<Bachelorarbeit> am Institut für Informatik der Freien Universität Berlin

Human-Centered Computing (HCC), AG NBI

Semantic Similarity of Concepts for a Human-Centered Idea Recommendation Feature in the Clustering Application Orchard

– Exposé –

Luka Stärk

Matrikelnummer: 374532

luka.staerk@campus.tu-berlin.de

Betreuer: Michael Tebbe

Berlin, February 6, 2020

1 Motivation

The research project Ideas2Market explores the innovation process for applications of new technologies. A central task is to generate many ideas, to cover most possible solutions on how to apply the technology. This procedure is implemented using collaborative innovation approaches to crowd-source ideas. These ideas introduce great variety and creative value because they are created by different persons with diverse backgrounds. Nevertheless, these ideas are not yet fully evolved and considered to be on a brainstorming level, in the following they will be referred to as idea sparks. Therefore, in the further innovation process experts evolve, refine and transform promising idea sparks into product opportunities to deploy onto the market. Still, finding valuable idea sparks has proven challenging and due to their large number, it becomes unfeasible to check every idea spark manually and to derive benefits from them for advanced ideas. The project Ideas2Market aims to solve these problems with software support and by researching the human needs in creative processes, in which this thesis extends existing software, namely Orchard, with a new feature. The software supported collaborative-ideation process can be described in three phases as illustrated below in Figure 1:

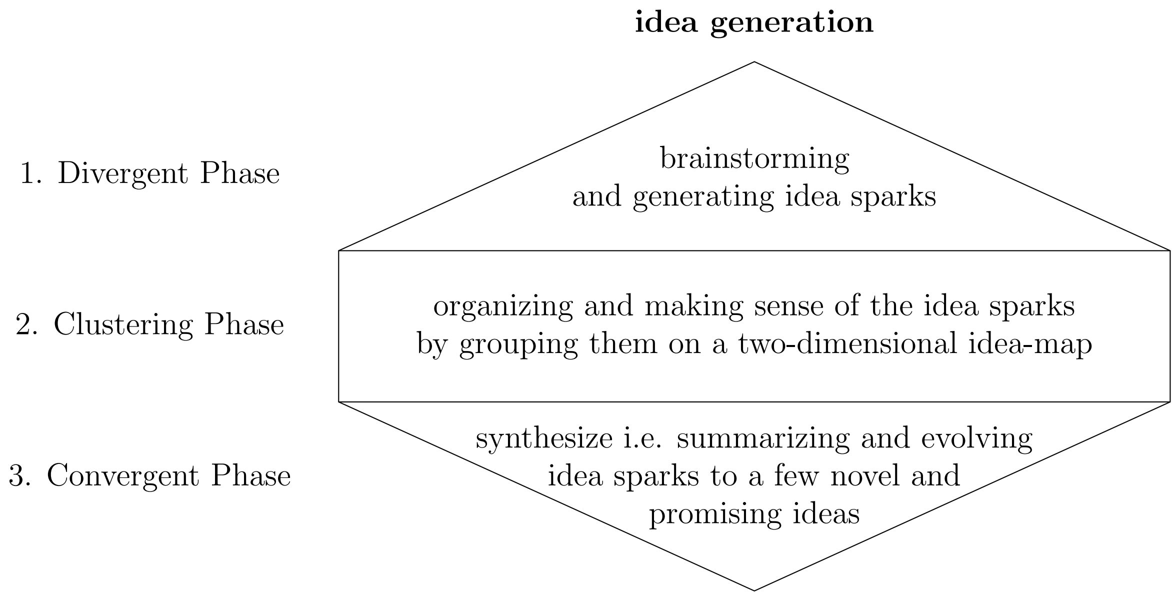

Figure 1: Three Phase Diamond of the Innovation Process [Tassoul and Buijs, 2007]

When clustering, the categories of the emerging clusters and the connections between idea sparks are not always so clear to us. The decision of creating a cluster is based on feeling and intuition and can be reversed any time. During the process, an ordering develops and the relationships between idea sparks become more visible, so far the theory. Clustering is ought to be beneficial as an activity in acquiring a more profound understanding of the idea-space [Siangliulue et al., 2016] and producing more valuable ideas in the Convergent Phase. However, for growing numbers of idea sparks, it becomes more challenging to organize the idea-space and to take into account all potential idea

Figure 2: Clustering view of the Orchard interface, where the recommendation frame (1) displays similar idea sparks to *SPARK 2* (4) from the Cluster named *PET* (3) with two idea sparks. (5) The right column displays the full content of the idea spark, that the user selects by mouse click and further the caption is set in bold, see *SPARK 2* (4).

sparks for one cluster. This task can then be monotonous and time-consuming, which can decrease the quality of the idea syntheses Siangliulue [2017]. My thesis is about counteracting this problem and increasing efficiency in the Clustering Phase, by extending Orchard with an interactive recommendation feature, as shown in Figure 2 on the left (1). So that the user can walk through the idea sparks led by their changing interest of categories, clusters, and topics.

2 Thematic Classification of the Thesis

2.1 Orchard Clustering Application

In the research project Ideas2Market, the clustering Web-Application Orchard has been developed to support the Clustering Phase of the ideation process, the user interface is shown in Figure 2. Orchard is inspired by the IdeaHound project [Siangliulue et al., 2016] and a tool for creative ideation to effectively synthesize ideas from numerous idea sparks. For the Clustering Phase, the user can drag and drop ideas from the *Spark Stack* onto the whiteboard. To create clusters or add to an existing cluster, the user drops one idea spark onto another or an existing cluster. The user can inspect an idea spark in detail by clicking on it. In that case, the complete description and labels of the idea spark are displayed in the right column, as displayed in Figure 2 (5). For the Convergent Phase, experts can write and archive their idea syntheses in the application.

The findings of my thesis on how to improve the clustering process with an interactive recommendation feature will be implemented and integrated into the Orchard application. For the interaction whit the feature, the user needs to specify their interest for particular spark ideas and topics by selecting an idea spark or a highlighted term on the Orchard application, see (4) in Figure 2. The recommended ideas ought to be related to the user's selection, thus a measurement of similarity between spark ideas is needed, which will be further elaborated in section 2.3 and 2.4.

2.2 Collaborative Ideation at Scale

In the dissertation "Supporting Effective Collective Ideation at Scale", Siangliulue [2017] is discussing solutions to increase efficiency in synthesizing numerous amounts of ideas. One possibility is to introduce a predefined idea-map, where the idea sparks are organized in clusters sorted by similarity score so that related and similar idea sparks are positioned near to each other [Siangliulue, 2017, 124]. Besides, it is easier for the user to internalize the idea-space and thus interact more with rare ideas [Siangliulue, 2017]. That is beneficial for the user because ideas are often mundane or repetitive [Siangliulue et al., 2016]. Then again, the user is more fixated on the categories that were given by the clusters and might miss other possible syntheses that would have been created without the suggested clusters [Siangliulue, 2017].

The recommendation feature is a different approch to support and accelerate the Clustering Process, which avoids the mentioned drawback of fixation on given categories, that have been observed for generated clusters [Siangliulue, 2017].

2.3 Knowledge Graph

The recommendation feature uses semantic similarity metrics to measure the similarity between defined concepts in a Knowledge Graph (KG). A KG records the relations between concepts. The occurring terms in the idea spark's descriptions are assigned to such concepts in the KG. A concept can be described by various terms and applies to a group of instances. E.g. synonyms such as *car* and *automobile* are referencing the same concept. The label of an instance, for example, is *Peugeot 104* and therefore is *car* one concept that applies to that instance. A KG makes theses informations accessible.

In the context of semantic web and linked data many Knowledge Graphs like DBpedia and Wikidata, are freely accessible and gain increasing popularity. KGs are semantic networks where relations between concepts and entities are recorded as triples (subject, predicate, object). These Information Networks are used for different tasks in the field of Natural Language Processing and Information Retrieval such as Word Sense Disambiguation, Topic Modeling, and Question Answering [Nastase, 2008]. The semantic similarity of concepts can be measured through hierarchical relations in Knowledge Graphs. The advantage of this approach is that the similarity between two concepts becomes interpretable when looking up the connecting path or the lowest common ancestor concept in the directed acyclic Knowledge Graph, e.g. Figure 3 where the edges are directed from the root of the tree to the leaves. Also tracing back presumably false results to the

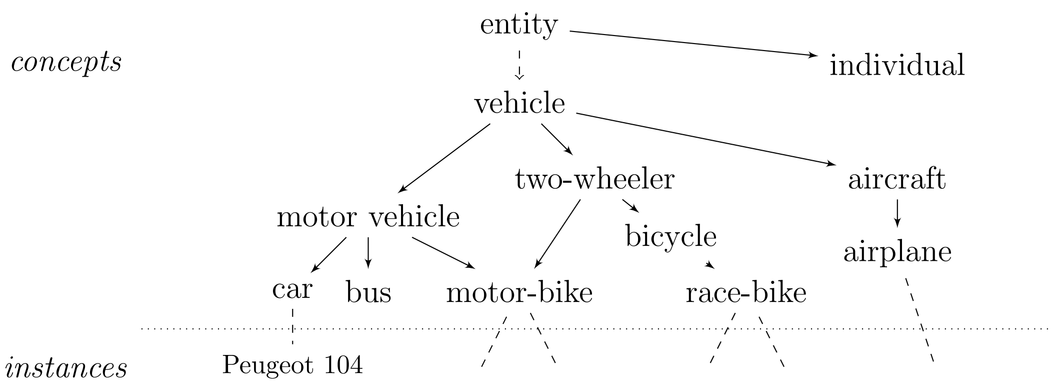

Figure 3: Part of the Knowledge Graph from Wikidata

origin by identifying incorrect relations in the KG is feasible. In statistical approaches, this information is more difficult to extract because the semantic relationships of concepts are not accessible as facts, like in a KG, rather as distances in high-dimensional spaces. The Wikidata ontology records more than 69 million items[1] and covers most of the real-world entities and is considered useful as KG in this approach.

2.4 Similarity of Concepts

In the field of Semantic Similarity, there are several metrics to measure the similarity of terms and concepts, and instances. Measuring semantic similarity divides up into mainly corpus-based and knowledge-based approaches. Corpus-based semantic similarity metrics use statistical relations of words in large text collections. Two words are similar when their surrounding text context is similar. This approach relies on the occurrence of words and ignores the different meanings a word can have, namely word sense disambiguation [Zhu and Iglesias, 2017]. In contrast, knowledge-based semantic similarity metrics, measure similarities between defined concepts in a KG. The most simple similarity metric takes the shortest path distance between two concepts and transforms it into a similarity score $s \in [0, 1]$, where 0 describes no similarity and 1 applies for identical concepts. For two concepts c_i, c_j let $length(c_i, c_i)$ denotes the length of the shortest path between c_i, c_j, then the similarity is calculated as

$$sim_{path}(c_i, c_j) = \frac{1}{1 + length(c_i, c_j)}. \tag{1}$$

Another widely used measurement is the Information Content (IC). The IC of a concept indicates how abstract or specific a concept and how much information the entities of a concept share in common. Intuitively more abstract concepts hold lower IC values and more specific ones higher values of IC [Resnik, 1995]. There are two different ways of measuring the IC, namely corpus-based or knowledge-based metrics. Zhu and Iglesias [2017] propose the following definitions:

[1]https://www.wikidata.org/wiki/Wikidata:Statistics

Definition 2.1. Information Content corpus-based: Let c_i be a concept, given a large general text-corpus, $Prob(c_i)$ is the probability to encounter a word from the set of $words(c_i)$ that are subsumed or associated with c_i. $Prob(c_i) = \frac{\Sigma_{w \in words(c_i)} count(w)}{N}$, where $count(w)$ is the occurrence of the word w and N is the total number of occurrences of concepts in the text-corpus. For the KG in Figure 3, the occurrence of the noun "automobile" would be counted towards the frequency of *car*, *motor-vehicle* and so forth. The Information Content can be quantified as negative the log likelihood $-log_e Prob(c_i)$ [Resnik, 1995], then the $IC_{corpus}(c_i) = -log_e Prob(c_i)$, so the IC of concept c_i increases when the probability decreases and if there would be one concept subsumming all other concepts its IC would be 0, as illustrated by the diagram in Figure 4.

Definition 2.2. Information Content graph-based:
Let c_i be a concept, then the $IC_{graph}(c_i) = -log_e Prob(c_i)$, where the $Prob(c_i) = \frac{|entities(c_i)|}{N}$ and $entities(c_i)$ is the set of entities for which concept c_i applies, so they all reach c_i through ancestral relations. N is the total number of entities in the KG. E.g. resolves *entities(two-wheeler)* to {*two-wheeler, bicycle, motor-bike, race-bike*}, is shown in Figure 3.

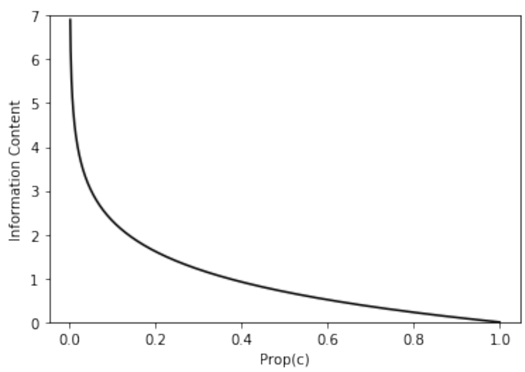

Figure 4: Decrease of the $IC(c)$ for increasing probability of the occurrence of concept c and its associated concepts.

The publication *Computing Semantic Similarity of Concepts in Knowledge Graphs* of Zhu and Iglesias [2017] discusses different metrics of concept similarity in Knowledge Graphs and compares them to their approach *wpath* with gold standard data sets of human judgements of similarity. Their metric *wpath* for semantic similarity measures is outperforming other widely used metrics by a small margin. It considers shortest-path length between two concepts in the Knowledge Graph and the Information Content (IC) of their least common subsumer (LCS). The LCS of two concepts is the most specific ancestral concept that is shared by both concepts. Therefore the LCS is the concept with the highest IC among the shared ancestors. E.g in the KG shown in Figure 3, the LCS of *motor-bike* and *bicycle* is the concept *two-wheeler*, and not *vehicle*, because it is less specific and ancestral to *two-wheeler*. [Zhu and Iglesias, 2017] define their semantic similarity method as

$$sim_{wpath(c_i,c_j)} = \frac{1}{1 + length(c_i, c_j) \cdot k^{IC(lcs)}}, \tag{2}$$

where the parameter $k \in (0, 1]$ determines the impact of the IC of the LCS in weighting the path length of two concepts. If $k = 1$ the IC does not influence the path length. Otherwise, the IC weights the path length, so that concepts with the same path length but different LCS can have different similarities. E.g. *car* and *bus* are more similar than *two-wheeler* and *aircraft*. Though for both pairs, the path length equals two, as shown in Figure 3. But the IC of *motor-vehicle* as their LCS is greater than the IC of *vehicle*, because *motor-vehicle* is more specific.

For the computation of the IC, the corpus-based method has shown robust results using modest amounts of text without annotated sense-tags, even outperforming the results for the largest available sense-tagged corpora [Pedersen, 2010]. The graph-based method has proven excellent results by Zhu and Iglesias [2017] and its further exploration is of interest in this thesis. Thus, for the semantic similarity measures both IC_{graph} and IC_{corpus} are calculated and analyzed on their performance.

2.5 Human-Centered Approach

In the field of machine learning and beyond, human-centered approaches have gained extensive attention. As machine learning discover relations and patterns in data instead of programming explicit rules, the solution may reflect the bias and incompleteness of the used data and contain uncertainties. Viewing the problem through the human lens and considering human needs, ensures that the problems stays grounded.

The mentioned concerns apply to this thesis as well. The spark idea recommendations are based on knowledge graph states and other assumptions such as the metric *wpath* and its parameter k. When interacting with a so-called intelligent-system, the user will naturally form a mental model of it and adjust interaction and behavior to the assumptions being made. E.g. users might experience spark idea recommendations as unrelated for some concepts, so that they will try to identify and then avoid using these concepts. When the user has a good mental model of the system, the interaction is more effective. Which gives reason to consider the interpretability of the interactive system when designing the User Interface. Furthermore, leaving tasks to the user, that the user performs best, makes the system more adaptive and gives the user the feeling of being in control, so that the user will be more likely to interact with the system [Abdul et al., 2018]. E.g in the Orchard application the user selects a concept of an idea spark, to be more specific in what they are interested in. An addi-

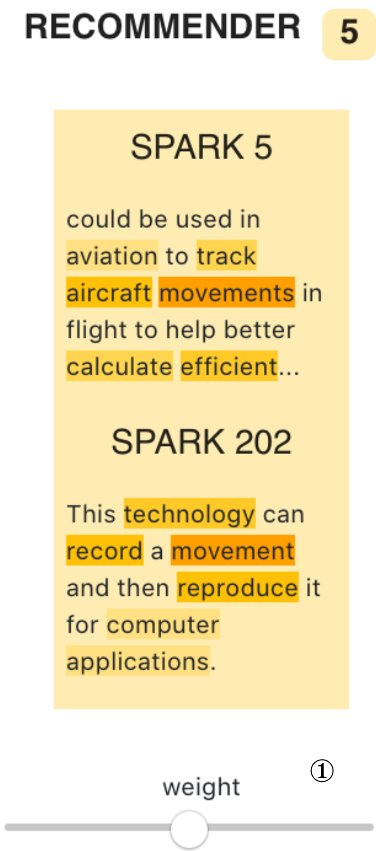

Figure 5: Recommendation Feature that highlights concepts with saturation depending on the similarity to the selected object. (1) Slider to change weight in *wpath* metric.

tional feature for the User Interface will be a Slider in a range, as shown in Figure 5 (1), to change the k-parameter, weighting between the impact of path length and Information Content of the LCS. Thus the user can experiment and adjust the weight to their needs. For more interpretability and a better understanding of the recommendations, the idea sparks in the recommendation frame are visualized with highlighted concepts as well, as Figure 5 illustrates. In which the color saturation depends on the similarity to the selected idea spark or concept. Thereby it becomes easier for the user to create a mental model of the recommendation feature and to understand why a certain idea spark is ranked as most similar.

3 Goal

The goal of my thesis is to improve the synthesis of ideas, by suporting the clustering process in the Orchard application. My approch is an recommendation feature that increases efficiency in clustering and thereby aids the user to interact more with rare and valuable idea sparks, without the fixation on predefined categories, as mentioned in section 2.2.

I will evaluate this approch in a qualitative user-study. Therefore, 5 experts with knowledge in the field of ideation will test the Orchard application with the recommendation feature and then refelct on their experience in a questionary or interview.

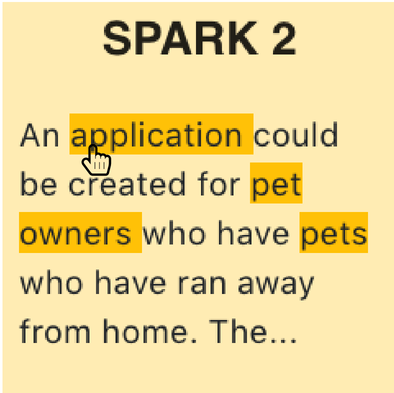

Figure 6: Spark with highlighted concepts

For the user-study 60 gold-standard idea sparks are used, created whithin the Ideas2Market project. The recommendation feature that I implement contains the following functionality. In Orchard, as Figure 6 illustrates, the user can click on an idea spark or a highlighted concept of its description to select the content, based on that idea sparks with similar concepts are recommended. The recommended idea sparks are displayed in the recommendation frame sorted by highest-similarity and the user can scroll through and drag them onto the whiteboard, as shown in Figure 2 (1).

Given the similarity measures between concepts, concepts and ideas, and a similarity score of the Word Mover's Distance [Kusner et al., 2015] between ideas, the recommendation feature for the Orchard clustering application can be implemented to support the functionality as described.

The proposed metric *wpath* of Zhu and Iglesias [2017] combines meaningful semantic similarity measures for the application on a Wikidata KG and has shown outperforming results for the DBpedia-Ontology for common English nouns. As the Wikidata KG is especially large and complex it is of interest answering, how well semantic similarity metrics perform on such KGs compared to the DBpedia-Ontologie. And if *wpath* therefore leads to novels results on a wider range of less prominent concepts. Therefore part of my thesis is a proof of concept on the metric of Zhu and Iglesias [2017] applied onto the Wikidata KG, for graph and corpus-based IC.

4 Planned Procedure

The Procedure for my thesis is divided into three substantial parts: implementation, integration into the Orchard application and the validation of the results.

4.1 Implementation of wpath

The requirement for the input data of idea sparks is that the concepts of the idea descriptions are assigned to Wikidata items. In the prepossessing all stop-word concepts, such as "I, my, You,..." and concepts that do not connect to the KG, are excluded.

The following functions will be implemented to apply the *wpath* metric for concept similarity.

- A function to generate a subgraph of the Wikidata Knowledge Graph. The subgraph is extracted with *SPARQL* queries to the *Wikidata* SPARQL-endpoint. The set of concepts that occur in the idea sparks build the bottom child layer of the KG. All existing connections in Wikidata from each idea-concept to *entity*, over the three predicates *subclass of (P279), instance of (P31), part of (P361)* are extracted for the KG. In which *entity* is the highest concept in the hierarchy of the Wikidata KG. (done)
- A function to resolve a directed graph into a directed acyclic graph (DAG). (done)
- A function to calculate the Information Content graph-based for all concepts in the DAG, over the edges *subclass of (P279), instance of (P31).*
- A function to calculate the Information Content corpus-based, using WordNet with the Brown Corpus [Francis and Kučera, 1964].
- A function that finds the least common subsumer for all pairs of two concepts in the DAG. (done)
- A function that calculates the all-shortest ancestral distance in the DAG, i.e. the shortest path between two nodes over their LCS. (done)

For the similarity measures from a concept to an idea spark, I will implement a function that returns the idea sparks which contain the concepts that are most similar to the concept given as input.

To calculate the similarity between idea sparks I will implement the word mover's distance (WMD) [Kusner et al., 2015] based on the similarity of their concepts.

4.2 Integration into Orchard

For the Orchard application, I will integrate the similarities, provided by the implementation for a given data-set of idea sparks, into the database of Orchard.

For the client-side I will add a recommendation frame, as shown in Figure 2 (1) and highlight the annotated concepts in the idea-description in the detail view (see Figure 2 (5)). When the user hovers over any idea spark on the whiteboard the concepts in the description become highlighted as well, as Figure 6 indicates. The highlighted concepts are clickable and a function updates the recommendations for the new source.

A React Component will be programmed that displays the idea sparks with highlighted concepts, with different color saturation, depending on the similarity to the selected object, as depicted in Figure 5.

4.3 Validation

The validation consists of the following two steps:

1. To validate the concept similarity measure, the implementation of *wpath* based on the *Wikidata* KG will be evaluated with the *R&G* dataset [Rubenstein and Goodenough, 1965] of human-judged word similarity and the results of the implementation of *wpath* from Zhu and Iglesias [2017] for DBpedia ontology. *R&G* is a widely used dataset containing human assessments of word similarity for 65 word-pairs of common English nouns. The results of *wpath* for *Wikidata* will be evaluated for different parameters $k \in (0, 1]$, to optimize the measurements for *R&G*. This is done for the implementation with the corpus-based IC and the graph-based IC, to analyze both variants.

2. As the last step, three or more persons, who are familiar with the clustering process, will test the recommendation feature in Orchard on a qualitative level. By evaluating the research question, if the recommendation feature in Orchard is helpful for the Clustering Phase and provides useful recommendations for the user's interactive requests. Furthermore, the experts evaluate the Human-Centered interface design choices. Such as discussing the color saturation of concepts and the functionality of the k-parameter adjustment, as shown in Figure 5, and if these meet their purpose, as described in Section 2.5.

5 Technical Implementation

The software is realized in python and the Knowledge Graph is extracted from Wikidata. For the calculation of the corpus-based IC, the implementation of Zhu and Iglesias [2017] is used. The Interface of the recommendation feature in Orchard is implemented in JavaScript, React and Redux.

6 First Schedule

8-14.February	Presentation of the thesis topic at HCC
8-14. February	Find a primary supervisor from TU-Berlin.
14-19. February	Implementation of IC_{graph}
20-22. February	Implementation of IC_{corpus}
23. February	Annotate and connect *R&G* concepts to Wikidata
26. February	Registration of the bachelor thesis at the TU-Berlin
30. February	Implementation of *wpath*
1. February	Validation of *wpath* and comparison with dataset of Rubenstein and Goodenough [1965] and implementation of Zhu and Iglesias [2017].
7. February	Implementation of WMD for similarity measures.
2-9. March	Test concept similarity implementation of *wpath* for Wikidata, to select best parameter k.
12. March	Finish technical implantation, take measurements for similarity.
13-17. March	The recommendation feature is integrated into Orchard for gold standard ideas.
20. March	Discussion of the results with Prof. Dr. Claudia Müller-Birn, Michael Tebbe and Maximilian Mackeprang.
30. March	The thesis are written, time for correction.
15. April	Submission of the bachelor thesis

References

Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y. Lim, and Mohan Kankanhalli. Trends and Trajectories for Explainable, Accountable and Intelligible Systems: An HCI Research Agenda. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–18, Montreal QC, Canada, 2018. ACM Press. ISBN 978-1-4503-5620-6. doi: 10.1145/3173574.3174156. URL `http://dl.acm.org/citation.cfm?doid=3173574.3174156`.

W. Nelson Francis and Henry Kučera. A Standard Corpus of Present-Day Edited American English, for use with Digital Computers (Brown). Providence, Rhode Island, 1964. Brown University.

Matt J Kusner, Yu Sun, Nicholas I Kolkin, and Kilian Q Weinberger. From Word Embeddings To Document Distances. page 10, 2015.

Vivi Nastase. Topic-driven Multi-document Summarization with Encyclopedic Knowledge and Spreading Activation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 763–772, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1613715.1613812`. event-place: Honolulu, Hawaii.

Ted Pedersen. Information Content Measures of Semantic Similarity Perform Better Without Sense-Tagged Text. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, page 4. Association for Computational Linguistics, Los Angeles, California, June 2010.

Philip Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. page 6, November 1995.

Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, October 1965. ISSN 00010782. doi: 10.1145/365628.365657. URL `http://portal.acm.org/citation.cfm?doid=365628.365657`.

Kanya (Pao) Siangliulue. Supporting Effective Collective Ideation at Scale. May 2017. ISSN http://nrs.harvard.edu/urn-3:HUL.InstRepos:40046559. URL `https://dash.harvard.edu/handle/1/40046559`.

Pao Siangliulue, Joel Chan, Steven P. Dow, and Krzysztof Z. Gajos. IdeaHound: Improving Large-scale Collaborative Ideation with Crowd-Powered Real-time Semantic Modeling. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16*, pages 609–624, Tokyo, Japan, 2016. ACM Press. ISBN 978-1-4503-4189-9. doi: 10.1145/2984511.2984578. URL `http://dl.acm.org/citation.cfm?doid=2984511.2984578`.

Marc Tassoul and Jan Buijs. Clustering: An Essential Step from Diverging to Converging. *Creativity and Innovation Management*, 16(1):16–26, 2007. ISSN 1467-8691. doi: 10.1111/j.1467-8691.2007.00413.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8691.2007.00413.x`.

G. Zhu and C. A. Iglesias. Computing Semantic Similarity of Concepts in Knowledge Graphs. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):72–85, January 2017. ISSN 1041-4347. doi: 10.1109/TKDE.2016.2610428.

Anhang I: Auszug Prüfungsordnung Bachelor

FU-Mitteilungen

§ 5
Bachelorarbeit und mündliche Abschlussprüfung

(1) Die Bachelorarbeit soll zeigen, dass die Studentin oder der Student in der Lage ist, ein Thema aus dem Bereich der Informatik unter Anleitung nach wissenschaftlichen Methoden in einer vorgegebenen Zeit zu bearbeiten und seine Arbeit und die Ergebnisse selbständig darzustellen, wissenschaftlich einzuordnen und zu dokumentieren.

(2) Die Bearbeitungsdauer einer Bachelorarbeit beträgt zwölf Wochen.

(3) Studierende werden auf Antrag zur Bachelorarbeit zugelassen, wenn sie

1. die Module
   - Datenstrukturen und Datenabstraktion
   - Grundlagen der Theoretischen Informatik
   - Logik und Diskrete Mathematik
   - Analysis oder Analysis I
   - Lineare Algebra oder Lineare Algebra I sowie
   - Rechnerarchitektur

   erfolgreich absolviert haben,
2. im Bachelorstudiengang Informatik zuletzt an der Freien Universität Berlin immatrikuliert gewesen sind.

(4) Dem Antrag auf Zulassung zur Bachelorarbeit sind Nachweise über das Vorliegen der Voraussetzungen gemäß Abs. 3 beizufügen, ferner die Bescheinigung einer prüfungsberechtigten Lehrkraft über die Bereitschaft zur Übernahme der Betreuung der Bachelorarbeit sowie eine Erklärung, dass die oder der Studierende nicht an einer anderen Hochschule im gleichen Studiengang, im gleichen Fach oder in einem Modul, welches einem der im Bachelorstudiengang Informatik studierten Modulen vergleichbar ist, Leistungsnachweise endgültig nicht erbracht oder Prüfungsleistungen endgültig nicht bestanden hat oder sich in einem schwebenden Prüfungsverfahren befindet. Der zuständige Prüfungsausschuss entscheidet über den Antrag.

(5) Der Prüfungsausschuss gibt in Abstimmung mit der Betreuerin bzw. dem Betreuer das Thema der Bachelorarbeit aus. Thema und Aufgabenstellung müssen so beschaffen sein, dass die Bearbeitung innerhalb der Bearbeitungsfrist abgeschlossen werden kann. Ausgabe und Fristeinhaltung sind aktenkundig zu machen.

(6) Als Beginn der Bearbeitungszeit gilt das Datum der Ausgabe des Themas durch den Prüfungsausschuss. Das Thema kann einmalig innerhalb der ersten drei Wochen zurückgegeben werden und gilt dann als nicht ausgegeben. Ausnahmsweise kann der Prüfungsausschuss auf begründeten Antrag im Einvernehmen mit der Betreuerin bzw. dem Betreuer die Bearbeitungszeit der Bachelorarbeit um bis zu vier Wochen verlängern. Bei der Abgabe hat die bzw. der Studierende schriftlich zu versichern, dass sie bzw. er die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt hat.

(7) Die Bachelorarbeit ist von zwei Prüfungsberechtigten zu bewerten, die vom Prüfungsausschuss bestellt werden. Einer der beiden Prüfer soll die Betreuerin bzw. der Betreuer der Bachelorarbeit sein. Mindestens einer der beiden Prüfer muss dem Kreis der Professorinnen und Professoren des Instituts für Informatik angehören.

(8) Die Ergebnisse der Bachelorarbeit werden im Rahmen einer mündlichen Abschlussprüfung, bestehend aus einem etwa 15-minütigen Vortrag mit anschließender etwa 15-minütiger Diskussion und Prüfungsgespräch, vorgestellt und wissenschaftlich eingeordnet und verteidigt.

(9) Voraussetzung für die Teilnahme an der mündlichen Abschlussprüfung ist die Abgabe der Bachelorarbeit. Der Prüfungstermin wird rechtzeitig in geeigneter Form bekannt gegeben.

(10) Die mündliche Abschlussprüfung wird von denjenigen Prüfungsberechtigten, welche die Bachelorarbeit bewertet haben, abgenommen.

(11) Ist die Note der Bachelorarbeit oder die Note der mündlichen Abschlussprüfung nicht mindestens „ausreichend" (4,0), so dürfen Bachelorarbeit und mündliche Abschlussprüfung einmal wiederholt werden.

84 FU-Mitteilungen 6/2007 vom 08.02.2007

Figure 7: Auszug Prüfungsordnung Bachelor