Benchmarking of Multimodal Time Series

An Interoperable Evaluation System for Machine Learning Algorithms

Martin Ebner

Benchmarking of Multimodal Time Series

An Interoperable Evaluation System for Machine Learning Algorithms

Master's Thesis

at

Graz University of Technology

submitted by

Martin Ebner

Institute for Theoretical Computer Science,
Graz University of Technology
A-8010 Graz, Austria

March 16th 2006

Advisor: o.Univ.-Prof.Dr.Wolfgang Maass

TUG Graz University of Technology
Erzherzog-Johann-University

Benchmarking von multimodalen Zeitreihen

Ein interoperables Evaluierungssystem für Algorithmen des maschinellen Lernens

Diplomarbeit

an der

Technischen Universiät Graz

vorgelegt von

Martin Ebner

Institut für Grundlagen der Informationsverarbeitung.
Technische Universität Graz
A-8010 Graz

16. März 2006

Diese Arbeit ist in englischer Sprache verfasst.

Betreuer: o.Univ.-Prof.Dr.Wolfgang Maass

TUG Technische Universität Graz
Erzherzog-Johann-Universität

Abstract

This thesis describes the implementation of a platform-independent *benchmarking system* for machine learning and data mining algorithms. The purpose of such a system is benchmarking distributed systems of *feature* extraction units, and providing the according datasets. Results are *visualized* on a dynamic web page. Input can be given by loosely coupled web services. So *usability* and algorithmic *awareness* are improved. A review of the underlying technology is given.
The state of the art in benchmarking methods, performance measures and benchmarking challenges is presented. Related psychological, *neurobiological* and sociological findings shed light onto the nature of *spacetime trajectories* and *multimodality*.
Dynamic time warping (DTW) and other algorithms capable of *classifying time series* are explored. DTW is implemented and evaluated on a number of test sets using the *benchmarking system*.

Kurzfassung

Diese Diplomarbeit beschreibt die Implementierung eines plattformunabhängigen Benchmarking Systems für Algorithmen des maschinellen Lernens und des Data Mining. Der Zweck eines solchen Systems ist das Erstellen von Vergleichstests von Systemen, die aus Featureextraktionsmodulen bestehen, sowie das Bereitstellen der zugehörigen Testdaten. Die Ergebnesse werden auf einer dynamischen Webseite visualisiert. Die Dateneingabe erfolgt durch loose gekoppelte Webservices. So werden Verwendbarkeit und das Bewustsein für die Algorithmen erhöht. Ein Überblick der zugrundeliegenden Technologie wird geboten.

Aktuelle Benchmarking Methoden, Performanceevaluierung und Benchmarkingwettbewerbe werden präsentiert. Ergebnisse aus der Psychologie, *Neurobiologie* und Soziologie werfen neues Licht auf *Multimodalität* und *Raumzeittrajektorien*.

Dynamic Time Warping (DTW) und andere Algorithmen, mit denen *Zeitreihenklassifikation* möglich ist, werden erforscht. DTW wird implementiert und mittels des *Benchmarking Systems* auf mehreren Testsets evaluiert.

I hereby certify that the work presented in this thesis is my own and that work performed by others is appropriately cited.

Ich versichere hiermit, diese Arbeit selbständig verfasst andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient zu haben.

Acknowledgements

I want to thank all people who gave me ideas through discussion and social context. (mental image entanglement;) Especially i want to thank Prof. Maass, for giving insight on so many levels, Andreas Juffinger for sharing his experience, and Michael Granitzer and all other members of the mistral project for building a shared knowledge.

Finally, i would like to thank my friends and spiritual guides to ground me, and remind me of what is really important.

Martin Ebner
Graz, Austria, March 2006

Contents

List of Figures

List of Tables

Chapter 1

Introduction

" Zeit ist das, was man an der Uhr abliest. "

[Albert Einstein]

" ...wenn die Zeit das Bewusstsein erreicht, ist sie schon vergangen. "

[anonym]

1.1 Motivation

Benchmarking datasets have been important in the development of machine learning algorithms. A prominent example is the UCI machine learning repository of the university of Irvine. The attributes of such a dataset consist of time invariant measured properties of a certain object. An example is the Iris flower dataset, which has 2 attributes, notably the width and the height of the blossom. Now, there are three types of Iris, name it IrisA, IrisB, and IrisC. In machine learning this is called a classification problem, with an input dimensionality of 2, and 3 classes. The algorithm learns the mapping between the 2-dimensional real valued input space, and the discrete class label with cardinality 3. (i.e. 1,2,3) This mapping is called hypothesis. After learning, the hypothesis can classify new, previously unseen data. Of course these machine classifications are not perfect. They have an error rate, which is defined precisely. So, different algorithms have different error rates. One can say "algorithm x" has the best performance on dataset Iris, if its error rate is smaller than all other algorithm's error rates. More than that, algorithms perform differently on different datasets. So it's good to know the algorithm's performance on a number of datasets.

That was all before **time** came into play. Even the width and height of an Iris flower is a function of time. Plants have a genetic program, and an environment. Blossoms come into existence and fade away. They also have an intrinsic biological function.

The problem mentioned above now has an input space of 2xT dimensionality, where T is the length of the time series. It can also be seen as a 2 dimensional trajectory with length T. Unraveling the temporal structure and biological rhythms, classification can be done more accurate. Repeated temporal motifs are an intrinsic source of information.

Time series benchmarks need *larger datasets*, more *flexible data structuring*, and *appropriate performance measures*.

This three requirements are incorporated into the ”Benchmarking System”, which is presented in this diploma thesis.

1.2 Walkthrough

This thesis describes the theoretical foundations, the benchmarking system and its applications.

Next chapter, Chapter 2 presents the current state of the art in benchmarking, and analyzes existing work.

Chapter 3 explores the field of multimodal analysis. Technical aspects are contrasted to psychological and neurocomputational viewpoints.

Chapter 4 descibes the different types of multimodal databases.

Chapter 5 examines the benchmarking system itself. The emphasis is on openness and extensibility.

Chapter 6 gives a brief overview of existing time series classification algorithms, and an introduction to the dynamic time warping algorithm.

Chapter 7 puts the whole process in action. Benchmarking is shown in an example time series classification algorithm.

Finally, Chapter 8 discusses current trends, describes work in progress, and outlines some ideas for future work and research.

The Appendix A presents the Mistral project. Several ideas emerged from being in this group.

Chapter 2

Benchmarking State of the Art

2.1 Introduction

Benchmarking datasets were used in hundreds of articles, and have build the ground for comparing machine learning algorithms. The University of Irvine provides a repository for datasets since many years. However, the state of the art in benchmarking time series data is not yet in a mature phase. This chapter provides a unified view on standard methods, measures and benchmarks that were conducted in the past. The emphasis is on large datasets and audiovisual meeting scenarios. Also the usage of terms, like classification and motif detection, and possible extensions are discussed.

2.2 Definitions

Data Set In the traditional sense a data set consists of a list of examples. Each example consists of:

1. The feature vector, which is the input time series or trajectory.
2. A label (observation), which can be:
   (a) An integer valued class label, in the case of *classification*
   (b) A real number, in the case of *regression*
   (c) A time series of real values or class labels, in the case of *dynamic or strong classification* / regression. This time series has usually the same length or an integer fraction of the input trajectory length. (i.e. fixed rate action recognition) In the case of *prediction* the time index of the labels are future observations.
   (d) A list of segments or start and end times, in the case of segmentation. The segments must not overlap and cover the whole input time span. (i.e. segmentation in TRECVID)
   (e) A list of non-overlapping subsequences in the input time series (or trajectory). Frequently occuring subsequences, under a certain distance measure (with a certain threshold), are called motifs. (Chiu et al., 2003) (motif discovery in EEG, robotics,...)
   (f) Of course, other label definitions are possible. The two paradigms of fixed sampling rate streams, and sparse events coexist.

By scientific method, each example equals to a repeatable experiment. The number of examples indicates the statistical significance of the results.
First, the aquisition of the example data, measuring natural or artificial processes, must be conducted

under controlled conditions. The recording equipment must be defined, producing time series or trajectories. Also the social ”setup” must be defined, i.e. when meetings are recorded. In many datasets, the repeatability of the data aquisition is not granted. For instance, a meeting scenario will never take the same course two times.
Second, the data is observed by human assessors, who label the data. A machine learning algorithm has the purpose to find the same labels as the experimenters. Therefore the data is splited into a training set, an optional validation set and a test set.
The labeling must also be done according to a protocol. In the case of social setups, determining the aspects, which should be labeled, may be difficult. One has to look for certain aspects, which do not change from meeting to meeting, but are bound to a specific person, group, society, or to all humans. Such aspects are defined by the ongoing research in psychology and sociology.

Training Set The training set includes both data and labels. It is used to train the machine learning algorithm, which learns a hypothesis from the data.

Validation Set The validation set includes both data and labels. It is used to validate the generalisation capability of the hypothesis (its performance on previously unseen data).

Test Set The test set consists only of the data. (from the algorithm view) The algorithm never gets the test set labels. It produces an output, which is then compared independently to the test set labels.

2.3 Performance Measures

Performance Measures are important for both machine learning and data mining. Several standard performance measures (TREC, 2001) have emerged during the last decades. The terms must be differentiated and clearly defined.

2.3.1 Slots

Insertions, deletions, and substitutions have been first stated by (Levenshtein, 1966). According to the slot hypothesis (Makhoul et al., 1999) a reference data structure is aligned to a hypothesis data structure. The atomic elements of the data structures are called ”slots”. The elements of a fixed length feature vector may be slots. Time slots may be defined by the sampling rate. Space slots may be defined by physical location or location within a feature space. (where the class decision boundaries separate the slots)

Figure 2.1 shows all possible outcomes, when a reference set N is compared to a hypothesis set M. Machine learning classically only views corresponding elements, namely the set C of correct elements, the set S of incorrect elements. The figure shows the division into two classes pos and neg. The confusion matrix is For the general N-class problem, $C \cup S$ would be divided into the NxN confusion matrix, with all possible (mis)classifications. C is the sum of confusion matrix' main diagonal and S is the sum of all other confusion matrix elements. The deletions D of reference slots are ”unknown” to the hypothesis. The insertions I of hypothesis slots ”unknown” to the reference, into account. (see also (Milch et al., 2005))

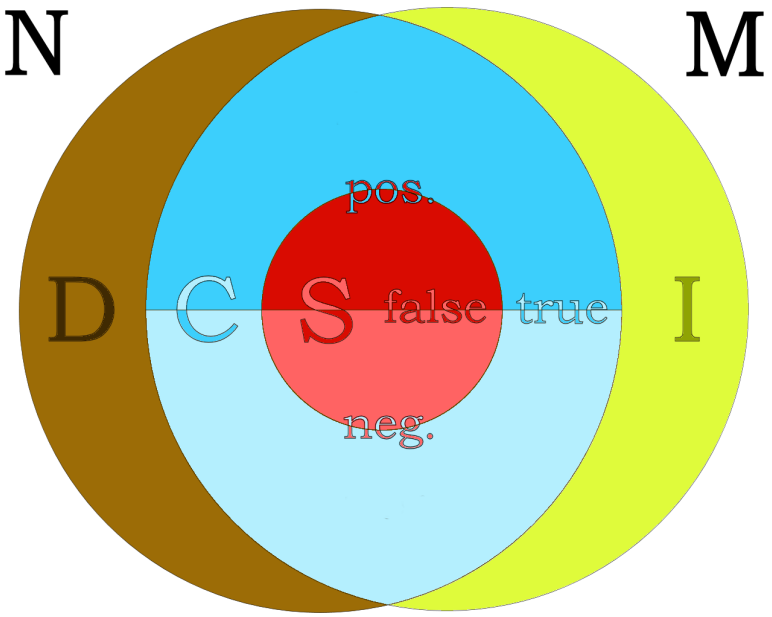

Figure 2.1: Reference set N and hypothesis set N. C are the correct, corresponding slots, where the hypothesis is true. S are the substitutions, corresponding slots, where the hypothesis is false. pos and neg are the classes of a two class problem. D are the deletions, references with no corresponding hypothesis. I are the insertions, hypothesis with no corresponding reference.

2.3.2 Recall

Recall is the number of correct slots divided by the number of reference slots.

$$R = \frac{C}{N} = \frac{C}{D + S + C} \tag{2.1}$$

2.3.3 Precision

Precision is the number of correct slots divided by the number of hypothesis slots.

$$P = \frac{C}{M} = \frac{C}{I + S + C} \tag{2.2}$$

2.3.4 F-Measure

The F-Measure is a function of both recall and precision.

$$F_\beta = \frac{(\beta^2 + 1)RP}{(\beta^2 P + R)} \tag{2.3}$$

with

$$\lim_{\beta \to 0} F_\beta = P \tag{2.4}$$

and

$$\lim_{\beta \to \infty} F_\beta = R \tag{2.5}$$

The most common used F-Measure is

$$F_1 = \frac{2RP}{R+P} = \frac{2C}{M+N} = \frac{C}{\frac{I}{2} + \frac{D}{2} + S + C} \tag{2.6}$$

2.3.5 Example

Figure 2.2 shows example features of a reference set M and a hypothesis set N. This could be spatial features (2D image features or audio cepstral features) or time-series. In either case the space-time *slot*, which defines correspondence between two elements must be defined. (segmentation) The features 'A', 'B', and 'C' occur both in the reference and the hypothesis set. (thus, C=3) 'E' and 'F' are substituted by 'H' and 'J'. (thus, S=2) 'G' is deleted. (D=1) 'K' and 'L' are inserted by the hypothesis. (I=2)

$R = \frac{C}{D+S+C} = \frac{3}{1+2+3} = \frac{1}{2} = 0.5$ (3 of 6 reference elements are "recalled" correctly by the hypothesis)
$P = \frac{C}{N} = \frac{C}{I+S+C} = \frac{3}{2+2+3} = \frac{3}{7} = 0.429$ (3 of 7 hypothesis elements are correct)
$F_1 = \frac{C}{\frac{I}{2}+\frac{D}{2}+S+C} = \frac{3}{\frac{2}{2}+\frac{1}{2}+2+3} = \frac{6}{13} = 0.462$ (average measure for recall and precision)

The most widely used measures in benchmarking are recall / precision (also the precision over recall diagram, see TREC-10 Proceedings appendix on common evaluation measures).

2.3.6 Slot Error Rate

The slot error rate (SER) (Makhoul et al., 1999) consists of all possible errors normed to the reference set. It is equal to the word error rate (WER) used in speech recognition. This measure was also used for meeting event classification (Gatica-Perez et al., 2003), where it is also called event error rate.

$$WER = SER = \frac{S+D+I}{N} = \frac{D+S+I}{D+S+C} \tag{2.7}$$

2.4 Similarity Measures

Data mining uses parameterless similarity measures, such as Lp-Norm, DTW, LCSS, Chebychef-Polynomials, and Euklidian Distance to cluster, classify, and detect motifs in offline and online time series. (Chen, Lei, 2005)

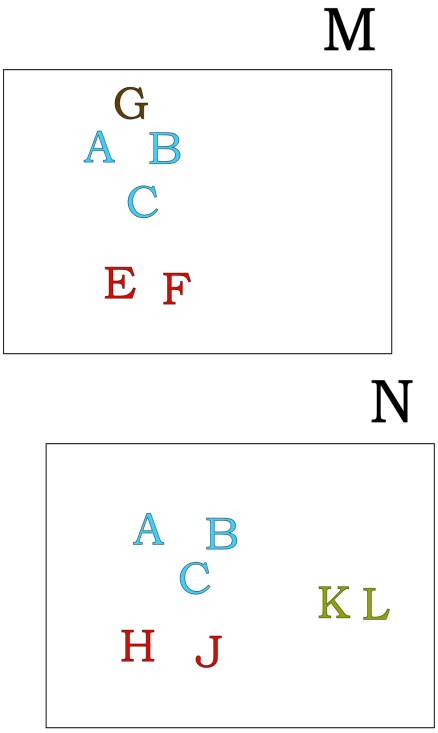

Figure 2.2: Example features of a reference set M and a hypothesis set N, with insertions, deletions, substitutions, and correct elements. This can be spatial features (2D image features or audio cepstral coefficients), time-series, or a combination of these. In either case the space-time *slot*, which defines correspondence between two elements must be defined. (segmentation) The features 'A', 'B', and 'C' occur both in the reference and the hypothesis set. (thus, C=3) 'E' and 'F' are substituted by 'H' and 'J'. (thus, S=2) 'G' is deleted. (D=1) 'K' and 'L' are inserted by the hypothesis. (I=2)

2.5 Example Benchmarking Efforts

2.5.1 TRECVID

Introduction

The TREC conference series is sponsored by the National Institute of Standards and Technology (NIST) with additional support from other U.S. government agencies. The goal of the conference series is to encourage research in information retrieval by providing a large test collection, uniform scoring procedures, and a forum for organizations interested in comparing their results. In 2001 and 2002 the TREC series sponsored a video "track" devoted to research in automatic segmentation, indexing, and content-based retrieval of digital video. Beginning in 2003, this track became an independent evaluation (TRECVID) with a 2-day workshop taking place just before TREC.

Data Corpus

Because of agreements with the data providers and the funding organization the data is available for the cost of shipping but only to TRECVID participants. Some portion of the TRECVID 2005 data may be available for purchase from the Linguistic Data Consortium.

A random sample of about 6 hours will be removed from the television news data, combined with

about 3 hours of NASA science programming, and the resulting data set used as shot boundary test data. The remaining 160 hours of television news will be split in half chronologically by source. The first halves will be combined and designated as development data for the search, high/low-level feature, and shot boundary detection tasks. The second halves will be combined and used as test data for the search and high/low-level feature tasks. Half of the BBC rush video will be designated as development data, the remainder as test data.

A description of the TRECVID benchmarking state of the art can be found under (TRECVID, 2005). The different tasks are:

- Shot Boundary Detection (time segmentation, ca. 2s per shot)
- Low Level Feature Extraction (camera motion)
- High Level Feature Extraction (semantic concepts)
- Interactive Search

Performance Measures used for all tasks are precision, recall, and average precision, which is the area under the recall-precision curve (TREC, 2001). The results of multiple runs of each system were committed to NIST and evaluated.

```
<videoFeatureExtractionResults>
-
<videoFeatureExtractionRunResult trType="A" sysId="SiriusCy1"
    priority="1" desc="This run uses the top secret x-component">
-
<videoFeatureExtractionFeatureResult fNum="01">
<item seqNum="1" shotId="shot118\_2"/>
<item seqNum="2" shotId="shot118_3"/>
<item seqNum="3" shotId="shot18_19"/>
<item seqNum="4" shotId="shot123_2"/>
<item seqNum="5" shotId="shot56_42"/>
<item seqNum="6" shotId="shot193_3"/>
<item seqNum="7" shotId="shot121_12"/>
<item seqNum="8" shotId="shot22_20"/>
<item seqNum="9" shotId="shot103_122"/>
<!-- ... -->
<item seqNum="2000" shotId="shot118_2"/>
</videoFeatureExtractionFeatureResult>
<!-- ... -->
-
<videoFeatureExtractionFeatureResult fNum="10">
<item seqNum="1" shotId="shot118_2"/>
<item seqNum="2" shotId="shot118_3"/>
<item seqNum="3" shotId="shot18_19"/>
<item seqNum="4" shotId="shot123_2"/>
<item seqNum="5" shotId="shot56_42"/>
<item seqNum="6" shotId="shot193_3"/>
<item seqNum="7" shotId="shot121_12"/>
<item seqNum="8" shotId="shot22_20"/>
<item seqNum="9" shotId="shot103_122"/>
<!-- ... -->
<item seqNum="2000" shotId="shot118_2"/>
</videoFeatureExtractionFeatureResult>
```

```
33 </videoFeatureExtractionRunResult>
34 -
35 <videoFeatureExtractionRunResult trType="B" sysId="SiriusCy6"
       priority="2" desc="This run does not use the x-component">
36 -
37 <videoFeatureExtractionFeatureResult fNum="01">
38 <item seqNum="1" shotId="shot118_2"/>
39 <item seqNum="2" shotId="shot118_3"/>
40 <item seqNum="3" shotId="shot18_19"/>
41 <item seqNum="4" shotId="shot123_2"/>
42 <item seqNum="5" shotId="shot56_42"/>
43 <item seqNum="6" shotId="shot193_3"/>
44 <item seqNum="7" shotId="shot121_12"/>
45 <item seqNum="8" shotId="shot22_20"/>
46 <item seqNum="9" shotId="shot103_122"/>
47 <!-- ... -->
48 <item seqNum="2000" shotId="shot118_2"/>
49 </videoFeatureExtractionFeatureResult>
50 <!-- ... -->
51 -
52 <videoFeatureExtractionFeatureResult fNum="10">
53 <item seqNum="1" shotId="shot118_2"/>
54 <item seqNum="2" shotId="shot118_3"/>
55 <item seqNum="3" shotId="shot18_19"/>
56 <item seqNum="4" shotId="shot123_2"/>
57 <item seqNum="5" shotId="shot56_42"/>
58 <item seqNum="6" shotId="shot193_3"/>
59 <item seqNum="7" shotId="shot121_12"/>
60 <item seqNum="8" shotId="shot22_20"/>
61 <item seqNum="9" shotId="shot103_122"/>
62 <!-- ... -->
63 <item seqNum="2000" shotId="shot118_2"/>
64 </videoFeatureExtractionFeatureResult>
65 </videoFeatureExtractionRunResult>
66 </videoFeatureExtractionResults>
```

Listing 2.1: Example of TRECVID video feature extraction results for two runs

2.5.2 M4

See Appendix A.

2.5.3 MUSCLE WP3

See Appendix A.

2.5.4 NIPS Eye Challenge Benchmark

See (Pfeiffer et al., 2005).

2.5.5 CATS Time Series Prediction Benchmark

See (CATS, 2004). This is a time series prediction benchmark with missing subsequences, which have to be predicted. The test data was published after submission of the predicted subsequences by each group. The performance measure used is the mean squared error (MSE). Each method is described in a paper. The results are ranked by the MSE on the web page.

2.5.6 Comparison

Introduction

All the above mentioned projects try to extract features from multimodal data corpuses. Which features are extracted depends on the domain, and the current state of the art in the multimodal information retrieval. Performance Measures are important for observing and controlling the development (or evolution) of an algorithm consciously. Such measures shall represent a good objective estimation of the users subjective experience of system performance, like the recall and precision of a web search or the word error rate of a speech to text system. To point out the centering on the users subjective experience, here is another example: NIST uses several human accessors to measure high level features and search performance. They have a time limit to conduct a certain task. This reduces the bias introduced by different human assessors.

Data Corpus

Mostly, data is divided into a public and a private part. Data is available for download, or shipped by DVDs or loaned harddisks. Data is divided into development (training,validation) data and test data (on which the benchmark is defined) Data is provided with metadata and annotation.

Benchmarking System

Benchmarking basically performs a comparison between extracted information and annotated information. So different retrieval systems can be compared objectively. The performance measure most useful for the domain is chosen. The extracted information (the benchmarking system's input) is generally specified as XML document. There are certain standards how to conduct a benchmark. The current state of the art is semi-automatically benchmarking.

Performance Measures

Granularity is certainly an issue. TRECVID/(MUSCLE) uses shots as video primitives, analogous to cuts in films. Features and Search Topics are defined as a boolean value on each shot. Event classification is usually conducted by segmentation and classification of these bins. TRECVID uses a 2 step process, whereas M4 uses combined segmentation and classification approaches. Both use fixed sampling rate input data. None uses sparse events yet.

2.6 Conclusion

Standard Benchmarking Efforts for large, multimodal timeseries data are restricted and clearly defined. The input data is provided with fixed sampling rate, fixed amount of features. The labels are provided in XML, using predefined classes. The labels usually denote the class of a certain time segment, which is either predefined or learned. Common evaluation measures are recall, precision, F_1 and word error rate.

Sparse events and multi valued objects are not yet standard representations. Data is usually shipped or preprocessed and downloaded.

Chapter 3

Forms of Multimodal Analysis

" All human beings and things depend on their causes and parts, and cannot exist independent of these. They all arise in dependance; consequently they are empty of inherent existence. Because all phenomena have arisen in dependancy, they are empty by nature. "

[XIV. Dalai Lama, explaining the heart sutra's "form is emptiness" statement.]

3.1 Introduction

This chapter presents some psychological effects regarding multimodality, which is related to regions of the brain integrating multiple modalities. Next the information theoretical problem of low level versus high level features is discussed, which is related to early sensory versus non-sensory brain regions. Next, the paradigm of socially sensitive computing is presented.

3.2 Definitions

Multimodal Concerning multiple senses and modes of interaction.

Stimulus In physiology, a stimulus is something external that elicits or influences a physiological or psychological activity or response. In psychology, anything effectively impinging upon any of the sensory apparatuses of a living organism, including physical phenomena both internal and external to the body. In other fields, a stimulus is anything that may have an impact on a system; an input to the system.

Social Network Analysis "A social network is a social structure between actors, mostly individuals or organizations. It indicates the ways in which they are connected through various social familiarities ranging from casual acquaintance to close familial bonds. The term was first coined in 1954 by J. A. Barnes (in: Class and Committees in a Norwegian Island Parish, "Human Relations").

Social network analysis (also sometimes called network theory) has emerged as a key technique in modern sociology, anthropology, Social Psychology and organizational studies, as well as a popular topic of speculation and study. Research in a number of academic fields have demonstrated that social networks operate on many levels, from families up to the level of nations, and play a critical role in determining the way problems are solved, organizations are run, and the degree to which individuals succeed in achieving their goals.

Social networking also refers to a category of Internet applications to help connect friends, business partners, or other individuals together using a variety of tools. These applications, known as online social networks are becoming increasingly popular." (wikipedia)

Recently, there is a quite large data corpus of **mobile phone data available**. It was automatically collected, with acknowledgment of the participants. The data, models, and evaluation can be found here.

3.3 Multimodal Sensory Integration in Living Systems

3.3.1 Introduction

Two major factors are important for multimodal integration of mental events. First, the recognition or what system. Second, the localisation, the where system. Both systems have temporal and spatial aspects. According to (Lewald, 2002) the localisation system can be trained by disparate audio and visual events in such a way that the offset error remains in a following pointing experiment. This suggests short-term-plasticity of auditory localisation in relation to disparate visual information. This and other psychological effects can be useful in three applications:

1. Building better audiovisual recognition and benchmarking systems
2. Building sort of a mp3 codec for multimodal objects. mp3 exploits psychoaccoustic phenomena. Why not exploit psycho-multimodal effects?
3. Building computational models to understand the multimodal brain processes.

HUMAINE From (Grandjean, 2004): "HUMAINE aims to lay the foundations for European development of systems that can register, model and/or influence human emotional and emotion-related states and processes - 'emotion-oriented systems'. Such systems may be central to future interfaces, but their conceptual underpinnings are not sufficiently advanced to be sure of their real potential or the best way to develop them. One of the reasons is that relevant knowledge is dispersed across many disciplines. HUMAINE brings together leading experts from the key disciplines in a programme designed to achieve intellectual integration. It identifies six thematic areas that cut across traditional groupings and offer a framework for an appropriate division of labour - theory of emotion; signal/sign interfaces; the structure of emotionally coloured interactions; emotion in cognition and action; emotion in communication and persuasion; and usability of emotion-oriented systems. Teams linked to each area will run a workshop in it and carry out joint research to define an exemplar embodying guiding principles for future work in their area. Cutting across these are plenary sessions where teams from all areas report; activities to create necessary infrastructure (databases recognising cultural and gender diversity, an ethical framework, an electronic portal); and output to the wider community in the form of a handbook and recommendations of good practice (as precursors to formal standards)."

McGurk Effect According to Lawrence Rosenblum: "The Mc Gurk effect (McGurk and MacDonald, 1976) demonstrates of how we all use visual speech information. The effect shows that we can't help but integrate visual speech into what we *hear*. It shows that visual articulatory information is integrated into our perception of speech automatically and unconsciously. The syllable that we perceive depends on the strength of the auditory and visual information, and whether some compromise can be achieved.". See Figure 3.1.

Ventriloquism Effect The Ventroliquism effect is shown in Figure 3.2.

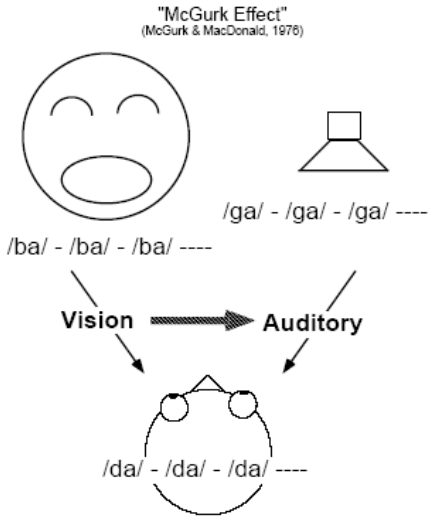

Figure 3.1: McGurk effect. Conflicting visual and auditory information is integrated. The person perceives ”da da da”, while he physically hears ”ba ba ba” and sees ”ga ga ga”. Test the experiment video yourself.

Ventriloquism Aftereffect The so-called ventriloquism aftereffect is a remarkable example of rapid adaptive changes in spatial localization caused by visual stimuli. After exposure to a consistent spatial disparity of auditory and visual stimuli, localization of sound sources is systematically shifted to correct for the deviation of the sound from visual positions during the previous adaptation period. **Short-term plasticity** is supposed to play a role in this process. (Lewald, 2002)

Bounce-Inducing Effect The Bounce-Inducing effect is shown in Figure 3.3.

3.3.2 Localisation

Events are localized at the same position in certain space and time windows. (Thurlow and Jack, 1973) revealed in experiments with humans a maximum divergence of 10% and a maximum asynchronity of 200ms. Other studies come to differing, but similar results, based on the experimental setup. (Lewald et al., 2001) yields 3°/100ms.

3.3.3 Recognition

The time windows for object identity are around 40°/400ms. This is similar to the limits of the McGurk and Bounce-Induction effects. (Watanabe, 2001)

A possible neural correlate of multimodal integration is the superior colliculus.

3.3.4 Superior colliculus

According to experiments of Stein et al. in anesthetized (Stein and Meredith, 1993) and alert (Wallace et al., 1998) cats, 55% of the neurons in superior colliculus had higher excitation, when a multimodal stimulus was presented.

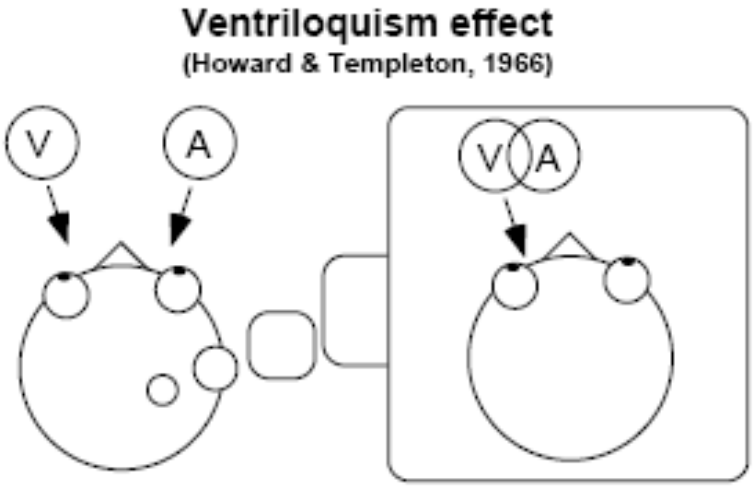

Figure 3.2: Ventriloquism effect. When the auditory and visual events are at the same time, the audio location is perceived to be attracted towards the visual location.

Spatial Rule The responses of neurons are higher when multimodal spatial stimuli occur compared to unimodal stimulus or the sum of unimodal stimuli. When the spatial occurrence of stimuli are disparate these neurons do not discharge or show a decrease of spontaneous activity.

Temporal Rule Time seems to be less important than space for the multimodal neurons. The maximum of responses is achieved, when the stimuli overlap in time.

Rule of inverse efficiacy When two stimuli are spatially near and temporally synchronized, the response of multimodal neurons is superior to the maximum of the unimodal response. The lower unimodal responses are the higher multimodal response is. (Some kind of tunedness to multimodality)

Multimodal integration can yield both detection and recognition of events. These two processes can modulate the attention process. Shedding the light of awareness onto events means orienting resources. Recognition increases and response time decreases, when the stimulus is multimodal.

The influence and process structure of time in multimodal brain research is still unclear.

3.3.5 Unity

In (Wallace et al., 2004) auditory and visual stimuli are presented to the human participants with different disparities (0°,5°,10°,15°; 200ms,500ms,800ms). If two modalities are perceived as unified, they are reported to be at the same location, regardless of their disparity.(Figure 3.4) Otherwise, if they are perceived as not unified, there is an offset between auditory and visual localisation, (Figure 3.5) and the variance is also greater.(Figure 3.6) Interestingly the variance has its maximum at zero disparity, whereas in the unified case, the variance has its minimum at zero disparity.
This minimal variance can be interpreted as the saliency of the event. When the auditory and visual events are *inferred* from the same source, the auditory localisation is *attracted* by the visual localisation. If the two events cannot be *inferred* from a common source, the auditory localisation is *repelled* from the visual localisation.

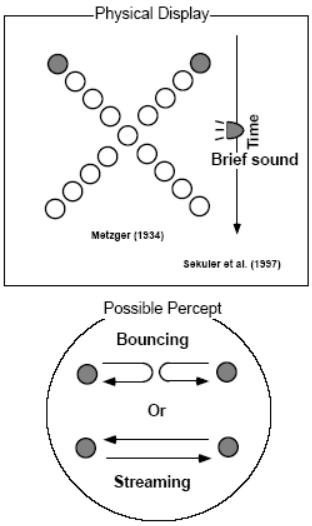

Figure 3.3: Bounce-Inducing Effect. There are two conflicting interpretations of the visual event. The balls can cross or bounce. If in addition a auditory event is played at the critical time instance, then the perception is biased toward bouncing.

3.4 Semantic Gap

" Rules are for obedience of fools and for guidance of wise men. "

[Douglas Bader (1910 - 1982)]

This section gives a short overview of the term "semantic gap" used in computer science. Between low-level machine features like words, auditory, and visual features on the one hand and high level semantics used by humans there exists a gap, the so-called *semantic gap*. This term also relates to the generalisation problem in machine learning. For the mistral project this means, that after the feature extraction of the different modalities, there are just low-level features, which must somehow be integrated by machine learning and semantically enriched by rule-based machine reasoning, in order to derive better, more human understandable semantic concepts. Concepts like "indoor", or "participant is attending", or even "chair" (see Figure 3.7) can never be determined unambiguously. In principle this is shown in (Addis et al., 2005).

NOE K-Space The IST Network of Excellence *K-Space* (The Knowledge Space of Technology to Bridge the Semantic Gap) deals with feature extraction, building MPEG-7 representations, and inferring semantic knowledge to reduce the semantic gap.

3.5 Socially Sensitive Computing

Irrational Set "An irrational set is where no finite set of rules can be constructed that can include unambiguously any member of that set, and at the same time, unambiguously exclude any non-member of that set." (Addis et al., 2005)

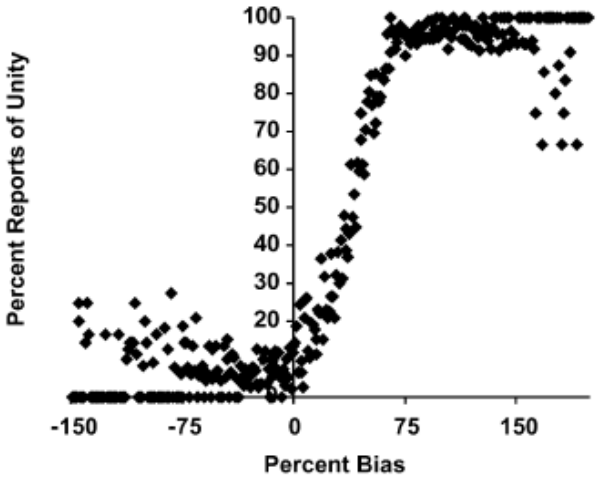

Figure 3.4: Correlation of perceived unity of multimodal events and localisation bias. Plotted are percent of unity reports for each 1% bin of localisation bias. If unity is reported, the bias tends to be 100%. (same location) If no unity is reported, bias is zero or negative.

Irrational sets can lead to decreasing performance of a machine learning algorithm, although the amount of training examples is increasing.

Human judgment is still necessary, because what is an acceptable behavior or performance depends on time and social context. There must exist a purpose or goal of a person, which changes the set boundaries or hypothesis dynamically. This paradigm shift in computer science is based upon Wittgenstein and solved by the *abductive loop*.
There are two interpretations of a program: One from the bits up to the program. This computation is a rational set. The other one is from the program up to the problem domain. It connects real world objects (persons, companies, departments, music albums) to data structures, classes. The problem is that real world objects are irrational, in a sense that they are not fully separable, and changing (like the facial features of a person undergoing an emotion). This results in a never ending cycle of human made software evolution, because the software objects and context is changing. This suggests the development of software adaptive to the intentions and purpose of persons or groups, changing the ”perceptions” of the software by user feedback. So the software would be socially sensitive by exchanging the meaning of semantic concepts by the common meaning for the given social context. (what Wittgenstein called family resemblance) (Addis et al., 2005) **Of course, the social values of such changing concepts can be abused to influence users, like media or advertisement does today. One must be aware of the ethical value connected to such concepts. In order to benefit people rather than harm them, the foundation of such ontologies must be ethical. That means the values and the intentions behind them must support well being independent of social group, company, status, nationality and race. Therefore such systems must contain an awareness of ethics - the guarantee and monitoring of beneficial effects for all humans - in their core. Only this constraint can control their destructive power.**

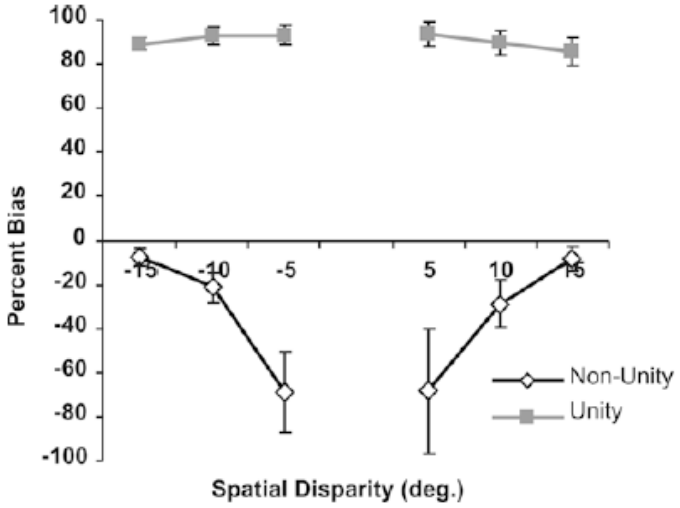

Figure 3.5: Localisation bias as a function of spatial disparity. If unity is reported (curve with gray squares), the bias is nearly 100% (perceived at same location, regardless if physically separated). If no unity is reported, the bias is near zero for large disparities, (perceived equals physical location) and grows negatively, if disparity gets smaller. (perceived at different location, regardless if disparity is small)

3.6 Conclusion and Suggestions

Results from Neurobiology and Psychology suggest that different modalities can influence each other, with the aftereffect of biased localisation trough short-term plasticity. Both localisation (the *Where* information) and recognition (the *What* information) reveal certain spacetime windows for multimodal integration. Certain brain regions like the superior colliulus are specialized for multimodal integration. There is a difference, whether a disparate audio and visual stimuli are perceived as a unified object. If they are perceived as unified, they are also perceived on the same location, despite being at different physical locations. If they are perceived as not unified, they are perceived as even further away from each other, than they really are.

This could be useful to model the human audiovisual experience, and thus yield better machine learning results for multimodal integration. On the other hand, auditory and visual systems seem to bias each other. It is difficult to imagine what this would mean for a benchmarking system, because there would be no absolute values of position any more, which can be compared.

Another aspect is the semantic gap, the gap between low-level and high-level features. This strongly relates to the binding problem. Eastern cultures long ago had the notion of the third eye or sixth sense. In current philosophy and upcoming consciousness research this is called the non-sensory fringe of consciousness. This raises the question, weather there are regions of the brain, which explicitly recognize qualities like familiarity, rightness (of the current audiovisual and somatosensory scene) in order to determine the validity of the current perception. Such a modality would be useful also for machine learning of complex scenes.

Another topic is the completely missing emotion detection in mistral project. (The humaine project explicitly deals with emotions) They certainly form the basis of human interaction, and thus of social action recognition. Although the mistral recording agent is not able to show any emotional expression (it is just perceiving), emotion recognition of the participants could be useful for modeling social interaction. At last, the paradigm of socially sensitive computing should be mentioned, which tries to build socially adaptive computer systems.

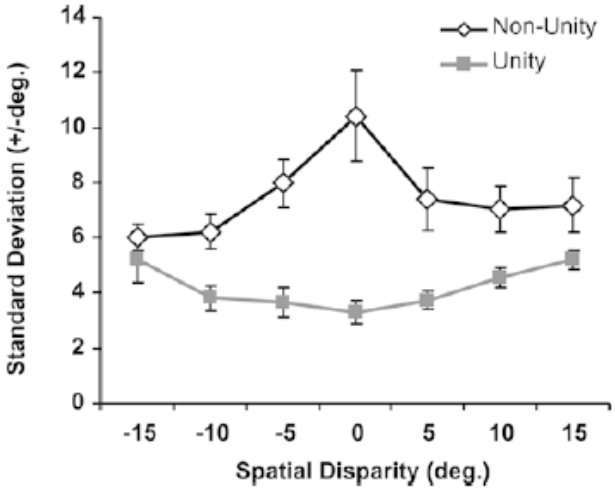

Figure 3.6: If reported as unity, standard deviation of location has its minimum at zero disparity (spatial coincidence). If reported as not unity, standard deviation of location is larger and has its maximum at zero disparity.

Of course even the current meeting scenario of mistral raises the question, what could be the use of such systems, without violating the notion of ethics.

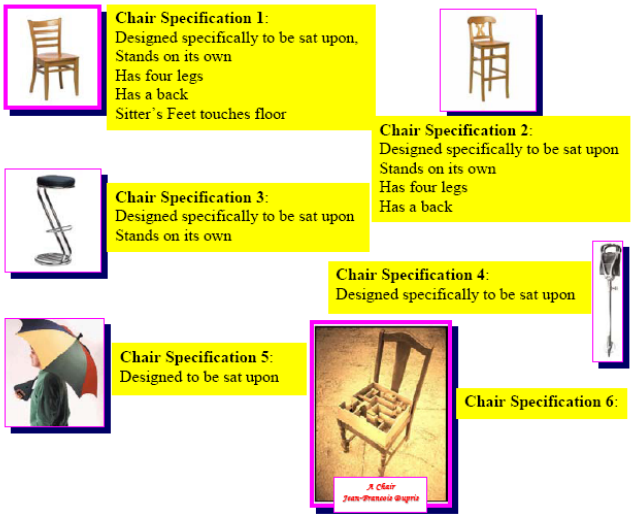

Figure 3.7: An attempt at identifying a chair using rules.

Chapter 4

Multimodal Databases and Interoperable Systems

4.1 On the problem of data storage

The problem of integrating databases and programming languages (Cook and Ibrahim, 2005) is open for about 40 years. New solutions based on Java are now in a rather mature phase. But are all the requirements for such an integration forfilled? Clearly, no. What would be an optimal solution? This can only be addressed in terms of defined properties, such as orthogonal persistence. This dilemma produces a real jungle of available and upcoming persistence solutions and standards. Globally, at least one important issue is currently being solved, by the spirit of open source. It is the evolution of programming language and platform independent mechanisms through XML.
So two parallel lines of development are co-evolving. The true spirit of open source and science, which boils down to worldwide social networks, delivers free products, which are interoperable by its social nature. Global companies easily identify with the spirit of money, which tends to grow larger and larger. They must deal with the upcoming ”integration issues”, form alliances, join standardization groups and so on.

There are different types of databases.

1. Classical RDBMS (Relational database management systems) are the most common databases today. They are structured by tables, which contain columns of different type. One row of information is a record. Each record has a unique identifier or primary key, which can also be used to link different tables. Relations are usually put in terms like: Table A has zero or more tables B. Several constraints can also be managed by the database, like referential integrity through foreign keys. Programming languages access the data through the query language SQL, which can select subsets of a table, delete, or update rows. (The SQL statement is transmitted to the database through JDBC/ODBC connectors) The basic information entity is always a record, which is a mixed-type tree of depth one.

2. In XML, each information entity is tree-like. In addition, identifiers and references are possible, so graphical models can be represented. There exists a mapping from XML to any major programming languages' data structures. The process of converting an object oriented class to a XML string is called serialization. Recently, there are several open source and commercial XML databases, which can be accessed and transformed by the new W3C standards like XQUERY and

XSLT.

3. The third approach is object/relational mapping. O/R mappers like Hibernate (Hibernate, 2005) can map from an object-oriented programming language to a classical RDBMS. This is somewhat unnatural, but meets several of the integration aspects, like mapping plus a query language. (Cook and Ibrahim, 2005)

4. The fourth and last approach are object databases, which can store and search programming languages' objects directly. The data is stored in a single file. They are nearly perfect for mobile devices, but have some limitations regarding large amount of data.

The chosen storage method for the benchmarking system combines *XML and O/R mapping*. The programming language was chosen *Java*, for its platform independence. Now, data can be stored, but how is it transmitted through the internet? There are several possible ways to do this. Classical means of transportation are e-mail(smtp), (s)ftp, http, and sockets. In the past several years a new technology, named *web services* (WS) was appearing. There were several development steps from Apache SOAP, and Axis to *Axis2*, which is now available in its basic form. (open source) The used protocol is SOAP. It separates the transportation of the data in an configurable layer. Web services are often associated with the upcoming "*semantic web*", a language based paradigm for worldwide information services, which can be looked up in service directories.
WS are used mainly for secure communication between worldwide companies now. In the commercial dimension, the future is in electronic contracts (policies) between companies, automatic discovery of markets, and data mining in general. An interesting fact is, that the open source community is developing core electronic business facilities. The open-source company JBoss for instance builds a complete environment for business applications, including the newest web servers, portal, servlet, and web services support. A highly motivating fact is also, that efforts in that direction are often maintained by people from developing countries, which have previously worked for leading international companies. Science has the same interest than developing countries, that ideas are shared. This can reduce both the north-south and the science-industry technology gap.

4.2 Java Data Mining Standard

Java Data Mining (JDM) (Kadav et al., 2003)(Hornick, 2004) is a Java API and web service definitions for data mining. The standard was specified in 2004 by the Java Community Process (JSR). Oracle, Hypernion, IBM, Sun, and BEA were involved. Simulation Tools like Matlab have rich functionality, but often have less performance and are difficult to integrate into commercial products. JDM offers to possibility to easily integrate data mining algorithms into distributed applications. Different vendors implement a core data mining functionality and parts of the optional features. Support of an optional feature can be determined also at runtime. The risk of choosing a proprietary product of a single vendor is thus reduced. If a specific feature is not supported by one vendor, other vendors using the same JDM API are available. JDM specifies functions like regression, classification, association and clustering. Specified algorithms are decision trees, neural networks, naive bayes, support vector machines, and k-means. Operations like build, test, apply, import and export are executed asynchronously. There is no unified benchmarking available in JDM. Time Series Algorithms will be supported in version 2.0, which is currently in its early specification phase. The final release of JDM 2.0 is expected not before 2006.

4.3 Variable Feature Structure

Not only data may change, also data structures (the feature types, cardinality, i.e. number of persons in a meeting) may change. (in contrast to fixed rate, fixed feature count)

- the data can be *sparse* (not all possible features may exist)
- the structure may change (schema evolution, structure learning)
- previously unknown features may occur (schema addition)

Thinking of a general MPEG-7 audiovisual scene, each object may have

- different modalities (features from different extraction units)
- a life cycle (starting to exist, existing, end of existence)
- simple, hierarchical or graph structure

Algorithms for aligning such representations still have to be explored.

4.4 Conclusion

Programming languages and databases still have no common ground. There exist solutions for mapping programming language structures to databases, in order to persist the data. (Hibernate) A trend in data exchange is XML for programming-language independent storage and Web Services for XML transportation.

Chapter 5

The Benchmarking System

5.1 Introduction

The benchmarking system is being developed within the mistral project. The main objective of the software is the benchmarking of machine learning and data mining algorithms.
In a more general view the multimodal data streams of mistral are time series or trajectories, in the multidimensional case (with dimensionality d). Also the output of any mistral algorithm is a trajectory through feature space.
Different Features are represented as different XML Structures. There are two reasons, why the data is not simply a series of double vectors of dimensionality d, or a matrix Nxd, where N is the number of points of the trajectory.
First data needs to be structured into semantic meaningful units or attributes. The application developer or user should speak the same language as the algorithms. The contents of our phenomenal consciousness are often viewed as structured and interconnected. (Dainton, 2000) For example, person A has a red T-shirt and a Nokia Handy, which is the source of a ring tone, which has a pitch trajectory Table 5.1. So objects can be part of other objects (Fergus et al., 2005) or occurring at the same time. Both kinds of structure are poured into a tree structured XML, which is the established way of communicating data between web applications, large companies, using some internet protocol.
Second, XML also allows flexible amount of objects (2, 3....n people are attending the meeting) and can even contain or link "unknown" data types, for instance specified by the 1st order probabilistic language

5.2 Machine Learning Process

The basic procedure of benchmarking a machine learning algorithm is as follows.

1. **Training**
   Train algorithm on training set (data plus labels) A part of the training set may be used as validation set, in order to find model parameters for generalisation.

2. **Cross Validation**
   Optional. The training set is split into k parts. Using one of the k parts as validation set, the algorithm is trained on the rest and evaluated on the validation set. This is done k times, the average performance is statistically more significant than for the case k=1. (k-fold cross validation, k is usually 10)

3. **Testing**
   Run algorithm on test set. This yields a set of output (XML) elements. (Hypothesis Set)

Datetime	pitch
2005-11-08T19:11.1	440Hz
2005-11-08T19:12.1	560Hz
2005-11-08T19:13.1	650Hz
2005-11-08T19:13.6	440Hz

Table 5.1: Pitch Trajectory of Mobile Phone Ring Tone

4. **Benchmarking**
   Compare the hypothesis set to the reference set. Compute performance measures.

Now the benchmarking system intercepts this procedure at certain points:

1. Only once: Get a the MATLAB, Java, C++ client interface for the benchmarking web service.

2. **GetStatus**
   Test, if the benchmarking system is available.

3. **GetTrainSet**
   Get the training data manually via web page or automatically via web service

4. **(Training)**

5. **GetTestSet**
   Get the test data manually via web page or automatically via web service

6. **(Testing)**

7. **Benchmark**
   Benchmark the output features via web service. The benchmarking result will be obtained by the centralized benchmarking system, and viewed, among with inter-algorithmic comparisons, on the benchmarking dynamical web page (within a few seconds).

This procedure delivers automated benchmarking and visualisation, which is beneficial, when a number of algorithms have to be tested on a number of test sets. Note that GetTrainSet and GetTestSet are not yet specified and implemented.

5.3 Use Case

A use case would be for instance: Alex is member of the cognitive xxx/mistral project. His new version 0.97 of the ”magic algo” is ready now. It’s friday afternoon. Utilizing the limited matlab licenses of the institute, he starts the benchmarking of his algorithm on a really huge dataset. Then he leaves into weekend. On Saturday afternoon - no, he does not get a notification message yet - he looks at the benchmarking site and, - yes - his algorithm is outperforming Bernadette’s ”super trooper algo” (she is working at another university actually) by 3 percent. It’ goin’ to be really fun tonight!

5.4 Definitions

- **Feature Type**
  A feature type (or just feature) specifies the type or structure of information. (i.e. Event, TimeSeriesPointFeature, Word) It is specified as xml schema.

- **Feature Instance**
  A feature instance is specified by a xml text following the Feature schema. It can be either observed by humans or extracted my machines.

- **Module Instance**
  Identifies a reproducible extraction algorithm/unit. This includes its source code version, and its internal state or knowledge. It produces Feature Instances, which may be of multiple Feature Types.

- **Hypothesis Set**
  A Hypothesis Set consists of the Feature Instances produced by a Module Instance.

- **Reference Set**
  A Reference Set or Test Set is a collection of different Feature Instances, which are annotated by a human or a Module Instance. (also Target Instances) It typically describes a whole media corpus.

- **System Instance**
  Dynamical System. A System Instance is a distributed configuration of Module Instances producing Feature Instances. Note that multiple Module Instances can produce the same Feature Type. Thus a single Module Instance must be selected, which produces the Hypothesis Set of the whole System Instance.

- **Benchmarking Result**
  Quantitative evaluation of a Hypothesis Set according to a Reference Set.

5.5 Requirements

This section describes requirements regarding the Benchmarking System. Before requirements are listed a short summary describing the framework is provided. (Taken from the mistral requirements document)

5.5.1 Overview

The Benchmarking Framework (BM-FW) will provide a database for time-varying and multimodal data streams. The data will be used for benchmarking the performance of different algorithms. It will contain, but not exclusively contain, data for Mistral specific applications and algorithms. The database will be accessible via an easy-to-use web interface, a database API, and a TCP/IP interface to the benchmark server. Part of the data will be public with the aim to establish the database as a standard benchmarking database for time varying data. Sensible data will be accessible to project partners only. Together with the data, meta information and learning targets will be provided. Detailed Requirements are listed in Table 5.2.

The Requirement Ben6.2 was rated higher (1) afterwards, because of the interest from video and audio algorithm developers for performance and progress visualisation. During the next phase also the requirement of inter-modal and inter-feature performance comparisons appeared.

Nr	Name	Description	Priority
Ben1	**Benchmark Data**		
Ben1.1	Benchmark Data	The BM-FW will provide data for benchmarking learning algorithms with time-varying input streams and for learning algorithms on multimodal data.	1
Ben1.2	Project-Relevant Data	The BM-FW will provide data relevant for benchmarking Mistral-specific applications and algorithms. This includes all information entities (IE) that are generated by the Mistral-system.	1
Ben2	**User Interface**		
Ben2.1	Web Access	The BM-FW will provide an easy to use web-interface to access data.	1
Ben2.2	API and TCP/IP Access	The BM-FW will provide a consistent storage, and search functionality for IE's. The interface will be application and TCP/IP based.	1
Ben2.3	Open Access	Part of the data will be public accessible.	1
Ben2.4	Restricted Access	Sensible data (and possibly also other project-relevant data) will be accessible to project partners only.	1
Ben3	**Data**		
Ben3.1	Modalities	Data of different modalities will be stored in the database: Video, Audio, Text, possibly also other data streams (e.g. EEG recordings, etc.).	1
Ben3.2	Document-Corpuses	Documents which belong together logically will be provided in a document corpus. This will be especially important for benchmarking of Mistral-specific applications.	2
Ben3.3	Data Formats	Data will be provided in common formats (but not each data set will be provided in all formats), and in IE package format.	1
Ben3.4	Dataset documentation	All datasets will be documented extensively in text-files (PDF) which will be available for the users.	2
Ben3.5	Metadata	Datasets with metadata in common formats will be provided.	2
Ben3.6	Learning Targets	In order to validate the performance of algorithms on the data, learning targets will be provided (e.g. Time points when a dog appears in a video).	2
Ben4	**Security**		
Ben4.1	Secure Data	Sensible data will be stored securely and will not be accessible to non-authorized persons.	1
Ben5	**Maintenance**		
Ben5.1	Easy Maintenance	Maintenance (e.g. Add new datasets to the database) of the BM-FW will be easy.	3
Ben6	**Feedback**		
Ben6.1	User Feedback	Users will be able to give feedback on individual datasets.	4
Ben6.2	Performances	Performances of different algorithmic approaches will be made visible for datasets. Users will be able to add their results	3
Ben7	**Tools**		
Ben7.1	Evaluation Tools	Tools will be provided which help the users to evaluate the performance of their algorithms.	4

Table 5.2: Requirements of the Benchmarking System

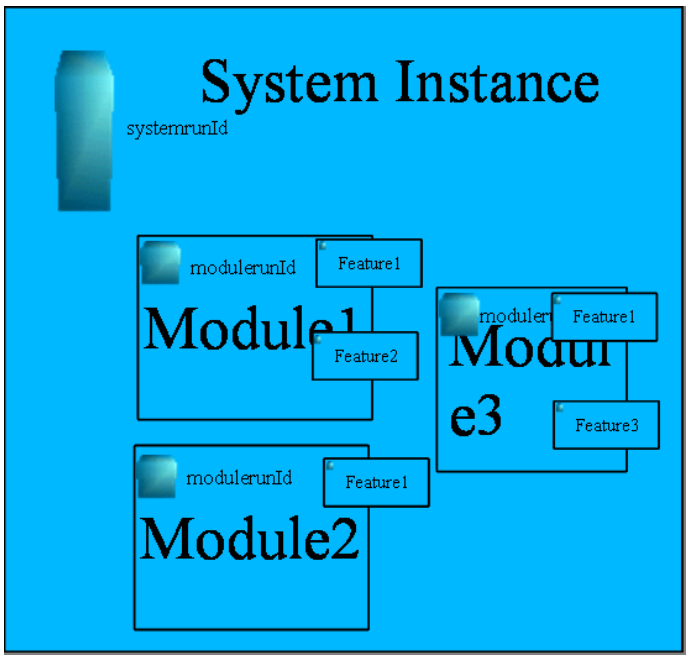

Figure 5.1: Benchmarking from the dynamical systems perspective. Each subsystem (module) draws the input feature instances from a common pool. (media corpus, preprocessed feature instances, or extracted feature instances) A module can have different feature types, with several feature instances each, as output. "Unimodal" modules extract information from the raw data. "Multimodal" modules integrate information from different modules or modalities. System and Module Instances have 128bit UUID's, which are truly unique identifiers. Hypothesis Sets can be compared to the Reference Set. Thus Benchmarking Results can be obtained for each Module Instance. The best performing Module Instance (usually the multimodal integration or semantic enrichment unit) determines the Hypothesis Set and Benchmarking Result of the whole System Instance.

5.6 Architectural Design

Figure 5.1 views the benchmarking system from the dynamical systems perspective. Figure 5.2 shows the underlying persistence and benchmarking mechanisms.

5.7 Software Design

Java 1.5 Java 1.5 will be used as main programming language.

eclipse.BIRT BIRT (Figure 5.3) is an Eclipse-based open source reporting system for web applications, especially those based on Java and J2EE. BIRT has two main components: a report designer based on Eclipse, and a runtime component that you can add to your app server. BIRT also offers a charting engine that lets you add charts to your own application.

The BIRT Report Engine is packaged as a JAR file that you add to your J2EE application. The Report Engine is a series of POJOs (Plain Old Java Objects) that your JSP pages can call to integrate reporting into your application.

BIRT 1.0.1 is used for generating dynamic web reports from the benchmarking data. The reports include benchmarking results viewing, comparing different system instances, time series view, recall-

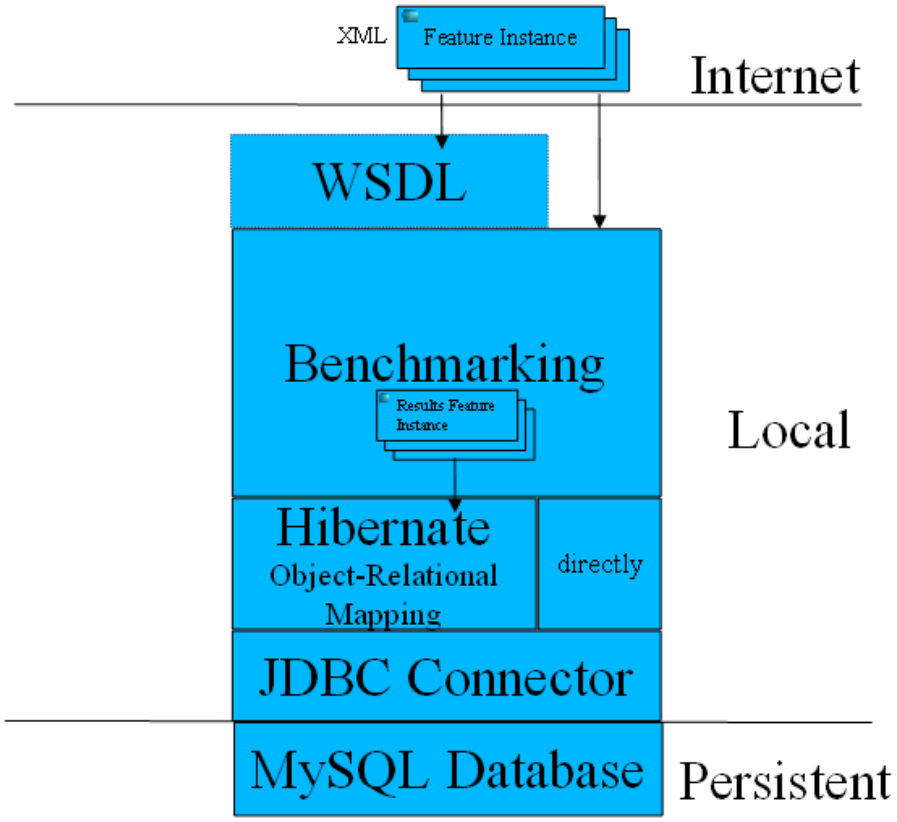

Figure 5.2: Persistence Architecture of the Benchmarking System. The Hypothesis Set is submitted via XML file upload or web service. Then it is are benchmarked with respect to the Reference Set. The Benchmarking Results and the Hypothesis Set are represented as Java objects. The Java objects are mapped to database tables and stored using the Hibernate O/R mapper and the JDBC Connector. The use of Hibernate is optional. One can go back to plain JDBC, if performance is critical.

precision diagram, and media catalog.

Tomcat Apache Jakarta Tomcat 5.5.9 is being used for web deployment. Tomcat is a servlet container, which translates Java Server Pages (.jsp; dynamical web pages) into html producing Java classes. The report generating Java classes are called from within JSP. An external security filter is integrated, to protect the restricted area files.

Ant Apache Ant deployment scripts are used. Two web archive(.war) files, for the web application and the web service, are built and deployed automatically.

MySQL The MySQL 5 open source database is used as relational database backend.

Hibernate The Hibernate 3.0 open source object relational mapper is used to store java objects to MySQL.

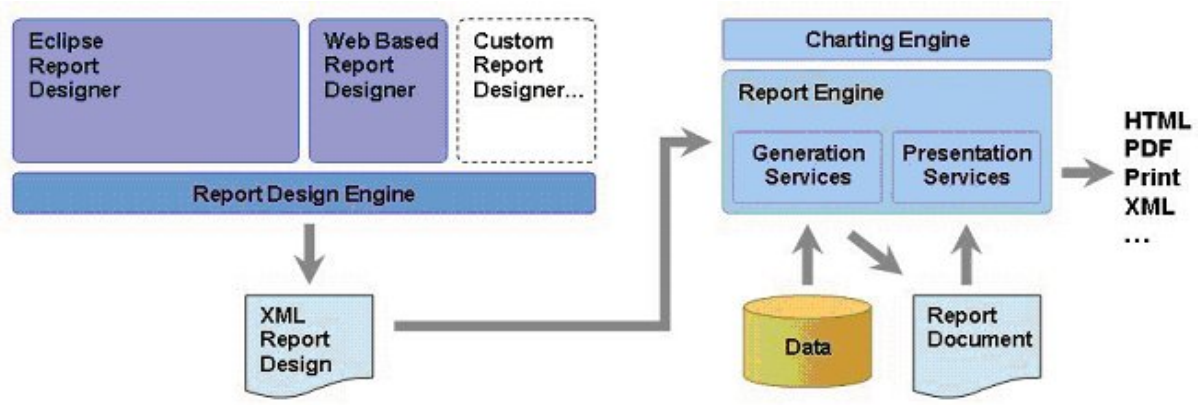

Figure 5.3: Architectural overview of Eclipse BIRT

Axis2 Apache Axis2 0.92 web services are used. MTOM file attachments are used to transfer the XML file, which must be according to the BenchmarkingInput.xsd schema. (available on the web page) BenchmarkingInput.xsd defines all used feature types, and the SystemRun type, which includes system information, extracted features, and test set to test on. A new test set can be defined, otherwise an existing test set identifier is necessary. A Java and MATLAB interface is provided.

Overview

The current solution contains the benchmarking core (**org.mistral.benchmarking**, **org.mistral.benchmarking.measures**), and an underlying persistence mechanism (**org.mistral.persistence**) layer to write/read/update the Java structures (**org.mistral**) into/from the MySQL database.
The dynamic web application (**org.mistral.benchmarking.webapp** + JSP pages + Eclipse.BIRT + static content) accesses both the MySQL database (for Eclipse.BIRT visualisation) and the benchmarking core functionality. (XML Results file upload) The private part of the web page is secured by the security filter (**org.securityfilter**).
The axis2 web service (**org.mistral.benchmarking.ws**) accepts the incoming service requests, and handles the XML file to the benchmarking core. The benchmarking core verifies and benchmarks the algorithm's extracted features according to the stated test set. The benchmarking measurement method for a feature XYZ (i.e. Event, TimeSeriesPointFeature,...) is determined through the **org.mistral.benchmarking.measures.XYZMeasurement** class. The Measurement class uses the predefined performance measures of **org.mistral.benchmarking.measures**. The XML data is imported into the database through the autogenerated XML DOM parser. (**com.altova** and **com.Benchmarking**). **org.mistral.benchmarking.test** contains the benchmarking system test cases. Web application and web service logging; import, and export are done with **org.mistral.benchmarking.util**.

For adding a new Feature Type 'XYZ' to the benchmarking system, one has to:

1. Define a XYZFeature element in the BenchmarkingInput.xsd schema

2. Generate the Java Code (parser) for the BenchmarkingInput.xsd and the SQL create statement for the XYZFeature element (with xmlspy software)

3. Replace the parser packages in the source code, and create the new database table by executing the SQL query.
4. Create a hibernate mapping class with the eclipse 'hibernate artifact generation' tool, using the 'reverse engineer from db' option.
5. Test the persistence of the new hibernate class.
6. Write a **org.mistral.benchmarking.measures.XYZMeasurement** class, which defines the benchmarking. Reuse existing performance measures.
7. Redeploy the web application and web service using the ant task 'all' of build.xml.
8. Benchmark an algorithm, which produces XYZFeature labels.

Java Class Documentation

- **com.altova** Auto generated XML DOM Parser
  - AltovaException
- **com.altova.types** Auto generated XML DOM Parser
  - SchemaType
  - SchemaTypeBinary
  - SchemaTypeCalendar
  - SchemaTypeNumber
  - SchemaAnyURI
  - SchemaBase64Binary
  - SchemaBinaryBase
  - SchemaBoolean
  - SchemaByte
  - SchemaCalendarBase
  - SchemaDate
  - SchemaDateTime
  - SchemaDecimal
  - SchemaDouble
  - SchemaDuration
  - SchemaEntity
  - SchemaFloat
  - SchemaHexBinary
  - SchemaID
  - SchemaIDRef
  - SchemaInt
  - SchemaInteger
  - SchemaLanguage
  - SchemaLong

  - SchemaName
  - SchemaNCName
  - SchemaNMToken
  - SchemaNormalizedString
  - SchemaShort
  - SchemaString
  - SchemaTime
  - SchemaToken
  - SchemaTypeFactory
  - NotANumberException
  - SchemaTypeException
  - StringParseException
  - TypesIncompatibleException
  - ValuesNotConvertableException

- **com.altova.xml** Auto generated XML DOM Parser
  - Document
  - Node
  - XmlException

- **com.Benchmarking** Auto generated XML DOM Parser (XSD to Java)
  - BenchmarkingDoc
  - BenchmarkingInputDoc
  - BenchmarkingResultsType
  - SystemRunResultsType
  - TestSetResultsType

- **com.Benchmarking.av** Auto generated XML DOM Parser (from XSD to Java)
  - ActionType
  - EventType
  - ObjectIndexType
  - ObjectType
  - TestType

- **com.Benchmarking.bmi** Auto generated XML DOM Parser (from XSD to Java)
  - FeatureRunType
  - FeaturesType
  - FeatureType
  - ModuleType
  - SessionDataType
  - SystemHeaderType
  - SystemRunType

  - TargetsType
  - TimeSeriesFeatureType
  - VersionType

- **com.BenchmarkingTest** Auto generated XML DOM Parser (from XSD to Java)
  - BenchmarkingTest

- **org.mistral** Public Feature Types (can be stored by Hibernate from Java to DB)
  - *BenchmarkingResultsFeature* BenchmarkingResultsFeature generated by hbm2java
  - *CausalityFeature* CausalityFeature generated by hbm2java
  - *Doubles* TrajectoryPointFeature generated by hbm2java
  - *Event* Event generated by hbm2java
  - *Feature* Feature generated by hbm2java
  - *HibernateQueryTest* Retrieve data as objects
  - *MediumFeature* MediumFeature generated by hbm2java
  - *SessionData* SessionData generated by hbm2java
  - *SystemRunHeader* SystemRunHeader generated by hbm2java
  - *TimeSeriesPointFeature* TimeSeriesPointFeature generated by hbm2java
  - *TrajectoryPointFeature* TrajectoryPointFeature generated by hbm2java

- **org.mistral.benchmarking** Benchmarking System Core
  - *Benchmarking* central benchmarking class.
  - *BenchmarkingSystemProperties* Access to the bm-system.properties map.
  - *Status* bm-system status.

- **org.mistral.benchmarking.measures** Benchmarking Performance Measurement
  - *FMeasure* $F_\beta = \frac{(\beta^2+1)RP}{(\beta^2 P+R)}$, see 2.3.4.
  - *Measure* abstract base class for measures.
  - *Measurement* abstract base class for measurement of certain performance measures of a 'Feature' type The whole 'System Instance' data structure is given as input.
  - *PrecisionMeasure* $P = \frac{C}{M} = \frac{C}{I+S+C}$, see 2.3.3.
  - *RecallMeasure* $R = \frac{C}{N} = \frac{C}{D+S+C}$, see 2.3.2.
  - *TimeSeriesPointFeatureMeasurement* Measurement for TimeSeriesPointFeature.

- **org.mistral.benchmarking.test** Benchmarking System Test Cases
  - *BenchmarkingDocTest* BenchmarkingDoc read/write Testcase
  - *BenchmarkingInputDocTest* BenchmarkingInputDoc read/write Testcase.
  - *BenchmarkingTest* Benchmarking Test Application.
  - *GenerateReportTest* Generate eclipse.BIRT report Testcase
  - *HibernateCreateTest* Create sample data, letting Hibernate persist it for us.
  - *MeasureTest* Test measures.

- **org.mistral.benchmarking.util** Logging, Import and Export Utilities

  - *HtmlLoggerFactory* Factory Pattern.
  - *ImportFilter* Import Filters for Media/Feature Data
  - *ImportTimeSeriesDataLibrary* import the TSDL by parsing the web page of Hyndman.

- **org.mistral.benchmarking.webapp** Dynamic Web Application Core
  - *BenchmarkingInput* webapp bean for uploading and benchmarking an xml file.
  - *GenerateReport* Uses eclipse.BIRT to generate a .html file from a .xml report definition.
  - *HyperlinkReplacer* hyperlink support for eclipse.BIRT
  - *StartupContextListener* listens to webapp context startup.

- **org.mistral.benchmarking.ws.client**
  - *BenchmarkingClient* BenchmarkingService axis2 client.
  - *BenchmarkingCmdClient* BenchmarkingService client test case.
  - *BenchmarkingGuiClient* Benchmarking webservice client with GUI.
  - *BenchmarkingInput* Class for easily creating a xml file according to BenchmarkingInput.xsd
  - *UserInterface* Swing GUI for BenchmarkingService.

- **org.mistral.benchmarking.ws.service**
  - *BenchmarkingService* BenchmarkingService axis2 webservice.

- **org.mistral.persistence**
  - *HibernateAccess* Hibernate persistence layer.
  - *HibernateUtil* Basic Hibernate helper class, handles SessionFactory, Session and Transaction.
  - *MySQLDatabaseAccess* MySQL database access layer.

- **org.securityfilter.example** Security for web application restricted area
  - *Constants* Constants for the tomcat security filter.

- **org.securityfilter.example.realm**
  - *TrivialSecurityRealm* Implementation of the SecurityRealmInterface.

- **org.securityfilter.example.realm.catalina**
  - *TrivialCatalinaRealm* Catalina Realm implementation.

Database Layout The four main entities in the database are **systems**, **modules**, **features**, and **test-sets**.
Figure 5.4 shows the relations between those entities. Systems *consist of* modules, which in turn *compute* features. Systems are *benchmarked on* test sets, which *consist of* (human annotated) features.
Based on the computed, annotated, and benchmarked data, queries can be made. A Module can be part of different systems. A Feature can be part of different test sets. A test set has possibly many benchmarks of systems, to determine which system performed best on that test set. A feature is computed by different modules, querying i.e. which modules can detect cars.
As the four entities may change over time (test set enlargement, new system configuration, new algorithm version, new feature definition, new computed feature), each entity has a unique 128bit UUID

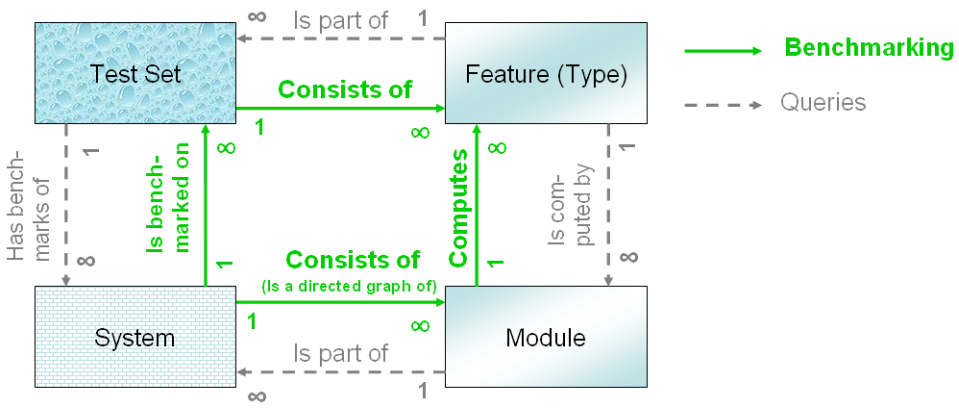

Figure 5.4: Basic enity relationship diagram of the benchmarking system. A system consists of modules. A Module computes features. A testset consists of features. A system is benchmarked on testsets. (Benchmarking, green solid arrows) Afterwards queries can be made. (Queries, grey dashed arrows)

(Universally Unique Identifier for distributed systems).
Features have a concrete type. *TimeSeriesPointFeature*, consisting of a value plus time stamp, is the simplest, and unstructured feature type. (time series)More complex features, like audiovisual *Event*, contain several attributes of different basic type (string, int, double, classes,...). This is equivalent to the notion of an *object* in OOP.

5.8 Demonstration

Web Application Figure 5.5 shows the Benchmarking System start page, with its main objective. Figure 5.6 shows the Catalog of available data. Figure 5.7 shows the benchmarking results page for algorithm comparison.

5.9 Conclusion

The Benchmarking System can store and deliver training sets, test sets, labels and algorithm result labels (Requirement *Ben1*). The datasets are stored in common formats (jpeg, wav), but not yet in a MPEG-7 representation. (*Ben3.3*) Datasets are stored in a zipped version. (*Ben3.2*) Labels are XML files (defined as XSD schemes) and persisted through Hibernate into a relational database (MySQL). The user interface delivers web access of both data and benchmarking results. Benchmarking can be done via web service (axis2, from any programming language) or XML file upload. A security filter splits the web page into a public and a private area. Thus Requirements *Ben2* and *Ben4* are forfilled. Generated events

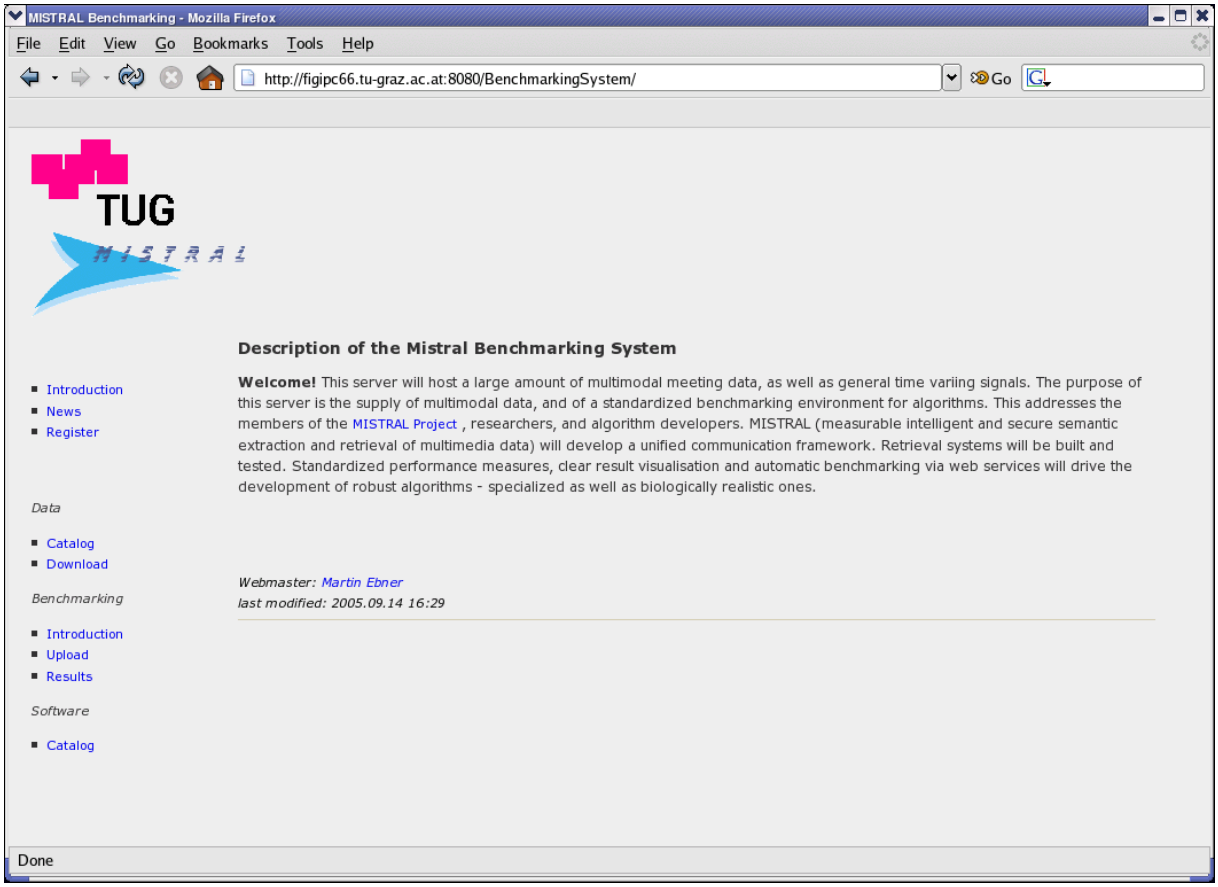

Figure 5.5: Startpage of the Benchmarking System Web Application

from different modalities are stored separately. (*Ben3.1*)
Thus all priority 1 requirements (except *Ben3.3* only partially) **are forfilled.**
Among the lower priority requirements, *Ben6.2*, algorithm performance visualisation was implemented. Document-corpuses, (*Ben3.2*), are roughly implemented (zipped corpuses). Dataset documentation (*Ben3.4*) is only included in the metadata, with a maximum of 256 text characters. Metadata for each corpus is included in the database, but not in a MPEG-7 form yet. *(Ben3.5)* Learning targets can be provided in the corpus, but yet no annotation effort was conducted. *(Ben3.6)* Maintainance *(Ben5.1)* is quite easy. For instance, The web application plus the web service can be installed (deployed) within two .war files. The MySQL database can be backuped easily, with common tools. Test set labels can be submitted via web service. User feedback was not implemented. *(Ben 6.2)* Finally, tools *(Ben7)* like the annotation framework (least priority) are missing, but are in planning stadium. (all requirements discussed)
In addition, the notion of distributed modules building a system in incorporated into the Benchmarking System.

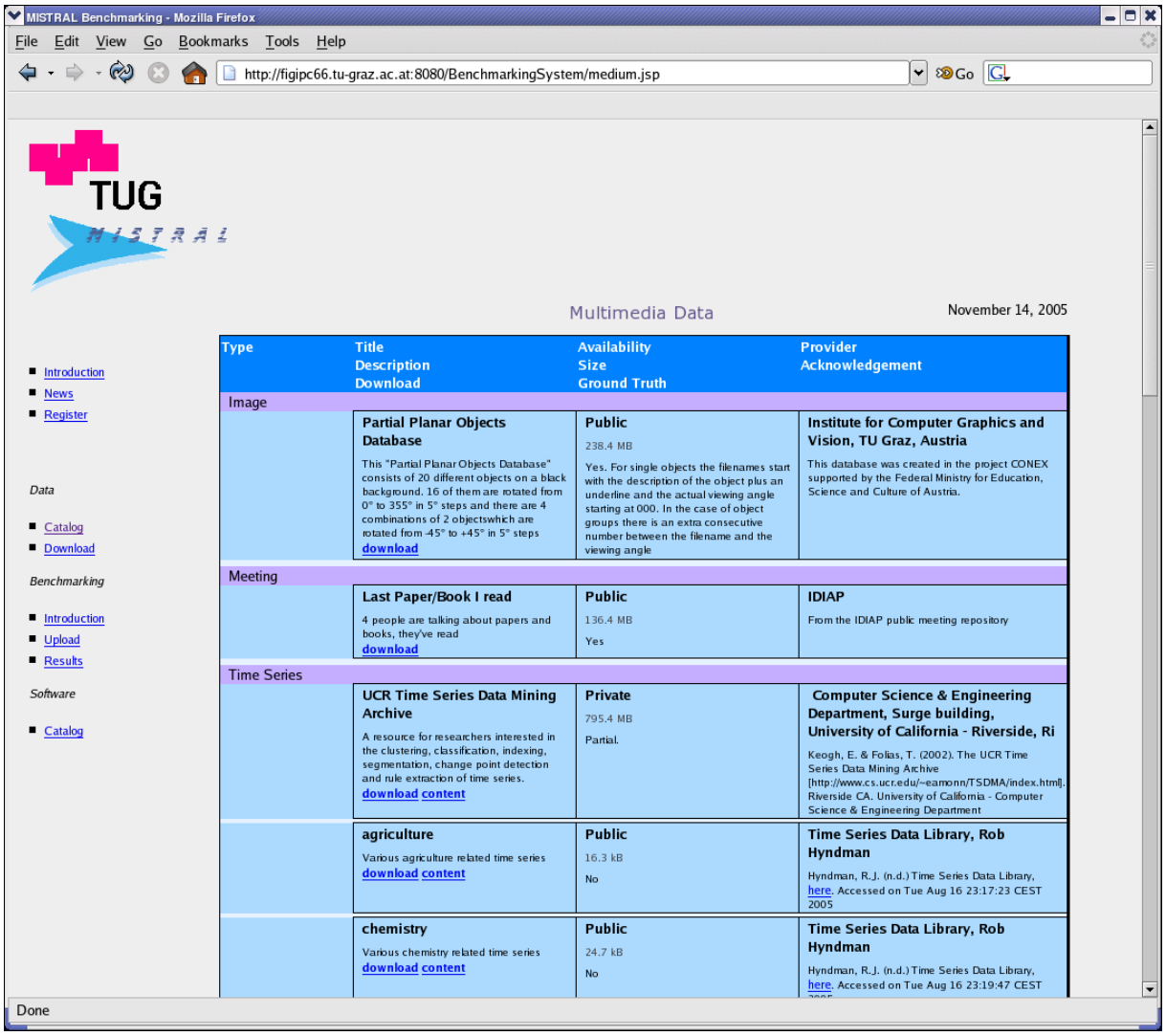

Figure 5.6: Data Catalog of the Benchmarking System Web Application. Datasets are divided into categories. (Image, Meeting, Time Series) Each dataset is provided with metadata. (Title, Description, Download link, Private/Public Availability, Size, Ground Truth (Label exist), Data Provider, and Provider Legal Acknowledgment)

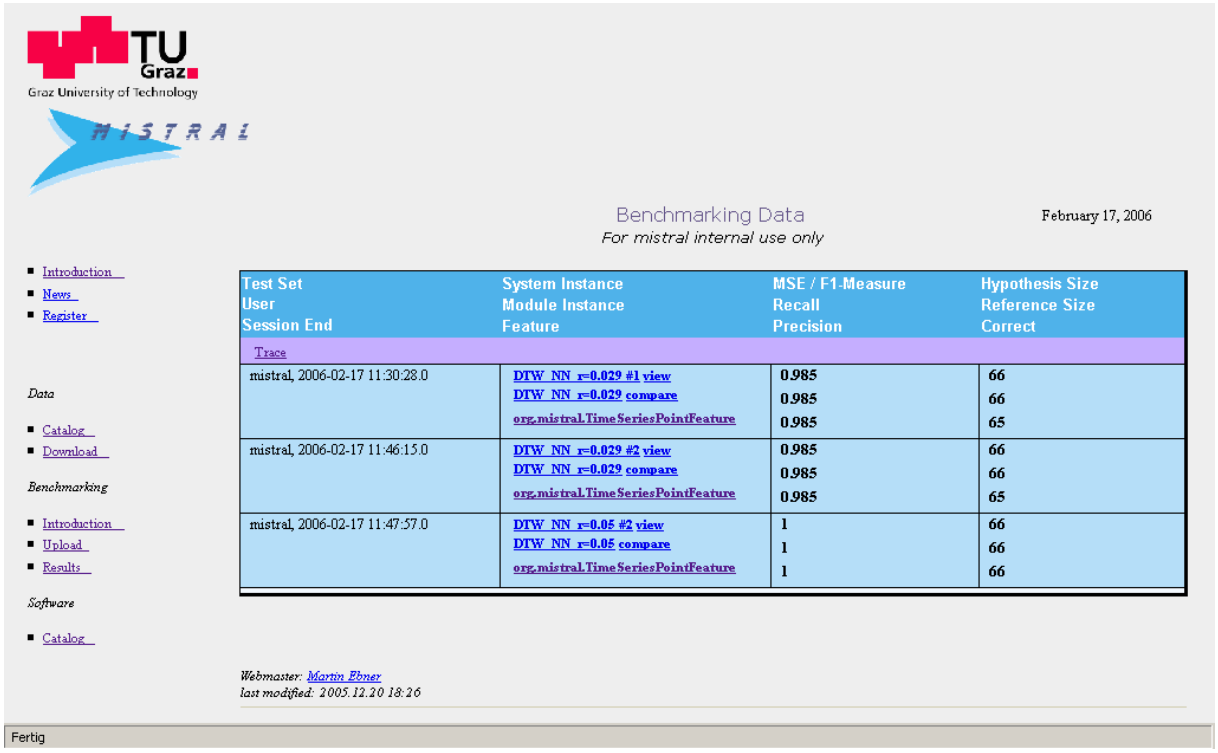

Figure 5.7: Online Benchmarking Results on the Web Page. See the Experimental Results chapter for more detail.

Chapter 6

Algorithms for Time Series Classification

6.1 Overview

Classification and clustering of timeseries become more important in the recent years. The most basic similarity measure is euclidean distance. Dynamic Time Warping, introduced later in this chapter, resolves problems of time dependency by fixed algorithmic aligning, time warping. K-nearest neighbor method and the distance measure are used to build a classification hypothesis out of the input data.
Standard machine learning techniques like Support Vector Machines or Artificial Neural Networks suffer from the curse of dimensionality for long time series.
Models with internal state, like *Hidden Markov Models* (HMM) or *Dynamical Bayesian Networks* (DBN) provide a possibility of integrating multiple modalities, and further have a dynamical output, which makes them capable of not just classifying whole time series, but of producing the most likely stream of classification labels. This is often notated as *subsequence classification*, *event detection*, or *motif detection*. A problem may lie in the *Markov Property* itself, which asserts the state of time t to be only dependent on the state of time t-1. This constraint is practical, if each input variable and each internal model state have the same time scale or sampling frequency. Variables with lower sampling frequency still must be sampled at the highest signal's sampling frequency. A similar problem occurs with sparse events. For data inherent processes running on a vast range of timescales, the markov property introduces a huge amount of unnecessary model structure dependencies. Not to mention the adaptivity of the model to biological, chemical, physical rhythms. A step in that direction are *Continuous Time Bayesian Networks*. This model can give the distribution over time, when a particular event occurs.

6.2 Dynamic Time Warping

6.2.1 Introduction

Dynamic time warping (DTW) (Ratanamahatana and Keogh, 2004) performs well on not just a few, but many different benchmarking datasets. Dynamic time warping was first used by the speech processing community (Itakura, 1975), and then introduced to the data mining community. (Berndt and Clifford, 1994) DTW has quadratic time complexity. It is used for classification, motif discovery, rule discovery and anomaly detection. There are various applications in bioinformatics, EEG analysis, robotics, music and many more fields.

6.2.2 Definitions

Time Series Q and C Given *two time series*, the query sequence Q of length n, and the candidate sequence C of length m,

$$Q =< q_1, ..., q_n > \tag{6.1}$$

and

$$C =< c_1, ..., c_m > \tag{6.2}$$

Distance Matrix D We construct a *distance matrix* D with its elements

$$d_{(i,j)} = (q_i - c_j)^2, \quad i = \{1, ..., n\} \wedge j = \{1, ..., m\} \tag{6.3}$$

It contains the (squared) euklidean distance between any two points q_i and c_j of the sequences.

Warping Path W Then we define a warping path W (see Figure 6.1) as a sequence of contiguous path elements w_k,

$$W = w_1, ..., w_k, ...w_l, \forall k = \{1, ..., l\} \wedge w_k = (i, j)_k \wedge max(m, n) \leq l < m + n - 1 \tag{6.4}$$

Dynamic Time Warping $DTW(Q, C)$ Finally, DTW is defined as the length of the minimum length warping path W^*.

$$DTW(Q, C) = \min\{\sqrt{\sum_k d_{w_k}}\} = \sqrt{\sum_k d_{w_k^*}} \tag{6.5}$$

Dynamic Programming This can be done by dynamic programming. We calculate the *cumulative distance* $\gamma_{(i,j)}$ by the minimum previous cumulative distance, which is the minimum of the left, lower-left and lower neighbor's cumulative distance , plus the current distance $d_{(i,j)}$.

$$\gamma_{(i,j)} = \min\left\{\gamma_{(i-1,j)}, \gamma_{(i,j-1)}, \gamma_{(i-1,j-1)}\right\} + d_{(i,j)} \tag{6.6}$$

These constraints reduce the number of possible paths:

Boundary Condition The path has to start at $w_1 = (1, 1)$ and must end at $w_l = (m, n)$. Beginning and end points of the two time series are aligned.

Continuity Condition The indices i and j must not increase by more than 1 every step. This means,

$$w_k = (i, j) \Rightarrow w_{k-1} = (i', j'), \quad (i - i') \leq 1 \wedge (j - j') \leq 1 \tag{6.7}$$

Monotony Condition The indices i and j must not decrease. This means,

$$w_k = (i, j) \Rightarrow w_{k-1} = (i', j'), \quad (i - i') \geq 0 \wedge (j - j') \geq 0 \tag{6.8}$$

Slope Constraint Condition The slope of the path must not be steeper or shallower than a certain ratio $x = a/b$. After a steps in direction i one step in direction j must be taken, and vice versa.

Adjustment Window The path always stays near the diagonal.

$$w_k = (i, j)_k, j - r \leq i \leq j + r \tag{6.9}$$

The *reach* r is either constant (Sakoe-Chiba band) or a function of i. (Itakura parallelogram)
In the special case of $w_k = (k, k), \forall k = \{1, ..., l\} \wedge m = n$, the $DTW(Q, C)$ is equal to the euklidean distance of Q and C. Time and space complexity of the DTW are of $O(mn)$. The above constraints decrease the time complexity only by a constant factor.

6.2.3 LB Keogh

$LB\_Keogh(Q, C)$ is a lower bound for the $DTW(Q, C)$ measure 6.5, as shown in (Keogh and Ratanamahatana, 2003).
Given the adjustment window 6.9, the upper and lower envelopes are defined.

$$U_i = \max\{q_{i-r}, ..., q_i, ..., q_{i+r}\} \tag{6.10}$$

$$L_i = \min\{q_{i-r}, ..., q_i, ..., q_{i+r}\} \tag{6.11}$$

with

$$\forall_i L_i \leq q_i \leq U_i \tag{6.12}$$

From the upper and lower envelopes U_i and L_i, $LB\_Keogh(Q, C)$ can be defined.

$$LB\_Keogh(Q, C) = \sqrt{\sum_{i=1}^{n} \begin{cases} (c_i - U_i)^2 & for \quad c_i > U_i \\ (c_i - L_i)^2 & for \quad c_i < L_i \\ 0 & else \end{cases}} \tag{6.13}$$

Lower Bounding Lemma Given any sequences Q and C of equal length the inequality

$$LB\_Keogh(Q, C) \leq DTW(Q, C) \tag{6.14}$$

holds. The proof is shown in (Keogh and Ratanamahatana, 2003).
It has been shown empirically (Keogh and Ratanamahatana, 2003), that LB_Keogh reduces time complexity of a DTW nearest neighbor search.

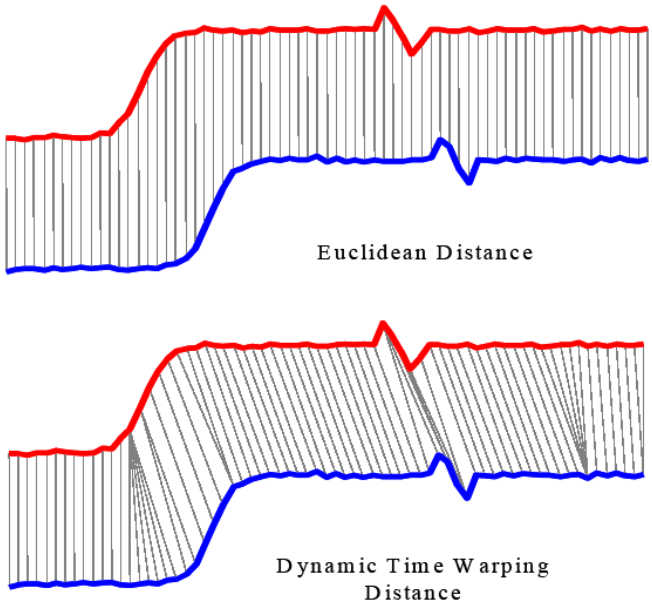

Figure 6.1: Nonlinear Time Alignment using Dynamic Time Warping

6.2.4 Exact Indexing

Exact indexing for faster DTW nearest neighbor search can be done using LB_Keogh. The database sequences of length n are divided into N frames of equal size. The index size N is usually 16 or less. Figure 6.2 shows CPU cost results for different benchmarks and benchmark sizes. The result suggests a linear speedup in the range between 2^9 and 2^{13} database size, and constant speedup for larger databases.

6.3 Conclusion

This chapter gives an introduction to time series classification algorithms. The DTW algorithm is explained in detail. Also the lower bounding LB_Keogh for DTW is presented, and a short introduction to DTW exact indexing is given.

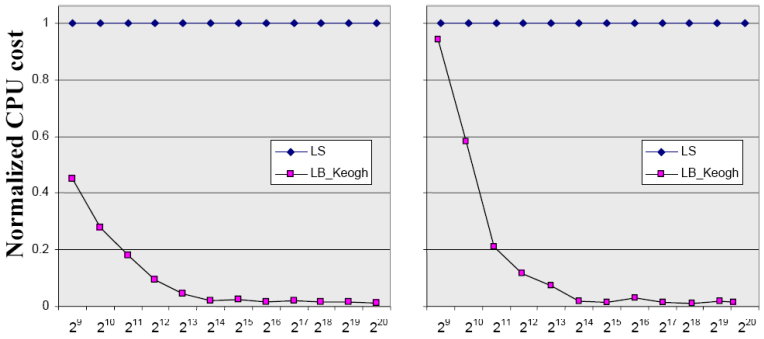

Figure 6.2: CPU cost of nearest neighbor search with indexed DTW using LB_Keogh normalized to the CPU cost of linear scan. The database size is drawn on the x-axis. The used index size is 16. The timeseries length is 256. Left diagram shows results for random walk data, on the right a mixed bag of 32 datasets is used.

Chapter 7

Experimental Results

7.1 Matlab Client

Additionally, a matlab client for accessing the benchmarking web service is provided. The client consists of the java web service client and matlab embedding scripts:

- `./classes` contains the java classes
- `./lib` contains the java libraries
- `./matlab` contains the matlab files
- `./matlab/benchmark.m` benchmarking of test set results via web service
- `./matlab/bm_status.m` get the status of the web service ('configured',...)
- `./matlab/testset.m` uploads a new test set via web service
- `./matlab/examples/dtw_class` is a benchmarking example for the DTW/nearest neighbor algorithm. (see next section)
- `./matlab/examples/timeser_pred` is a benchmarking example for a simple time series prediction algorithm.
- `./src` contains the java source code
- `./README` *contains installation instructions*

7.2 Benchmarking the DTW_NN Algorithm

This chapter shows the benchmarking results of the DTW_NN algorithm on three different datasets. This algorithm uses DTW as distance measure and nearest neighbor method for classification. The DTW is calculated using the distance matrix and dynamic programming in MATLAB, with a C language core for dynamic programming. LB_Keogh was implemented for further reducing time complexity.

testset_id	classes	#training instances	#test instances	#total instances	time steps
Gunx	2	133	66	200	150
Trace	4	133	66	200	275
twopat	4	3333	1666	5000	128

Table 7.1: Properties of the used timeseries test sets

7.2.1 Test Sets

Table 7.1 lists the properties of the 3 different classification datasets used. The original datasets were divided into a public training set of $frac23$ and a private test set of $frac13$ of the dataset. This ensures, that the algorithm is not optimized for the test set. This advantage only applies to datasets, which are first published by the benchmarking system, keeping the test set within the black box.

Gunx

See article (Keogh and Ratanamahatana, 2003).

This 2-class dataset comes from the video surveillance domain. The dataset has two classes, each containing 100 instances. All instances were created using one female actor and one male actor in a single session. The two classes are:

1 Gun-Draw: The actors have their hands by their sides. They draw a replicate gun from a hip-mounted holster, point it at a target for approximately one second, then return the gun to the holster, and their hands to their sides.

2 Point: The actors have their hands by their sides. They point with their index fingers to a target for approximately one second, and then return their hands to their sides.

For both classes, the centroid of the right hand in both the X- and Y-axes are tracked; however, this dataset only contain X-axis for simplicity. Each instance has the same length of 150 data points (plus the class label), and is z-normalized (mean = 0, std = 1).

Classification Error Rates:

Euclidean: 5.50%
DTW with 10% warping window size: 4.50%
DTW with the best (3.25%) uniform warping window size: 1.00%

Trace

See article (Keogh and Ratanamahatana, 2003).

This 4-class dataset is a subset of the Transient Classification Benchmark (trace project). It is a synthetic dataset designed to simulate instrumentation failures in a nuclear power plant, created by Davide

Roverso. The full dataset consists of 16 classes, 50 instances in each class. Each instance has 4 features.

The TRACE subset only uses the second feature of class 2 and , and the third feature of class 3 and 7. Hence, this dataset contains 200 instances, 50 for each class. All instances are linearly interpolated to have the same length of 275 data points, and are z-normalized.

Classification Error Rates:
Euclidean: 11.00%
DTW with 10% warping window size: 0.00%
DTW with the best (3.375%) uniform warping window size: 0.00%

twopat

TWO-PAT dataset - 5000 cases - 128 time steps - 4 classes

This synthetic dataset was designed by Pierre Geurt in his Phd thesis, (Geurts, 2002) in Appendix B.2.3.

class labels:

1 down-down (1306 cases)

2 up-down (1248 cases)

3 down-up (1245 cases)

4 up-up (1201 cases)

7.3 Benchmarking Method

The Benchmarking System was used for evaluation of the algorithm's results on the reference set. The benchmarking can be done automatically, with the testset_id as parameter. (Benchmarking in other programming languages can be done similarly)

```
67 function dtw_nn_bench(testset_id, modulerun_id, runnumber, r)
68
69 % DTW, nearest neighbor classification
70 % ********************************************************
71 %
72 % testset_id      test set identifier
73 % modulerun_id    module desciption (incl. version+parameters)
74 % runnumber       benchmarking run number
75 % r               window size of the DTW algorithm
76 %
77 % i.e.:
78 % testset_id='Gunx';
79 % modulerun_id='DTW_NN_r=0.027';
80 % runnumber=1;
81 % r=0.027
82 % see Ratanamahatana and Keogh, 2004
83 % ********************************************************
84
85 % 1.) GETTING TRAINING SET 'X' + LABELS 'C',
```

```
86 %      AND TEST SET 'Xtest'
87
88
89 filename=strcat(testset_id,'-data.mat');
90 load(num2str(filename));
91
92 % 3.) TRAINING
93
94 % optional: parameter search...
95
96 % 4.) TESTING
97
98 Ctest_algo = dtw_nn(X, C, Xtest, r);
99
100 % 5.) BENCHMARKING
101
102 benchmark(Ctest_algo, testset_id, modulerun_id, runnumber);
```

Listing 7.1: Example of TRECVID video feature extraction results for two runs

The Benchmarking System calculates F_1, Recall and Precision. There are no deletions or insertions, as the reference set is congruent with the hypothesis set. Both are vectors of same length. The length is the number of test set examples. Thus, all three measures are equal to $(1 - MAE)$. MSE equals to MAE for classification. Note, that information retrieval, sparse event-, and motif detection may deliver non-congruent reference and hypothesis sets.

7.4 Results

Figure 7.1 shows the results visible on the benchmarking website. For the first two results, the *original results (Ratanamahatana and Keogh, 2004) can be reproduced.*
The $precision$ on the 'Gunx' test set is 0.985 instead of 0.99 in (Ratanamahatana and Keogh, 2004). The $precision$ on the 'Trace' test set is 1.0 (original: 1.0 also), but with slightly different value of $r = 0.05$ (instead of $r = 0.0291$) The reason is, because the original evaluation method *Evaluate()* uses the leave-one-out cross-validation (k-fold cross validation with $k = |ReferenceSet|$) on the whole dataset, instead of a separate test set. Keogh's approach may be criticized for not being tested on new data. My comparison here must also be criticized for comparing different data. Thus, from not hiding the test data (forever), there always is the possibility of different learning and evaluation. In this case the results are empirically almost equal.
The $precision$ on the 'twopat' test set is 1.0. The original text's best algorithm, Pruned DT (S learned) reaches $1 - 0.0322 = 0.9678$ accuracy.

7.5 Conclusion

This chapter showed the benchmarking of a time series classification algorithm, using the benchmarking system's matlab interface. The DTW_NN algorithm is benchmarked on three medium size datasets. For the first two datasets, the original results could be reproduced. The third test set yields 100% precision, the original precision is 0.9678. One principle problem remains: The original data must be splitted into public training set and hidden test set (labels). For very small datasets, the decreased training set size may decrease performance.

Test Set User Session End	System Instance Module Instance Feature	MSE / F1-Measure Recall Precision	Hypothesis Size Reference Size Correct
Gunx			
mistral, 2006-02-17 11:23:09.0	DTW_NN_r=0.027 #1 view DTW_NN_r=0.027 compare org.mistral.TimeSeriesPointFeature	0.985 0.985 0.985	66 66 65
Trace			
mistral, 2006-02-17 11:46:15.0	DTW_NN_r=0.029 #2 view DTW_NN_r=0.029 compare org.mistral.TimeSeriesPointFeature	0.985 0.985 0.985	66 66 65
mistral, 2006-02-17 11:30:28.0	DTW_NN_r=0.029 #1 view DTW_NN_r=0.029 compare org.mistral.TimeSeriesPointFeature	0.985 0.985 0.985	66 66 65
mistral, 2006-02-17 11:47:57.0	DTW_NN_r=0.05 #2 view DTW_NN_r=0.05 compare org.mistral.TimeSeriesPointFeature	1 1 1	66 66 66
twopat			
mistral, 2006-02-16 13:31:16.0	DTW_NN_r=0.05 #1 view DTW_NN_r=0.05 compare org.mistral.TimeSeriesPointFeature	1 1 1	1666 1666 1666

Figure 7.1: Performance measures for the DTW algorithm on three time series test sets.

Chapter 8

Outlook

The benchmarking system is flexible, because of the extensible feature types and measurement definitions. Classification, regression, prediction, and even motif detection can be implemented in the benchmarking system.
A future extension could be the automatic benchmarking of whole systems, which consist of modules. Each module has to be made *available online* through web services. (the modules must contain online algorithms, in order to enable interaction) This could be done through a standardized input data interface, which should be multidimensional sampled trajectories, or even sparse events, if the module is capable of processing such inputs. Configurations of modules and input data could be described in a separate XML document, processed and benchmarked at once. (after each time step, the outputs of all modules have to be synchronized, and distributed to their dependent modules) For the mistral project, the first step will be web services with locally stored data, and no feedback (no online algorithms required). A *more complex data structure* like the the time and space varying components of a human face could be benchmarked, deriving a measure for the integration capabilities of face detection algorithms. The most important face component (eye?) could be identified by leaving one component out, and determining, which hidden data component lets the performance drop most.
Any distributed system of feature extraction algorithms could be benchmarked in an analog experiment, leaving one module out. This should reveal the most important module for performance. (but leaving out more than one module could in principle yield an increasing performance again)
When distinct modules process non-overlapping spacetime parts (like in the face example, using one module for each face part), the data-leave-one-out benchmark implies the module-leave-one-out benchmark. Imagine, for instance the 'R-eye-algo' module, getting the visual features of the right eye region ('R-eye-visual-feature') and producing an eye blink event ('R-eye-blink'). In the data-leave-one-out benchmark, 'R-eye-visual-feature' would be inhibited. This would imply that the 'R-eye-algo' could not extract any 'R-eye-blink' data, which is equivalent to module-leave-one-out of 'R-eye-algo'. The reverse implication is not true.
There is still much to do for segmenting and classifying complex multimodal scenes.
Another extension of the benchmarking system could provide *online data streams*, for instance web cam video-, skin conductance-, or heart rate variability-streams could be used to model and user satisfaction and health.

Chapter 9

Concluding Remarks

The benchmarking system was developed in the context of a *meeting recording scenario*. This naturally lays the emphasis on recognition and localisation, what is ”out there”. This is true for physical objects, but not for mental objects like action, identity, or behavior. Thus one has to look what is ”in there”, and study physical brain processes, multimodality, psychology.
The benchmarking system compares human labeled to machine labeled data. Thus relative performance of machine learning algorithms can be determined.

Hidden Information For existing datasets in the literature there may exist two problems:

1. If the test data was published, a human programmer may get it and infer knowledge from it.
2. If a different evaluation method was used, the results might not be comparable.

For new datasets the benchmarking system is optimal, because the test set is hidden.

In my current personal opinion, there is always leaking information, through social context (non-local processes, which are ”out there”, or local processes, which are ”in there”). What if a human inside the Schroedinger's cat black box would know the state of the cat? He could clearly never tell anyone! That's for sure.

Appendix A

The Mistral Project

A.1 Introduction

Multimedia data has a rich and complex structure in terms of inter- and intra-document references and can be an extremely valuable source of information. However, this potential is severely limited until and unless effective methods for semantic extraction and semantic-based cross-media exploration and retrieval can be devised. Today's leading-edge techniques in this area are working well for low-level feature extraction (e.g. color histograms), are focusing on narrow aspects of isolated collections of multimedia data, and are dealing only with single media types. MISTRAL follows the following lines of radically new research: MISTRAL will extract a large variety of semantically relevant metadata from one media type and integrate it closely with semantic concepts derived from other media types. Eventually, the results from this cross-media semantic integration will also be fed back to the semantic extraction processes of the different media types so as to enhance the quality of the results of these processes. MISTRAL will focus on most innovative, semantic-based cross-media exploration and retrieval techniques employing concepts at different semantic levels. MISTRAL addresses the specifics of multimedia data in the global, networked context employing semantic web technologies. The MISTRAL results for semantic-based multimedia retrieval will contribute to a significant improvement of today's human-computer interaction in multimedia retrieval and exploration applications. New types of functionalities include but are not limited to cross-media-based automatic detection of objects in multimedia data: For example, if a video contains an audio stream with barking together with a particular constellation of video features, the system can automatically consider the features in the video as an object 'dog'. semantic-enriched cross-media queries: A sample query could be 'find all videos with a barking dog in the background and playing children in the foreground'. cross-media synchronisation: The idea is to synchronize independent types of media according to the extracted semantic concepts. For example, if users see somebody walking in a video, they should also hear footfall from an audio.The amount of multimedia data available world-wide and its network-based linkage will continue its rapid growth in the foreseeable future: According to a study conducted at the University of California, Berkeley an amount of 800 Megabyte of new data is created per year and per capita. The same study estimates that of a total amount of 5 Exabytes of information available world-wide, about 92% exists in electronic form, with a 170 Terabyte share being available on the Internet. Image, video and audio data is becoming the predominant form of information forming this global asset. This deluge of multimedia data calls for new semantic extraction and most innovative retrieval and exploration techniques. Awareness of the importance of semantic extraction and semantic-based retrieval is slowly beginning to enter mainstream business thinking, as evidenced in a recent study of Gartner which notes that "Contrary to many enterprises' expectations, search technology hasn't settled into a stable commodity sold by a few giant enterprises" and advises enterprises to "select products with robust functions to examine semantic structures in corpora". In order to produce systems that live up to the high expectations, substantial and highly innovative research is still required. Taken together, these

two points mean that the first implementations of the envisaged system within organisations can be expected within the next 3 to 5 years (because the market will be ready by then and semantic technologies will not be ready before then), and rollouts on a larger scale in a 5 to 8 years time frame. (Parts of the mistral description is taken from (Mistral Project, 2005) an working documents)

A.2 Relevant Existing Results and Methods

A.2.1 General Information

The purpose of this document is to summarize projects, methods and techniques which may be relevant for Mistral. Every project partner may contribute to the document by adding sections.

A.2.2 M4 Project

The overall objective of the M4 (multimodal meeting manager) (M4 Project, 2005) project is the construction of a demonstration system to enable structuring, browsing and querying of an archive of automatically analysed meetings. The archived meetings will have taken place in a room equipped with multimodal sensors.

- Word-level transcription (word error rate of 20-50%)
- Recognition of gestures and actions
- *No text extraction unit*
- Multimodal identification of person, intent (and emotion)
- Source localization and tracking
- Multimodal Integration (HMM, DBN approaches)
- Construction of a demonstrator system for browsing and accessing information from an archive of processed meetings.

IDIAP File Server A public corpus of 60 annotated short meetings is available at the IDIAP File Server . (60 meetings with 5 minutes duration make 5 hours of annotated multichannel meeting data) The protected corpus includes longer annotated meetings and an annotated video from the dutch parliament. It is only available for the project M4 partners. See also the documentation. The dataset was originally described in (McCowan et al., 2003), a hidden markov model (HMM) approach to group action recognition.

Used Meeting Setup The available information on the server includes:

- Video
  - 3 channels (files)
  - PAL (720x576x25)
  - DivX 5.1.1 (.avi, 800kbps, 30MB per file)
  - RealMedia (.rm8/.rm9)
- Audio

  - 8 circular array microphones
  - 4 lapel microphones
  - PCM format (.wav, 16kHz, 16bit)
- Pdf-Slides
- Xml/html files
  - group actions
    - discussion
    - monologue1
    - monologue2
    - monologue3
    - monologue4
    - notetaking
    - presentation
    - whiteboard
  - person position
  - emotion
  - interactivity
  - activity
  - physical state
- speaker segmentation
- words transcription
- Speech annotation (annotated words, ASR output)

Different collections of video / audio can be played from the server by selecting the appropriate stream sources. A SMIL file is generated. SMIL, a W3C Recommendation, is a XML-based language for positioning and synchronizing multimedia contents. '.smil' files can be played by RealPlayer. Annotation of video streams is used for providing a ground truth for the speech, gesture, face, and object recognizers. Second it can be used as 'ideal' outputs of the recognizers for testing multimodal integration. The annotation was performed using the Anvil Software. (see screen shot below)

BRNO parabolic mirror conference system The BRNO parabolic mirror conference system has only 4 lapel microphones, 1 parabolic mirror camera, and is annotated with whole sentences. 189 minutes of annotated meeting data is publicly available.

EPFL mobile USB camera The EPFL mobile USB camera can track the face position automatically.

A.2.3 IBM's MARVEL

MARVEL (described in the whitepaper) is very close to MISTRAL, but they are focusing on *Video only*. Text is extracted from Videos only. No separate repository of text data is used. MARVEL assumes a given training set. For now its not clear if MARVEL supports Training set selection. MARVEL focuses on speech rather than audio/sounds. Goal: 'The MARVEL system uses multimodal machine learning techniques for bridging the semantic gap for multimedia content analysis and retrieval. MARVEL

automatically annotates multimedia making it possible to later search and retrieve content of interest. The objective of MARVEL is to help the media industry, including stock photo/video and broadcast companies, as well as libraries, organize large and growing amounts of multimedia content much more efficiently and automatically.'

- Data Base
  - Focus on Video Data (News)
  - Text is extracted by transcription.
- Tasks
  - Annotation
  - Retrieval
- Approach
  - Automatic Video Metadata Extraction (MPEG-7) by extracting text (speech, closed captions, transcript), visual features and semantic concepts using statistical models (Taken from Marvel Whitepaper)
  - Create a semantic library by learning from training data. Learning is performed with text, visual and audio as input. Human interaction is needed during training process. Exploiting Relationships: e.g. confidence of detecting an Airplane is boosted by the finding that a scene is outdoor.

A.2.4 Prous Science MAVS

MAVS (Multilingual Audio Visual Search) is design for searching in archives of scientific content. MAVS allows the location of words, phrases and concepts within any multimedia content, typically videos. The results are produced similar to a web search, with a list of hits and rankings. By clicking on a link, the user gets the exact moment in the video.

Although the system is said to be capable of handling different type of data (see picture) the focus is on voice recognition technology that indexes multimedia content without a specific acoustic training to match a given speaker's voice. MAVS performed best of its class in a TREC 2002 conference sponsored by NIST & DARPA to foster research in Information Retrieval and allow cross comparison between different systems. For video search shots were requested with specific/generic: people (George Washington, football players), things (Golden Gate Bridge, sailboats), locations (overhead views of cities), activities (rocket taking off) and combinations of the above (people spending leisure time at the beach, microscopic views of living cells, locomotive approaching the viewer).

Features of the system are:

- **Speaker independence**
  There is no need for specific acoustic training to match a given speaker's voice.
- **Vocabulary independence**
  There is no need to incorporate names or terms specific to a given professional field into the system before it can recognize them.
- **Full-text indexing**
  The system indexes everything that is said and not just specific keywords.
- **Low system overhead**
  The system can operate on a medium-class personal computer (Pentium III with 512 MB RAM).

- **Accuracy**
  The word recognition rate is over 95% on standard conference content that has not been recorded in a professional studio without need of speaker training.

Ease of adaptation to different languages: English and Spanish are the first languages available, Japanese is currently under development.

For speech-to-text capability the system won following prizes:

- Wall Street Journal Technology Innovation Winners 2004 in multimedia category
- Winner of European IST (Information Society Technologies) Prize 2004

A.2.5 2M2Net Framework

2M2Net is a generic framework for multi-modal retrieval of multimedia data (text, image, video and audio) in multimedia digital libraries. The retrieval is conducted based on the integration of multi-modal features including both semantic keywords and media-specific low-level features. 2M2Net framework progressively improves its retrieval performance, by applying the learning-from-elements strategy to propagate keyword annotations, as well as the query profiling strategy to facilitate effective retrieval using historic information of the previously processed queries. Each multimedia document is represented by its semantic skeleton, which maintains the metadata of both high-level semantics and low-level features for each element in the document. Each document and media object has a list of weighted keywords attached to it as their semantic annotation. When a document is pre-processed semantic analysis is performed to obtain the keyword annotation. This is straightforward for a textual element, but semantics of non-textual object (image, video) cannot be extracted from its content and must be acquired indirectly using a heuristic: in a multimedia document, the elements of a document are semantically correlated to each other, so that semantics of a non-textual element can be inferred from related textual elements. Besides this intra-document correlation, there is also inter-document semantic correlations, which widely exist in digital libraries, indicated by structural neighborhood and hyperlinks between documents or media objects. Learning-from-elements strategy is used to propagate and improve the keyword annotations progressively and interactively during relevance feedback. It is a kind of semantic feedback, because it is triggered when the user submits a set of documents or media objects as feedback examples for a given query. In addition to keyword-based retrieval the system also supports feature-level retrieval. The user submits a media object as the query example, and results are retrieved based on the similarity of low-level features of each media object (such as colour and texture feature for image, structural and motion feature for video, etc.), which are extracted in the pre-processing phase. Certain media specific feedback techniques are also incorporated in the framework. Query profiling is the counterpart of learning-from-elements strategy at the feature level (details are out of the scope of this document). It is an incremental feedback technique that memorizes the history of previous user feedbacks to improve the processing of future queries. Semantics keywords and low-level features are seamlessly integrated throughout the whole working flow of the framework to enhance performance by combining the two search paradigms: keyword search and low-level feature-based search (search by example). For example when one keyword search yields just one or few good matches feature-based search with 'good' objects is started to receive more similar objects (this also works when the roles of keyword search and feature based search are exchanged). This second pass provides many results, which are typically not very precise and the system must collect feedback from the user and conducts feedback process in parallel at the semantic level and the feature level to propagate semantic keywords and readjust weights of keywords and features.

A.2.6 Grounding Language: Learning, Generation, and Understanding

In (Roy, 2005) a theoretical framework (together with some implementation details based on the construction of numerous grounded language systems) about language learning, generation and understand-

ing is outlined. The framework concentrates on concrete semantics (as opposed to abstract semantics), motivated by two observations:

- The language acquisition of children begins with conversation about their concrete environment. Semantics develops from the ground up.
- Communication is based on the alignment of conceptual systems across agents, which rests upon a shared external reality.

The main focus of the framework is the connection of words and the physical environment, and how an agent can understand speech acts about the environment. This work and previous ones may be of interest for MISTRAL's merging of semantic concepts.

'The relationship between words and the physical world, and consequently our ability to use words to refer to entities in the world, provides the foundations for linguistic communication. Current approaches to the design of language processing systems are missing this critical connection.' (Roy, 2005)

'Language processing systems that rely on human mediated symbolic knowledge have no way to verify knowledge, nor any principled direct way to map language to physical entities in the world. An NLP system that is told what the world is like will fail in the same ways that we know that a robot will fail.' (Roy, 2005)

A.2.7 Other Systems

Google Video Search Google does a continuous search of videos on the internet by applying ASR (Automatic Speech Recognition) engine on the audio-video tracks and transcribing what was said during the video. Then, through Google search, parts of the (TV program)- videos could be retrieved based on keywords.

Yahoo Video Search The indexing is done based on the available textual description (mostly file-name) of the video files. No Video-, neither ASR is applied in this case.

Blinks Basically the same as Google, but with worse recognition rate. While Google applies its engine on the whole American TV program, Blinks only have apps. 5-6 channels with TV documentaries and news broadcast.

A.2.8 Internet Databases and Benchmarking

WP3 Benchmarking The WP3 Benchmarking is a work package of the MUSCLE NoE, a project involved in automatic extraction of semantic information from multimedia data. The WP3 work package aims to develop objective methods for comparing these algorithms, and to encourage the use of these methods.

- **Provided software**
  Tools for annotating images and segmenting videos.
- **Provided data**
  Two image datasets (coins, planar objects) and several video datasets (e.g. Basketball, Gestures, TV, Tracking evaluation).

Data is available only after registration, and the amount of provided data is limited.

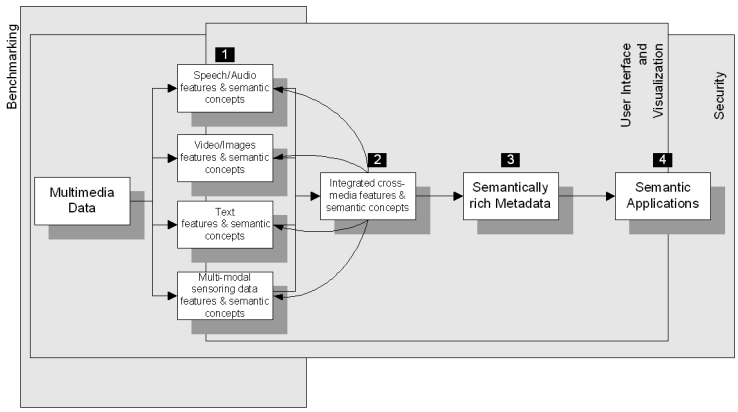

Figure A.1: Block diagram of the mistral services.

TRECVID The TRECVID data repository provides video data used in the TRECVID workshops. Part of the data is public (up to TRECVID 2002). Data of 2003/2004 workshops is available for a fee of 2000/20000 Dollars for non-profit/profit organisations.

The TREC conference series is sponsored by the National Institute of Standards and Technology (NIST) with additional support from other U.S. government agencies. The goal of the conference series is to encourage research in information retrieval by providing a large test collection, uniform scoring procedures, and a forum for organizations interested in comparing their results. In 2001 and 2002 the TREC series sponsored a video ”track’ devoted to research in automatic segmentation, indexing, and content-based retrieval of digital video. Beginning in 2003, this track became an independent evaluation (TRECVID) with a 2-day workshop taking place just before TREC.’.

The homepage simply points to the Open Video Project where the videos are available. Furthermore, annotations and some evaluation tools related to the actual tasks are provided.

IDIAP MMM File server This is a database for multimodal meeting data with relations to the M4 project. Part of the data is public. The database contains lots of video, audio, and text data of small meetings. Annotations are also provided (e.g. Who speaks when, automatic transcription of audio). There is also data from IDIAP meetings available. The meeting corpus is also described in (McCowan et al., 2003). The audiovisual data is provided in several formats. Additionally pdf slides, text and annotations are provided.

A.3 Mistral Architecture

Figure A.1 shows a block diagram of the mistral services.
Horizontal Services:
From the multimedia corpus audio-, video-, and text features are extracted. (1) From these features, multimodal features and semantic concepts are extracted. (2) From these features and concepts, semantically enriched concepts are derived. (3) These concepts are then exploited in semantic applications.
Vertical Services:
The extracted features are benchmarked, in order to evaluate the performance of the algorithms. Security is applied to restricted parts of all services. User interfaces are built for both data extraction and application domains.

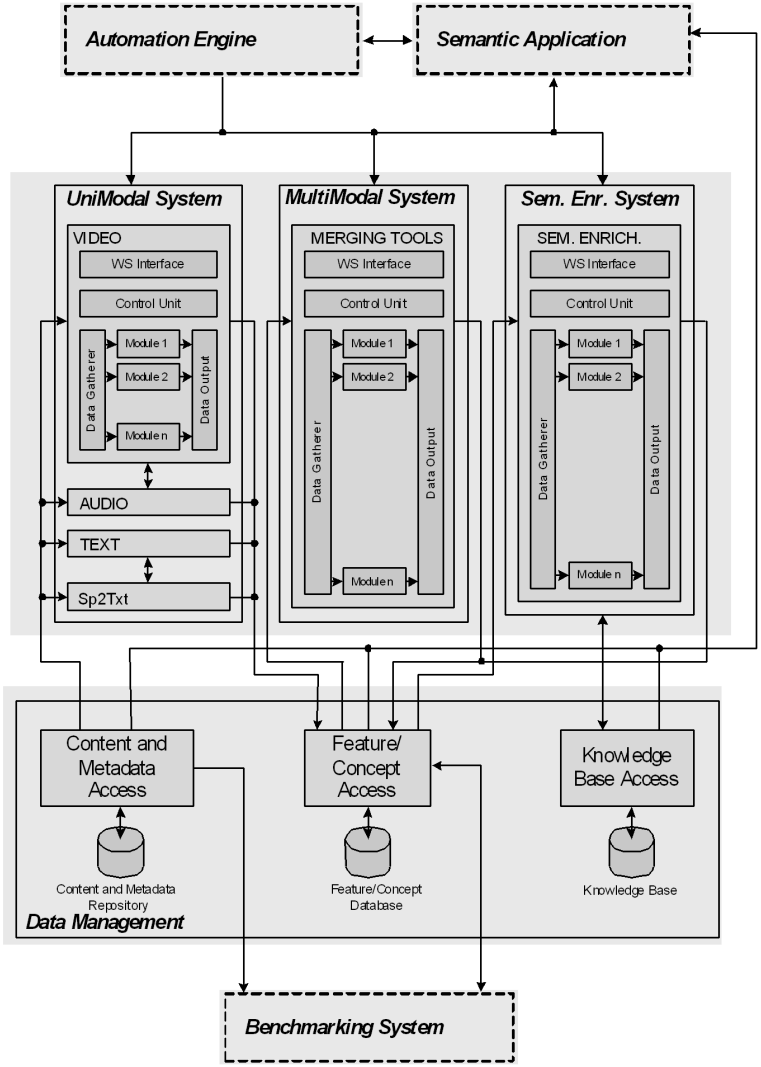

Figure A.2: Architectural diagram of the mistral units.

Figure A.2 shows the architectural design of the mistral system. The unimodal system, the multimodal system, the semantic enrichment system, with pluggable units each are depicted. Information is stored in the data management layer. Three repositories for data/metadata (input data), features (extracted features/concepts), and a knowledge base for semantic enrichment have to be established. The benchmarking system measures the performance of the extraction units. The automation engine controls different processes like training and testing. Semantic applications use the stored knowledge and provide user feedback.
A complete position paper of the mistral project is available. (Sabol et al., 2005)

Bibliography

Addis, T. R., Visschera, B.-F., Billinge, D., and Gooding, D. C. (2005). Socially sensitive computing, a neccesary paradigm shift for computer science. *Grand Challenge in Non-Classical Computation International Workshop*. online. 17, 18

Berndt, D. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *KDD Workshop on Knowledge Discovery in Databases*, pages 359–370, Seattle, WA, USA. 43

Boujemaa, N., Fauqueur, J., and Gouet, V. (2003). What's beyond query by example? *Project IMEDIA - INRIA*. online.

CATS (2004). Time series prediction competition - the cats benchmark. International Joint Conference on Neural Networks, online. 10

Chen, Lei (2005). Literatures on similarity-based time series retrieval, online article collection. online. 6

Chiu, B., Keogh, E., and Lonardi, S. (2003). Probabilistic discovery of time series motifs. In *9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 493–498, Washington, DC, USA. online. 3

Cook, W. R. and Ibrahim, A. H. (2005). Integrating programming languages & databases. what's the problem? *Submitted for Publication*. online. 23, 24

Dainton, B. (2000). *Stream of Consciousness; Unity and Continuity in Conscious Experience*. Routledge. 27

Fergus, R., Perona, P., and Zisserman, A. (2005). A sparse object category model for efficient learning and exhaustive recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, USA. online. 27

Gatica-Perez, D., McCowan, I., Barnard, M., Bengio, S., and Bourlard, H. (2003). On automatic annotation of meeting databases. In *IEEE International Conference on Image Processing (ICIP)*, Barcelona, Spain. 6

Geurts, P. (2002). *Contributions to decision tree induction: bias/variance tradeoff and time series classification*. PhD thesis, University of Liege, Belgium. online. 51

Grandjean, N. (2004). Multimodal sensory integration: Questions and suggestions. NoE HUMAINE Workshop - From Signals to Signs of Emotion and Vice Versa, online. 14

Hibernate (2005). *Hibernate Reference Documentation*. online. 24

Hornick, M. F. (2004). Java data mining (jsr-73) final release published. *DM Direct Special Report, October 5, 2004 Issue*. online. 24

Itakura, F. (1975). Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 23*, pages 67–72. online. 43

Kadav, A., Kawale, J., and Mitra, P. (2003). Data mining standards. *Data Mining Grid digital library*. online. 24

Keogh, E. and Kasetti, S. (2002). On the need for time series data mining benchmarks: A survey and empirical demonstration. In *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 102–111, Edmonton, Alberta, Canada. online.

Keogh, E. and Ratanamahatana, C. (2003). Exact indexing of dynamic time warping. *Knowledge and Information Systems, Vol. 7*, pages 358–386. online. 45, 50

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady, 10(8)*, pages 707–710. 4

Lewald, J. (2002). Rapid adaptation to auditory-visual spatial disparity. *Learning and Memory, Vol. 9*, pages 268–278. online. 14, 15

Lewald, J., Ehrenstein, W., and R.Guski (2001). Spatio-temporal constraints for auditory-visual integration. *Behavioural Brain Research, Vol. 121*. online. 15

M4 Project (2005). M4 project - work programme, online documentation. online. 60

Makhoul, J., Kubala, F., Schwartz, R., and Weischedel, R. (1999). Performance measures for information extraction. In *Proceedings of the DARPA Broadcast News Workshop*, Herndon, VA, USA. online. 4, 6

McCowan, I., Bengio, S., Gatica-Perez, D., Lathoud, G., Monay, F., Moore, D., Wellner, P., , and Bourlard, H. (2003). Modeling human interaction in meetings. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hong Kong, China. 60, 65

McGurk, H. and MacDonald, J. (1976). Hearing lips and seeing voices. *Nature, Vol. 264*, pages 746–748. online. 14

Milch, B., Marthi, B., Sontag, D., Russell, S., Ong, D. L., and Kolobov, A. (2005). Blog: Probabilistic models with unknown objects. In *Proceeding IJCAI-05, Edinburgh, Scotland*, pages 1352–1359. online. 4

Mistral Project (2005). Projektantrag: Forschungsprojekt. mistral (measurable intelligent and reliable semantic extraction and retrieval of multimedia data). 60

Pfeiffer, M., Saffari, A. R., and Juffinger, A. (2005). Predicting text relevance from sequential reading behavior. In *NIPS Workshop on Machine Learning for Implicit Feedback and User Modeling*, Whistler, Canada. online, competition. 9

Ratanamahatana, C. and Keogh, E. (2004). Making time-series classification more accurate using learned constraints. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, Lake Buena Vista, Florida, USA. online. 43, 52

Roy, D. (2005). Grounding language in the world: Signs, schemas, and meaning. *In review, Artificial Intelligence*. online. 63, 64

Sabol, V., Granitzer, M., Tochtermann, K., and Sarka, W. (2005). Mistral - measurable intelligent and reliable semantic extraction and retrieval of multimedia data. In *EWMIT05*. 66

Stein, B. E. and Meredith, M. A. (1993). *The Merging of the Senses*. MIT Press. 15

Thurlow, W. and Jack, C. (1973). Certain determinants of the ventriloquism effect. *Perceptual and Motor Skills, Vol. 36*, pages 1171–1184. 15

TREC (2001). Common evaluation measures. In *The Tenth Text REtrieval Conference (TREC)*, pages A–14–A–23, Gaitherburg, Maryland, USA. online. 4, 8

TRECVID (2005). Guidelines for the trecvid 2005 evaluation, online manual. online. 8

Wallace, M. T., Meredith, M. A., , and Stein, B. E. (1998). Multisensory integration in the superior colliculus of the alert cat. *The Journal of Neurophysiology, Vol. 80*, pages 1006–1010. online. 15

Wallace, M. T., Roberson, G. E., Hairston, W. D., and Stein, B. E. (2004). Unifying multisensory signals across time and space. *Experimental Brain Research, Vol. 158*, pages 252–258. online. 16

Watanabe, K. (2001). *Crossmodal Interaction in Humans*. PhD thesis, California Institute of Technology. online. 15