HowTo ToBIT

Erstes Meeting

Tipps und Hinweise zum Vorgehen

Prof. Dr. Kaspar Riesen

Disclaimer

- Dies ist kein offizielles Dokument des Instituts fur Wirtschaftsinformatik (IWI) der Fachhochschule Nordwestschweiz (FHNW).
- Es handelt sich um eine persönliche Empfehlung, die ich Ihnen in meiner Funktion als betreuender Dozent weitergebe.
- Es ist anzunehmen, dass andere betreuende Dozenten am IWI einige Hinweise, die in diesem Dokument gemacht werden, anders als ich beurteilen.
- Ich empfehle Ihnen deshalb unter keinen Umständen, das vorliegende Dokument zu verwenden, falls Sie Ihre Arbeit nicht bei mir verfassen.

Key dates

17.09.19	themes handed to students	
Until 22.09.19 (23:59)	choose your 5 preferred themes	**Heute**
from 26.09.19	contact your supervisor	**!**
from 26.09.19	**1. group meeting**	
25.10.19	proposal in pdf format to your supervisor	
28.10.-31.10.19	**2. group meeting** (proposal presentation)	
25.11.-29.11.19	**3. group meeting**	**Do, 31.10 13.15Uhr**
01.12.19	last chance to cancel registration	**Fr, 29.11 13.15Uhr**
31.12.19 (23:59)	upload report in pdf format to moodle	
t.b.a.	**presentation** (Jan/Feb 2020)	
1 day in advance	presentation in pdf format to supervisor and academic writing coach	

Results

- **Proposal** (comprising summary, outline and references) written in English and presented
- **Report** written in English with **6000-7000 words** (not including cover sheets, table of content, bibliography. etc.)
- **Presentation** in English and last **20 minutes**
- (Presentation slides to supervisor one day before presentation)

Grading

- **Admission Requirement:**
  - Proposal
  - Participation in 2 additional presentations

all

- **Rated results**:
  - Report
  - Presentation

\+ Proposal

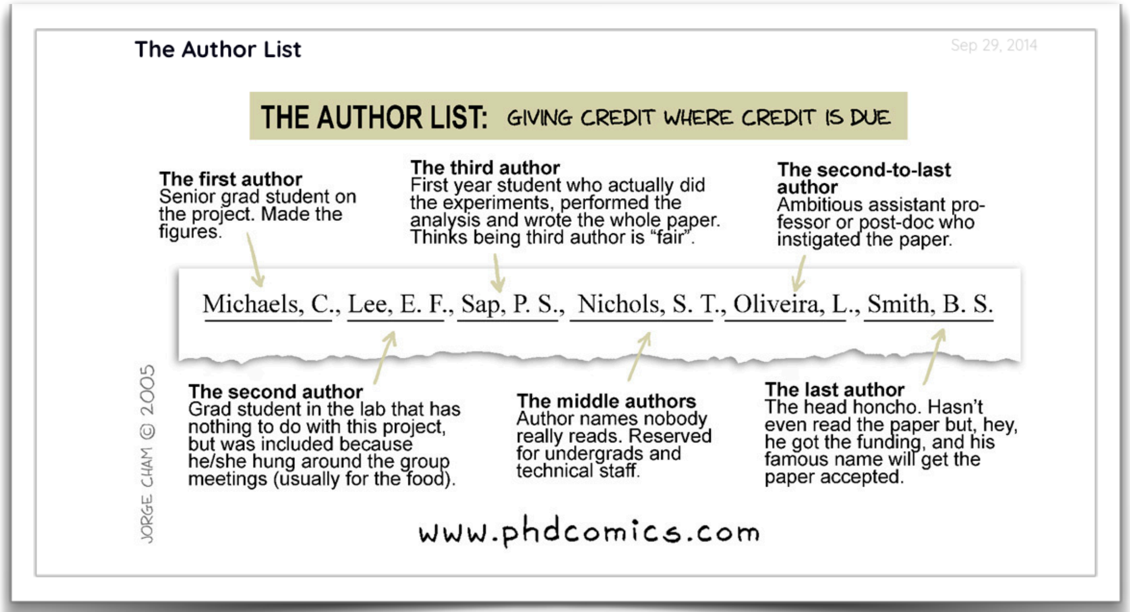
- **Total score ToBIT-Modul (Curriculum 2017)**:
  - ToBIT seminar 80%
  - Academic Writing 20%
- **Total score ToBIT-Modul (Curriculum 2012)**:
  - ToBIT seminar 100%
  - Academic Writing (Communication 4) – separate Grading

Paper Suchen

- Mindestens 20 Referenzen auf Artikel aus Fachzeitschriften/ *Proceedings* oder Fachbüchern
- Keine oder nur sehr wenige Referenzen auf Weblinks oder Vorlesungsunterlagen
- Suche über…
  - https://dblp.uni-trier.de
  - https://www.sciencedirect.com
  - https://www.worldscientific.com
  - https://www.springer.com/de
  - https://www.computer.org/csdl/
  - https://scholar.google.ch

Paper Auswählen

- Maximal 40 Referenzen
- Sortieren Sie die Suchergebnisse:
    - Relevanz
    - Anzahl Zitierungen
    - Jahrgang
- Achten Sie darauf, dass Sie *alle* relevanten Forschungsgruppen berücksichtigen!

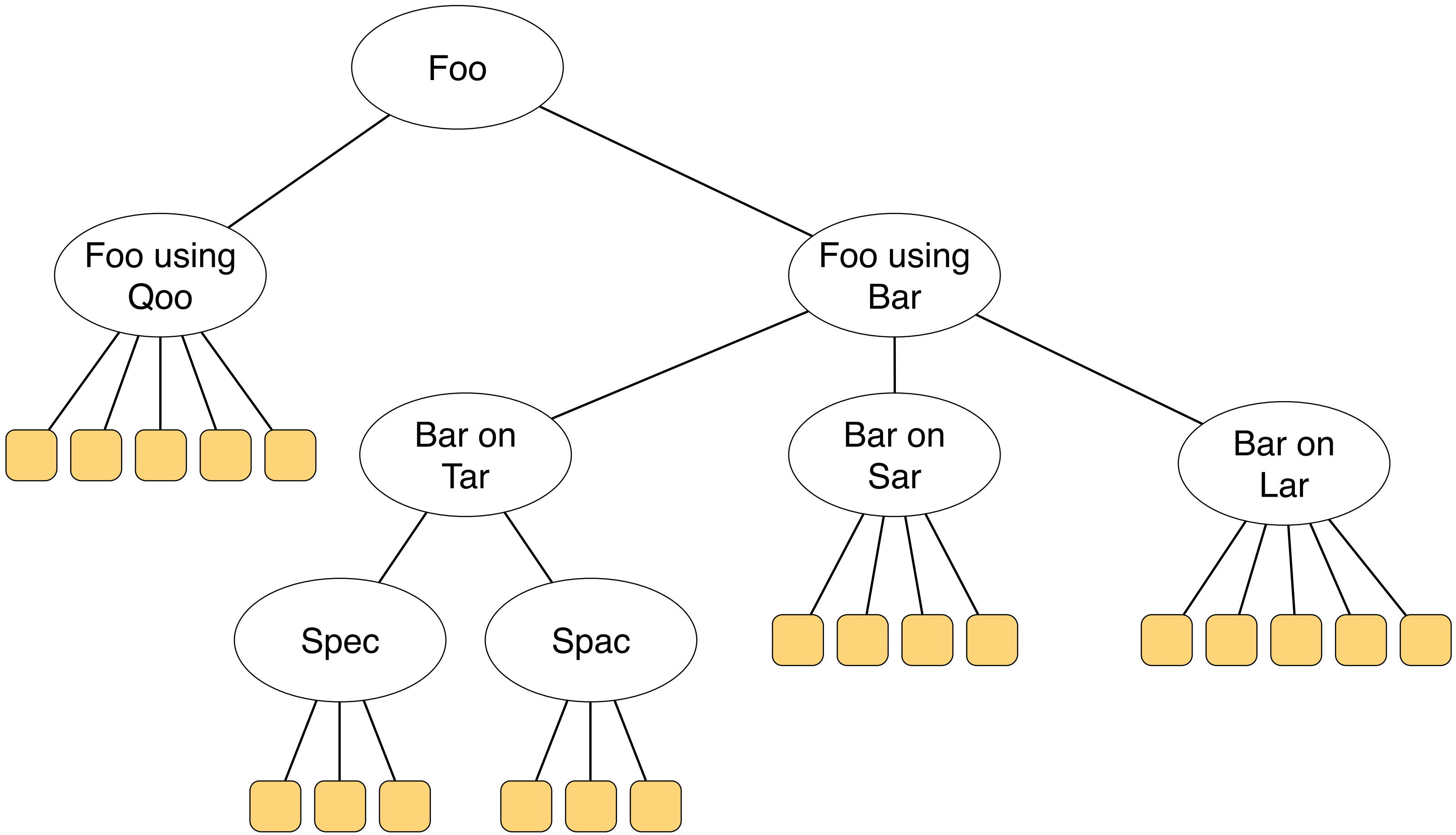


Paper Sichten

„Entschlüsseln“ über https://sci-hub.se (wechselt immer wieder: https://whereisscihub.now.sh)

Paper Sortieren I/II

Gruppieren Sie die Paper (Bottom-up vs.Top-Down):



Paper Sortieren II/II

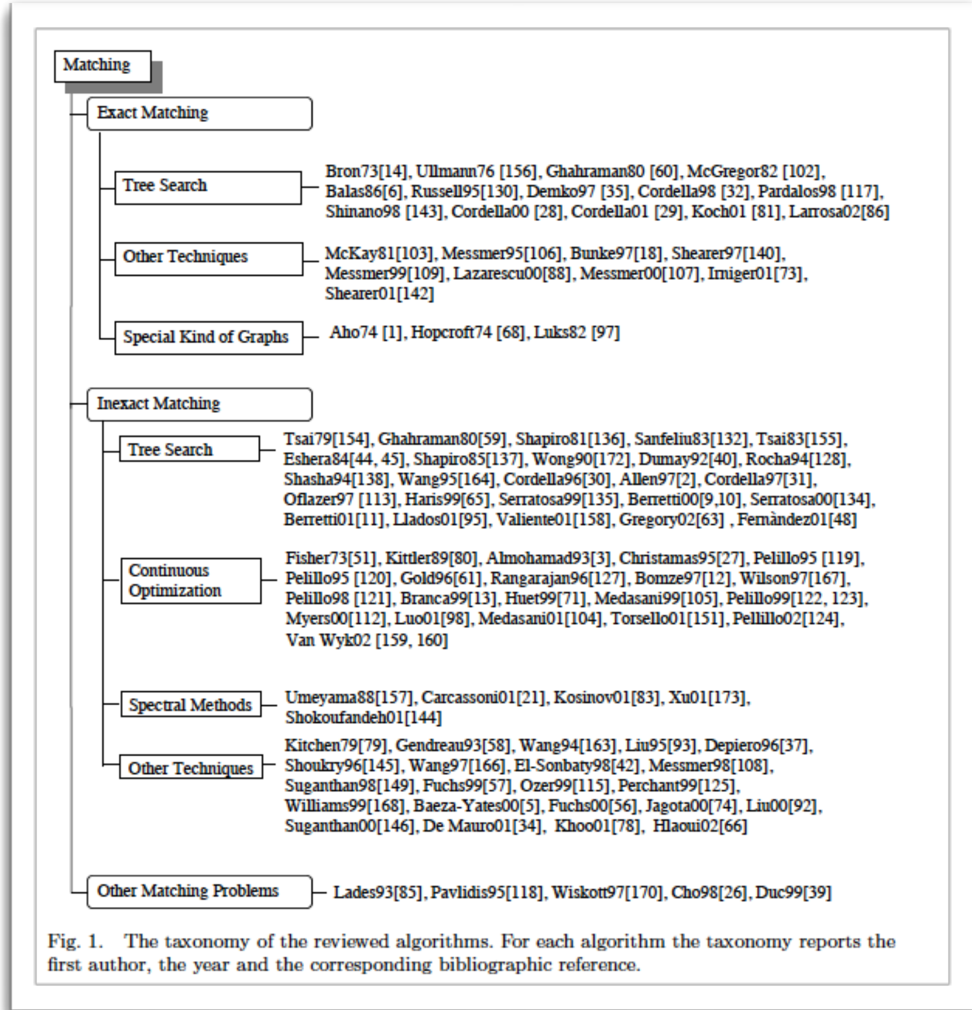


Fig. 1. The taxonomy of the reviewed algorithms. For each algorithm the taxonomy reports the first author, the year and the corresponding bibliographic reference.

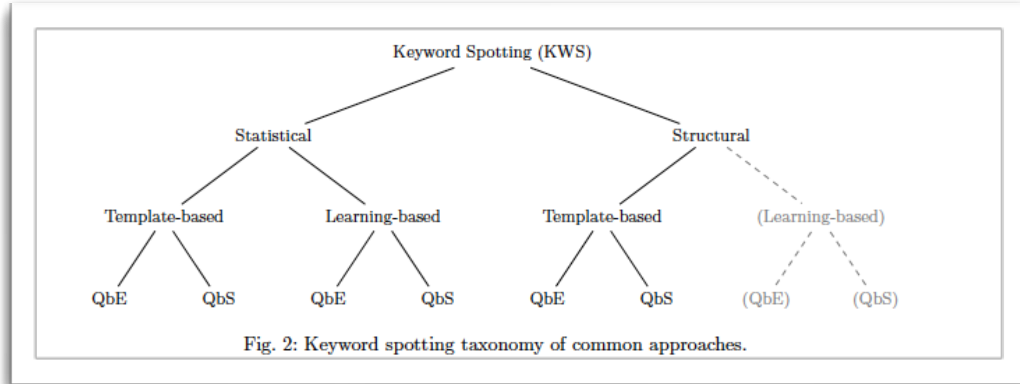
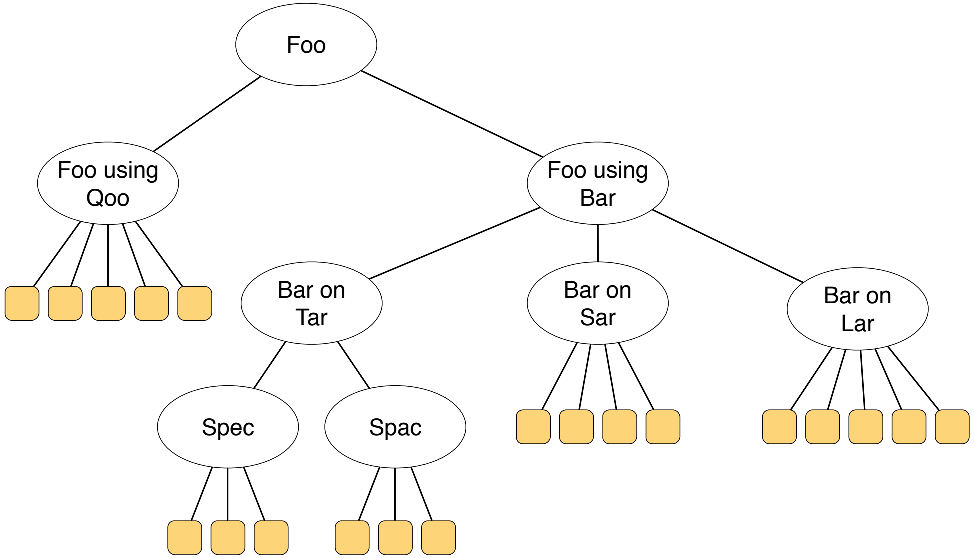
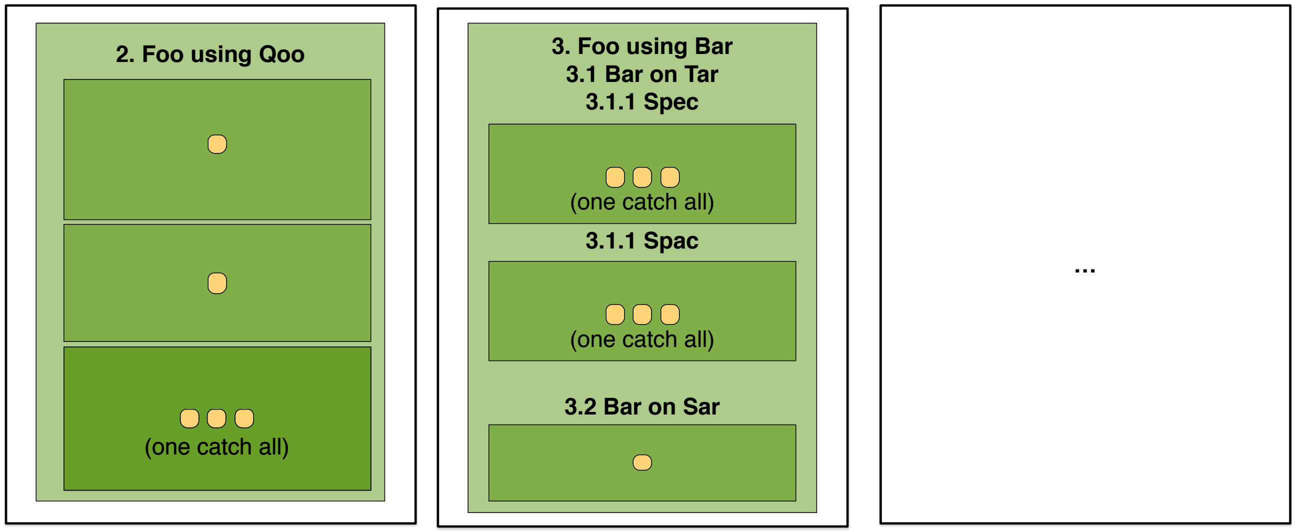


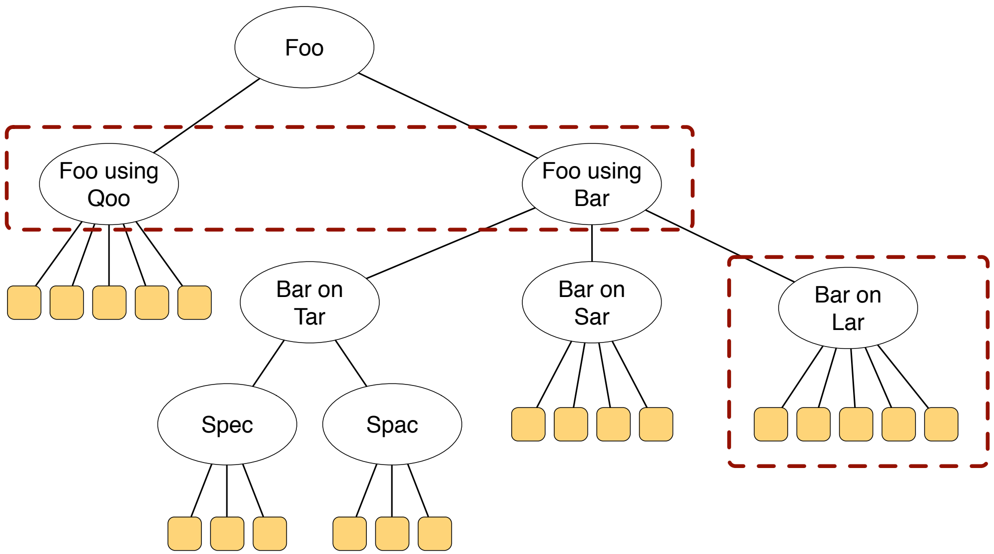
Fig. 2: Keyword spotting taxonomy of common approaches.

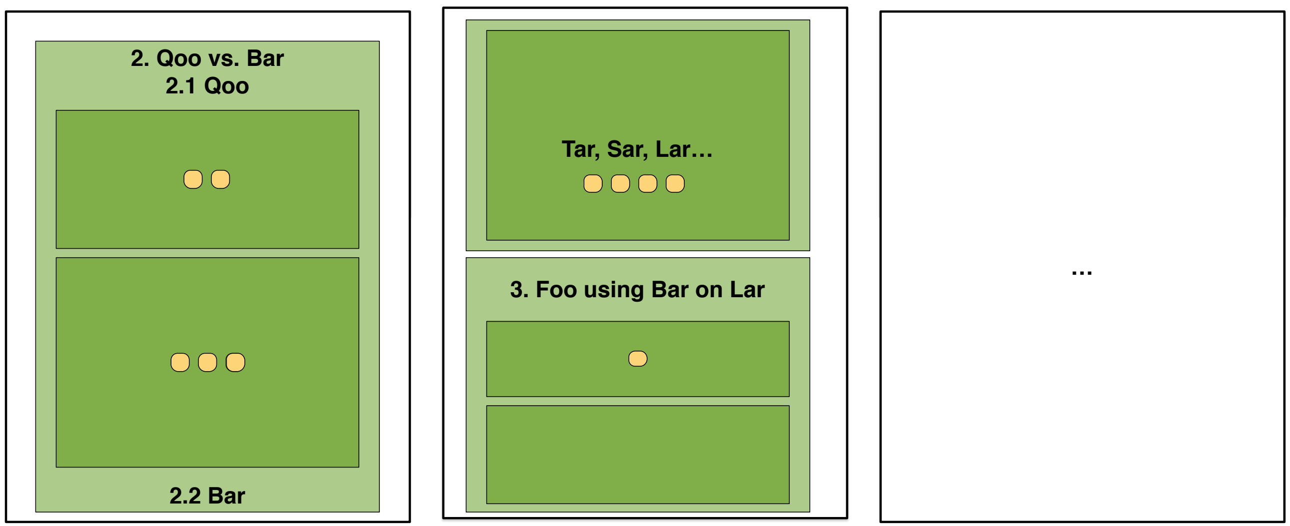
Survey Strukturieren I/III





Survey Strukturieren II/III





Survey Strukturieren III/III

Grundsatz:

Balancieren & Begründen

Proposal (25.10)

- Inhalt:
  - Taxonomie als Baumstruktur
  - Vorläufige Auswahl an *Papers* den korrekten Blättern des Baumes zugewiesen
  - Vorschlag für die Struktur des *Papers* (Fokus auf *Survey*) als ausführlich kommentiertes Inhaltsverzeichnis (inkl. Referenzen)
  - Optional: Weitere Ideen/Inhalte (z.B. ein *Vergleich* oder ein *Experiment* oder eine *Beschreibung einer Anwendung* oder ein *Tutorial für ein Tool* oder …). Zentrale Frage: Wie wollen Sie diese Inhalte mit Ihrem *Survey* verknüpfen?
- Am zweiten Meeting (31.10) präsentieren Sie Ihr Proposal in einer geeigneten Form (Feedback aus der Gruppe und von mir) …
- … und **danach** beginnen Sie mit dem Schreiben des *Papers*….

Tipps beim Schreiben

- Sprache: einfach, präzise und neutral (keine komplizierte und umständliche Satzbildung, keine saloppe, blumige, euphorische Ausdrucksweise). Ich empfehle Ihnen, die gesamte Arbeit im Präsens zu verfassen.

- Fachbegriffe definieren und einführen, bevor Sie diese in Ihrer Arbeit verwenden.

- Verwenden Sie nie Abkürzungen ohne diese vorher mit dem vollständigen Begriff eingeführt zu haben (*Support Vector Machines* (*SVMs*) sind Erkennungsmethoden, die …). Haben Sie eine Abkürzung definiert, verwenden Sie diese!

- Bevor Sie mit dem Verfassen eines neuen Abschnitts beginnen, überlegen Sie sich, welches die Hauptgedanken sind (Informationen/Ideen/Methoden/o.ä.), die Sie in diesem Abschnitt beschreiben wollen. Ordnen Sie diese Hauptgedanken so, dass der lineare Aufbau dieser Gedanken sinnvoll ist.

- Achten Sie darauf, dass pro Paragraph maximal *ein* Gedanke erläutert wird. Ist dieser Gedanke zu Ende geführt, wechseln Sie den Paragraphen. Oftmals kann man im letzten Satz eines Paragraphen eine Überleitung auf den nächsten Paragraphen machen oder alternativ im ersten Satz eines Paragraphen auf den letzten Paragraphen Bezug nehmen.

- Machen Sie Paragraphen nicht zu kurz aber auch nicht zu lang: Ein Paragraph sollte mindestens 5 Zeilen bis maximal 20 Zeilen umfassen (dies ist nur eine grobe Faustregel).

- Verwenden Sie ausschliesslich qualitativ hochwertige Bilder und Tabellen (achten Sie auf eine genügend hohe Auflösung). Die Bilder und Tabellen sind zwingend zu nummerieren. Beachten Sie, dass jedes Bild und jede Tabelle im Fliesstext erwähnt werden muss (*In Abbildung 17 wird verdeutlicht, dass…* oder *In Tabelle 3 sind die Vor- und Nachteile des … zu sehen*, etc.).

LaTex vs. Word

A First Step Towards Exact Graph Edit Distance Using Bipartite Graph Matching

Miquel Ferrer[1], Francesc Serratosa[2], and Kaspar Riesen[1]

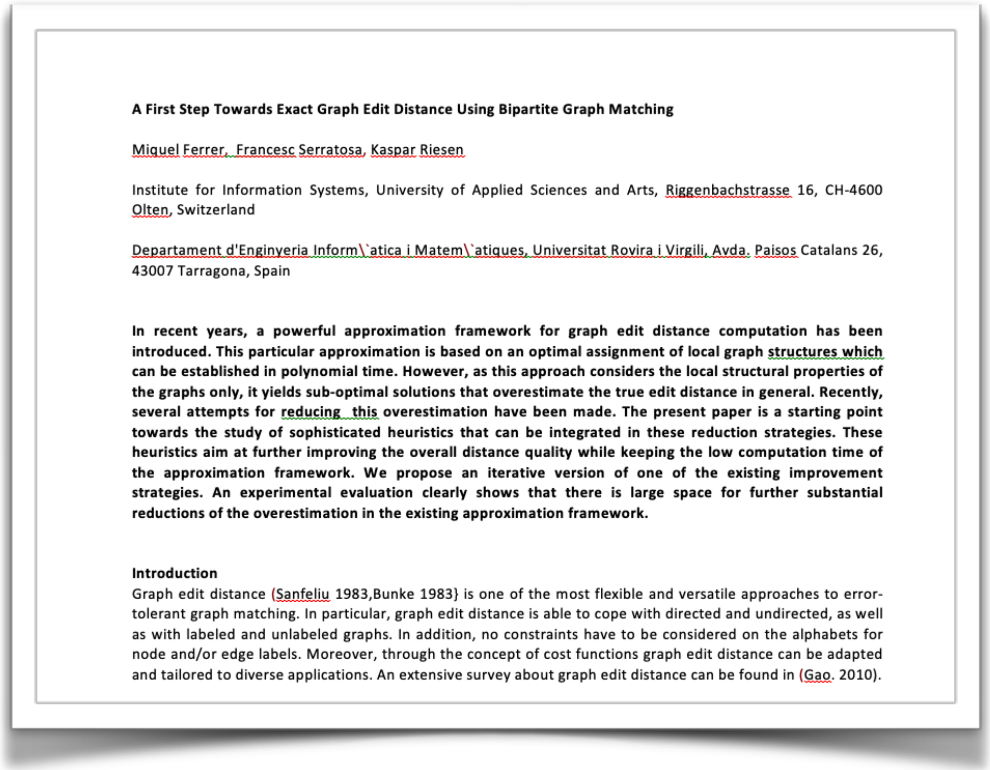
[1] Institute for Information Systems, University of Applied Sciences and Arts, Riggenbachstrasse 16, CH-4600 Olten, Switzerland
[2] Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Avda. Països Catalans 26, 43007 Tarragona, Spain

Abstract. In recent years, a powerful approximation framework for graph edit distance computation has been introduced. This particular approximation is based on an optimal assignment of local graph structures which can be established in polynomial time. However, as this approach considers the local structural properties of the graphs only, it yields sub-optimal solutions that overestimate the true edit distance in general. Recently, several attempts for reducing this overestimation have been made. The present paper is a starting point towards the study of sophisticated heuristics that can be integrated in these reduction strategies. These heuristics aim at further improving the overall distance quality while keeping the low computation time of the approximation framework. We propose an iterative version of one of the existing improvement strategies. An experimental evaluation clearly shows that there is large space for further substantial reductions of the overestimation in the existing approximation framework.

1 Introduction

Graph edit distance [1, 2] is one of the most flexible and versatile approaches to error-tolerant graph matching. In particular, graph edit distance is able to cope with directed and undirected, as well as with labeled and unlabeled graphs. In addition, no constraints have to be considered on the alphabets for node and/or edge labels. Moreover, through the concept of cost functions graph edit distance can be adapted and tailored to diverse applications [3, 4]. An extensive survey about graph edit distance can be found in [5].

The major drawback of graph edit distance is its high computational complexity that restrict its applicability to graphs of rather small size. In fact, graph edit distance belongs to the family of *quadratic assignment problems* (QAPs), which belong to the class of $\mathcal{NP}$*-complete* problems. Therefore, exact computa-



A First Step Towards Exact Graph Edit Distance Using Bipartite Graph Matching

Miquel Ferrer, Francesc Serratosa, Kaspar Riesen

Institute for Information Systems, University of Applied Sciences and Arts, Riggenbachstrasse 16, CH-4600 Olten, Switzerland

Departament d'Enginyeria Inform\`atica i Matem\`atiques, Universitat Rovira i Virgili, Avda. Paisos Catalans 26, 43007 Tarragona, Spain

In recent years, a powerful approximation framework for graph edit distance computation has been introduced. This particular approximation is based on an optimal assignment of local graph structures which can be established in polynomial time. However, as this approach considers the local structural properties of the graphs only, it yields sub-optimal solutions that overestimate the true edit distance in general. Recently, several attempts for reducing this overestimation have been made. The present paper is a starting point towards the study of sophisticated heuristics that can be integrated in these reduction strategies. These heuristics aim at further improving the overall distance quality while keeping the low computation time of the approximation framework. We propose an iterative version of one of the existing improvement strategies. An experimental evaluation clearly shows that there is large space for further substantial reductions of the overestimation in the existing approximation framework.

Introduction

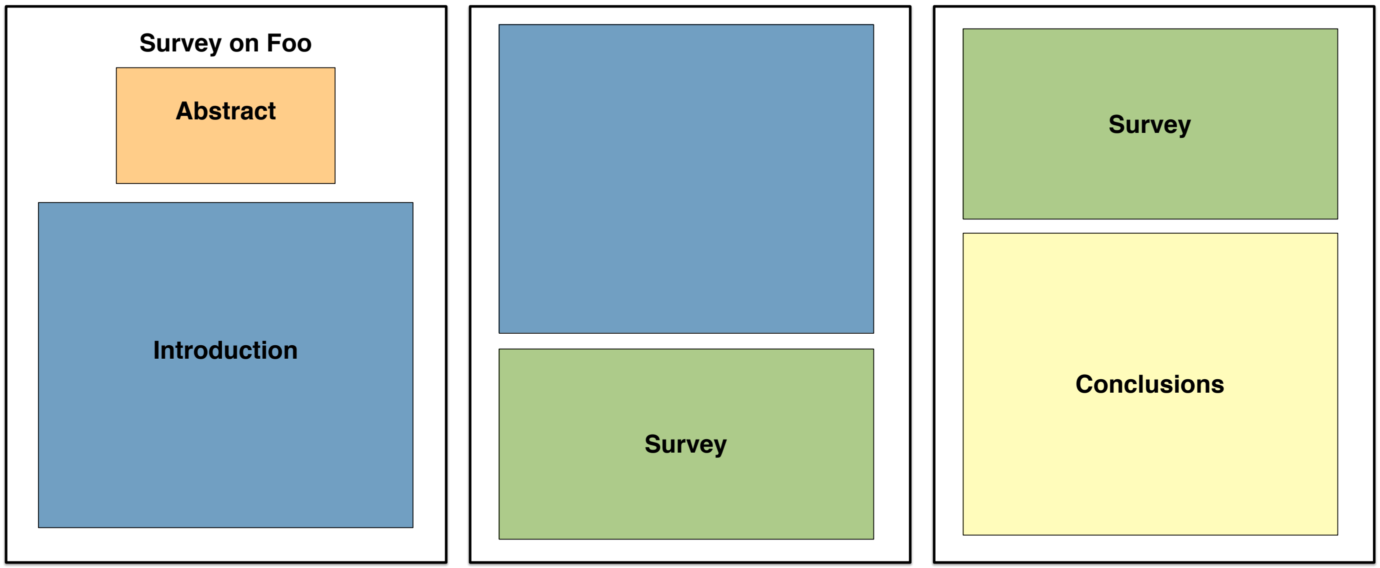
Graph edit distance (Sanfeliu 1983,Bunke 1983} is one of the most flexible and versatile approaches to error-tolerant graph matching. In particular, graph edit distance is able to cope with directed and undirected, as well as with labeled and unlabeled graphs. In addition, no constraints have to be considered on the alphabets for node and/or edge labels. Moreover, through the concept of cost functions graph edit distance can be adapted and tailored to diverse applications. An extensive survey about graph edit distance can be found in (Gao. 2010).

Empfehlung:

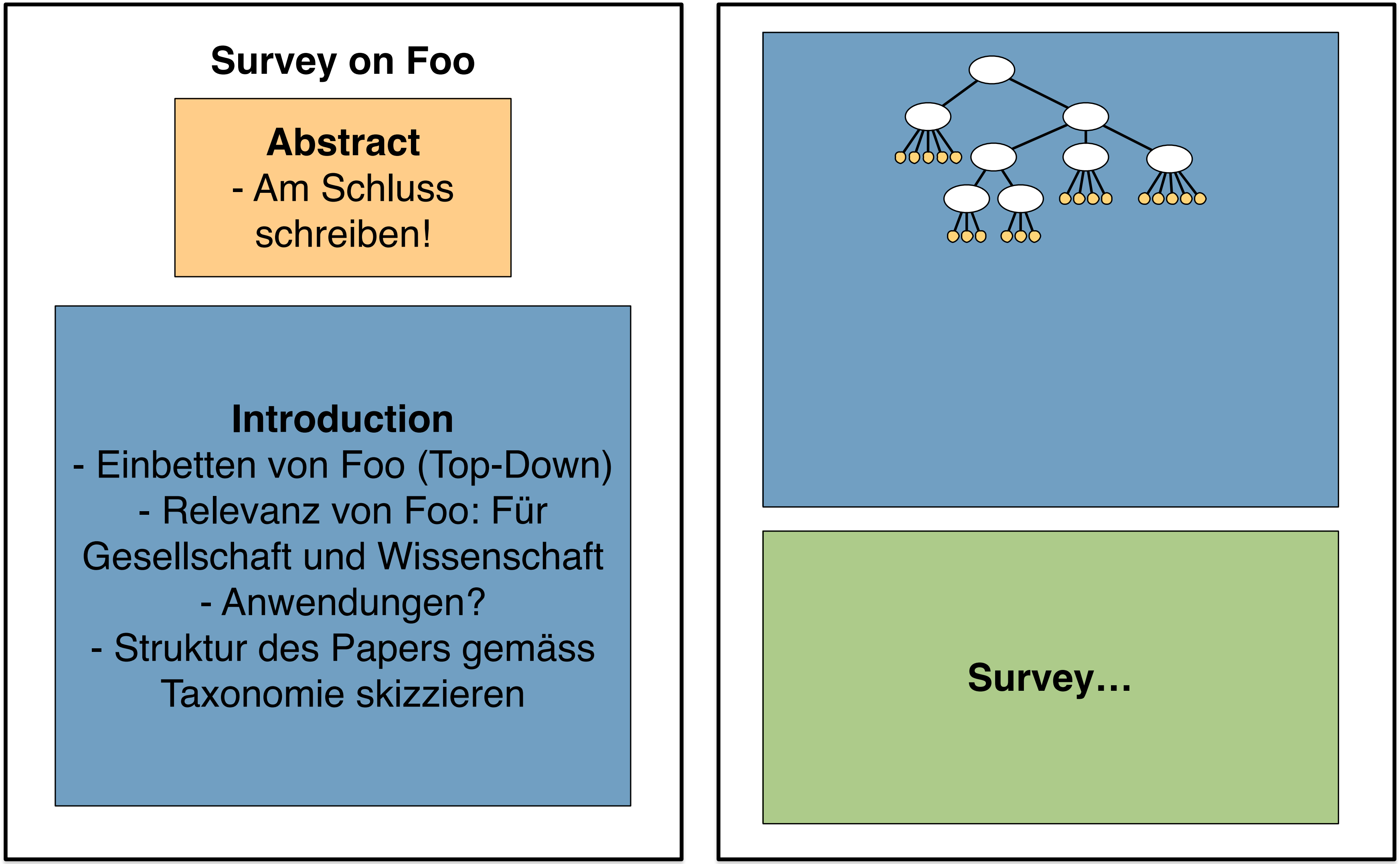
https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines

Survey Einbetten

- Der *Survey* bildet den Kern Ihres *Papers*.
- Weitere Teile (Minimalanforderung):
    - Abstract
    - Introduction
    - Conclusions



Abstract und Introduction



Abstract Beispiele

A recent paper posed the question: "Graph Matching: What are we really talking about?". Far from providing a definite answer to that question, in this paper we will try to characterize the role that graphs play within the Pattern Recognition field. To this aim two taxonomies are presented and discussed. The first includes almost all the graph matching algorithms proposed from the late seventies, and describes the different classes of algorithms. The second taxonomy considers the types of common applications of graph-based techniques in the Pattern Recognition and Machine Vision field.

Keywords: Graph matching algorithms; pattern recognition.

Abstract—Handwriting has continued to persist as a means of communication and recording information in day-to-day life even with the introduction of new technologies. Given its ubiquity in human transactions, machine recognition of handwriting has practical significance, as in reading handwritten notes in a PDA, in postal addresses on envelopes, in amounts in bank checks, in handwritten fields in forms, etc. This overview describes the nature of handwritten language, how it is transduced into electronic data, and the basic concepts behind written language recognition algorithms. Both the *on-line case* (which pertains to the availability of trajectory data during writing) and the *off-line case* (which pertains to scanned images) are considered. Algorithms for preprocessing, character and word recognition, and performance with practical systems are indicated. Other fields of application, like signature verification, writer authentification, handwriting learning tools are also considered.

Index Terms—Handwriting recognition, on-line, off-line, written language, signature verification, cursive script, handwriting learning tools, writer authentification.

Introduction Beispiel

1 Introduction

Graphs are recognized as versatile alternative to feature vectors and thus, they found widespread application in pattern recognition and related fields [1, 2]. However, one drawback of graphs, when compared to feature vectors, is the significant increase of the complexity of many algorithms. Regard, for instance, the algorithmic comparison of two patterns (which is actually a basic requirement for pattern recognition). Due to the homogeneous nature of feature vectors, pairwise comparisons is straightforward and can be accomplished in linear time with respect to the length of the two vectors. Yet, the same task for graphs, commonly referred to as *graph matching*, is much more complex, as one has to identify common parts of the graphs by considering all of their subsets of nodes. Regarding that there are $O(2^n)$ subsets of nodes in a graph with n nodes, the inherent difficulty of graph matching becomes obvious.

In the last four decades a huge number of procedures for graph matching have been proposed in the literature [1, 2]. They range from *spectral methods* [3, 4], over *graph kernels* [5, 6], to reformulations of the discrete graph matching

problem to an instance of a *continuous optimization problem* (basically by relaxing some constraints) [7]. *Graph edit distance* [8, 9], introduced about 30 years ago, is still one of the most flexible graph distance models available and topic of various recent research projects.

In order to compute the graph edit distance often A* based search techniques using some heuristics are employed (e.g. [10]). Yet, exact graph edit distance computation based on a tree search algorithm is exponential in the number of nodes of the involved graphs. Formally, for two graphs with m and n nodes we observe a time complexity of $O(m^n)$. This means that for large graphs the computation of the exact edit distance is intractable.

In [11] authors of the present paper introduced an algorithmic framework for the approximation of graph edit distance. The basic idea of this approach is to reduce the difficult problem of graph edit distance to a *linear sum assignment problem* (LSAP), for which an arsenal of efficient (i.e. cubic time) algorithms exist [12]. In two recent papers [13, 14] the optimal algorithm for the LSAP has been replaced with a suboptimal greedy algorithm which runs in quadratic time. Due to the lower complexity of this suboptimal assignment process, a substantial speed up of the complete approximation procedure has been observed. However, it was also reported that the distance accuracy of this extension is slightly worse than with the original algorithm. Major contribution of the present paper is to improve the overall distance accuracy of this recent procedure by means of an elaborated transformation of the underlying cost model.

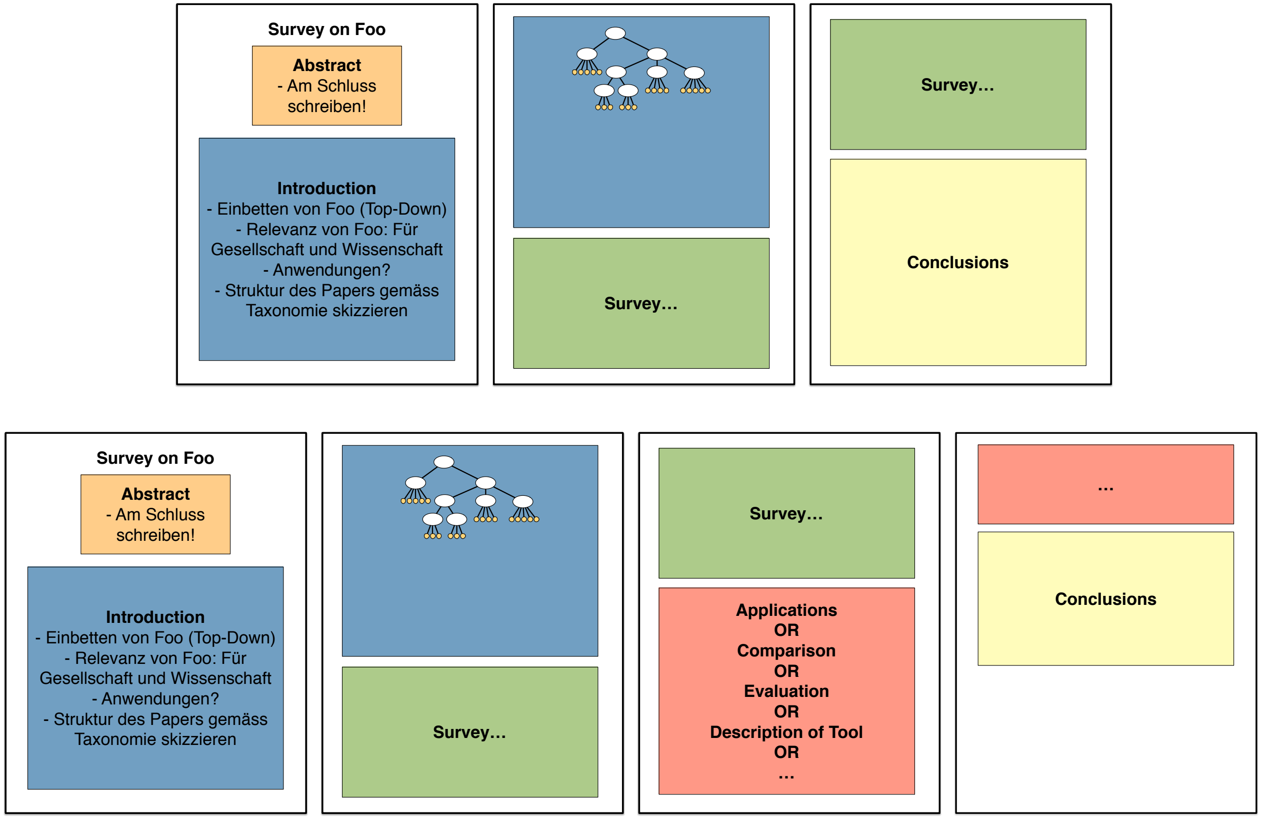
The remainder of this paper is organized as follows. Next, in Sect. 2, the computation of graph edit distance is thoroughly reviewed. In particular, it is shown how the graph edit distance problem can be reduced to a linear sum assignment problem. In Sect. 3, the transformation of the cost model into a utility model is outlined. Eventually, in Sect. 4, we empirically confirm the benefit of this transformation in a classification experiment on three graph data sets. Finally, in Sect. 5, we conclude the paper.

Conclusions

- Ihr *Paper* **muss** eine eigene Reflexion/Interpretation enthalten.
- Lösung 1: Einbetten in *Conclusions* (in der Sie z.B. die Vor- und Nachteile der beschriebenen Methoden beschreiben).
- Lösung 2: Separater Abschnitt im *Paper* (*Vergleich* oder *Experiment* oder *Beschreibung einer Anwendung* oder *Tutorial für ein Tool* oder …)

Drittes Meeting (29.11)

Sie präsentieren einen Vorschlag für einen „Abschluss“ des Papers, den wir im Plenum diskutieren



Grading

ToBIT Seminar

Written Part			90%
I.	Documentation & formal criteria	18%	
II.	Research (conceptual/theoretical) Framework	25%	
III.	Procedure & analysis	19%	
IV.	Interpretation & reflection	19%	
V.	Achievement of objectives & Critical appraisal	19%	
Oral Part			10%
I.	Delivery	40%	
II.	Content	60%	

Beurteilung Paper (90%)

	ungenügend	knapp genügend	genügend	besser als genügend	beinahe gut	gut	besser als gut	beinahe ausgezeichnet	ausgezeichnet
Qualität des initialen Proposals	0	1	2	3	4	5	6	7	8
Qualität des Abstracts	0	1	2	3	4	5	6	7	8
Introduction (Einbettung der Thematik)	0	1	2	3	4	5	6	7	8
Taxonomie/ Struktur/ Vollständigkeit	0	1	2	3	4	5	6	7	8
Beschreibung der Methoden	0	1	2	3	4	5	6	7	8
Interpretation/ Vergleich/ Conclusions	0	1	2	3	4	5	6	7	8
Referenzen (Qualität und Quantität)	0	1	2	3	4	5	6	7	8
Besonderes 1*	-	-	-	-	-	5	6	7	8
Besonderes 2*	-	-	-	-	-	5	6	7	8

*Darf den Durchschnitt nicht verkleinern

Beurteilung Präsentation (10%)

	ungenügend	knapp genügend	genügend	besser als genügend	beinahe gut	gut	besser als gut	beinahe ausgezeichnet	ausgezeichnet
Präsentations-stil	0	1	2	3	4	5	6	7	8
Strukturierung und Vollständigkeit	0	1	2	3	4	5	6	7	8
Korrektheit	0	1	2	3	4	5	6	7	8
Visualisierung	0	1	2	3	4	5	6	7	8
Diskussion/ Beantwortung der Fragen	0	1	2	3	4	5	6	7	8

Beurteilung

Grundsatz:

Default = gut

Betreuung

- Die Betreuung findet grundsätzlich an den Gruppenmeetings statt
- *Konkrete* Fragen per Mail (Keine Antwort: Reminder nach 3 Tagen)
- *No News are Good News*

```
/**
 * ?&!
 */
```