<Bachelorarbeit> am Institut für Informatik der Freien Universität Berlin

Human-Centered Computing (HCC), AG NBI

Recommender System for Idea-Clustering based on Semantic Similarity of Concepts in Knowledge Graphs

– Exposé –

Luka Stärk

Matrikelnummer: 374532

luka.staerk@campus.tu-berlin.de

Betreuer: Michael Tebbe

Berlin, January 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 are not yet fully evolved and considered to be on a brainstorming level, in the following they will be referred to as idea sparks. Nevertheless, these idea sparks introduce great variety and creative value because they are created by different persons with diverse backgrounds. 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. These ideas are evolved by experts in the further process and then become refined and transformed into product opportunities to deploy onto the market as the last step. The project Ideas2Market aims to solve these problems with software support and by researching the human needs in creative processes. The software supported collaborative-ideation process can be described in three phases as illustrated below in Figure 1:

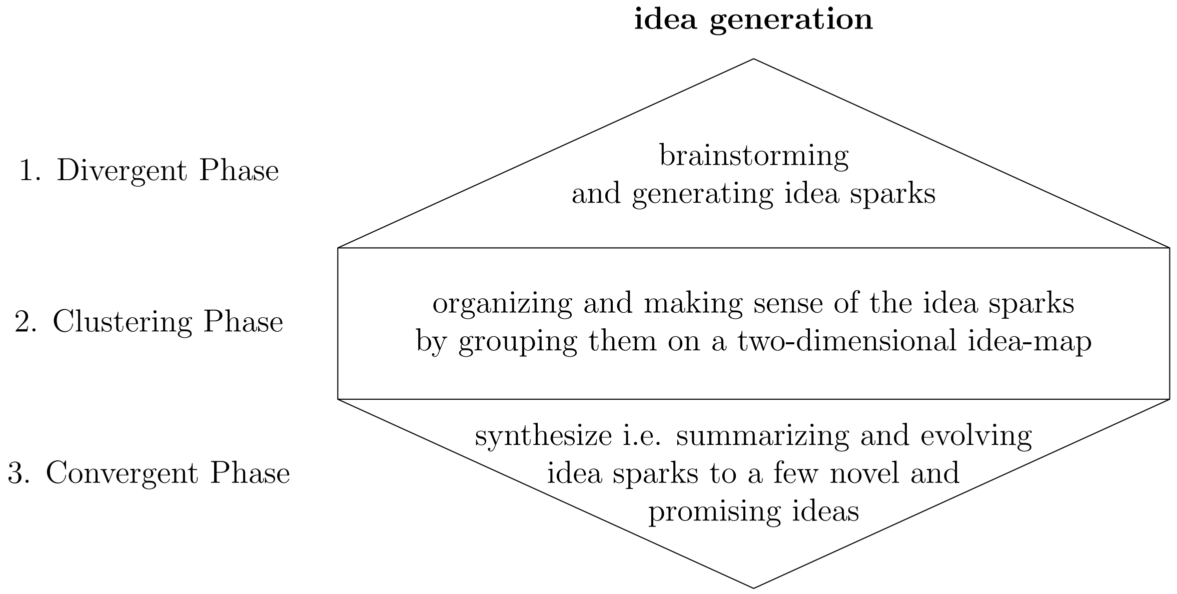
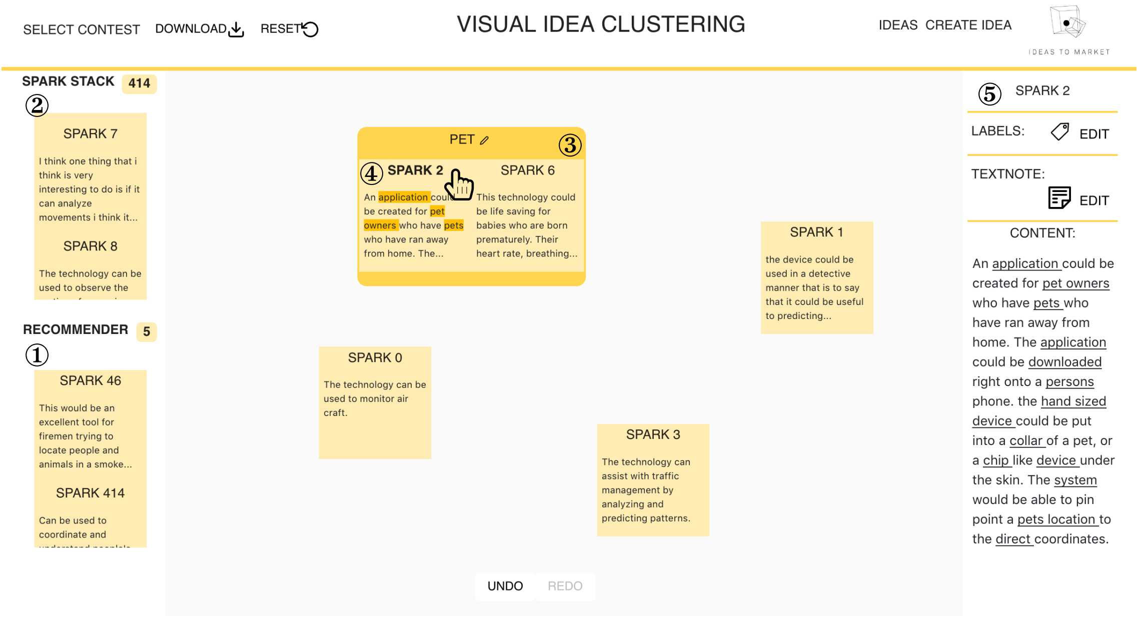

idea generation
1. Divergent Phase
brainstorming
and generating idea sparks
2. Clustering Phase
organizing and making sense of the idea sparks
by grouping them on a two-dimensional idea-map
3. Convergent Phase
synthesize i.e. summarizing and evolving
idea sparks to a few novel and
promising ideas

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 fare 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



SPARK 7: I think one thing that i think is very interesting to do is if it can analyze movements i think it...

SPARK 8: The technology can be used to observe the

SPARK 46: This would be an excellent tool for firemen trying to locate people and animals in a smoke...

SPARK 414: Can be used to coordinate and

SPARK 2: An application could be created for pet owners who have pets who have ran away from home. The...

SPARK 6: This technology could be life saving for babies who are born prematurely. Their heart rate, breathing...

SPARK 0: The technology can be used to monitor air craft.

SPARK 1: the device could be used in a detective manner that is to say that it could be useful to predicting...

SPARK 3: The technology can assist with traffic management by analyzing and predicting patterns.

CONTENT: An application could be created for pet owners who have pets who have ran away from home. The application could be downloaded right onto a persons phone. the hand sized device could be put into a collar of a pet, or a chip like device under the skin. The system would be able to pin point a pets location to the direct coordinates.

Figure 2: Clustering view of the Orchard interface, where the recommender 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. My thesis is about counteracting this problem and increasing efficiency in the clustering process.

2 Thematic Classification of Thesis

2.1 Orchard Clustering Application

In the research project Ideas2market the Clustering Web-Application Orchard has been developed to support the second and third phases of the ideation process (see Figure 2 for the clustering step). 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 (see Figure 2 (5)).

2.2 Supporting Effective Collective Ideation at Scale

One solution to increase efficiency in synthesizing ideas is to introduce a predefined idea-map, where the idea sparks are organized in clusters by similarity score, where related and similar idea sparks are positioned near to each other [Siangliulue, 2017, 124]. In addition, it is easier for the user to internalize the idea-space and thus interact more with rare ideas [Siangliulue, 2017], which is a benefit in quality 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]. To increase efficiency in the clustering phase and prevent fixation on given categories I propose an idea recommender system, as shown in Figure 2 on the left (1), for the Orchard Application. So that the user can walk through the idea sparks lead by his changing interest of categories, clusters, and topics.

2.3 Recommender System

Recommender systems (RSs) became popular in commercial platforms like Online Shops and Streaming Services and is a subclass of information filtering system, that predicts the user's interest for specific content. To navigate successfully through large information spaces RSs have been an important tool. There are different types of RSs that are often combined into hybrids. One of the most prominent types is collaborative filtering where predictions are derived from the behavior of other users. Another approach is content-based filtering, which is based on the description of the items. Through the user's interaction history with the RS, a classifier of his or her preferences is learned to predict the user's interest in the following interaction. Knowledge-based is a third type, that provides recommendations through reasoning about which items match the user's requirements. Thus it requires specific knowledge about the items (e.g. an underlying ontology or data-model with many properties) and the user's requirements. To map these requirements to a ranking of items, the recommendation criteria have to be defined. It can cause a knowledge bottleneck due to the consideration of only explicit knowledge in the RS. But in contrast to the first two types, the knowledge-based advantage is that it has no "cold-start" problem, where a lot of users data or ratings are required to provide accurate recommendations and gain implicit knowledge about users and items.

The RS in Orchard is knowledge-based. To provide recommendations the user's requirements are specified by selecting idea sparks and concepts of the user's current interest. Criteria for the RS is the semantic similarity of concepts, described in the following section 2.5. The Wikidata ontology contains the concepts of idea sparks and supplies the necessary information to apply measurements as recommendation criteria.

2.4 Knowledge Graph

For the similarity measures, I use the relations between defined concepts in a Knowledge Graph (KG). In the context of semantic web and linked data many Knowledge Graphs

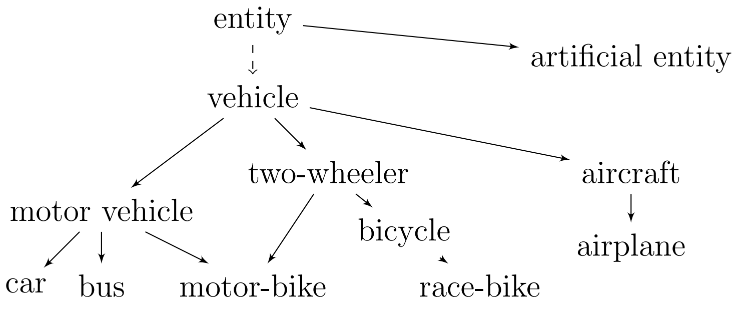


Figure 3: Part of the Knowledge Graph from Wikidata

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 like word sense disambiguation, topic modeling, and Question Answering [Nastase, 2008]. Semantic similarity is a metric to measure the similarity of concepts in the KG through their hierarchical relations [Zhu and Iglesias, 2017]. The advantage of this approach is that the similarity 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 edge are directed from the trees root to the leaves. 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. Wikidata records more than 69 million items[1] and covers most of the real-world entities and is considered useful as KG in this approach.

2.5 Similarity of Concepts

In the field of Semantic Similarity, there are several metrics to measure the similarity of terms, concepts and instances. Measuring semantic similarity divides up into mainly corpus-based and knowledge-based approaches. Corpus-based semantic similarities metrics use statistical relations of words in a 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 meaning 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 score $s \in [0, 1]$. For two concepts c_i, c_j let $length(c_i, c_i)$ denote the shortest path between c_i, c_j then the similarity is calculates as

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

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

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

Definition 2.1. Information Content corpus-based:

Let c_i be a concept, given a large general text-corpus $Prop(c_i)$ 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 N is the total number of concepts observed in the text-corpus. The Information Content can be quantify as negative the log likelihood $-log_e Prob(c_i)$ [**?**], 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. The upper boundery depends on N, the occurences of all concepts in the text-corpus. E.g. for 6 Million concepts the maxium IC would be about 15.

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 $Prob(c_i) = \frac{count(entities(c_i))}{N}$ and $entities(c_i)$ is a set of entities of type c_i in the KG, so they all reach c_i through ancestral realtions. N is the total number of entities in the KG. E.g. resolves $entities(two - wheeler)$ to $two - wheeler, bicycle, motor - bike, race - bike$, as show in Figure 3.

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-judged similarity in meaning. Their metric *wpath* for measures of similarity 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 one 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 more abstract and ancestral to *two-wheeler*. [Zhu and Iglesias, 2017] define the semantic similarity methode as

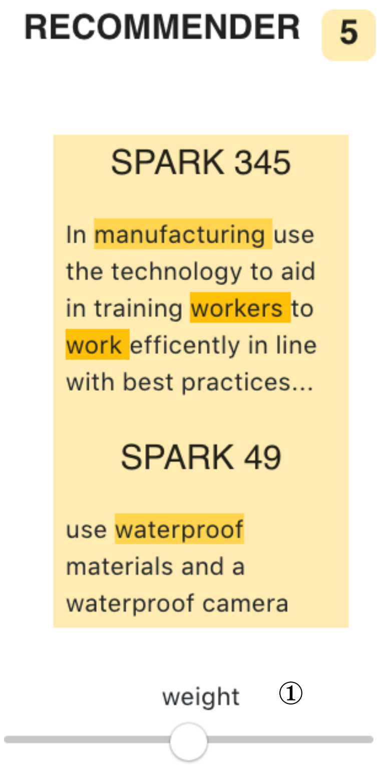
$$sim_{wpath(c_i,c_j)} = \frac{1}{1 + length(c_i, c_j) \cdot k^{IC(lcs)}}, \tag{2}$$

where the parameter $k \in (0, 1]$ weighs between IC of the LCS and the path length of two concepts. If $k = 1$ the IC has no influence to 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*, bacause it is less specific.

2.6 Human-Centered

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 stay's grounded.

The mentioned concerns apply to this thesis as well. The RS is 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. the RS in Orchard may respond with unrelated idea sparks for some concepts so that the user will try to identify and then avoid using these concepts. So when the user has a good mental model of the system, the interaction is more effective. Which gives reason to consider interpretability and transparency of the system when thinking about 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 the concept that is most interesting of an idea spark, to be more specific in what he or she is interested in. Another feature for the User Interface will be a Slider in a range, as shown in Figure 4 (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 get an idea which weight leads to good results. For more transparency and a better understanding of the RS, the idea sparks in the recommender frame are visualized with highlighted concepts as well, as Figure 4 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 RS and to understand why a certain idea spark is ranked as most similar.



SPARK 345

In manufacturing use the technology to aid in training workers to work efficently in line with best practices...

SPARK 49

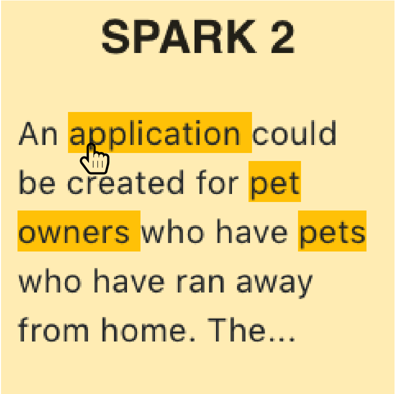
use waterproof materials and a waterproof camera

Figure 4: Recommender System with highlighted concepts of different saturation depending on similarity to the selected object. (1) Slider to change weight in *wpath* metric.

3 Goal

The goal of my thesis is to improve the clustering process in the Orchard application with a recommender system. So that the user interacts more with rear and valuable idea sparks and clusters more effectively without fixation on predefined categories. To apply the RS, concept similarity is calculated for all concepts extracted from 60 goldstandard idea sparks. The proposed metric *wpath* of Zhu and Iglesias [2017] combines meaningful sematic similarity measures for the application on a Wikidata KG and has shown outperforming results for the DBpedia ontology. As the Wikidata KG is especially large and complex it is of interest answering, how well semantic similarity metrics performe on such KGs compared to others and if *wpath* leads to novels results in this case. Therefore part of my thesis is a prove of concept on the mentric of Zhu and Iglesias [2017]. Given the similarity measures between concepts, concepts and ideas, and a similarity score of the Word Mover's Distance [Kusner et al.] between ideas, the RS for the Orchard clustering application can be implemented to support the functionality as discribed in the following.

In Orchard, as Figure 5 illustrates, the user can click on an idea spark or a highlighted concept of it's description to select the content, based on that idea sparks with similar concepts are recommended. The recommended idea sparks are displayed in the recommender frame sorted by highest similarity and the user can scroll through and drag them onto the whiteboard (see Figure 2 (1)).



SPARK 2

An application could be created for pet owners who have pets who have ran away from home. The...

Figure 5: Spark with highlighted concepts

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

The requirement for the input data of idea sparks is that the concepts of the idea descriptions are assigned to Wikidata items. In prepossessing all stop-word concepts, such as "I, my, You,. . . " and concepts that do not connect to get 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 an directed graph into an 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 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.] 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 recommender frame (see 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 (see Figure 5). 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, see Figure 4.

4.3 Validation

The validation consists of the following three steps:

1. To validate the measures of concept similarity, the implementation of *wpath* will be compared to 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]. *R&G* is a widely used dataset containing human assessment of word similarity for 65 word-pairs of common English nouns. Zhu and Iglesias [2017] uses the same dataset for *wpath* with corpus-based IC and graph-based IC with the DBpedia ontology. This comparison will be done for different parameter $k \in (0, 1]$ for *wpath*, to optimize the measurements for *R&G*.

2. To find the best weight k between path length and IC for the Wikidata KG, the *wpath* similarity will be applied on several idea-concepts for different parameters $k \in (0, 1]$.
3. As the last step, the recommender system will be tested with Orchard on a qualitative level with three or more persons who are familiar with the clustering process, to evaluate the research question, if the recommender system in Orchard is a helpful feature to effectively find clusters and creating a satisfying result.

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 recommender system in Orchard is implemented in JavaScript, React and Redux.

6 First Schedule

14. January	Implementation of IC_{graph}
6-15. January	Presentation of the thesis topic at HCC
14. January	Find a primary supervisor from TU-Berlin.
9. January	Annotate and connect *R&G* concepts to Wikidata
15. January	Registration of the bachelor thesis at the TU-Berlin
17. January	Implementation of *wpath*
23. January	Validation of *wpath* and comparison with dataset of Rubenstein and Goodenough [1965] and implementation of Zhu and Iglesias [2017].
30. January	Implementation of WMD for similarity measures.
4. Feburary	Test concept similarity implementation of *wpath* for Wikidata, to select best parameter k.
20. Feburary	Finish technical implantation, take measurements for similarity.
30. Feburary	The recommender system is integrated into Orchard for gold standard ideas.
3. March	Discussion of the results with Prof. Dr. Claudia Müller-Birn, Michael Tebbe and Maximilian Mackeprang.
25. 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`.

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

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.

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`.

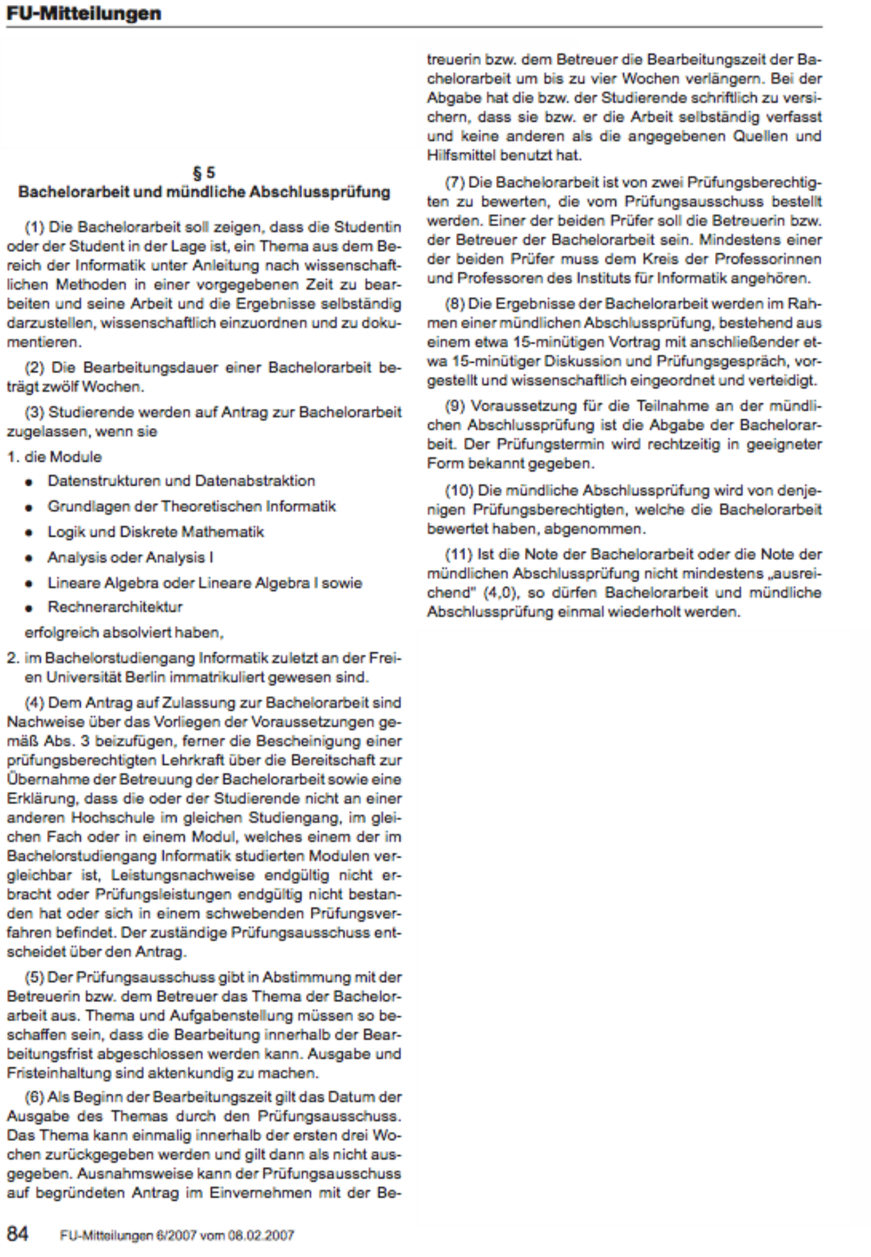
Kanya (Pao) Siangliulue. Supporting Effective Collective Ideation at Scale. May 2017. ISSN http://nrs.harvard.edu/urn-3:HUL.InstRepos:40046559. URL `https://dash.harvard.edu/handle/1/40046559`.

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



§ 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.

Figure 6: Auszug Prüfungsordnung Bachelor