

Towards Vectorial Web Search

Wolfgang Orthuber
UKSH
Arnold Heller Str. 16
24105 Kiel / Germany
+49 431 5972881

orthuber@kfo-zmk.uni-kiel.de

ABSTRACT

General similarity search of quantifiable resources is possible on the Web. For this a simple data structure called VRD ("Vectorial Resource Descriptor") is proposed, which contains a feature vector for representation of the object, and a VSI ("Vector Space Identifier") which uniquely identifies the kind of object which is represented by the feature vector. Feature vectors of VRDs with the same name are directly comparable using a given metric. At this similarities of the objects' data are mapped to spatial similarities of the feature vectors. So similarity search is possible by calculating distances: Objects are the more similar, the smaller the distance between the feature vectors of the representing VRDs is. This vectorial (numeric) similarity search could be efficiently combined with conventional word based search. We describe this here and give some examples.

Categories and Subject Descriptors

H.3.3 [Information search and retrieval]: *Search process – Selection process*; E.1 [Data structures]: *Distributed data structures*; I.5.2 [Pattern recognition]: *Design Methodology, Feature evaluation and selection*

General Terms

Algorithms, Standardization

Keywords

Pattern Search, numeric web search, vectorial web search, similarity measure, feature vectors, task sharing, metric databases, similarity search, VRDs, VSIs, QRIs

1. INTRODUCTION

Due to the enormous amount of data on the web there is an increasing need for information integration. This lead to the *Semantic Web* and the *Linked Data* approach which aim in meshing together meaningful machine readable data on the web. The mesh already allows definition of neighborhood: Data which are directly linked together can be regarded as neighboring in the mesh. At this it is necessary to create every link explicitly. There is, however, a well known mathematical concept for definition of neighborhood which is much finer than any mesh of links:

The vector space

It is no competitor, because it cannot be used in all cases. But in applications in which it can be used, e.g. representation of measurable objects, it is a very efficient extension to the existing mesh of linked information, using a compact data structure (figure 1) as connector. Searching within a vector space bases on numeric comparisons. This is completely different to current language based web search in which word combinations can be

found. Word based search is incomplete: If you have original data which represent e.g. sounds, pictures or measurement results, you cannot search for similar data directly, bypassing language. There are too many different kinds of binary representations and any kind needs its individual algorithm for comparison. Programming and maintenance of all these algorithms exceeds the working capacity of a search engine's personnel. But it is possible to make feature extraction from original data and to store the result in a searchable form. Subsequently we will call this form "VRD" (Figure 1 and chapter 2.2.2.1).

VRD

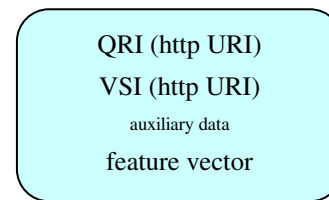


Figure 1. A Vectorial Resource Descriptor according to our definition, designed as simple as possible. QRI is the identifier of the (quantifiable) resource. The VSI ("Vector Space Identifier") identifies the meaning of the numerical content (feature vector) globally uniquely (2.1.2). So it is clear which kind of original data is represented. Similarities of original data are mapped to spatial similarities of the feature vectors. So to a given VRD similar VRDs with the same VSI can be found by direct comparison of the feature vectors using an appropriate metric.

VRDs are machine readable and could be embedded into the semantic web as linked open data ([7], 2.2.1). They are uniformly comparable and searchable, and it is possible to share the work for definition and generation of VRDs by efficient organization (2.1.2) which also makes it commercially attractive to participate. As a consequence it is possible to search with the same search engine not only for text, but also for an increasing number of well-defined numerical VRDs on the web. This bundling of the search activity into one crawler and web database for all VRDs is much more efficient than building and managing a database and a crawler for every kind of VRD. The benefits become clear when thinking about new possible searchable VRDs like

- Human diagnostic parameters and measurements [26] (blood parameters, MRI with feature extraction, results of echocardiography, heart sounds (5.2.2),

EKG, temperature chart etc.). The more comprehensive the causes leading to a disease are recorded, the better. Suppose the doctor could send the patient's parameters and measurements directly and anonymously into the Internet and find completed medical reports (with real treatment results) of patients with similar VRDs. Obviously with this information he could perform better treatment¹. The consequence would be more objectivity in medicine, which is very important.

- Measurements and classifications in all areas of daily life and of science
- Digital representations of various items of daily life and their direct recognition (e.g. melodies, faces etc.)
- Measurement and classification schemes of products and services and with this more individual and reproducible adaptation of products and services to the customers needs.
- Numeric data which are up to now stored in separated databases in the hidden web, not accessible for search engines. The proposed VRD structure gives motivation to pack such numeric data in a globally accessible and interchangeable form. So it could also help to make hidden (numeric) data from the deep web accessible for all people

It would exceed the scope of this article to list more than a few examples, but already these show that there is not only scientific, but also practical and commercial potential. The examples also indicate that such a universal approach to direct VRD search is a comprehensive task and requires a clear reply to the following questions:

1. How can the user provide a VRD for which similar VRDs are to be found on the web?
2. How can the search engine recognize the kind of the VRD provided by the user and so isolate the set of comparable VRDs on the web?
3. How can the search engine quantify the similarity between the provided VRD and the comparable VRDs on the web to calculate their rank in the search result?

It turns out that there are satisfying answers to all these questions. They are abbreviated:

1. In case of concise VRDs the user can enter the VRD directly by keyboard, e.g. as sequence of numbers, together with a "VSI" (figure 1, 2.1.2), which uniquely specifies the kind of the VRD. In other cases the user can provide the VRD as file generated by software which is designed for handling of this VRD kind. If appropriate, this software may be connected with some digitizing device.
2. The search engine recognizes the kind of the VRD by the VSI. On the web VRDs are stored within special strings, within hypertext, or within RDF or XML files (see 2.2.2). They contain in their header the associated VSI, auxiliary

information and the feature vector. Comparable are only files with the same VSI.

3. Quantification of similarity is done by direct comparison of the feature vectors of the VRDs using a short distance function (e.g. Euclidean distance, Manhattan distance).

2. Realizing vectorial (numeric) web search

The here suggested organizational details should represent an efficient possibility for realization. Variants are conceivable. Important is that responsibility and necessary work are clearly shared in a way that it is attractive to participate.

2.1 Names and conventions

First of all it is appropriate to explain some frequently used abbreviations:

2.1.1 VRD

A VRD ("Vectorial Resource Descriptor") consists of the VSI ("Vector Space Identifier") and the feature vector as shown in figure 1, plus additional information (date, links, see 2.2.2.1). The feature vector is usually a sequence of numbers. The length of the sequence is variable; it depends on the VRD definition.

2.1.2 VSI

The VSI uniquely identifies the kind of data which are represented by the feature vector. It is a http URI [7] [32] which (like a URL) permanently refers to a unique, permanent web address and which differs if and only if the web address differs. So it is a unique name and also a unique reference. It points to a file with links (VSD, see 2.2.1) to all defining and further associated information. **This guarantees also that there is exactly one determining definition (anchor) for this name, and indirectly well defined task sharing among all domain name owners for definition of VRDs.**

2.1.3 VRD representation

It is possible to add the VRD directly to hypertext (similarly like RDFa). VRDs can be also represented as special files (2.2.2.1).

2.1.4 Comparison of VRDs, deviation d

If a VRD is given for a search, it is necessary to quantify the similarity to other VRDs on the web with the same VSI. The result of such comparison is a number $d \geq 0$ ("deviation"), in which $d=0$ if the feature vectors of the compared VRDs are identical, else $d > 0$; d is the greater, the more they deviate. As result of the search the URLs of those VRDs with the smallest d are shown first, together with d , which makes the search result more transparent.

2.1.5 Feature vector generalized

The feature vector is usually a sequence of numbers, more generally it is a data structure designed for efficient comparison, so that the result of comparison is a number $d \geq 0$. This convention makes the VRDs adaptable to the programmers' needs.

2.2 Organizational prerequisites

¹ The best treatment could be found using global experience; repetition of ineffective treatment experiments could be avoided.

2.2.1 VSIs

VSIs are http URIs as described in 2.1.2, they not only identify the numeric content, they simultaneously point to a Vector Space Descriptor (the "anchor") which contains all necessary information (definitions, templates, metrics, links to related web content). Subsequently we will call this file VSD.

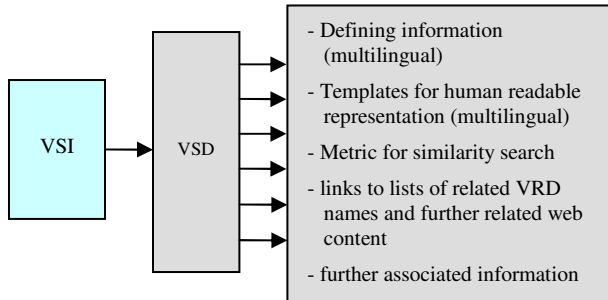


Figure 2. The VSI points to the Vector Space Descriptor VSD which contains all determining information of VRDs with this name

The owner of the internet domain dn in which the Vector Space Descriptor VSD is placed has the privilege to determine all defining and further information which is associated to VRDs with names dn/*². If he wants that his VRDs are useful and not ignored, he should

- provide efficient and useful definitions of the VRD structure, so that they (their numerical representation as feature vectors) are directly 1:1 comparable (and with this searchable) using a short default algorithm (e.g. weighted Euclidean distance, Manhattan distance),
- if necessary, give information to software for creation of VRD files and/or donate or sell it,
- if necessary, give information about associated digitizing devices, he may also sell them.
- if necessary, provide for every kind of VRD a template which can be used for viewing the feature vector's content.

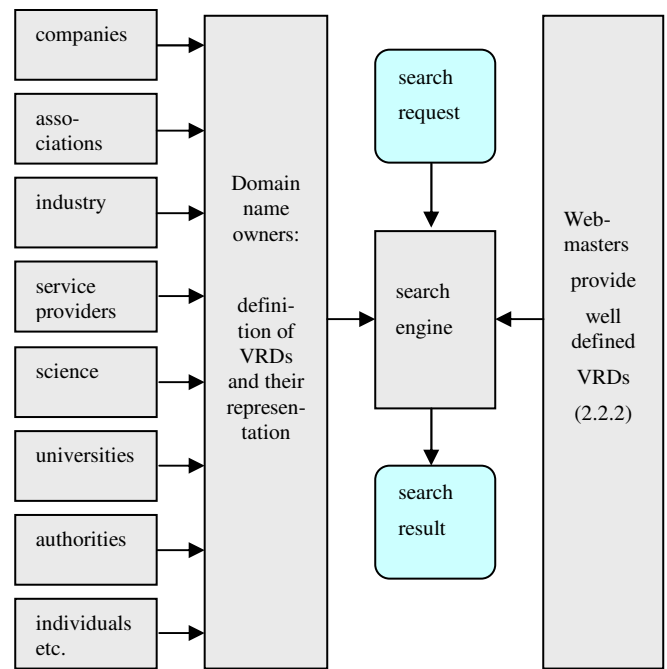


Figure 3. Task sharing in VRD search.

Someone who invests much work in optimizing his kind of VRD can gain from this, because an efficient VRD is more frequently used. Some consequences:

- His domain dn is more attractive
- The (own) products and services which can be found via his well-defined VRDs are more attractive.
- The (own) software and/or digitizing devices which are necessary for generation of the dn/* VRDs are more attractive.
- Communication in the own special field is more efficient.

Webmasters are motivated to integrate well-defined VRDs (see 2.2.2) in their pages, because this makes them better detectable. At this **they have to adhere to the standard, else the search engine does not recognize the VRDs**. So many parties are involved, and there is motivation to keep to the standard.

The subsequent examples are written in XML for illustration purposes. This can help for development of the final standard.

2.2.1.1 patdef.xml

The Vector Space Descriptor VSD points among others to a file "patdef.xml" which is used by the search engine. It contains:

```
<VRD_entry> //start of entry which
//describes one VRD

<VRD_name> ... </VRD_name> //the VSI

<keywords> ... </keywords> //keywords associated with
//this VSI and its definition; most
//important keywords should occur first; the search
//engine can use this information

<fvlen> ... </fvlen> //maximal length of the feature
//vector or -1, if open

<cmode> ... </cmode> //the VRDs are represented
//by sequences of numbers and cmode is the number
```

² It is convenient to call the VRDs with these names a "VRD group".

```

//of the metric used for comparison;
//it determines the calculation of the
//deviation d (see 2.1.4) between
//two compared VRDs

//1: d = 1 - correlation coefficient
//2: d = Euclidean distance (sqrt(sum of squared
// differences))
//3: d = sum of absolute differences
//      (1D-point differences, Manhattan metric)
//4: d = sum of 2D-point differences
//      (VRD numbers are paired)
//5: d = sum of 3D-Point differences
//      (VRD numbers are triples)
//6: d = GPS metric
//7: d = discrete metric
//      (to search for only identical VRDs)
//...

<fvweight>// optional weighting vector (a sequence of
... //numbers), contains for all dimensions of the
... //feature vector multipliers which are
... // applied before comparison; default for every
//dimension: e.g. 1/(standard deviation)
</fvweight>

<URLde> ... </URLde> //URL of a complete definition
//and description of the VRD

<URLview> ... </URLview> //optional URL of a template
//which can be used to show the
//content of the VRD;
//above URLs can be also stored
//in the Vector Space Descriptor

VSD

... //further information to this kind of VRD

</VRD_entry> //end of entry which describes the
//VRD

```

2.2.2 VRDs on the web

2.2.2.1 VRDs within hypertext or as XML -files

One possibility for VRD representation is a XML file (see 2.1.3):

```

<VRDfile> // start of VRD-file
<VRD> // start of VRD
<VRD_name> ... </VRD_name> // the VSI

<subname> ... </subname> // optional name extensions
// which can be used for selection

<date> ... </date> // Date

<URLa> ... </URLa> // one or more URLs associated
// with this VRD; the primary URL
// is listened in the first entry

<URLorg> ... </URLorg> // if appropriate the URL of a
// file which contains original data

// further data
// also place for expansions and improvements

<feature_vector> ... </feature_vector>
// feature vector (a sequence of numbers)
// which represents the VRD

</VRD> // end of VRD
... // optionally further VRDs
</VRDfile> // end of VRD-file

```

Of course also other formats are possible. If wished, the <VRD>...</VRD> structure can be embedded in hypertext files (similarly like RDFa).

2.2.3 Semantic Web

The Resource Description Framework (RDF) [33] can be used for VRD representation.

The usage of RDF for the *definition* of VRDs (instead of natural language) is more ambitious. There would be far-reaching possibilities, e.g. for description of relations between VRDs with different VSIs. Future research and practice can show details about this.

2.3 The search

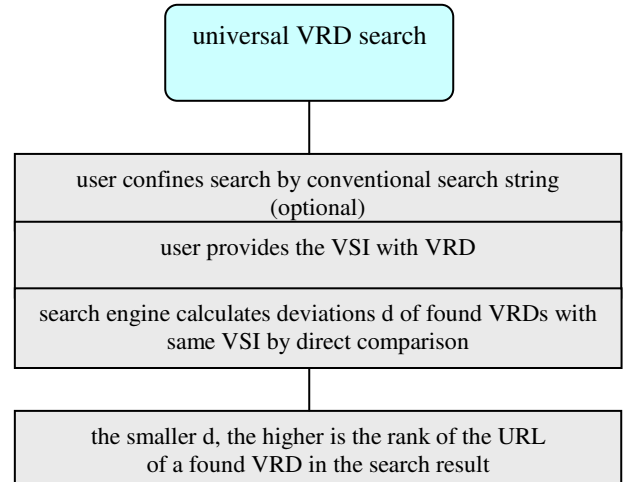


Figure 4. The search process.

The user must provide the VRD (VSI and feature vector), optionally AND combined with a conventional search string. The search engine reads the file "patdef.xml" which contains the variable cmode. Depending on cmode and optional weighting coefficients the search engine calculates the deviation d of every VRD (of its feature vector as defined in 2.2.2). In the search result the VRDs with smallest deviation d are listened first, together with d (Fig. 4).

It is possible that the user defines the metric for comparison. This can be advantageous in case of good knowledge about the VRD's feature vector structure.

2.3.1 Combination of VRDs

It is also possible to combine several VRDs for a search. At this the weight (for determination of the search result order) of every VRD could be predefined by an additional number, e.g. as relative percentage. If not predefined, as default value for weighting e.g. the reciprocal of the distance's standard deviation can be used, or (dependent on the chosen metric) the reciprocal of the distance's mean deviation.

2.3.2 Structured queries

The components of the feature vector (numbers) are addressable using an index. Structured queries (SPARQL [34], XQuery [35]) could be used to specify additional requirements for the search, for example the condition that certain components of the feature vector must be smaller or greater than given limits. Queries with very complex conditions could be generated by software tools.

2.3.3 *Templates for human readable representation*

Templates are sentences in which the numeric content is inserted. They are accessible via the Vector Space Descriptor VSD (Fig. 2, 2.2.1) and can make the numeric content human readable in all wished languages. Long feature vectors can be made visible by templates which describe the design of diagrams or other graphic representation. This is a comprehensive and interesting topic (standardized representation with available original data) which can be focused in further publications.

3. Experimental evaluation

A local nontrivial prototype for an application in healthcare (Similarity search of heart sounds) has been shown in Fig 6-8 of [26]. Though it is not possible for us to emulate the global web application, its feasibility can be derived from the fact, that the search result is nothing else than a list of web references (to VRDs with the same VSI, see 4.3.2) which is sorted according to the distances of the VRDs' feature vectors, and calculation of distances between vectors is quickly possible: We made a small (and well reproducible) experiment using a C++ compiler on a single PC (2,1 GHz Pentium). The time for calculating 1 million weighted Euclidean distances of vectors with dimensionality 10 in double accuracy (without file IO) was between 0,20 and 0,21 seconds. If the dimensionality is greater or smaller, the time is nearly proportional to the dimensionality. Such calculation would be only necessary if similarity search is done within 1 million VRDs of the same kind (with the same VSI) without tree structure.

4. Discussion, possible problems and solutions

4.1 Reliability of VRD definitions

Problems could occur, if a domain name owner wants to redefine a widespread VRD structure, e.g. because of improvements. A simple expansion of a VRD definition (a definition of additional feature vector components) would be possible without changing the VSI - in this case the old VRDs would be regarded as partially completed, they would remain comparable using a metric which considers only the existing dimensions, if this is wished in the search request. In case of relevant (possibly incompatible) changes, however, a modification of the VSI (see examples in 2.1.2) is necessary, so that there is no overlapping. If a domain name owner makes arbitrary redefinitions of his VRD structures, he would damage his reputation. Therefore he should avoid this. Frequency of irresponsible behavior will probably be similar to other areas of software business and economy.

Redefinitions can be avoided completely if VRD definitions are stored in a non modifiable way. Furthermore it should be discussed whether we establish an institution which very early introduces officially recommended VSIs with definitions. We recommend integration of frequently used basic VRD definitions in official domains to guarantee their quality and reliability.

4.2 Redundant VRD definitions

Due to the URL convention 2.1.2 every VSI will have exactly one meaning (not multiple meanings). But there is the possibility that there are many VSIs whose feature vectors have (nearly) the same meaning. To avoid this, definitions of VRDs (Fig 2) could also contain lists of keywords. Text search restricted to VRD definitions discloses existing definitions about a specific topic to prevent from redefinitions. If there are nevertheless many VSIs with identical feature vector definition, webmasters can ask web search engines for the most frequently used VSI, and integrate only VRDs with these name in their web pages. So we can get concentration to one name again.

4.3 Complexity

Though the technical feasibility of searching in metric spaces is well investigated [10], it is advisable to think about possible barriers concerning the complexity of the project.

4.3.1 Complexity of data storage

The feature vectors of appropriately defined VRDs represent very compressed information (a sequence of numbers; units and definition of every number are stored once on the web (2.1.2) for all VRDs with the same name, not repeatedly). So VRDs can even contribute to more efficient usage of web space.

4.3.2 Time complexity

VRDs which are comparable have the same VSI. So to a given VRD all comparable VRDs can be found on the total web as quickly as today words can be found on the web which are identical to a given word.

To minimize the time for comparison (see 3) it should be tried to minimize the dimensionality of the VRD's feature vector. This should be also done because the sparsity increases exponentially with the dimensionality given a constant amount of data, with points tending to become equidistant from one another ("Curse of Dimensionality" [4][19]). The search time also depends on the number of comparable VRDs on the web (those with the same VSI) and on further confinement of the search by a regular expression (2.3.2) or a conventional search string (e.g. "box" in 5.1.3). Due to such preselection the subset of concerned VRDs and time for search is reduced. Further enhancement of performance is possible using an appropriate tree structure [3][4][5] or hashing [16]. Only if very many high dimensional VRDs with the same VSI are stored on the web, and if the search is not enough confined, and if there is not enough hardware for parallel processing, the search time can become critical. If we accept small errors, even in this case it can be possible to get an acceptable search result after dimension reduction [15] or by using approximation methods [34].

4.4 Novelty, hen and egg problem

Up to now there is no data structure on the web which is designed for general (also multidimensional, vectorial) numeric similarity search over the total web. The VRD structure (VSI and feature vector) is consciously designed as simple as possible to close this gap. It is not restricted to a special application (e.g. multimedia) and can be evaluated in uniform way, because its format is consistent and not dependent on the kind of represented original data. This allows to minimize the expense for programming similarity search.

But up to now there are no VRDs on the web, so there is no direct motivation for a search engine to program the extensions necessary for their similarity search. And if there is no search engine which supports VRD search, there is no direct motivation for webmasters to insert VRDs in their domains.

Therefore main intention of this article is to show the potential of VRD search, so that already the indirect motivation is enough to do the first step, so that we can overcome the hen and egg problem.

5. Some examples

The following examples illustrate very different applications. They are knowingly chosen so to show the adaptability of the VRD structure.

5.1 Concise VRDs

The lower the dimensionality, the easier is the handling. As convention we suggest, that VRDs with (feature vectors with) less than 9 dimensions can be called "concise VRDs".

5.1.1 GPS coordinates

A feature vector with GPS coordinates is obviously attractive. Alone this definition would allow combination of conventional text search with radial search:

The File "patdef.xml" (2.2.1.1) contains the following entry

```
<VRD_searchpar>
  <VRD_name>dn/gps.htm </VRD_name>
  <fvlen>3</fvlen>
  <cmode>6</cmode>
  <URLde>gpsdef.htm</URLde>
</VRD_searchpar>
```

The file "gpsdef.htm" explains the feature vector of VRDs with name "dn/gps.htm" and could give some examples, which show that the 3 ordered numbers of the feature vector describe the latitude, longitude and altitude of GPS coordinates in decimal degrees.

In case of such concise VRDs instead of file-upload direct keyboard input is adequate. At this it is convenient to include VSI and VRD directly into the search string using the special sign "#" as delimiter. An example of a search string is

```
#dn/gps.htm -35.283130 149.113580#
```

This string can be used to search for objects near the given GPS coordinates (here latitude -35.283130 and longitude 149.113580, 2D comparison because altitude has been omitted). Usually this is combined with conventional text search, e.g. searching for the string "hotel". The combined search string would be

```
hotel #dn/gps.htm -35.283130 149.113580#
```

This string can be used to search³ for hotels near the given GPS coordinates. Similarly it would be possible to search for near

³ More precisely: This string causes a search in all URLs associated with VRDs with VSI "dn/gps.htm" for pages which contain the word "hotel". The pages with minimal deviation d are listened first. If d=0 (minimal), the feature vector must be "-35.283130 149.113580" or "-35.283130 149.113580 x", in which x is an arbitrary number (altitude is ignored because the search string contains only two numbers).

doctors, specialists, restaurants, shops etc., and to make further evaluation [37].

The definition of coordinate systems can be very helpful also on other objects. For example a simple three dimensional Cartesian coordinate system on a representative anatomical model of the human body (with zoom function) can be used for conversation in medicine. For localization doctors can exchange the coordinates, the specification of the affected body tissue, and the size of a finding.

5.1.2 Price-d

This VRD can be used to search for objects which are for sale. The File "patdef.xml" contains the following entry

```
<VRD_searchpar>
  <VRD_name>price-d.htm </VRD_name>
  <fvlen>1</fvlen>
  <cmode>3</cmode>
  <URLde>price-def.htm</URLde>
</VRD_searchpar>
```

The file "price-def.htm" explains the feature vector structure associated with VSI "dn/price-d.htm": Only one number which is the price of some object in dollar.

An example of a search string is

```
#dn/price-d.htm 0#
```

to search for as cheap as possible objects, or

```
#dn/price-d.htm 100#
```

to search for objects with prices near 100 dollar.

An example of a combined search string is

```
suitcase #dn/price-d.htm 100#
```

This can be used to search for suitcases with prices near 100 dollar. Obviously it would be desirable to specify further characteristics of the suitcase. This can be done by combination of the "price-d" VRD with other VRDs, e.g. with the "lwh" VRD:

5.1.3 Length, width, height

The File "patdef.xml" :

```
<VRD_searchpar>
  <VRD_name>dn/lwh.htm </VRD_name>
  <fvlen>3</fvlen>
  <cmode>3</cmode>
  <URLde>lwhdef.htm</URLde>
</VRD_searchpar>
```

The file "lwhdef.htm" explains the feature vector structure associated with VSI "dn/lwh.htm" and should give some examples which show that the three ordered numbers of the VRD describe the length, width and height of some object in meters.

An example of a combined search string is

```
box #dn/lwh.htm 3 2 1#
```

This search string can be used to search for boxes with 3 m length, 2 m width and 1 m height. It is useful to define "?" as placeholder for free numbers (which are ignored when calculating the deviation) and to assume "?" for all omitted trailing numbers. So the search string

```
box #dn/lwh.htm 3 ? ? #
```

or synonymously

```
box #dn/lwh.htm 3#
```

can be used to search for boxes with 3 m length and variable width and height.

5.1.4 Color

The File "patdef.xml":

```
<VRD_searchpar>
  <VRD_name>dn/color.htm </VRD_name>
  <fvlen>2</fvlen>
  <cmode>3</cmode>
  <URLde>colordef.htm</URLde>
</VRD_searchpar>
```

The file "colordef.htm" explains the feature vector structure associated with the VSI "dn/color.htm": The feature vector has two dimensions (a, b) in which

a = percentage of red (650 nm),

b = percentage of green (510 nm);

Percentage of blue (475 nm) = 100 - a - b;

An example of a combined search string is

```
LED #dn/color.htm 0 50 #
```

This string can be used to search for a blue-green light emitting diode.

5.1.5 Shoe size

The File "patdef.xml":

```
<VRD_searchpar>
  <VRD_name> dn/shoesize-usm.htm </VRD_name>
  <fvlen>2</fvlen>
  <cmode>3</cmode> //comparison by summation
  //of absolute differences
  <URLde>usmdef.htm</URLde>
</VRD_searchpar>
```

In this case the file "usmdef.htm" explains the structure of the VRD. An exemplary VRD-file may contain:

```
<VRD>
  <VRD_name>dn/shoesize-usm.htm </VRD_name>
  <URLa>model_marathon.htm</URLa> //link to file
  //with description of an associated shoe
  <feature_vector>
    9 //size of shoe (US men's shoe size)
    1 //width of shoe (1=narrow, 2=medium, 3=broad)
  </feature_vector>
</VRD>
```

After reading this example we could assume that the file "model-marathon.txt" describes a narrow version of a shoe with US men's shoe size 9. A combined search string for such shoes is

```
runner #dn/shoesize-usm.htm 9 1#
```

or, if width does not matter

```
runner #dn/shoesize-usm.htm 9#
```

The examples show that the usage of concise VRDs is an easy and flexible possibility to search qualified for objects which have one or a few quantifiable properties. Because of their constant meaning and probably frequent usage it is recommendable to define basal VRDs (which concern e.g. measurements of size, area, volume, weight and other combinations of SI units) in an official reliable domain (4.1).

5.1.6 Discrete metric for feature vectors which represent words

It is possible to use the discrete metric for a search, so that only VRDs are listened in the search result, whose feature vectors have zero distance. This is e.g. appropriate if the feature vector represents a name. We can define that in case of the discrete metric the feature vector is the lowercase Unicode representation of words, separated by "-", and that also the search string is automatically converted into lowercase Unicode. This allows

direct (non-numeric) input of words in the search string. An exemplary file "patdef.xml":

```
<VRD_searchpar>
  <VRD_name>dn/photo.htm</VRD_name>
  <fvlen>-1</fvlen> //open length
  <cmode>7</cmode> //discrete metric
  <URLde>photodef.htm</URLde>
</VRD_searchpar>
```

The file "photodef.htm" explains that the feature vector is the Unicode representation of an object's name, and that all <URLa>...<URLa> entries of VRDs point to photos of this object.

An exemplary searchable VRD

```
<VRD>
  <VRD_name>dn/photo.htm</VRD_name>
  <URLa>photo1.jpg</URLa> //links to photos
  <URLa>photo2.jpg</URLa> //of the object
  <URLa>photo3.jpg</URLa>
  ...
  <feature_vector>
    ... //Unicode representation of the object's name
  </feature_vector>
</VRD>
```

We may also define, that person's names should be given in the form "forename-surname". So an example of a search string is

```
#dn/photo.htm sepp-maier#
to search for photos of Sepp Maier.
```

This convention is an exception here, because the feature vector's representation in the search string is language based. There is the difficulty that often different words are used for the same object. For example one may search for photos of "sepp-meier" or "josef-maier" or "joseph-maier" etc. Nevertheless this additional possibility can be useful in many cases.

It should be mentioned here that the feature vector can also represent the position within a semantic tree, or choices of text modules within structured text.

5.1.7 VRDs with zero length

It is also possible to define VRDs whose feature vectors have zero length. This can be used to associate terms, which are described in the associated definition (Fig. 2), to URLs. If for example a VRD contains

```
<VRD_name>dn/face/topview.htm </VRD_name>
...
<URLa>john.jpg</URLa>
```

and the file "patdef.xml" contains the record

```
<VRD_searchpar>
  <VRD_name>dn/face/topview.htm </VRD_name>
  <fvlen>0</fvlen>
  <cmode>1</cmode> //not used
  <URLde>topviewdef.htm</URLde>
</VRD_searchpar>
```

and the file "topviewdef.htm" explains that the VSI "face/topview.htm" denotes the frontal photo of a human face with 10° angle from the top, then we could assume that the file "john.jpg" is such a photo.

5.2 Medium-length and long VRDs

5.2.1 Melodies

The file "patdef.xml":

```
<VRD_searchpar>
  <VRD_name>dn/melody/voicel.htm </VRD_name>
  <fvlen>-1</fvlen>
```

```

<cmode>1</cmode>
<URLde>voicedef.htm</URLde>
</VRD_searchpar>

```

The file "voicedef.htm" contains a description which explains that the "melody/voicel.htm" VRD contains relative durations and frequency of tones. This description must also define the structure of the associated VRD. An example of this is:

```

<VRD>
<VRD_name>dn/melody/voicel.htm </VRD_name>
<URLa>PicOfExp1.mp3</URLa>
<URLa>promenadel.mp3</URLa>
<feature_vector>
... // the components of the feature vector
</feature_vector>
</VRD>

```

We can define the feature vector as a sequence of numbers $fv[0]...fv[p-1]$, in which is:

```

fv[0] = frequency of 2nd note divided by
        reference frequency4
fv[1] = duration of 2nd note divided by
        reference duration
...
fv[2j-2] = freq. of (j+1)nd note divided by
           reference frequency
fv[2j-1] = duration of (j+1)nd note divided by
           reference duration
...
until j=p/2;

```

If for a search only the first part of a melody is given, only the first part of the feature vector is compared. Similarly it is possible to search only for the rhythm. If weighting of frequency quotients should be enlarged resp. shortened (relatively to duration quotients), this is possible by multiplying them by a number greater than 1 resp. smaller than 1. The VSI owner of "dn/melody.htm" may invest some money for building software which converts keyboard input into sounds and perhaps later invest into development of a small USB-piano.

VRDs with medium length are also interesting for detailed classifications or measurements in science and medicine (examples: diagnostic data in dependence of time, e.g. temperature chart; see also [26]).

5.2.2 Complex VRDs, nontrivial software example and resulting output

Comparison of complex data like pictures (e.g. fingerprints) and sounds is usually done after feature extraction. An appropriate transformation of the original data is often the first step in the feature extraction process. For example in case of heart sounds (Figure 5) a wavelet transformation [23] allows analysis of the signal at different scales and times. After some further arithmetic the resulting coefficients (Figure 6) can be used for building a searchable feature vector which is directly comparable using a concise distance function.

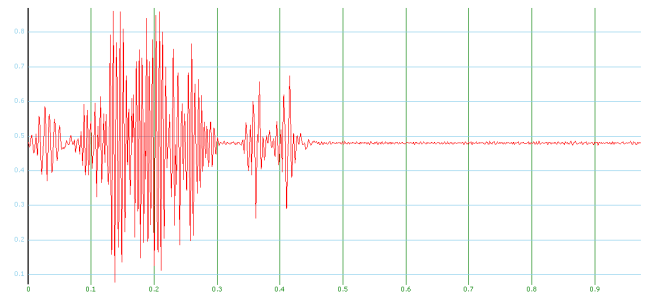


Figure 5. A heart sound in case of pulmonary valve stenosis; vertical axis: relative amplitude, horizontal axis: time in seconds.

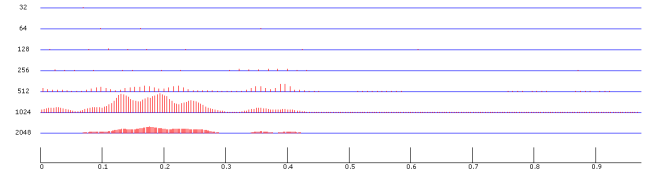


Figure 6. Smoothened norms of the transformation coefficients of Figure 5 for seven different scales.

Home
 © Pattern Search © Text Search © Login © Logout © Forgot Password © Change Password © New Account © Edit Account © Upload Publ © Config © Userlist © Options © Help

Here you can upload your pattern containing zip file to search for publications with similar patterns.

Durchsuchen

Results to 2006_01_29_pulmsten.ZIP :

[/p/6/2006_01_29/index.htm](#)
 title: pulmsten_6: a heartsound in case of pulmonary valve stenosis
 userid: author_userid_pulmsten_6
 comment: example of comment pulmsten_6
 keywords: keyword_1_pulmsten_6; keyword_2_pulmsten_6; keyword_diagnosis_pulmsten_6;
 sv: here optional text for searchview: searchviewtxt_pulmsten_6
 d: 2.9105E-07

[/p/1/2006_02_19_3/index.htm](#)
 title: hs_aoinsuff: also a heartsound in case of aortic insufficiency
 userid: author_userid_hs_aoinsuff
 comment: example of comment hs_aoinsuff
 keywords: keyword_1_hs_aoinsuff; keyword_2_hs_aoinsuff; keyword_diagnosis_hs_aoinsuff;
 sv: here optional text for searchview: searchviewtxt_hs_aoinsuff
 d: 0.23125

[/p/1/2006_01_17/index.htm](#)
 title: aosten: a heartsound in case of aortic valve stenosis
 userid: author_userid_aosten
 comment: example of comment aosten
 keywords: keyword_1_aosten; keyword_2_aosten; keyword_diagnosis_aosten;
 abs: text of abstract aosten ...
 d: 0.29098

Figure 7. Exemplary output of our software prototype which performs similarity search of heart sounds. The uploaded VRD represents the heart sound in case of pulmonary valve stenosis after wavelet transformation as shown in figure 6. Links to articles with most similar stored VRDs are listened first. The links are accompanied by structured information for test purposes. The distance d quantifies the deviation from the uploaded VRD.

As shown in figure 7, similarity search works also in case of complex VRDs. This can be done by one and the same search engine. But the work for VRD definition and VRD generation has to be shared, remembering the variety of useful VRD structures. This is possible using the URL convention 2.1.2.

6. Acknowledgements

Many thanks to Gunar Fiedler, Axel Boldt, Daniel Keim, Hans-Peter Kriegel and all others who encouraged us and who provided useful and helpful advices.

7. Conclusions

⁴ reference frequency: e.g. mean frequency of the first 4 notes; reference duration: e.g. mean duration of the first 4 notes.

Conventional language based web search requires exact matching. VRD search, however, has to find to a given VRD not only identical, but also the most similar VRDs. Quantification of similarity is done by direct comparison of the VRDs' feature vectors using a short distance function. Identification of the VRDs is done by a unique VSI - only VRDs with the same name are comparable. It is necessary to share the work connected with VRD development (connected with optimal representation of original data by feature vectors). Therefore we suggest VSIs which are simultaneously web addresses (http URLs), so associated information (e.g. definition, metrics for comparison, templates for human readable representation) is at once accessible, the work for VRD design is automatically world wide shared among all domain name owners, and VRDs are linked open data. Appropriately defined VRDs are directly comparable and with this searchable. As a consequence it is possible to search for an increasing number of quantifiable objects which are represented by well-defined VRDs on the web. In the search result the URLs of those VRDs with the same VSI and smallest deviation from the search VRD have highest rank. If wished, this can be combined with conventional text search.

Besides search there are additional possibilities because the VRDs are machine-readable like variables of a computer program. They could be evaluated directly by computer software, e.g. for statistics, modeling - it would exceed the scope of this article to deepen this here.

Direct usage of well-defined VRDs can facilitate efficient communication.

8. References

- [1] Ankolekar A., M. Krötzsch, T. Tran, D. Vrandečić. The Two Cultures: Mashing up Web 2.0 and the Semantic Web. In Proc. 16th International World Wide Web Conference 2007, ACM 825-834.
- [2] Arasu A., H. Garcia-Molina. Extracting structured data from web pages. In Proc. ACM SIGMOD 2003, 337-348.
- [3] Beckmann N., H.-P. Kriegel, R. Schneider, B. Seeger. The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. SIGMOD Conference (1990), 322-331.
- [4] Berchtold S., C. Böhm, H.-P. Kriegel. The Pyramid-Tree: Breaking the Curse of Dimensionality. SIGMOD Conference (1998), 142-153.
- [5] Berchtold S., D. A. Keim, H.-P. Kriegel. The X-tree: An Index Structure for High-Dimensional Data. VLDB (1996), 28-39.
- [6] Berners-Lee T., W. Hall, J. Hendler, N. Shadbolt, D. J. Weitzner. Creating a science of the web. Science 313 (2006), 769-771.
- [7] Bizer, C., R. Cyganiak, T. Heath. How to Publish Linked Data on the Web. <http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/>, viewed 2009-05-27.
- [8] Böhm C., B. Braunmüller, M. Breunig, H.P. Kriegel. High Performance Clustering Based on the Similarity Join. In Proc. of the 2000 ACM CIKM International Conference on Information and Knowledge Management (2000), 298-305.
- [9] Chaudhuri S., V. Ganti, R. Kaushik. A Primitive Operator for Similarity Joins in Data Cleaning. In Proc. of the 22nd Int'l Conf on Data Engineering (2006), 5.
- [10] Ciaccia P., M. Patella, P. Zezula. M-Tree: An Efficient Access Method for Similarity Search in Metric Spaces. In Proc. 23rd Int'l Conf. Very Large Data Bases (1997), 426-435.
- [11] Demartini G., I. Brunkhorst, P.A. Chirita, W. Nejdl. Ranking Categories for Web Search (Electronic Edition). In Advances in Information Retrieval , 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, Proceedings (2008), 564-569.
- [12] Ding L., T. Finin, A. Joshi, R. Pan, R.S. Cost, Y. Peng, P. Reddivari, V.C. Doshi, J. Sachs. Swoogle: A semantic web search and metadata engine. In Proc. 13th ACM Conf. on Info. & Knowledge Management, 2004.
- [13] Ehrenfeucht A. and D. Haussler. A new distance metric on strings computable in linear time. Discrete Applied Math, 40, 1988.
- [14] Fagin R., R. Kumar, D. Sivakumar. Efficient Similarity Search and Classification via Rank Aggregation. In Proc. of the 2003 ACM-SIGMOD Int'l Conf. on Management of Data (2003), 301-312.
- [15] Fodor I.K. A survey of dimension reduction techniques, US DOE Office of Scientific and Technical Information, 2002
- [16] Gionis A., P. Indyk, R. Motwani. Similarity Search in High Dimensions via Hashing. In Proc. of the 25th Int'l Conf. on Very Large Data Bases (1999), 518-529.
- [17] Haveliwala T. H., A. Gionis, D. Klein, and P. Indyk. Evaluating strategies for similarity search on the Web. In Proc. 11th International World Wide Web Conference 2002, ACM 432-442.
- [18] Henzinger M.R., R. Motwani, and C. Silverstein. Challenges in Web Search Engines. Proc. ACM Special Interest Group on Information Retrieval Forum, ACM Press (2002), 11-22.
- [19] Indyk P., R. Motwani. Approximate Nearest Neighbors: Towards Removing the Curse of

- Dimensionality. In Proc. of the 30th Symposium on the Theory of Computing (1998), 604-613.
- [20] Internet Corporation for Assigned Names and Numbers, <http://www.icann.org/>
- [21] Internet Archive Wayback Machine <http://www.archive.org/index.php>
- [22] Joshi S., N. Agrawal, R. Krishnapuram, and S. Negi. A bag of paths model for measuring structural similarity in Web documents. In Proc. 9th ACM Intl. Conf. on Knowledge Discovery and Data Mining (SIGKDD 2003), 577-582.
- [23] Liang, H., L. Hartimo. A heart sound feature extraction algorithm based on wavelet decomposition and reconstruction. In Proc. of the 20th Annual International Conference of the IEEE-EMBS (1998), 1539-1542.
- [24] Lin T.C., T.R. Reed. Heart Sound Segmentation for Computer-Aided Auscultation Proc Signal and Image Processing (2005), 122-127.
- [25] Manber U. Finding similar files in a large file system. In Proc. of the 1994 USENIX Conference, May 1994.
- [26] Orthuber W., G. Fiedler, M. Kattan, T. Sommer, H. Fischer-Brandies. Design of a global medical database which is searchable by human diagnostic VRDs. The Open Medical Informatics Journal 2 (2008), 21-32.
- [27] Page L., S. Brin, R. Motwani, T. Winograd. The page rank citation ranking: Bringing order to the web. Technical report, Stanford University, 1999.
- [28] Salton G., A. Wong, and C. S. Yang. A vector space model for automatic indexing. Communication of ACM, 18(11) 1975, 613-620.
- [29] Sarawagi S., A. Kirpal. Efficient Set Joins on Similarity Predicates. In Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (2004), 743-754.
- [30] Schutyser F., A. C. Hausamen, J. E. Swennen. Three-dimensional cephalometry, G. R. J. A color atlas and manual. Springer, Berlin Heidelberg New York, 2006.
- [31] Spertus E., M. Sahami, O. Buyukkocuten. Evaluating Similarity Measures: A Large Scale Study in the Orkut Social Network. In Proc. of the 11th ACM-SIGKDD Int'l Conf. on Knowledge Discovery in Data Mining (2005), 678-684.
- [32] W3C. URIs, URLs, and URNs: Clarifications and Recommendations 1.0. <http://www.w3.org/TR/uri-clarification/>, viewed 2009-05-23.
- [33] W3C. RDF/XML Syntax Specification (Revised); Recommendation 10 February 2004. <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/>, viewed 2008-08-12.
- [34] W3C. SPARQL Query Language for RDF; W3C Recommendation 15 January 2008. <http://www.w3.org/TR/rdf-sparql-query/>, viewed 2009-06-03
- [35] W3C: XQuery 1.0: An XML Query Language; W3C Recommendation 23 January 2007. <http://www.w3.org/TR/xquery/>, viewed 2009-06-03.
- [36] Weber R. Similarity Search in High-Dimensional Vector Spaces. Theses to data bases and information systems, Vol. 74, ISBN 3-89838-474-8, 239 pages, 2001.
- [37] Zheng Y., L. Liu, L. Wang, X. Xie. Learning Transportation Mode from Raw GPS Data for Geographic Applications on the Web. In Proc. 17th International World Wide Web Conference 2008, ACM 247-256.

Addendum

Nomenclature since 10. October 2009:

Title "Numeric web search" changed to "Vectorial web search"

in next version:

"VRD" changed to "Vectorial Resource Descriptor" (VRD)

"VSI" changed to "Vector Space Identifier" (VSI)

"URLa" changed to "Identifier of quantifiable resource" (QRI); one QRI obligatory
VRD, VSI, QRI are all HTTP URIs

Linking file "LIF" Name changed to "Vector Space Descriptor" (VSD)

all sources in RDF now, please contact me in case of interest