors into this conceptual space where it computes similarity.

For LSA they chose the 200 top latent dimensions (concepts) to represent the vectors, in accordance with standard practise [15].

As discussed above, evaluating the effectiveness of unsupervised similarity schemes is not easy. This is due to the fact that, as far as we know, there exists no corpus with an explicitly evaluated similarity between each pair of papers. One possible solution would be to construct such a corpus for testing purposes but this option is clearly too costly and so instead MU evaluate their work by assuming that MSC equality implies content similarity. Accordingly, they evaluated how closely the computed LSA similarity between two papers corresponds to the similarity implied by them sharing the same MSC.

To avoid data sparseness, they only took note of the top MSC categories (first two letters of the MSC codes). Both *tfidf* and Latent Semantic Analysis produce clusters where articles with the same MSC group are clustered in the same postion while there is a low

overlap between groups with different MSC groups. There are also however exceptions which is to be expected from noisy real-world data.

3.3.1 MU's Experiments with Latent Semantic Analysis

One advantage of Latent Semantic Analysis [15] is that "concepts" can be easily analysed to determine what is being discriminated within clusters.

The corpus they used consisted of papers from the *Czechoslovak Mathematical Journal* (CMJ) which contains papers in several different languages and it is clear that the first thing the method distinguishes is language, as the first terms of top concepts are:

1. 0.3 "the" +0.19 "and" +0.19 "is" +0.18 "that" +0.15 "of" +0.14 "we" +0.14 "for" +0.11 "ε" +0.11 "let" +0.11 "then" +...
2. −0.41 "ist" −0.40 "die" −0.28 "und" −0.26 "der" −0.23 "wir" −0.21 "für" −0.17 "eine" −0.17 "von" −0.14 "mit" −0.13 "dann" +...
3. −0.31 "de" −0.30 "est" −0.29 "que" −0.27 "la" −0.26 "les" −0.2 "une" −0.2 "pour" −0.20 "et" −0.18 "dans" −0.18 "nous" +...
4. −0.36 "что" −0.29 "для" −0.23 "пусть" −0.19 "из" −0.19 "если" −0.16 "так" −0.16 "то" −0.14 "на" −0.14 "тогда" −0.131169 "мы" +...
5. −0.33 "semigroup" −0.25 "ideal" −0.19 "group" −0.18 "lattice" +0.18 "solution" +0.16 "equation" −0.16 "ordered" −0.15 "ideals" −0.15 "semigroups" +...
6. 0.46 "graph" +0.40 "vertices" +0.36 "vertex" +0.23 "graphs" +0.2 "edge" +0.19 "edges" −0.18 "ε" −0.15 "semigroup" −0.13 "ideal" +...
7. 0.81 "ε" −0.25 "semigroup" −0.16 "ideal" +0.12 "lattice" −0.11 "semigroups" +0.10 "i" −0.1 "ideals" +0.09 "ordered" +0.09 "ř" −0.08 "idempotent" +...
8. 0.29 "semigroup" −0.22 "space" +0.2 "ε" +0.19 "solution" +0.19 "ideal" +0.18 "equation" +0.16 "oscillatory" −0.15 "spaces" −0.16 "compact" +...

(Adapted from [43])

The first concepts clearly capture the language of the paper (EN, DE, FR, RU), and only then topical term-sets start to be grabbed. This is not surprising but it means that the classifiers then have to be trained either for every language, or the document features have to be chosen language-independently by mapping words to some common topic ontology. To the best of our knowledge, nothing like EuroWordNet for mathematical subject classification terms or mathematics exists.

Given the amount of training data, we face the sparsity problem for languages such as Czech, Italian, German and even French presented in the CMJ digital library. However, it is possible that the combined size of the EuDML collection will make this problem less of an issue.

MU also applied LSA on the monolingual corpora of *Archivum Mathematicum*, where mathematics formulae were used during tokenization (subcorpus created from original TEX files). In this case, we see that even in the first concepts, there was significant proportion of mathematical terms with high weights in concepts created by LSA:

1. −0.32 "t" −0.24 "ds" −0.17 "u" −0.17 "_" −0.17 "x" −0.15 "solution" −0.12 "equation" −0.11 "q" −0.11 "x_" −0.11 "oscillatory" +...
2. 0.28 "ds" +0.28 "t" −0.22 "bundle" −0.16 "natural" +0.15 "oscillatory" −0.15 "vector" +0.13 "solution" −0.13 "connection" −0.13 "manifold" +0.11 "t_0" +...

3. −0.22 "bundle" +0.19 "ring" −0.17 "natural" −0.16 "oscillatory" +0.15 "fuzzy" −0.15 "ds" +0.12 "ideal" −0.11 "t" −0.11 "r_0" −0.11 "nonoscillatory" +...
4. 0.29 "ring" −0.23 "x_" −0.21 "_" +0.21 "oscillatory" +0.18 "ideal" +0.17 "r" +0.16 "prime" +0.15 "rings" +0.13 "nonoscillatory" −0.12 "x_n" +...

This supports the idea that mathematical formulae have to be taken into account—having robust math OCR and finding its good discriminative feature representation we may get much better similarity and classification results in the future. In addition, [23] report early results in word-based context windows to disambiguate mathematical terms and symbols. We believe this work could potentially improve the above results in unsupervised clustering of mathematical documents.

3.3.2 MU's Implementation of Streamed Clustering

A relatively new way of clustering is *streamed clustering*, where number of documents covered is virtually unlimited [24]. A new NLP tool GenSim has been developed recently. Its aim is to make unsupervised "semantic analysis" (in the mundane statistical sense, no psychology/linguistics) of texts. Features:

- can process corpora larger than RAM (streamed algorithms);
- simple to trivial interfaces: you can get going quickly, no Java-esque madness.

GenSim contains unique incremental implementations of popular algorithms like:

Latent Semantic Analysis takes 2.5 hours on a 2 billion corpus of 3.2M documents (the entire English Wikipedia), on a single laptop. LSA has not been used much in practical NLP due to its massive computational demands; it is now no longer an issue.

Latent Dirichlet Allocation a more recent but slower technique, can be run in distributed mode over a cluster of computers.

3.3.3 Future Directions in Research

This work is still at an early stage but there are several directions which potentially can improve performance and in particular, recall. For example, the pre-processing of vectors representing the documents can be improved using NLP techniques (characteristic words, bi-words, etc.) or use higher order models (deep networks). Mainstream machine learning research has concentrated on using "convex", shallow methods (SVM, shallow neural networks with back-propagation training) so far. State-of-the-art fine tuned methods allow very high accuracy even on large scale classification problems. However, the training of these methods is exceptionally high and the models are big. Using the ensembles of classifiers makes the situation even less satisfactory (size even bigger), and the final models need to be regularized.

In future, new algorithms for a hierarchical text classification [16] might be tried and training large models with non-convex optimization [8] may give classifications that does not exhibit overfitting.

Further studies will encompass a fine-grained classification trained on bigger collections (using MSC tagged mathematical papers from `ArXiv.org`), growing NUMDAM and DML-CZ libraries etc.), and a rigorous measure confidence evaluation [20].

For final large scale applications scaling issues, and fine-tuning the best performance by choosing the best set of preprocessing parameters and machine learning methods remains to be done. We watch Apache Lucene Mahout project's code when scalability of machine learning will arise as a serious issue. GenSim's scalability might be sufficient for EuDML scale.

3.4 Summary of Document Classification and Clustering

MU's results convincingly demonstrate the feasibility of a machine learning approach to the classification and clustering of mathematical papers. In addition, the approaches can be easily tweaked to favour a different trade-off between higher recall and/or precision. Results in the form of guessed MSC codes and automatically generated lists of similar documents are already directly used in the DML-CZ project thus providing a preexisting proof of concept.

4 Towards a Prototype

In the report above we have reviewed several technologies which have proven useful in association analysis: citation indexing and document classification and clustering. As reviewed, citation indexing is an established feature within digital library-based information retrieval and there are various algorithms capable of doing this task with good accuracy. However, there has been little analysis comparing different approaches on the same collection and using comparable metrics.

There has been less application of document classification and clustering technologies within digital libraries. However, automatic classification and clustering based on free text allows documents to be retrieved for which there is no pre-existing citation link or association. Therefore these technologies provide the potential to *discover* relevant research which may not be previously associated with the original document or search query.

The next step in Work Package 8 is to produce a prototype which supports both types of association analysis. We have decided to produce a suite of APIs for partner's existing technology from previous projects including:

- ICM's work on Citation Indexing and Matching;
- UJF/CMD's work on Citation Indexing and Matching;
- MU's work on Unsupervised Document Clustering.

This prototype will be released in month 15 of the project. Once the prototype is completed, in months 16–18, we will perform an evaluation comparing the technologies on the same document collection (a representative sample of the EuDML collection). Finally in months 19–33, based on this evaluation, we will make enhancements to our range of APIs to move beyond the current state of the art in association analysis. Finally we will release the toolset as a standard service for the system architecture in month 33.

5 Summary

This report has provided an overview of the state of the art in association analysis in digital libraries focused on two main technologies: citation indexing and document classification and clustering. Each technology offers a distinctive facility for building associations between documents within a collection and we already have working proof of concept applications and expertise from previous projects. Our next step is to first build a working prototype toolset which can be properly evaluated and then further enhanced beyond the state of the art.

References

[1] A. Accomazzi, G. Eichhorn, M. J. Kurtz, C. S. Grant, and S. S. Murray. The ads bibliographic reference resolver. In D. M. Mehringer, R. L. Plante, and D. A. Roberts, editors, *Astronomical Data Analysis Software and Systems VIII*, volume 172 of *ASP Conference Series*, 1999.

[2] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[3] R. Amsler. Application of Citation-based Automatic Classification. Technical report, The University of Texas at Austin, Linguistics Research Center, 1972.

[4] Douglas N. Arnold and Kristine K. Fowler. Nefarious Numbers, 2010. `http://arXiv.org/abs/1010.0278`.

[5] arXiv. `http://arXiv.org/`.

[6] Miroslav Bartošek, Martin Lhoták, Jiří Rákosník, Petr Sojka, and Martin Šárfy. DML-CZ: The Objectives and the First Steps. In Jonathan Borwein, Eugénio M. Rocha, and José Francisco Rodrigues, editors, *CMDE 2006: Communicating Mathematics in the Digital Era*, pages 69–79. A. K. Peters, MA, USA, 2008.

[7] J. Beel and B. Gipp. Google Scholar's Ranking Algorithm: An Introductory Overview. In Birger Larsen and Jacqueline Leta, editors, *Proceedings of the 12th International Conference on Scientometrics and Informetrics (ISSI'09)*, volume 1, pages 230–241, Rio de Janeiro, Brazil, July 2009. International Society for Scientometrics and Informetrics.

[8] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 153–160. MIT Press, Cambridge, MA, 2007.

[9] J. Bichtler and E. A. Eaton. The Combined Use of Bibliographic Coupling and Cocitation for Document Retrieval. *Journal of the American Society for Information Science*, 31(4):278–282, 1980.

[10] Thierry Bouche. Towards a Digital Mathematics Library? In Jonathan Borwein, Eugénio M. Rocha, and José Francisco Rodrigues, editors, *CMDE 2006: Communicating Mathematics in the Digital Era*, pages 43–68. A.K. Peters, MA, USA, 2008.

[11] C. C. Chen, K.-H. Yang, H. Y. Kao, and J.-M. Ho. BibPro: A Citation Parser Based on Sequence Alignment Techniques. In *Proceedings of the International Conference on Advanced Information Networking and Applications, AINA*, pages 1175–1180, 2008.

[12] I. G. Councill, H. Li, Z. Zhuang, S. Debnath, L. Bolelli, W. C. Lee, A. Sivasubramaniam, and C. L. Giles. Learning Metadata from the Evidence in an On-line Citation Matching Scheme. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, Chapel Hill, NC, USA, June 11–15 2006.

[13] I. G. Councill, H. Li, Z. Zhuang, S. Debnath, L. Bolelli, W. C. Lee, A. Sivasubramaniam, and C. L. Giles. Learning Metadata from the Evidence in an On-line Citation Matching Scheme. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, Chapel Hill, NC, USA, 2006.

[14] CrossRef. `http://crossref.org/`.

[15] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[16] Andrea Esuli, Tiziano Fagni, and Fabrizio Sebastiani. Boosting multi-label hierarchical text categorization. *Information Retrieval*, 11(3), 2008.

[17] George Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[18] Luigi Galavotti, Fabrizio Sebastiani, and Maria Simi. Experiments on the use of feature selection and negative evidence in automated text categorization. In José Luis Borbinha and Thomas Baker, editors, *Research and Advanced Technology for Digital Libraries, 4th European Conference, ECDL 2000, Lisbon, Portugal, September 18–20, 2000, Proceedings*, volume 1923 of *Lecture Notes in Computer Science*, pages 59–68. Springer, 2000.

[19] C. Galvez and F. Moya-Anegón. Approximate Personal Name-matching Through Finite-state Graphs. *Journal of the American Society for Information Science and Technology*, 58(13):1960–1976, 2007.

[20] Simona Gandrabur, George Foster, and Guy Lapalme. Confidence Estimation for NLP Applications. *ACM Transactions on Speech and Language Processing*, 3(3):1–29, October 2006.

[21] E. Garfield. Citation Analysis As a Tool in Journal Evaluation. *Science*, 178:471–479, 1972.

[22] Claude Goutorbe. Document Interlinking in a Digital Math Library. In Petr Sojka, editor, *Towards a Digital Mathematics Library*, pages 85–94, Grand Bend, Ontario, Canada, 2008. Masaryk University, Brno.

[23] Mihai Grigore, Magdalena Wolska, and Michael Kohlhase. Towards context-based disambiguation of mathematical expressions. *Math-for-Industry Lecture Note Series*, 22:262–271, December 2009.

[24] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. Clustering data streams: Theory and practice. *IEEE Transactions on Knowledge and Data Engineering*, 15:515–528, 2003. `http://doi.ieeecomputersociety.org/10.1109/TKDE.2003.1198387`.

[25] D. Gupta, B. Morris, T. Catapano, and G. Sautter. A New Approach Towards Bibliographic Reference Identification, Parsing and Inline Citation Matching. In S. Ranka, S. Aluru, R. Buyya, Y. Chung, S. Dua, A. Grama, S. K. S. Gupta, R. Kumar, and V. V. Phoha, editors, *Contemporary Computing*. Springer, 2009.

[26] H. Han, H. Zha, and C. Lee Giles. Name Disambiguation in Author Citations Using a K-way Spectral Clustering Method. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 334–343, New York, NY, USA, 2005. ACM.

[27] E. Hetzner. A Simple Method for Citation Metadata Extraction Using Hidden Markov Models. In *JCDL '08: Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 280–284, New York, NY, USA, 2008. ACM.

[28] K. Sparck Jones. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[29] M. M. Kessler. Bibliographic Coupling Between Scientific Papers. *American Documentation*, 24:123–131, 1963.

[30] Robert Krovetz. Viewing morphology as an inference process. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Linguistic Analysis, pages 191–202, 1993.

[31] Lukáš Lalinský. Citation Crawling, 2009. Bachelor Thesis Masaryk University, Brno, Faculty of Informatics, `https://is.muni.cz/th/158017/fi_b/?lang=en`.

[32] S. Lawrence, C. L. Giles, and K. D. Bollacker. Autonomous Citation Matching. In *Proceedings of the Third Annual Conference on Autonomous Agents*, pages 392–393, Seattle, Washington, United States, 1999.

[33] Joon Ho Lee. Analyses of multiple evidence combination. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Combination Techniques, pages 267–276, 1997.

[34] X. Lin, H. D. White, and J. Buzydlowski. Real-time Author Co-citation Mapping for Online Searching. *Inf. Process. Manage*, 39(5):689–706, 2003.

[35] Apache Lucene. `http://lucene.apache.org/`.

[36] G. S. Mann and D. Yarowsky. Unsupervised Personal Name Disambiguation. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL*, pages 33–40, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[37] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[38] K. W. McCain. Mapping Authors in Intellectual Space: A Technical Overview. *Journal of the American Society for Information Science*, 41(6):433–443, 1990.

[39] A. Noruzi. Google Scholar: The New Generation of Citation Indexes. *LIBRI*, 55(4):170–180, 2005.

[40] Carl Ohrtmann and Felix Müller, editors. *Jahrbuch über die Fortschritte der Mathematik (1868–1942)*, volume 1–68. Druck und Verlag von Georg Reimer, Berlin, 1871–1942. electronic version available by project ERAM `http://www.emis.de/projects/JFM/`.

[41] ParaCite. `http://paracite.eprints.org/`.

[42] Jan Pomikálek and Radim Řehůřek. The Influence of Preprocessing Parameters on Text Categorization. *International Journal of Applied Science, Engineering and Technology*, 1(4):430–434, 2007.

[43] Radim Řehůřek and Petr Sojka. Automated Classification and Categorization of Mathematical Knowledge. In Serge Autexier, John Campbell, Julio Rubio, Volker Sorge, Masakazu Suzuki, and Freek Wiedijk, editors, *Intelligent Computer Mathematics—Proceedings of 7th International Conference on Mathematical Knowledge Management MKM 2008*, volume 5144 of *Lecture Notes in Computer Science LNCS/LNAI*, pages 543–557, Berlin, Heidelberg, July 2008. Springer-Verlag.

[44] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, 2010. software available at `http://nlp.fi.muni.cz/projekty/gensim`.

[45] Royal Society of London. *Catalogue of Scientific Papers 1800–1900*. London, 1908. Volumes 1–19 and Subject Index in 4 vols. published 1867–1925; free electronic version available by project Gallica `http://gallica.bnf.fr/`.

[46] G. M. Salton, editor. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, 1971.

[47] Gerard Salton and Chris Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523, 1988.

[48] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002. `http://citeseer.ist.psu.edu/sebastiani02machine.html`.

[49] H. G. Small. Co-citation in the Scientific Literature: A New Measure of the Relationship Between Two Documents. *Journal of the American Society for Information Science*, 24:265–269, 1973.

[50] Petr Sojka. From Scanned Image to Knowledge Sharing. In Klaus Tochtermann and Hermann Maurer, editors, *Proceedings of I-KNOW'05: Fifth International Conference on Knowledge Management*, pages 664–672, Graz, Austria, June 2005. Know-Center in coop. with Graz Uni, Joanneum Research and Springer Pub. Co.

[51] C. Sutton. GRaphical Models in Mallet, 2006. `http://mallet.cs.umass.edu/grmm/`.

[52] C. Sutton and A. McCallum. *Introduction to Statistical Relational Learning*. MIT Press, 2006.

[53] Wojtek Sylwestrzak, José Borbinha, Thierry Bouche, Aleksander Nowiński, and Petr Sojka. EuDML—Towards the European Digital Mathematics Library. In Petr Sojka, editor, *Proceedings of DML 2010*, pages 11–24, Paris, France, July 2010. Masaryk University. `http://dml.cz/dmlcz/702569`.

[54] V. I. Torvik, M. Weeber, D. R. Swanson, and N. R. Smalheiser. A Probabilistic Similarity Metric for Medline Records: A Model for Author Name Disambiguation: Research Articles. *Journal of the American Society for Information Science and Technology*, 56(2):140–158, 2005.

[55] Yiming Yang. A Study on Thresholding Strategies for Text Categorization. In W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors, *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pages 137–145, New York, September 9–13 2001. ACM Press.

[56] Yiming Yang and Thorsten Joachims. Text categorization. *Scholarpedia*, 2008. `http://www.scholarpedia.org/article/Text_categorization`.

[57] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In Douglas H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

[58] W. Yu, A. Yesupriya, A. Wulf, J. Qu, M. Gwinn, and M. Khoury. An Automatic Method to Generate Domain-specific Investigator Networks Using PubMed Abstracts. *BMC Medical Informatics and Decision Making*, 7, 2007.