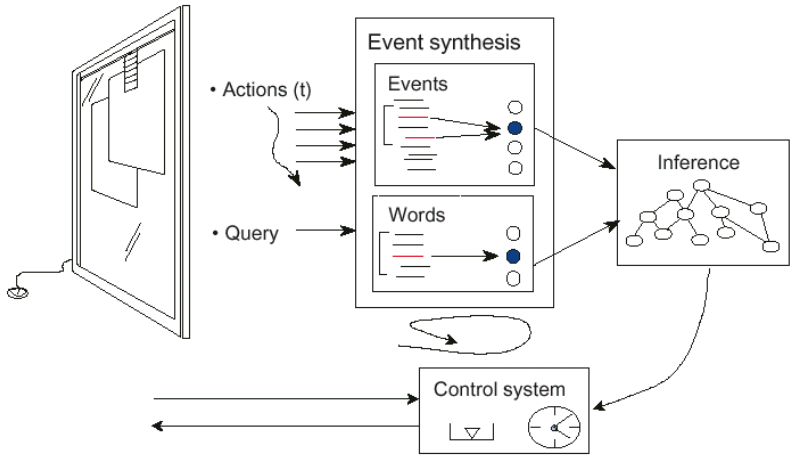
ML & DM Applications



Univ.-Lektor Dr.techn. Alexander K. Seewald
Österreichisches Forschungsinstitut
für Artificial Intelligence

© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

RoboSail

- Autopilot for 1-person sailing: controls sail & rudder
- Includes many state-of-the-art AI and ML components.
- Human jargon as background knowledge, disambiguated by learning from examples.
- Water wave sensors and prediction, wind sensors, GPS navigation...
- Race-proven: Won 2002 Dual Round Britain & Ireland

© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

RoboSail (2)

Shows the complexity of building a system that shows human-like intelligence for a specific domain.

- Three years to build first prototype, another three years for commercial exploitation.
- Human background knowledge plays an important part, but techniques from Machine Learning glues all together.

Human knowledge: *If you are sailing close-hauled and there is a gust of wind then steer the boat a bit windward.*

RoboSail's translation: If the apparent wind angle is between *dLow* and *dHigh* degrees and the apparent wind speed average *meanS* increases by a factor of *f* for more than *t* seconds, then steer the ship *E* degrees windward.

Learn unknown constants from examples!

© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

Spam Filtering

Problem

- Spam : Nonspam = 17 : 1; ~200 spams/day...
- Local installation of SpamAssassin: Combines 900+ regular expression rules plus NaïveBayes learner. Each rule (including NB learner) has a score. Mails are classified as Spam when sum scores exceed threshold.
- Works quite well after extensive fine-tuning. Not feasible to do this for all my colleagues!

⇒**Idea: Formulate fine-tuning as a ML task.** *Multi-view learning*: Mail can be described via NB learner, or via applying 900+ rules. Learn scores via *Linear Regression*, then train misclassified examples via NB, and repeat until convergence, or limit of 13 iterations is reached.

© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

Spam Filtering (2)

Results – Qualitative

- Performance seems even better than hand-tuned system(!)
- Adapted for seven of my colleagues: widely appreciated
- SysAdmin: **Stop,** because this will overload mail server.
  Currently training a single model for the whole institute.

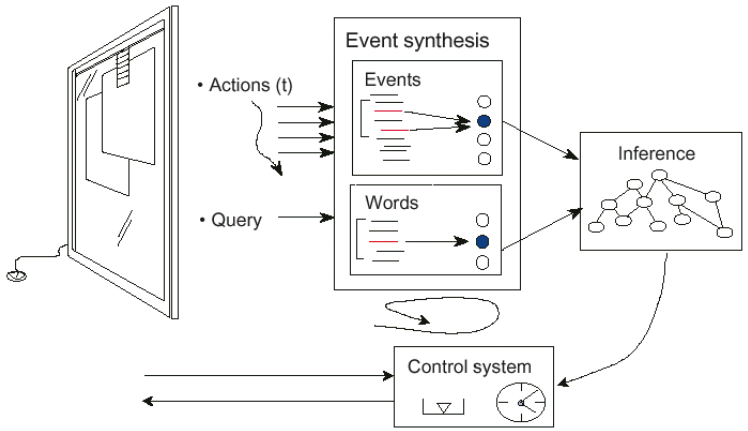
Results – Quantitative

- Fewer hams misclassified than CRM114 (= this year's best paper @ Spam-Conference), but more spams misclassified. Ham/Spam error can be traded via cost factor.
- Training time is comparable; Testing is faster for SA; CRM114 updates faster and needs far less training data.
- Remarkable to have such a simple system perform competitively to such a complex one – but not surprising.

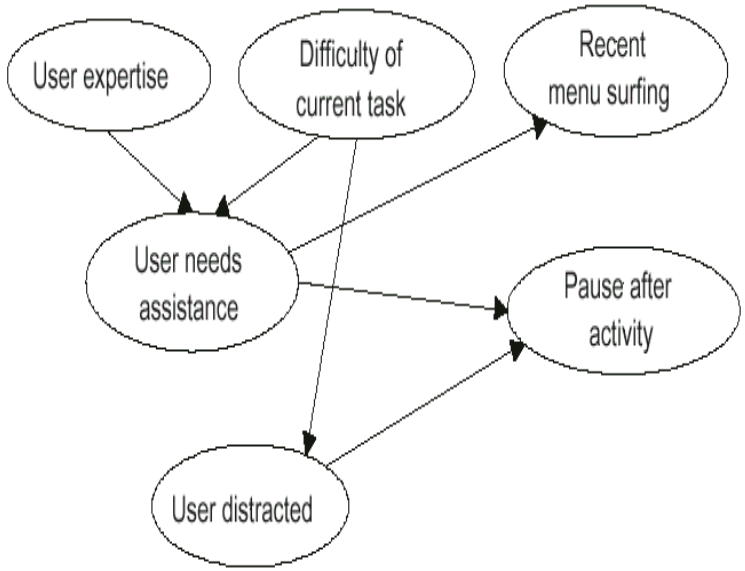
© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

The Lumière Project - MS Office Assistant

- Based on usability research (1998) with a human expert giving advice. Aim: To create automated assistants with similar performance.



- Manually created Bayesian networks to predict User's goals from his Actions and explicit Queries.
- For explicit queries, experts assessed cond. probabilities for words within a query, separately for forty specific goals and 600 words.



© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

The Lumière Project - MS Office Assistant

A good idea! Why it may not work so well...

- Mainly a static model. Does not take specific user into account even if his needs differ. User expertise modelling was removed in final version which worsens this problem.
- *A hard problem:* If you try to achieve something which is not one of the forty modelled goals, the system's advice will always be distracting. The system does not know its limits.
- Most probabilities seem to have been estimated manually and normalized in a complex way. However, humans are notoriously bad at assigning probabilities to rare events.

Microsoft is currently working on utilizing Bayesian learning for spam filtering, where it should work better.

© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

BioMinT: Biological Text Mining

Research project funded by the EU (2003 – 2005)

- Develop a generic text mining tool for content-based and knowledge-intensive information retrieval and extraction
- To be applied to the annotation of the Swiss-Prot and PRINTS proteomics databases with information mined from scientific papers; and to build human-readable reports
- Adapted to the needs of biological researchers in general and specifically for SwissProt / PRINTS annotation.

Useful metaphor: In-silico research / curator assistant

www.biomint.org

© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

The BioMinT Tool

General workflow

- User enters protein / gene name
- Name is looked up in comprehensive Gene and Protein Synonym Database (GPSDB). Selection criteria: species, taxonomic range, source database and source field.

  This expands Name with (almost) all known synonyms.

3. Generate & execute PubMed query with all synonyms.
4. Retrieve references, filter and rank by relevance.
5. Extract information for annotation purposes (PRINTS,SP)

1. & 2. have recently been made available to the public on biomint.oefai.at (BioInformatics Journal - Application Note was recently submitted)

© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

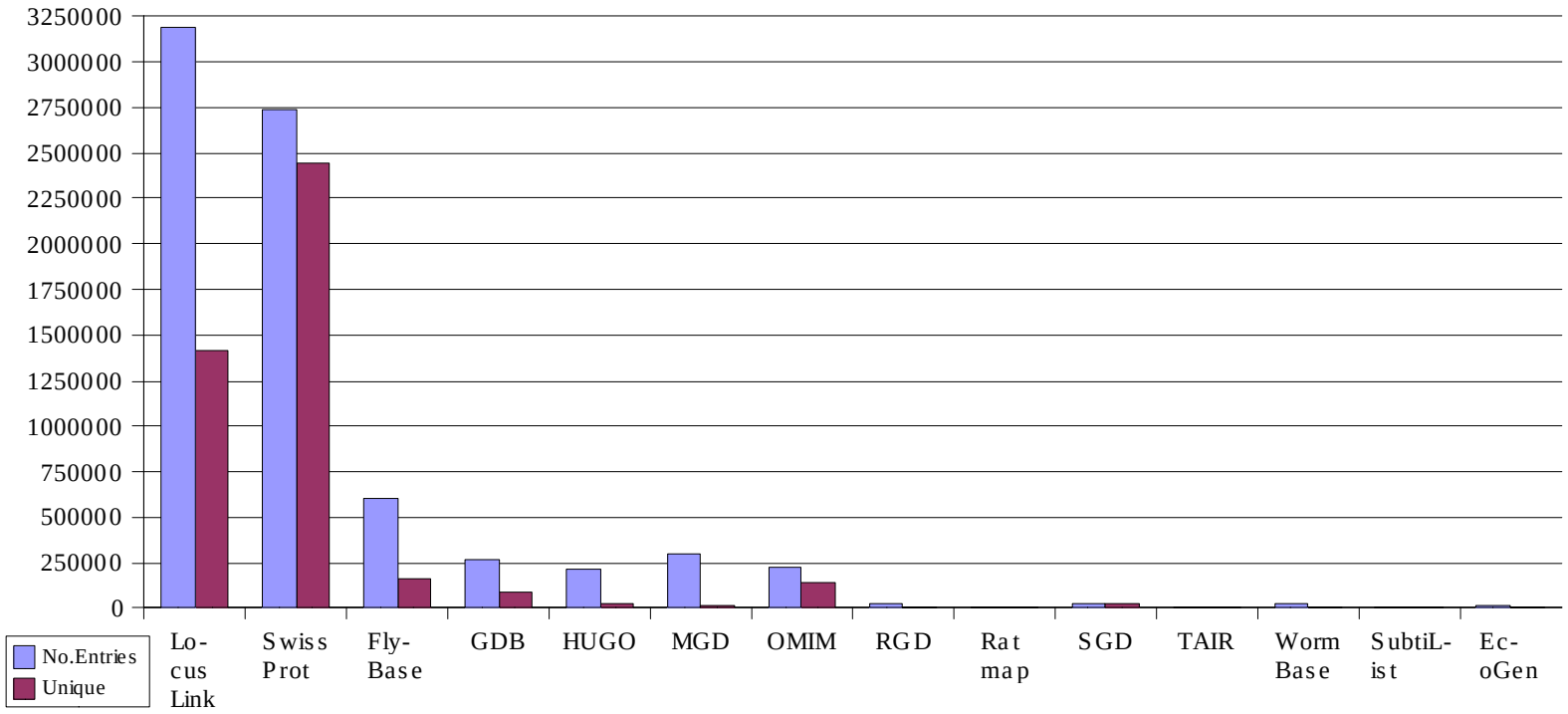
BioMinT - GPSDB

Download all 14 databases according to SIB (+ SwissProt)

Extract all relevant fields & links from each DB separately

Create all pairs of synonyms (noting Source DB, field, ID)

10,059,614 synonym pairs; 562,628 unique names (Dec.04)



© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

BioMinT - Ranking Evaluation

All ranking comparisons are based on the *medical annotation dataset* provided by SIB. This should be sufficiently close to the real-life application for the BioMinT, but only deals with medical (disease-related) annotation of H. sapiens.

	LR	SR	NBR	ORR	RND	LR_M
wt1	0.118	0.118	0.294	**0.353**	0.135	**0.353**
ump s.	**1.000**	**1.000**	0.667	**1.000**	0.629	**1.000**
xpa	**0.750**	0.667	**0.750**	0.510	0.521	0.625
vhl	0.414	0.431	0.586	**0.690**	0.192	0.534
wrn	**0.667**	0.333	0.444	0.333	0.109	0.556
xpc	0.375	0.350	**0.500**	**0.500**	0.311	**0.500**
wfs1	0.727	0.727	**0.818**	0.727	0.647	0.727
GCDH	0.889	**1.000**	0.889	0.778	0.814	0.889
tulp1	0.545	0.556	0.667	**1.000**	0.594	0.550
Avg.	0.549	0.518	0.562	0.589	0.395	**0.637**

Precision/Recall Break-Even Point

	LR	SR	NBR	ORR	RND	LR_M
wt1	0.199	0.164	0.267	**0.366**	0.165	0.364
ump s.	**1.000**	**1.000**	0.333	**1.000**	0.245	**1.000**
xpa	**0.500**	0.333	**0.500**	0.019	0.043	0.250
vhl	0.449	0.407	0.617	**0.677**	0.209	0.604
wrn	0.698	0.438	0.462	0.282	0.096	**0.699**
xpc	0.292	0.171	0.500	0.559	0.106	**0.700**
wfs1	0.874	0.884	**0.930**	0.873	0.700	0.907
GCDH	0.977	**1.000**	0.878	0.792	0.863	0.977
tulp1	0.091	0.111	0.333	**1.000**	0.211	0.100
Avg.	0.564	0.501	0.536	0.619	0.293	**0.622**

Average Precision

Ranking based on word occurrence of *missense* worked very well, so query-dep. rankers performed best. Adding this single word to best q.-indep. ranker (LR) shifts it to the top(!)

Conclusion: Usually better than random shuffling. Occurrence of a single word makes very good ranker(!). More data needed.

© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

BioMinT - Homonymy Recognition

Synonym Group = A group of database entries connected by inter-database links, all dealing with same gene/protein entity.

Homonym = Name which appears in more than one *Syn.Grp*

Each of ten queries was expanded with all synonyms, and then checked for homonyms. All found homonyms were verified by domain experts: *Accuracy*=100%.

Removing homonyms: almost no change in ranking performance by two measures.

Conclusion: Homonyms have little impact, at least for medical annotation.

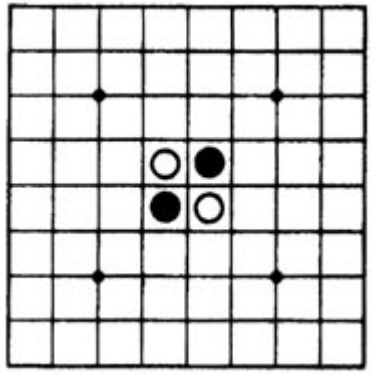
Query	Homonyms
vhl	HRCA1,RCA1
xpc	p125
wrn	RECQL2,RECQL3
tulp1	RP14
wt1	WAGR

© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

Logistello - Logistic Regression for Othello

Othello/Reversi

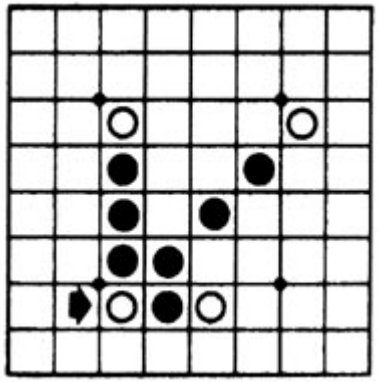
- White and Black stones on an 8x8 board
- In each step, *outflank* an enemy stone
- Repeat until no more moves possible.
- Player with most stones wins.



Standard AI Approach to Game Playing

- State-Evaluation Function: *How good is this position? I.e. how likely am I to win?*
- Alpha-beta Search: *Simulate all reasonable moves, evaluate leaf positions.*

Usual approach to model state-evaluation functions: Assume linear combination of input features with learned weights.



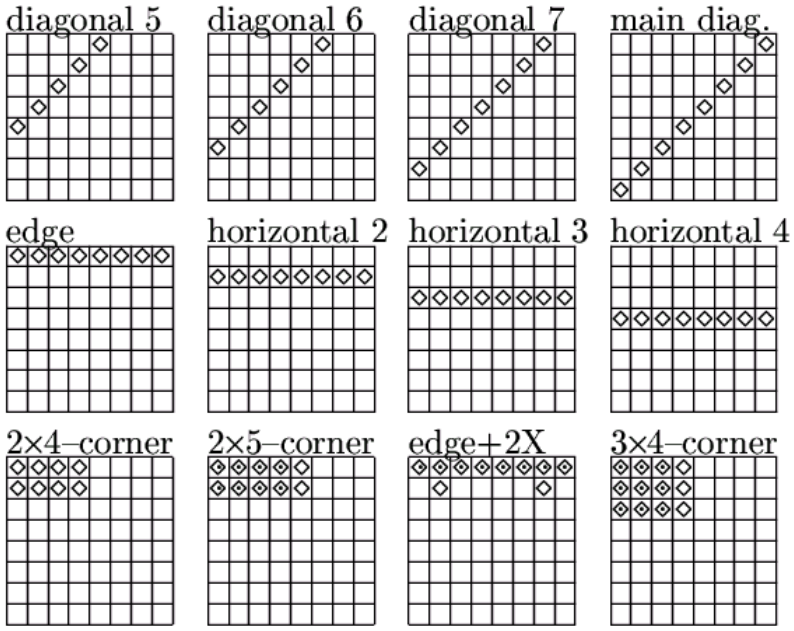
© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

Logistello - Logistic Regression for Othello

Logistello

- One of the strongest Othello programs (as of 2002)
- Evaluation function was learned via Logistic Regression using 2.5 million examples and 1.5 million features. Data from playing with other programs, experts and self.
- Mobility features: *How many actual and potential moves do I have? (vs. opponent)*
- Pattern features: *Diagonals*, *Edges*, *Corners*... (see right)

Experimentation and background knowledge essential.



© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

Natural Language Processing

Classical approach to NLP: Linguists create universal grammar for a given language, which is then used to parse arbitrary sentences (syntactic→semantic→pragmatic level)

Universally applicable, but intractable for real-life tasks.

State-of-the-Art approach: For specific application, create dataset of training sentences with relevant information *marked up* (= corpus); then train ML system to predict relevant information.

Much effort needed to create corpora, but feasible. E.g.

- Speech recognition: Mobile phone speech dialing, dictation systems (e.g. Philips SpeechMagic for Radiologists)
- Text Mining, Inf. Retrieval & Extraction (e.g. BioMinT)

© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

Computer Vision

Computer Vision has previously been investigated by manual programming while state-of-the-art approaches utilize machine learning techniques. Similar to NLP, this is transforming the field and creating new challenges for ML:

- High data volume, high speed processing. For some applications, direct hardware implementation is needed.
- Temporal structure of input/output, e.g. for tracking.
- Amount of (costly!) training data for acceptable accuracy. Unsupervised learning is not working very well right now.
- Integrating background knowledge on scene and patterns.

© Alexander K. Seewald
alex@seewald.at / alex.seewald.at

Current Topics in ML Applications

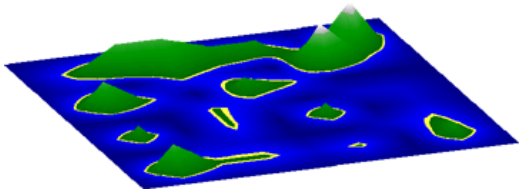
Self-healing Systems (e.g. Solaris 10, MS Zero Admin Kit)

- Very basic – mainly structured log files. The real challenge will be inferring the primary error from thousands of error messages (e.g. telecomm. network: ~ 500,000 errors daily, mostly non-critical misconfiguration and transient errors)

Intrusion / Novelty detection (~One-Class learning)

- Very little training data. Needs new learning systems!
- Potentially useful to design systems knowing their limits.

Music Clustering (e.g. Islands of Music)



- Display songs by similarity (MusicGroup @ ÖFAI)

© Alexander K. Seewald
alex@seewald.at / alex.seewald.at