Comparing hierarchical mathematical document clustering against the Mathematics Subject Classification tree

Tomasz Kuśmierczyk, Michał Łukasik, Łukasz Bolikowski, and Hung Son Nguyen

Interdisciplinary Centre for Mathematical and Computational Modelling,
University of Warsaw
{t.kusmierczyk, m.lukasik, l.bolikowski}@icm.edu.pl

Faculty of Mathematics, Informatics and Mechanics,
University of Warsaw
son@mimuw.edu.pl

Abstract. Mathematical publications are often labelled with Mathematical Subject Classification codes. These codes are grouped in a tree-like hierarchy created by experts. In this paper we posit that this hierarchy is highly correlated with content of publications. Following this assumption we try to reconstruct the MSC tree basing on our publications corpora. Results are compared to the original hierarchy and conclusions are drawn.

Keywords: documents clustering, documents representation, clustering similarity, measures of agreement, Mathematical Subject Classification

1 Introduction

1.1 Research Problem

There are several established classification schemes for scholarly literature, for example: Mathematics Subject Classification (MSC), Physics and Astronomy Classification Scheme (PACS), Journal of Economic Literature (JEL) Classification System, ACM Computing Classification System, or a much broader Dewey Decimal Classification. All these systems are created by human experts (rather than generated by an algorithm), all are hierarchical, and many undergo periodical updates which result in minor-to-moderate differences between revisions.

In this research, we are primarily interested in building algorithms that would, as far as it is possible, recreate a classification system for a given domain. In a wider sense, we are interested in studying the process that governs the development of such classification systems, in particular, understanding which features of the classified documents have the largest impact on the final hierarchy. While most of our theoretical work is applicable to any hierarchical classification scheme, our experiments are conducted on the 2000 revision of Mathematics Subject Classification.

This paper is structured as follows. In the remainder of this section we briefly summarize the MSC 2000 system, similarity as it is understood in computer science, and the data set used in our experiments. In Section 2 we investigate approaches to measuring structural similarity of objects, we outline state-of-the-art, present our original ideas and analyze the results of our experiments. In Section 3 we focus on similarity of documents. Section 4 describes methodology of evaluation of similarity matrices, and Section 5 presents various experiments related to reconstructing MSC 2000 hierarchy. The last sections contain summary and conclusions.

1.2 MSC Codes

MSC codes [1] are a hierarchical system for multi-tagging of mathematical documents. It was created by experts from Mathematical Reviews and Zentralblatt MATH [2]. There are two slightly different version of codes: MSC2000 and MSC2010.

In MSC hierarchy there are three levels of codes:

- leaves (denoted as L) - named with 5 characters (2 digits + letter or special character '-' + 2 digits) - for example: $05C05$ means 'Trees'
- middle level (denoted as M) - named with 3 characters - for example: $05C$ means 'Graph theory'
- higher level (denoted as H) - named with 2 characters - for example: 05 means 'Combinatorics'

Special character '-' is used for special purpose documents (instructional exposition, proceedings etc.).

In MSC every single document can have one or more codes assigned. First code is the most important and is called 'primary'. Subsequent are called 'secondary'. What is more, not only leaf-codes can be assigned but also codes from upper levels.

1.3 Similarity

Similarity is an intuitive and subjective concept. Many different approaches to this idea exist in psychology.

One of the earliest and the one that has most in common with computer science is mental distance approach [18]. In this approach objects are represented as points within the space and similarity is represented by some distance function.

The second most influential approach is featural approach [19] (in formalism closely related to Jaccard Coefficient). In this method objects are represented as sets of features. Similarity is then measured by comparing two sets of features. It increases with the number of common features and decreases with the number of differences. This approach deals with several psychological aspects of

[1] http://www.ams.org/mathscinet/msc/msc2010.html
[2] http://www.zentralblatt-math.org/zbmath/

similarity but has several disadvantages e.g. an assumption that commonalities and differences are independent.

In computer science, properties of similarity measures are formulated closely to the featural approach [11]:

F1) similarity is a value in [0, 1]
F2) similarity reaches its maximum when two objects are identical
F3) the more differences two objects have, the less similar they are
F4) the more commonalities two objects share, the more similar they are
F5) $1.0 - similarity$ has metric properties apart from triangle inequality

During our work we dealt with similarity of objects of different kinds e.g. documents and elements localized in different structures. Details are described in further sections.

1.4 Data Description

We used 13,609 documents tagged with MSC2000 codes. Documents originated from following digital libraries:

- ZentralBlatt-MATH [3]
- CEDRAM [4]
- NUMDAM [5]

Every document was represented by an abstract, keywords and a title that were merged into single list of words (apart from bigram calculation where bigrams are calculated before the merge). Length statistics (in words; after filtering - see section 5.1) of these fields are shown in table 1. Some symptoms of preprocessing problems can be found. Especially it is rather uncommon to have abstract containing over 35 thousands of words. The most probable explanation is that during extraction process some parts of document were glued to the abstract. Similar situations can happen in real, fully automatic systems. Due to this fact we decided to leave data after preprocessing without further modifications.

Table 1. Length statistics of documents.

field	min	max	avg	std
abstract	11	35522	509.23	493.66
keywords	7	374	77.17	39.13
title	10	318	64.59	34.74
merged	47	35846	651.00	505.39

[3] http://www.zentralblatt-math.org/zbmath/
[4] http://www.cedram.org/
[5] http://www.numdam.org/?lang=en

In our experiments we decided to consider only typical leaf codes composed of 5 letters and leave out special codes (with special character '-' instead of one letter in the name of a code). It should not influence our results much, as codes of different types account for less than 5% of all codes.

We also filtered out the codes that occurred less often than 10 times as a primary code. In the end, we had 345 non-special leaf-codes (level L). At level M of MSC tree we had 144 codes. Each of these codes had, on average 2.40, children (std=2.33). 76 codes had only single child and the maximum number of children was 13. On the H-level of MSC tree we had 37 codes out of which 15 had only a single child. Average number of children at this level was 3.89 (std=3.38) and the maximum number of children was 12. Statistics presented above show how diverse MSC subtree is and how complicated is the problem of its reconstruction.

The filtering left only $10,201$ documents in corpus: $7,575$ with primary code assigned and $7,032$ with at least one secondary code. Every leaf-code occurred as a primary code on average 21.96 times (with standard deviation 19.48; max=185) and as a secondary code 29.54 times (with standard deviation 24.01; max=204). What is worth noting, every code appeared at least once as a secondary code.

Before filtering, single document had averagely 2.98 codes assigned. After filtration, an average of only 1.74 codes was left (standard deviation = 0.88). The maximum noticed number of codes per document was 10. Every document had averagely 0.74 primary codes (std=0.44; before filtering=1.00) and 1.00 secondary codes (std=0.89; before filtering=1.98).

2 Similarity of Structures

To compare two structures (e.g. original MSC hierarchy and the reconstructed tree) we decided to adapt pairwise comparisons (in psychology called paired comparisons). In classical clusterings comparisons this group of methods is described as counting of pairs of elements (other two are: information-theoretical mutual information and summation of set overlaps). It was extended for purpose of comparing fuzzy clusterings. Later, we adapted it for hierarchical structures and introduced simple formalism.

Having two elements (e.g. leaves of MSC tree or publications) l_i and l_j, the bonding (introduced in [3]) between them is described by the function:

$$B_{ij} = b_T(l_i, l_j) \in [0, 1] \tag{1}$$

Index T denotes the structure in which leaves are positioned. It means that bonding in different structures T and T' can be different. Indexes i and j always denote the row and the column in a matrix.

Bonding measures how close two elements are to each other. Complementary value:

$$C_{ij} = 1 - B_{ij} \in [0, 1] \tag{2}$$

measures how much two elements (indexed by i and j) are separated according to structure T.

Having bondings (and complementary values) of two elements l_i and l_j in structures T and T' we need to decide how much these values ‘agree‘. It is performed by another function τ:

$$\Theta(\beta,\beta')_{ij} = \tau(\beta_{ij},\beta'_{ij}) \in [0,1] \tag{3}$$

where: β, β' can be either B, B' or C, C' matrices for T and T'. For example $\Theta(B,B')_{ij}$ measures how much two structures ‘agree‘ on how much elements l_i and l_j should be bonded.

Using four matrices:

$$\Theta(B,B'), \Theta(B,C'), \Theta(C,B'), \Theta(C,C')$$

we calculate four coefficients:

$$\begin{aligned} a &= h(\ \Theta(B,B')\ ) \\ b &= h(\ \Theta(B,C')\ ) \\ c &= h(\ \Theta(C,B')\ ) \\ d &= h(\ \Theta(C,C')\ ) \end{aligned}$$

where h is a function that aggregates values from matrices. It can be interpreted as summarizing over all pairs and therefore can be implemented as:

$$h(X) = \sum_{i,j>i} X_{ij}$$

Derived coefficients measure how much two structures ‘agree‘ (a, d) and ‘disagree‘ (b, c). Using them we can adapt similarity indexes designed for typical clusterings comparisons. The most common is the *Rand index* $\in [0,1]$ [15]:

$$\begin{aligned} RI &= \tfrac{a+d}{a+b+c+d} \\ RI &= \tfrac{a+d}{|LL|} \\ RI &= \tfrac{|LL|-(b+c)}{|LL|} \end{aligned} \tag{4}$$

where number of pairs of elements ($n = |L|$ stands for number of elements):

$$|LL| \equiv \frac{n(n-1)}{2}$$

The measure strongly depends on the number of clusters in the clustering structure [12] (we showed that this property is preserved for hierarchical structures). It was shown [5] that for some kinds of structures its value increases up to 1.0 with number of clusters. What is more, it was shown that RI for two random clusterings is not a constant. For these reasons *Rand Index* was modified to *Adjusted Rand Index* [3]:

$$\begin{aligned} ARI' &= \tfrac{2(ad-bc)}{c^2+b^2+2ad+(a+d)(c+b)} \in [-1,1] \\ ARI &= \tfrac{ARI'+1}{2} \in [0,1] \end{aligned} \tag{5}$$

Strong critic [4] is also given to RI for equal treatment of a and d. In some situations d dominates over a what can be a serious problem [8]. To overcome this problem *Jaccard coefficient* was introduced:

$$JI = \frac{a}{a+b+c} \in [0,1] \tag{6}$$

Presented above methods of similarity measurement depend on two functions: $b_T(l_i, l_j)$ and $\tau(\beta_{i,j}, \beta'_{i,j})$. Their selection changes the properties of measures.

In section 1.3 we assumed that $1.0 - similarity$ should have metric's properties (apart from triangle inequality). For above indexes (apart from *Rand Index*), it is not known what $b_T(l_i, l_j)$, $\tau(\beta_{i,j}, \beta'_{i,j})$ to use and how to modify them to fulfill this condition. Especially, when comparing T to itself we can obtain similarity lower than 1.0. Examples of such behaviour can be found in [7].

In [7] authors modified *Rand Index* in a way that $1.0 - RI$ is a metric (apart from the condition: $d(T, T') = 0 \implies T = T'$ which may not be fulfilled). During our research we showed that, assuming $\tau(a, b) \equiv min(a, b)$ and taking second formula for RI in equation 4, their result can be shown in our formalism. Assumption for $1 - RI$ to be (almost) metric is that $1 - b_T(l_i, l_j)$ must be a metric.

In the context of MSC-like trees our proposition for $b_T(l_i, l_j)$ is to use the common fraction ($\in [0,1]$) of two paths from root to leaves. In this case $b_T(l_i, l_j) = 0$ when two leaves have only root in common and $b_T(l_i, l_j) = 1 \; iff \; l_i = l_j$. Other metric properties were also proven.

An example of such measure is shown in the figure 1. For leaves l_1 and l_3 common fraction of paths has length q. For l_1 and l_2 it has the length $p + q$.

This measure is well-defined for trees where all leaves have the same depth. If we want to generalise to all trees, for two leaves: l_i and l_j, we have two fractions of paths f_i and f_j. Having f_i and f_j we can use $b_T(l_i, l_j) = F(f_i, f_j)$ where F can be *average*, *min*, *max* etc.

An alternative to 'path fraction' approach can be cosine-like measure:

$$b_T(l_i, l_j) \equiv 1 - \frac{2\cos^{-1}\left(\frac{\boldsymbol{v_i}}{|\boldsymbol{v_i}|} \cdot \frac{\boldsymbol{v_j}}{|\boldsymbol{v_j}|}\right)}{\pi} \in [0,1] \tag{7}$$

where $\boldsymbol{v_i}$, $\boldsymbol{v_j}$ are membership vectors assigned to l_i and l_j. Their length is equal to the number of nodes at the medium (M) level in MSC tree. The value of an m-th component in this vector describes the affiliation of the m-th element to a (M-level) node:

- 2.0 ⇔ element and node have common prefix of length 3 (for example '03A')
- 1.0 ⇔ element and node have common prefix of length 2 (for example '03')
- 0.0 ⇔ element and node have no common prefix

2.1 Experiment 1

To show behaviour of our indexes in different situations we tested them with randomly generated trees. We took $n = |L| \sim 350$ MSC leaves. These leaves describe part of MSC tree. Out of them we built 100 trees.

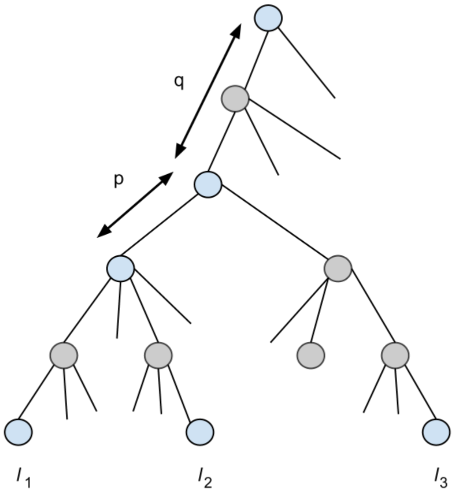


Fig. 1. Common fraction of paths from root to leaves.

In the first step, out of range $[n^{0.25}, n^{0.75}]$ we randomly selected number (m) of nodes at M level. Then, every leaf was assigned to one of the m nodes. In the next step, we randomly selected h out of range $[m^{0.25}, m^{0.75}]$ and assigned middle-level nodes to high-level nodes. In such procedure we generated single random tree. This procedure was repeated 100 times. In the end, we obtained 100 random trees.

In the picture 2 there are values of different indexes. We compared random trees and part of the original MSC tree. Prefixes of indexes stands for different configurations of b_T and τ:

- Hf - b_T - path fraction, $\tau(a, b) = min(a, b)$
- Hm - b_T - formula 7, $\tau(a, b) = min(a, b)$
- Bf - b_T - path fraction, $\tau(a, b) = ab$
- Bm - b_T - formula 7, $\tau(a, b) = ab$

Conclusions:

- ARI does not depend on number of nodes and is the most stable
- JI is also very stable but slightly decreases with number of nodes
- RI is very unstable and strongly increases with number of clusters
- τ does not influence much results (Hm-RI and Bm-RI give almost equal results; the same for Hf-RI and Bf-RI)
- b_T does not change order (plots for Hm-RI and Hf-RI are just translated and scaled; the same for Bm-RI and Bf-RI)

In the end, Bf-ARI and Hf-ARI seem to be the best measures. They take constant value (~ 0.5) for random tree no matter how much tree has nodes. The calculation of a common path is also more intuitive and faster than the calculation of a membership vectors. Having $n = |L|$ leaves overall complexity is $O(D \times n^2)$ where D stands for height of trees. To select between these two measures we designed further experiments.

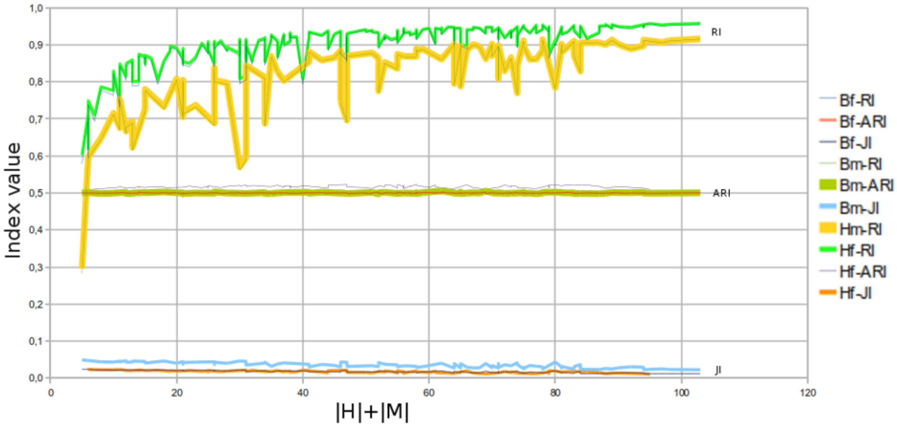


Fig. 2. Indexes' values for random trees.

2.2 Experiment 2

In the figure 3 behaviour of two indexes: Bf-ARI, Hf-ARI is shown. For each index 10 runs of an experiment is shown. There is also black, thick line that represents an average over all runs. In the single run we randomly selected nodes from H level and split them according to its child nodes. Every node from H level was replaced with c_x nodes, where c_x is a number of children of the node $x \in H$. Each new node has just single child (one of the previous children of x).

An analysis of figure 3 reveals that for comparing MSC subtree (described in the section 1.4) to itself (0 modifications) values of the indexes are different and smaller than 1.0 (0.71 for Bf-ARI and about 0.76 for Hf-ARI). This behaviour is consistent with description from the beginning of the section 2. Another conclusion is that both indexes decrease monotonically (in every single run) with number of such modifications. It is consistent with the intuition.

2.3 Experiment 3

In the figure 4 another experiment's results are shown. For each index 10 runs of an experiment is shown. In the single run we randomly selected nodes from

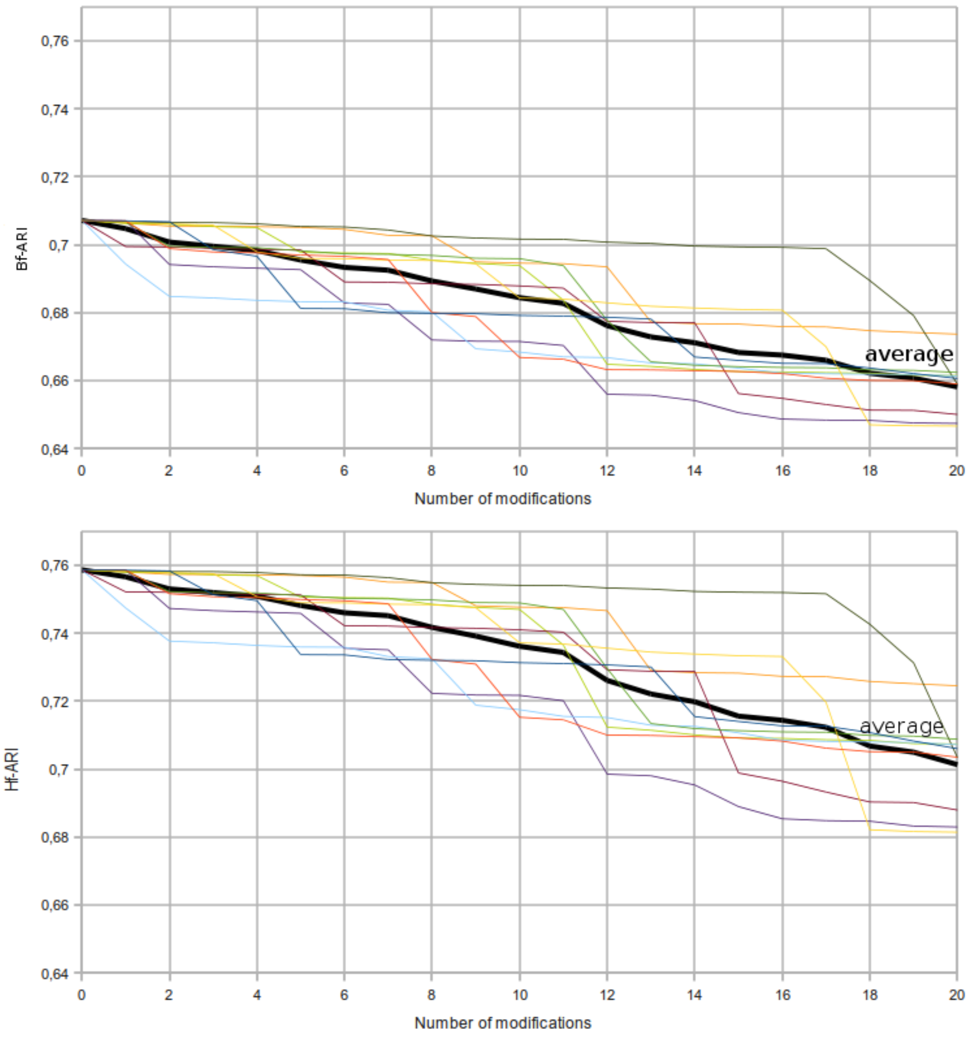


Fig. 3. Indexes' behaviour for splitting nodes at H level.

H level and merged their child nodes. For every node from *H*-level we took all its children and merged them into single child node. In the end, every node from *H*-level had only single child node.

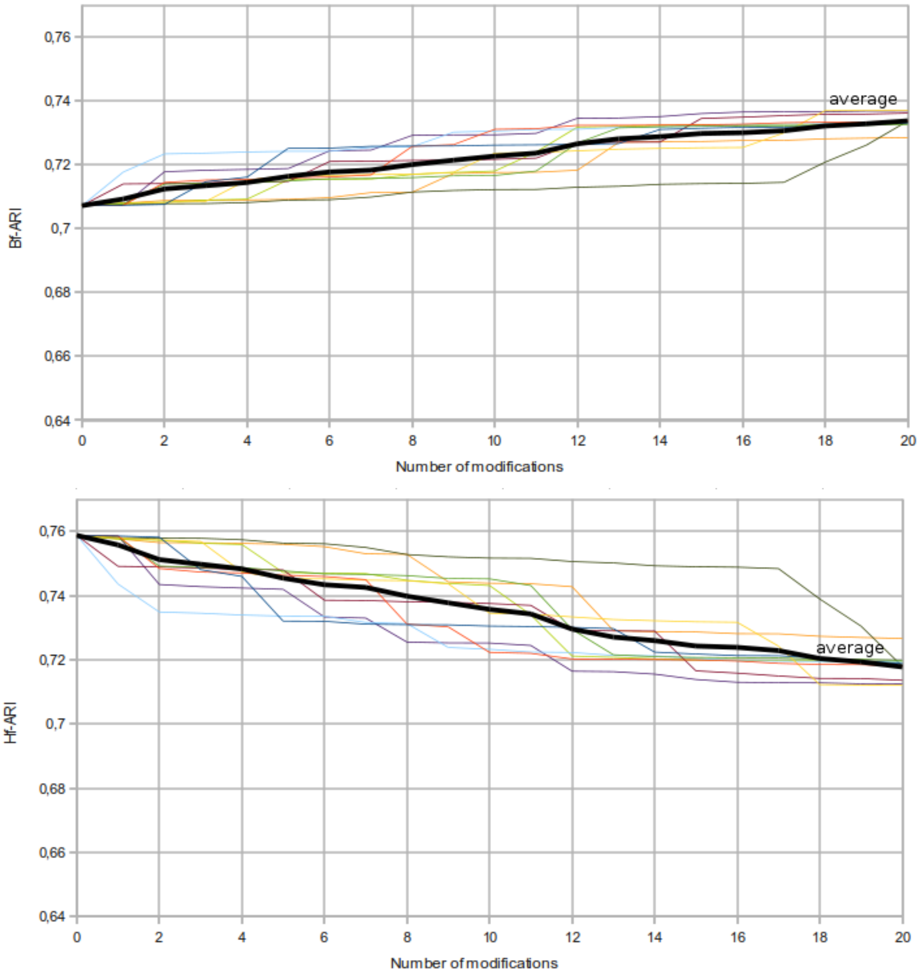


Fig. 4. Indexes' behaviour for merging nodes at M level.

An analysis of figure 4 reveals that Bf-ARI behaves counter-intuitively for such modifications what makes it useless for further use.

2.4 Measure Selection

An analysis of the behaviour of different indexes showed that the best, out of described indexes, is Hf-ARI. This index is resistant to change of number of nodes (section 2.1), to ‘pushing‘ nodes down (section 2.2) and up (section 2.3). The index have also relatively low computational complexity ($O(D \times n^2)$).

For the tree described in the section 1.4 the lower bound of Hf-ARI is equal to 0.5 (value for random trees) and the upper bound is equal to 0.759 (value for a comparison of the tree to itself). This two values state the range in which we expect to operate.

3 Similarity of Documents

Modern techniques of similarity determining split into two groups [2] [11]:

- content-based
- link-based

We focused on the first group.

For the reason that dealing with structure in documents is very demanding task, *bag-of-words model* is applied. In this model single document is represented as a set of pairs: (term, count). To compare such representations several measures was developed. One of the most successful [10] is *Ratio Model* [19] that is related to the first model described in section 1.3:

$$sim_{tv}(d_i, d_j) = \frac{|d_i \cap d_j|}{|d_i \cup d_j|} \tag{8}$$

where d_i - bag of terms (words, bigrams etc.) representing i-th document.

More advanced approaches apply *vector space model* [17] (related to the second model from section 1.3) in which documents are represented as vectors of numbers. Document d_i is thus represented by a vector $\boldsymbol{w_i}$. While vectors are often sparse, cosine-like measurements are used [12]:

$$sim_{cos}(d_i, d_j) = 1 - \frac{2\cos^{-1}(\frac{\boldsymbol{w_i}}{|\boldsymbol{w_i}|} \cdot \frac{\boldsymbol{w_j}}{|\boldsymbol{w_j}|})}{\pi} \tag{9}$$

The most popular method of constructing vector representation of documents is $TF \times IDF$ weighting scheme. $TF \times IDF$ belongs to wider group of methods called $LW \times GW$ where LW stands for local weight and GW for global weight. For every term t in document d single weight is generated:

$$\boldsymbol{w}^t = LW_{t,d} \times GW_t \tag{10}$$

Intuitively, weight should be higher if term is more important for document (e.g. occurs many times) but lower if is not very characteristic (e.g. occurs in many documents) (for further description see [12]). In $TF \times IDF$:

$$TF_{t,d} = \frac{d^t}{|d|} \tag{11}$$

$$IDF_t = log(\frac{N}{N^t}) \tag{12}$$

where:

- d^t - number of occurrences of t in document d
- $|d|$ - number of all terms in document d
- N - number of documents in corpus
- N^t - number of documents having term t

There are many doubts about $TF \times IDF$ scheme. Particularly, TF grows linearly with the number of term occurrences but psychologically the occurrence is more important than count. For example, it does not make big difference whether term t occurred 4 or 6 times. To overcome this problem another local weight can be introduced:

$$WF_{t,d} = \begin{cases} 1 + log(1 + TF_{t,d}) & TF_{t,d} > 0 \\ 0 \quad otherwise \end{cases} \tag{13}$$

Also for IDF there exist many replacements e.g. ENT [13] was reported as particularly efficient [10]:

$$ENT_t = 1 + \frac{\sum_d p_{t,d} \log(p_{t,d})}{log(N)} \tag{14}$$

$$p_{t,d} = \frac{TF_{t,d}}{GF_t} \tag{15}$$

where GF_t stands for t frequency in whole corpus.

The most advanced techniques (ESA, LSA, LDA etc.) that calculate vector representations try to discover some semantic behind documents. In this group the dominating approach is LSA (for details see [12]) in which original dimensions are linearly combined into new ones. Then, only the most 'informative' dimensions are kept. New dimensions are believed to be latent 'topics' of documents. The method is close to dimensionality reduction techniques.

LSA is calculated using Singular Value Decomposition. In our experiments we used *Gensim* implementation [16] because of its scalability and stream processing mode.

3.1 MSC Leaves Similarity

In our experiments we performed clustering (reconstruction of a MSC hierarchy) of MSC leaves. We assumed that leaves' similarity can be computed basing on the similarity of tagged documents. We considered several strategies of aggregation documents' similarity into MSC leaves similarity.

The first strategy is to consider only primary tags: MSC leaf is represented as a set of documents tagged with it at first place. Sets are disjoint. To estimate similarity between two leaves l_i and l_j aggregating function is calculated:

$$sim_{pr}(l_i, l_j) = A(\{sim(d_k, d_l) : primary(d_k) = l_i, primary(d_l) = l_j\}) \tag{16}$$

We considered $A \equiv average$ (denoted avg) and $A \equiv max$ (single linkage, denoted $single$). Complete linkage in this situation is pointless as in two sets of documents there are always two with similarity equal to 0.

The second strategy is to consider both: primary and secondary tags. It means that sets of documents overlap. In such case we assigned to every tag l_i of a document d_k weight: $e_{k,i}$. Now, similarity between l_i and l_j is calculated as a weighted average:

$$sim_{sec}(l_i, l_j) = \frac{1}{Z} \sum_{l_i \in tags(d_k), l_j \in tags(d_l)} (e_{k,i} + e_{l,j}) \cdot sim(d_k, d_l) \qquad (17)$$

$$Z = \sum_{l_i \in tags(d_k), l_j \in tags(d_l)} e_{k,i} + e_{l,j} \qquad (18)$$

Primary codes have always weight 1.0. Secondary codes can have constant weight $e_{k,i} = 0.5$ (strategy denoted as $avg - e0.5$) or $e_{k,i} = 0.75$ ($avg - e0.75$) or weight dependent on number of secondary codes assigned to particular document:

$$e_{k,i} = \frac{C}{|secondary(d_k)|} \qquad (19)$$

where $C = 0.75$ (denoted as $avg - s0.75$) or $C = 0.5$ (denoted as $avg - s0.5$).

4 Similarity Matrices Evaluation

Typical approach in clustering [9] is to perform evaluation in the end of the process - after clustering. For this strategy, in our case, having similarity matrices we should cluster using one of the clustering algorithms and then compare results.

Although, in practical applications, where the goal is to tune clustering process as much as it is possible, it is reasonable, this strategy does not give knowledge about similarity matrices themselves. To deal with this problem we decided to evaluate different features and similarity matrices just before clustering.

There are several intuitive assumptions about clustering algorithms. The most common and the most intuitive is that closer object should be joined more likely than distant. According to this rule, formal conditions can be set up [1]. Having three elements: l_i,l_j,l_k the requirement that l_i,l_j should be merged more likely than l_i,l_k and l_j,l_k can be written as inequalities:

$$sim(l_i, l_j) > sim(l_i, l_k) \qquad (20)$$

$$sim(l_i, l_j) > sim(l_j, l_k) \qquad (21)$$

In worst case number of conditions is bounded by $O(n^3)$ where $n = |L|$ - number of elements. That makes this method useful only if n is small (no bigger than ~ 1000).

Though its shortcomings, the approach has big advantage: can be interpreted as a simple thought experiment. Having n elements, we take three of them and show to an expert. He selects two the most similar or says that he does not know. If he knows, we check in our similarity matrix if the similarity conditions are held. The procedure is repeated for each tuple of size three. In the end, we have fraction of situations when we guessed correctly. Situations when the expert did not know do not count. For random similarity matrix result should be about 50%. For ideal data about 100%.

In our case, expert answers are read directly from MSC tree. The longer common prefix two elements have, the more similar should be. For example: $30A01$ and $30A02$ are the more similar than $30A01$ and $30B01$. Situations when elements have common prefix of equal length is treated as experts' answer: ‘do not know‘.

5 Experiments

To perform various kinds of experiments efficiently we designed modular framework. Sample configuration of the framework is shown in the figure 5. The data in most of components is processed in a stream. According to different input and output data type we have components of following types:

- document in/document out: documents' filtering, n-grams construction
- document in/vector out: term counts per document generation
- vector in/data model out: $LW \times GW$/LSA models construction
- vector and model in/vector out: $LW \times GW$/LSA vectors calculation
- vector in/similarity matrix out: Tversky/Cosine-like similarity calculation
- similarity matrix in/structure out: hierarchical/k-medoids clustering
- two structures in/similarity index value out: similarity indexes calculation

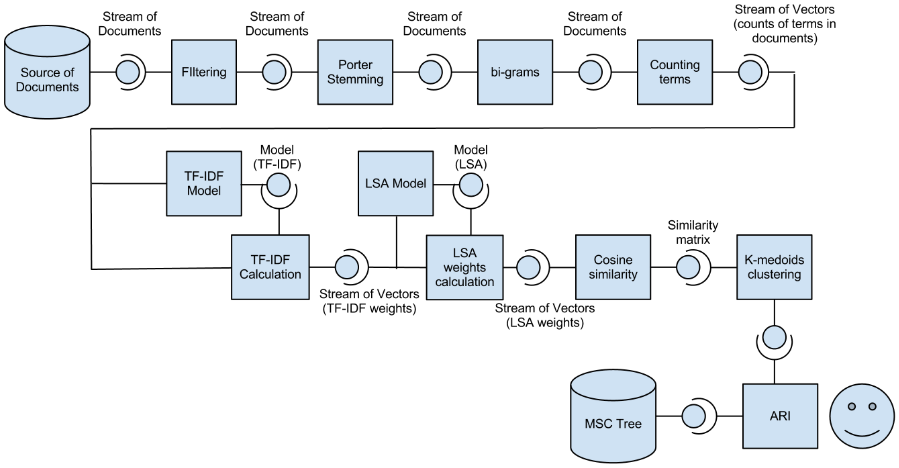


Fig. 5. Sample configuration of the experiment.

In different experiments some of the components were removed or replaced with another. For example $TF \times IDF$ can be replaced with $WF \times ENT$ or the component that calculates bigrams can be removed. The table 2 presents considered configurations of the framework.

Table 2. Possible configurations of the experiment.

Representation (section 3)	Similarity (section 3)	Aggregation (section 3.1)	Clustering (section 5.6)	Linkage (section 5.6)
Words Words-TFIDF Words-WFENT Words-TFIDF-LSA Words-WFENT-LSA Bigrams Bigrams-TFIDF Bigrams-WFENT Bigrams-TFIDF-LSA Bigrams-WFENT-LSA	Tversky (sim_{tv}) Cosine-like (sim_{cos})	average (avg) single (max) weighted average (avg-xYY)	hierarchical 3-level hierarchical 3-level k-medoids	average-linkage (avg) single-linkage (single) complete-linkage (complete)

In brackets numbers of connected sections are given.
In further text 'Words' is default and omitted.

Each column in the table represents single step in reconstruction process. First column describes possible representation in which documents could be prepared. Second - possible similarity measures. For bigrams and words we used Tversky measure (equation 8). In other cases Cosine-like measure was applied (equation 9). The third column shows possible similarity aggregation methods (for details see section 3.1). The last two columns are related to process of clustering of MSC leaves (section 5.6).

As it can be easily seen, most of the processing steps work on documents. Only in the last part (clustering/reconstruction of the tree, similarity index calculation) MSC-related data was considered.

5.1 Preprocessing

Every document in the corpus was preprocessed. Whole interpunction, brackets, numbers etc. were replaced with spaces. Then letters were changed to lower case. Also single letters and common words (stopwords from the list [6]) were removed. In the next step, words were stemmed using Porter algorithm [14] (believed to be the best stemming algorithm [6]).

[6] http://www.textfixer.com/resources/common-english-words.txt

5.2 Similarity Matrices Evaluation

Using the method described in section 4 we evaluated several variants of similarity matrices for MSC leaves.

We considered documents representations listed in the table 2. To calculate similarity between two documents we used techniques from section 3 (either Tversky for Words/Bigrams or Cosine-like in other cases). In the end, we used one of the strategies from section 3.1 to obtain MSC-leaves' similarity matrix.

An evaluation of different similarity matrices is shown in figures 6 and 7. The results of different variants of weighting strategies in aggregation (section 3.1) were consistent so we decided to show an average value of them (denoted as Avg-xYY)

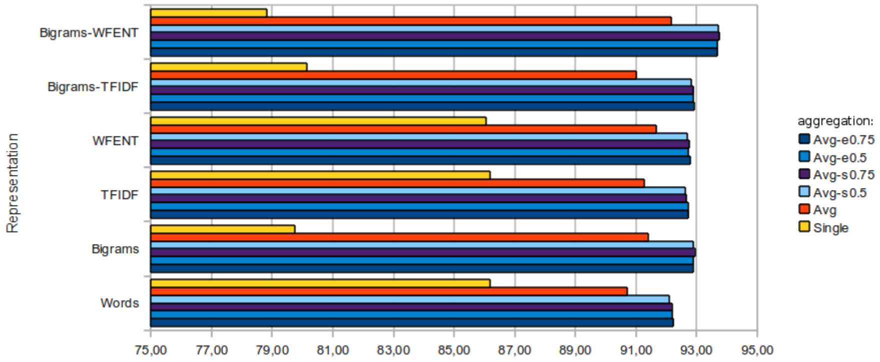


Fig. 6. Evaluation results for different MSC similarity matrices.

An analysis of the figures 6 and 7 showed that the best results (above 94% fulfilled conditions) were obtained for documents represented as bigrams with WF-ENT weighting scheme and LSA applied. Overall conclusions for the above figures are following:

- bigrams overcome representation as a bag of single words
- TF-IDF performs worse than WF-ENT
- single-linkage strategy used for aggregation of similarity is the weakest one: averaging gives better results
- considering both primary and secondary codes improves results
- optimal number of dimensions in LSA depends on weighting scheme and it is hard to find any rule for its selection

5.3 Extracted Topics

In tables 3, 4, 5, 6 LSA topics for different configurations (weighting schemes) are shown. Top ten topics is listed for every configuration. For every topic terms

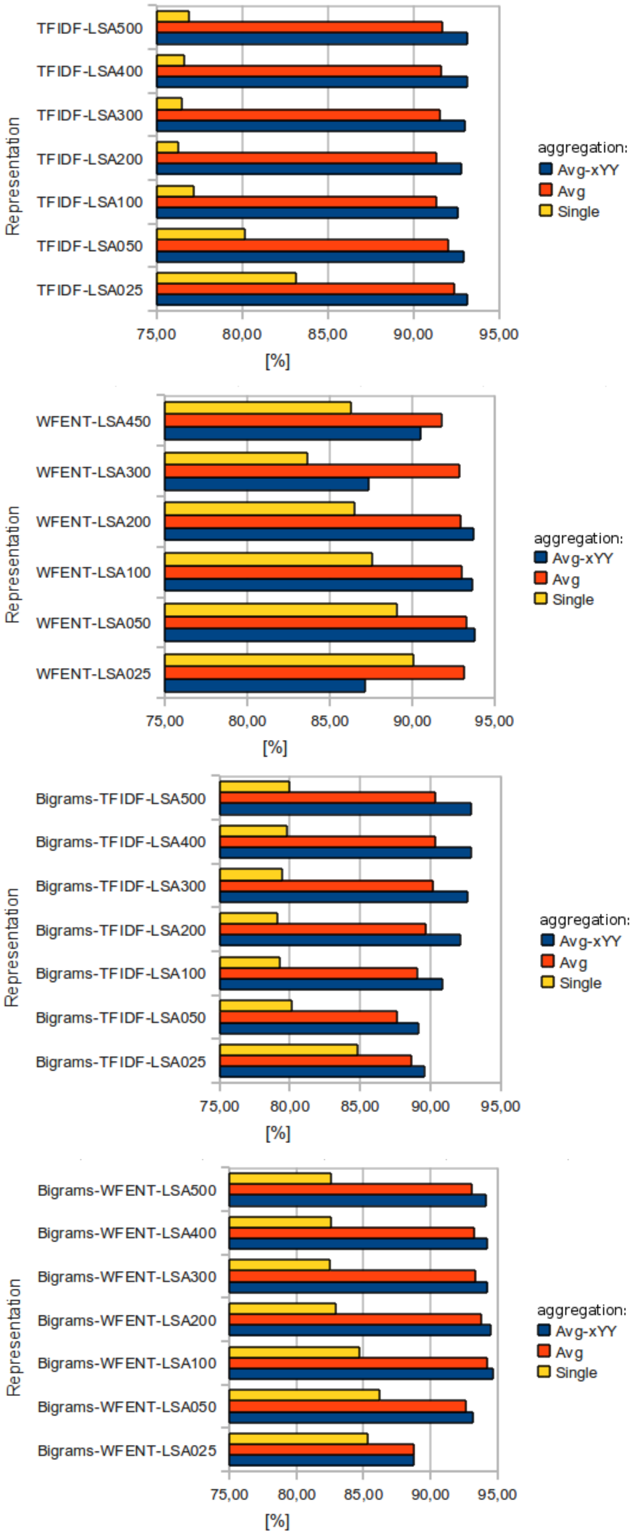


Fig. 7. Evaluation results for different MSC similarity matrices (LSA representations).

with highest weights are shown. Terms are sorted with weights so the leftmost are the most important.

The results show several problems in our data. First of them are TEX-tags (e.g. sb, sp, a̧l) that occurred in some documents. Another is that, some documents contain parts in other than English languages (e.g. French, German).

It can be easily seen that $WF \times ENT$ scheme managed much better than $TF \times IDF$. In the table 3 we see that TEX-tags were filtered out from the most important terms in the most influential topics. Also non-english words occur only in 4-th topic. Even better situation is shown in the table 4. Both: non-english words and TEX-tags were almost totally filtered out from top ten topics. It is important to remember that these ‘bad terms‘ were not completely removed. They were just pushed down to less important topics or given lower weights.

Table 3. Top ten LSA topics for $WF \times ENT$ weighting scheme.

1	-0.105*problem + -0.103*equat + -0.094*function + -0.093*gener + ...
2	0.987*obituari + 0.080*public + 0.076*jan + 0.064*reiterman + ...
3	0.197*equat + 0.196*solut + 0.151*problem + 0.141*nonlinear + ...
4	0.185*une + 0.183*de + 0.181*un + 0.173*la + 0.168*sur + ...
5	0.190*algorithm + 0.157*numer + 0.145*method + 0.144*comput + ...
6	0.293*process + 0.226*random + 0.192*brownian + 0.167*stochast + ...
7	-0.200*manifold + 0.137*number + 0.131*integ + 0.124*bound + ...
8	0.162*word + -0.156*field + 0.155*algorithm + 0.154*languag + ...
9	-0.200*schrödinger + -0.190*word + -0.170*languag + 0.139*error + ...
10	0.164*solut + -0.153*asymptot + 0.151*equat + -0.146*schrödinger + ...

Table 4. Top ten LSA topics for $WF \times ENT$ weighting scheme on bigrams.

1	0.670*list-public + 0.179*public-item + 0.013*comput-scientist + ...
2	0.759*public-item + -0.517*obituari + 0.349*list-public + ...
3	-0.984*order-determin + -0.048*physic-requir + -0.028*expect-uniqu + ...
4	0.482*differenti-ideal + 0.461*modul-field + 0.137*ring-modul + ...
5	-0.123*differenti-ideal + -0.112*modul-field + 0.110*math-zbl + ...
6	0.966*show-impli + 0.072*defin-notion + 0.038*theorem-survey + ...
7	0.653*preview-zbl + 0.653*see-preview + 0.255*extens-doubl + ...
8	0.531*applic-constitut + 0.456*inequ-base + 0.445*base-tetrahedra + ...
9	0.323*applic-constitut + 0.279*inequ-base + 0.259*base-tetrahedra + ...
10	-0.829*ring-proof + -0.471*present-formal + -0.065*eingeführt-und + ...

Table 5. Top ten LSA topics for $TF \times IDF$ weighting scheme.

1	-0.471*sb + -0.264*sp + -0.148*omega + -0.123*ąl + -0.114*group + ...
2	0.705*sb + 0.281*sp + -0.125*equat + -0.105*solut + ...
3	0.508*de + 0.216*la + 0.199*le + 0.182*group + 0.179*est + ...
4	0.328*group + -0.290*de + -0.177*solut + -0.174*omega + ...
5	0.595*omega + -0.271*sb + 0.160*ąl + 0.150*text + 0.146*partial + ...
6	0.355*process + 0.257*brownian + 0.234*random + 0.187*measur + ...
7	-0.650*ąl + 0.377*group + -0.206*categori + 0.133*subgroup + ...
8	0.487*ąl + -0.273*manifold + -0.230*surfac + 0.175*categori + ...
9	-0.387*group + 0.242*curv + 0.168*number + 0.159*alpha + -0.153*lie + ...
10	-0.406*omega + 0.235*alpha + 0.199*lambda + 0.193*equat + ...

Table 6. Top ten LSA topics for $TF \times IDF$ weighting scheme on bigrams.

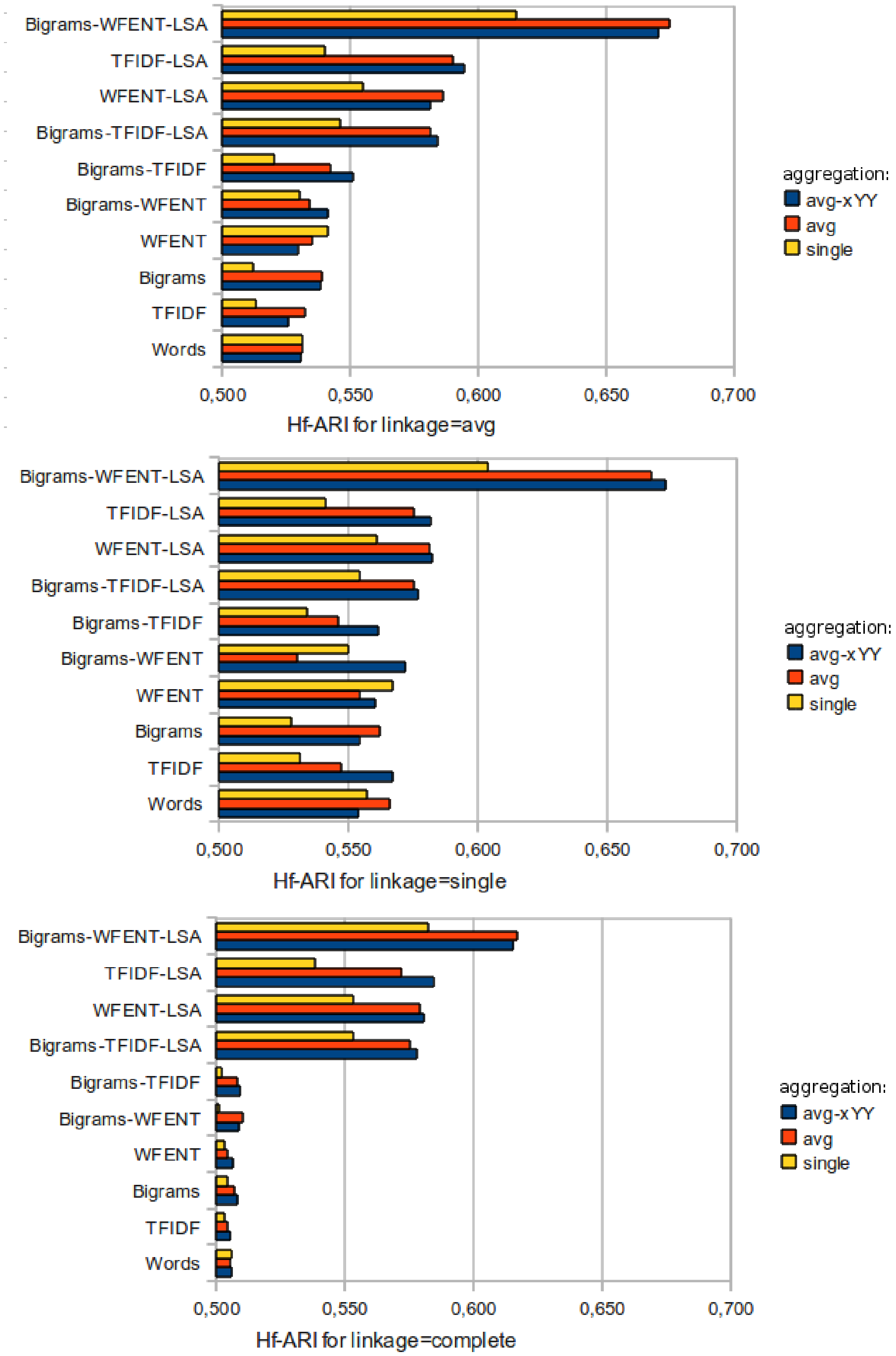
1	-0.581*sb-sb + -0.369*sp-sb + -0.281*sb-sp + -0.143*bbfr-sp + ...
2	0.421*finit-element + 0.256*error-estim + -0.245*sb-sb + ...
3	0.387*finit-element + 0.327*sb-sb + 0.226*element-method + ...
4	-0.496*navier-stoke + -0.424*stoke-equat + 0.262*finit-element + ...
5	-0.516*brownian-motion + -0.269*local-time + -0.234*random-walk + ...
6	0.370*navier-stoke + 0.322*stoke-equat + -0.257*bbfr-sp + 0.210*sb-sb + ...
7	0.263*de-la + 0.140*dan-le + 0.128*bbfr-sp + 0.117*sur-le + ...
8	-0.494*sb-sb + 0.277*sp-sb + 0.219*sb-sp + 0.218*ąl-sb + ...
9	0.535*random-walk + -0.288*differenti-equat + -0.254*brownian-motion + ...
10	-0.595*ąl-sb + 0.278*lie-algebra + 0.242*lie-group + ...

5.4 Linkage and Similarity Aggregation

Figure 8 shows comparison of different linkage (single/complete/average) and aggregation similarity (see section 3.1) strategies. The results only for 3-level hierarchical clustering (section 5.6) are shown but for other configurations are similar. Because results for different variants of weighted average aggregation method (section 3.1) were almost the same we decided to show an average of them (denoted as $avg - xYY$).

An analysis of the figure 8 leads to the following conclusions:

- the weakest aggregation methods is *single*
- an information about secondary codes, in most cases, improves results
- complete-linkage is the worst strategy for clustering
- for single-linkage results are similar for different representations (apart from the best one: Bigrams-WFENT-LSA)



Results for 3-level hierarchical clustering are shown.
$Avg - xYY$ stands for an average of $avg - e0.5$, $avg - s0.5$, $avg - e0.75$, $avg - s0.75$.
For LSA different number of topics were considered and the best (the one with the highest Hf-ARI value) was chosen for every configuration.

Fig. 8. Comparison of different linkage and similarity aggregation strategies against documents' representations.

5.5 Representations and Similarity

Results in the figure 8 can be interpreted as a comparison of different representations against aggregation and linkage strategies. Such analysis of figure 8 leads to following conclusions:

- Bigrams-WFENT-LSA is the best representation among considered
- the use of LSA improves results (it is especially apparent for complete-linkage)

For single-linkage it is hard to derive any consistent rules. Different representations lead to different results. For average-linkage we can assume that bigrams and more advanced representations generally give better results but there are some exceptions.

5.6 Clustering Method

In our experiments we considered two typical clustering approaches. First was hierarchical clustering. Second was k-medoids.

In hierarchical clustering we tested two strategies: either we compared MSC hierarchy to the binary tree (no modification in clustering results) or we compared MSC hierarchy to the tree reduced to just three levels (3-level hierarchical). Reduction was obtained by testing possible splits (number of nodes at M and H level; we tested values differing by 10) and selecting one with the highest Hf-ARI. Obtained value can be interpreted as an approximation of the upper bound for the clustering method.

For k-medoids clustering we used similar strategy. We tested possible values of k on two levels: M and H. Results of clustering from M level were used to compute input similarity matrix for clustering on H level. To aggregate similarity we tested three linkage method: single/complete/average. Because k-medoids method is non-deterministic the Hf-ARI value was averaged for 5 runs of clustering. It is worth mentioning here that differences between results in each run were very small and could not influence our conclusions.

Figure 9 presents Hf-ARI values for different clustering methods in different configurations. Three, described previously, clustering methods were considered. It is clear that the best result are obtained for hierarchical clustering with reducing to three levels (3-level hierarchical). For k-medoids and pure hierarchical clustering (binary tree) results are much worse. In some situations k-medoids obtains higher Hf-ARI value what shows that this measure strongly prefers hierarchies with the same numbers of levels.

5.7 Number of LSA Topics

In our experiments we tested LSA representation of documents. For LSA we tested different weighting schemes and different numbers of topics (25, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500). Table 7 presents what were the best values for different configurations. The column number four presents what weighting

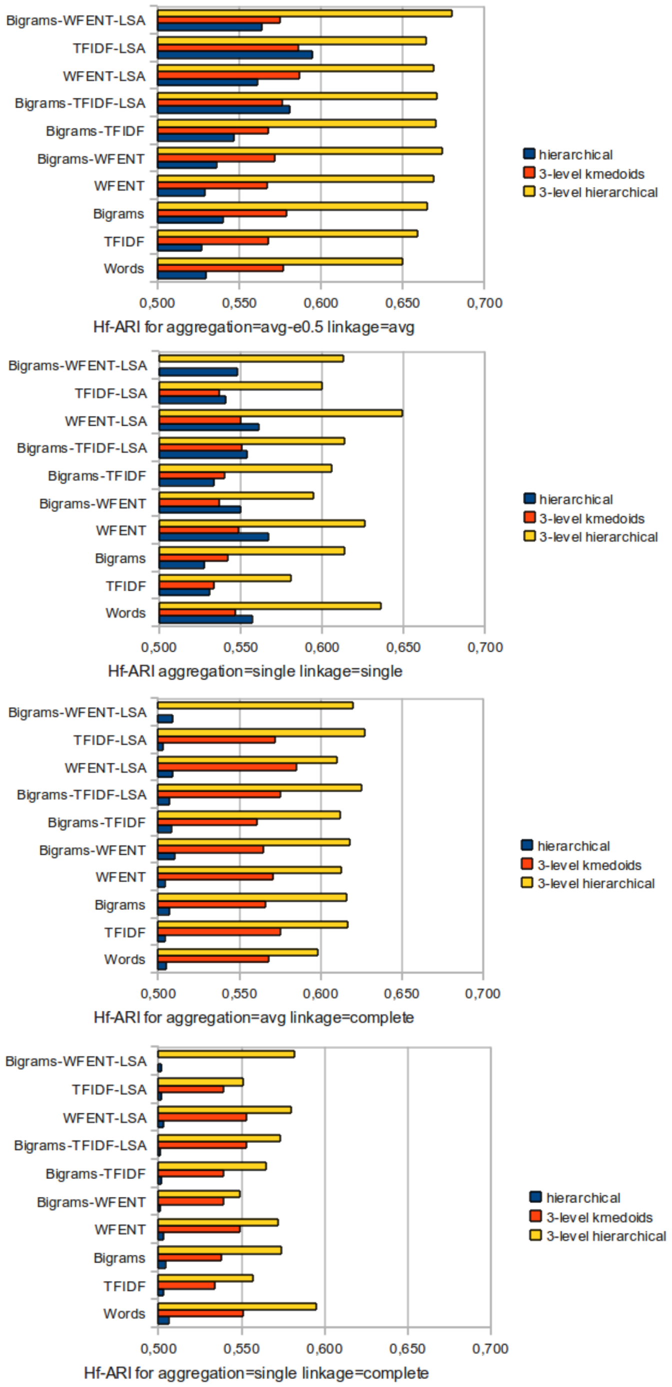


Fig. 9. Comparison of clustering methods.

scheme was used and whether bigrams were used or not. In fifth column the best found Hf-ARI value is shown. The last column presents number of topics assigned to the highest Hf-ARI value.

Table 7. The best number of topics for LSA representation in different configurations.

Aggregation	Linkage	Clustering	Representation	Hf-ARI	Num topics
avg-e0.5	avg	3-level hierarchical	Bigrams-TFIDF-LSA	0.671	500
avg-e0.5	avg	3-level hierarchical	Bigrams-WFENT-LSA	0.680	300
avg-e0.5	avg	3-level hierarchical	TFIDF-LSA	0.664	350
avg-e0.5	avg	3-level hierarchical	WFENT-LSA	0.669	150
avg-e0.5	avg	3-level kmedoids	Bigrams-TFIDF-LSA	0.576	150
avg-e0.5	avg	3-level kmedoids	Bigrams-WFENT-LSA	0.575	25
avg-e0.5	avg	3-level kmedoids	TFIDF-LSA	0.586	25
avg-e0.5	avg	3-level kmedoids	WFENT-LSA	0.587	200
single	single	3-level hierarchical	Bigrams-TFIDF-LSA	0.614	300
single	single	3-level hierarchical	Bigrams-WFENT-LSA	0.613	50
single	single	3-level hierarchical	TFIDF-LSA	0.600	25
single	single	3-level hierarchical	WFENT-LSA	0.649	25
single	single	3-level kmedoids	Bigrams-TFIDF-LSA	0.551	25
single	single	3-level kmedoids	TFIDF-LSA	0.537	25
single	single	3-level kmedoids	WFENT-LSA	0.550	25
single	single	hierarchical	Bigrams-WFENT-LSA	0.548	100

Generally, we can not give any simple advice for choice of number of topics. The best value strongly depends on representation and clustering algorithm what is consistent with observation from section 5.2. Anyway, some dependencies can be observed.

For 3-level kmedoids clustering low number of topics is preferred (especially for aggregation=single and linkage=single). For most configurations 25 topics was selected. For 3-level hierarchical clustering situation is more complicated. For aggregation=avg-e0.5 and linkage=avg high numbers were chosen. For aggregation=single and linkage=single low values are preferred.

5.8 The Best Hierarchy

In our experiments we tested over 2000 configurations (different representations, weighting schemes, similarity methods, clustering methods, number of LSA topics). Table 8 presents top 10 experiments' configurations with the highest Hf-ARI values.

First column stands for similarity aggregation method (section 3.1). Second describes linkage type in clustering. Three typical methods were considered: average, single and complete linkage. All the best results were obtained for 3-level

Table 8. Configurations with the highest Hf-ARI value.

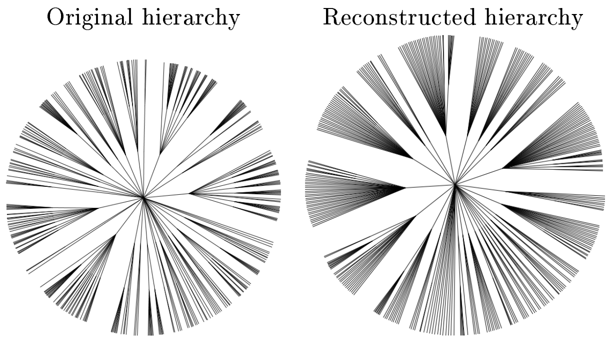
Leaves	Linkage	Representation	Clusters	Hf-ARI
avg-e0.5	avg	Bigrams-WFENT-LSA300	290 / 40	0.680
avg-s0.5	avg	Bigrams-WFENT	320 / 40	0.677
avg-s0.75	avg	WFENT-LSA150	250 / 40	0.677
avg	avg	Bigrams-WFENT-LSA400	300 / 40	0.676
avg-s0.75	single	Bigrams-WFENT-LSA500	220 / 30	0.676
avg	avg	Bigrams-WFENT-LSA350	280 / 40	0.675
avg-s0.5	single	Bigrams-WFENT-LSA250	320 / 30	0.675
avg-e0.5	avg	Bigrams-WFENT	310 / 30	0.674
avg-e0.5	avg	Bigrams-WFENT-LSA200	210 / 40	0.674
avg-e0.5	single	Bigrams-WFENT-LSA350	200 / 30	0.674

All the results were obtained for 3-level hierarchical clustering.
Hf-ARI for comparing MSC tree to itself is equal to 0.759.

hierarchical clustering (for details see section 5.6). In the third column document representations are shown. For LSA we tested different weighting schemes and number of topics between 25 and 500. To calculate similarity we used either Tversky (for words and bigrams) or Cosine-like method (in other cases). The fourth column stands for number of nodes (clusters) at different levels of obtained hierarchy. First number stands for number of nodes at M level and second at H level.

The best obtained hierarchy is for bigrams with $WF \times ENT$ weighting and LSA applied with $avg - e0.5$ similarity aggregation method. This result is consistent with the best result obtained in the section 5.2 what suggest that both measurements give similar results.

In the table 9 fragments of the best obtained tree are shown. In the second column identified problems are described. Simplified visualisation of the tree in comparison to the original MSC tree is shown in the figure 10.



Original hierarchy

Reconstructed hierarchy

Fig. 10. Simplified visualisations of trees.

Table 9. Problems in the best obtained tree.

Tree fragments	Problems
...((62G05) (62G07) (62G10) (62M05))...	some leaves are merged too late
...((20F36) (20D10 20D30 20E15 20F05 20K20)))...	some leaves are merged too fast
...((17B37) (22E40) (32M05) (32M15) (37D40) (43A80) (43A85) (53C15) (53C20) (53C21) ... (53C50) (53C55) (53D50) (58E20) (58J20) (58J35) (58J50) (58J60) (49Q05 53A10) (17B10 17B20 17B35 22E30 22E45 22E46 22E47))...	big groups of leaves of different kind are merged
... ((60K25 90B22))...	small clusters out of totally different codes are created
... ((91B14)) ((91B28)) ...	leaves that should be merged are left separated
...(65M15 65M60 65N15 65N25 65N30 65N55 74S05 76M10)...	single leaves are glued to groups of different type

5.9 Interesting Results

We reviewed our best hierarchy. We checked all clusters of size 2 and 3 at M level. From such clusters we extracted those pairs of MSC leaves that were in the same cluster but had no common prefix (e.g. 60K25 and 90B22). The list of extracted pairs is shown in the table 10.

An analysis of the table 10 leads to interesting conclusions. Our reconstruction process glued leaves that were strongly linked despite the fact that they were in different branches of the hierarchy. For example 60K25 (‘Queueing theory‘) was merged with 90B22 (‘Queues and service‘). It is clear that this two leaves must be very similar but the first was placed in 60 (‘Probability theory and stochastic processes‘) and the second in 90 (‘Operations research, mathematical programming‘).

Very similar analysis is shown in the table 11. We considered all pairs of MSC leaves and extracted those that were in the same cluster but had no common prefix. In the next step, we casted leaves to the highest level of the hierarchy (extracted prefixes of length 2) and counted pairs.

An analysis of the table 11 reveals similar conclusions. Some of the groups of leaves are placed in the same cluster even though they are in different branches. For example ‘Numerical analysis‘ very often co-occurred with ‘Fluid mechanics‘. This is connected to the fact that numerical methods are widely used in applications of fluid mechanics.

Table 10. 'Wrong' pairs in clusters of size 2 and 3.

Leaf	Explanation	Leaf	Explanation
60K25	Queueing theory	90B22	Queues and service
35P25	Scattering theory for PDE	47A40	Scattering theory
49Q05	Minimal surfaces	53A10	Minimal surfaces, surfaces with prescribed mean curvature
32S65	Singularities of holomorphic vector fields and foliations	37F75	Holomorphic foliations and vector fields
34C25	Periodic solutions	37J45	Periodic, homoclinic and heteroclinic orbits; ...
11F70	Minimal surfaces, surfaces with prescribed mean curvature	22E50	Representations of Lie and linear algebraic groups over local fields
35Q30	Stokes and Navier-Stokes equations	76D05	Navier-Stokes equations
35Q30	Stokes and Navier-Stokes equations	76N10	Existence, uniqueness, and regularity theory

6 Summary and Conclusions

In this paper we studied the problem of recreating the hierarchy of codes of a subject classification system. Our goal was to find a method of constructing, based on metadata of mathematical publications, a tree that would be as close to the original MSC 2000 tree as possible.

In order for the goal to be meaningful, we first had to decide what it means that two given trees are similar and to quantify that similarity. To this end we studied and developed novel methods of assessing tree similarity. After a series of experiments we have chosen Hf-ARI measure (cf. Section 2) as it had the best properties among all the evaluated candidates.

Next, we have selected a method of quantifying document similarity and devised optimization in computing similarity matrices. Finally, we have performed a series of experiments aimed at calibrating our solution. Each experimental result was accompanied by analysis and conclusions (cf. Section 5).

6.1 Future Work

During our research we have identified several problems occurring in reconstruction process e.g. some leaves are glued too early whereas other too late. The problems decrease quality of obtained results. We believe that their influence can be reduced by modifications in representation and clustering algorithms. Incorporating new features (e.g. link-based) can also help.

Another direction of our works would be to examine more deeply correlation between nodes in MSC hierarchy. This research could lead to the proposal of a

Table 11. The most often 'wrong' pairs.

Count	MSC leaf	Description	MSC leaf	Description
24	17*	Nonassociative rings and algebras	22*	Topological groups, Lie groups For transformation groups
12	65*	Numerical analysis	74*	Mechanics of deformable solids
12	65*	Numerical analysis	76*	Fluid mechanics For general continuum mechanics
4	35*	Partial differential equations	76*	Fluid mechanics For general continuum mechanics
2	90*	Operations research, mathematical programming	60*	Probability theory and stochastic processes For additional applications
2	76*	Fluid mechanics For general continuum mechanics	74*	Mechanics of deformable solids
2	37*	Dynamical systems and ergodic theory	32*	Several complex variables and analytic spaces For infinite-dimensional holomorphy
2	34*	Ordinary differential equations	37*	Dynamical systems and ergodic theory
2	22*	Topological groups, Lie groups For transformation groups	11*	Number theory
2	35*	Partial differential equations	47*	Operator theory
2	49*	Calculus of variations and optimal control; optimization	53*	Differential geometry For differential topology

modified structure. The structure could be more efficient in some applications such as automatic classification.

7 Acknowledgements

This work is supported by the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the Strategic scientific research and experimental development program: "Interdisciplinary System for Interactive Scientific and Scientic-Technical Information."

References

1. ICML 2010 Tutorial on Metric Learning (2010), http://www.eecs.berkeley.edu/\~{}kulis/icml2010_tutorial.htm
2. Ahmad, S.B., Cakmak, A., Ozsoyoglu, G., Hamdani, A.A.: Evaluating Publication Similarity Measures. IEEE Data Eng. Bull. 28(4), 21–28 (2005), http://sites.computer.org/debull/A05dec/bani-ahmad.pdf
3. Brouwer, R.: Extending the rand, adjusted rand and jaccard indices to fuzzy partitions. Journal of Intelligent Information Systems 32, 213–235 (2009), http://dx.doi.org/10.1007/s10844-008-0054-7, 10.1007/s10844-008-0054-7
4. Denœud, L., Guénoche, A.: Comparison of distance indices between partitions. In: Batagelj, V., Bock, H.H., Ferligoj, A., Žiberna, A. (eds.) Data Science and Classification, pp. 21–28. Studies in Classification, Data Analysis, and Knowledge Organization, Springer Berlin Heidelberg (2006), http://dx.doi.org/10.1007/3-540-34416-0_3
5. Fowlkes, E.B., Mallows, C.L.: A method for comparing two hierarchical clusterings. Journal of the American Statistical Association 78(383), 553–569 (1983), http://dx.doi.org/10.2307/2288117
6. Fuller, M., Zobel, J.: Conflation-based comparison of stemming algorithms. In: In Proceedings of the Third Australian Document Computing Symposium. pp. 8–13 (1998)
7. Hüllermeier, E., Rifqi, M., Henzgen, S., Senge., R.: Comparing fuzzy partitions: A generalization of the rand index and related measures. IEEE Transactions on Fuzzy Systems ((to appear))
8. Jiang, D., Tang, C., Zhang, A.: Cluster analysis for gene expression data: a survey. IEEE Transactions on Knowledge and Data Engineering 16(11), 1370–1386 (Nov 2004), http://dx.doi.org/10.1109/TKDE.2004.68
9. K.Sathiyakumari, V.Preamsudha, G.Manimekalai, M.Phil Scholar: A Survey on Document Clustering Using Different Techniques. International Journal of Computer Applications in Technology 2(5), 1534–1539 (2011)
10. Lee, M., Pincombe, B., Welsh, M., Bara, B.: An empirical evaluation of models of text document similarity. Lawrence Erlbaum Associates, Chicago (2005)
11. Lin, Z.: Link-based Similarity Measurement Techniques and Applications. Ph.D. thesis, The Chinese University of Hong Kong (2011)
12. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
13. Nakov, P., Popova, A., Mateev, P.: Weight functions impact on lsa performance. In: EuroConference RANLP'2001 (Recent Advances in NLP. pp. 187–193 (2001)
14. Porter, M.F.: Readings in information retrieval. chap. An algorithm for suffix stripping, pp. 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1997), http://dl.acm.org/citation.cfm?id=275537.275705
15. Rand, W.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66(336), 846–850 (1971)
16. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), http://is.muni.cz/publication/884893/en
17. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM 18(11), 613–620 (Nov 1975), http://doi.acm.org/10.1145/361219.361220

18. Shepard, R.: The analysis of proximities: Multidimensional scaling with an unknown distance function. i. Psychometrika 27, 125–140 (1962), http://dx.doi.org/10.1007/BF02289630, 10.1007/BF02289630
19. Tversky, A.: Features of similarity. Psychological Review 84(4), 327–352 (1977)