Bachelorarbeit

Fakultät IV - Elektrotechnik und Informatik der Technischen Universität Berlin

Quality and Usability Lab

Institut für Informatik der Freien Universität Berlin

Human-Centered Computing (HCC)

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

Luka Stärk

Betreuer: Michael Tebbe

Erstgutachter: Prof. Dr. Sebastian Möller

Zweitgutachterin: Prof. Dr. Claudia Müller-Birn

Berlin, 17. September 2020

Abstract

Orchard is a creative clustering application developed in the Idea2Market project. The task of bringing many ideas into spacial-relation, called clustering, is part of a more extensive ideation process and is considered highly creative and beneficial to humans in developing ideas. Hence, taking away this task with an automated cluster generation might harm the ideation process. Still, manually going through a stack of hundreds of ideas can be mundane. This thesis implements a recommendation feature that supports users in clustering many ideas to synthesize and evolve them into more advanced and promising ideas. The creative task of clustering remains in the user's hand. However, through interaction they can control which ideas are explored next when building their representation of ideas by clustering them. This is done by repeatedly specifying a target for which the recommendation feature provides similar ideas.

For the similarity measurements, to recommend similar ideas, the recommendation feature applies the information of knowledge graphs (KGs) that describe resources and encode relations and facts of concepts. This work includes experiments on the performance of five variants of concept similarity methods and one idea similarity method using the Wikidata KG to ensure the accuracy of the used similarity measures. The similarity method *path* uses the path-length between concepts of the KG, the methods *res, lin, jcn* differently apply a graph-based Information Content measurement and *wpath* combines the two. The evaluation points out the potential of Wikidata as a knowledge source for semantic similarity. The semantic similarity approach proved its suitability by a high correlation to human perceptions of word similarity for five well-studied datasets, namely R&G, M&C, WS353, WS353-Sim, and SimLex. *wpath* performed best in three out of five datasets and thus was applied in the recommendation feature. The idea similarity method is a variant of the Word Mover's Distance and proved its accuracy in further experiments.

A Think-Aloud user study with five participants conducted for this thesis shows that the recommendation feature achieves a targeted and more efficient iteration through the ideas for the clustering tool Orchard. With the novel feature, the participants reported they were able to explore the ideas for clusters of their current interest, which enabled an in-depth iteration on various topics. These observations could indicate an improvement in the idea synthesis, which could be looked at in further research.

Zusammenfassung

Orchard ist eine Clustering Applikation, die im Rahmen des Idea2Market-Projekts entwickelt wurde. Die Aufgabe, viele Ideen in einen räumlichen Zusammenhang zu bringen, das sogenannte Clustering, ist Teil eines umfassenderen Ideenfindungsprozesses und erfordert Kreativität und hilf bei der Weiterentwicklung von Ideen. Daher könnte es dem Ideenfindungsprozess schaden, wenn diese Aufgabe durch eine automatisierte Cluster-Generierung wegfällt. Dennoch kann es banal sein, manuell durch einen Stapel von Hunderten von Ideen zu gehen. Diese Arbeit implementiert ein Empfehlungsfeature, das die User*in beim Clustern vieler Ideen unterstützt, um diese zu synthetisieren und zu fortgeschritteneren und vielversprechenderen Ideen weiterzuentwickeln. Die kreative Aufgabe des Clusterings bleibt in der Hand der User*in. Durch Interaktion kann sie jedoch steuern, welche Ideen als nächstes untersucht werden, während sie ihre räumliche Darstellung der Ideen durch Clustering aufbaut. Dies geschieht durch wiederholtes Auswählen einer Idee oder eines Wortes, für das das Empfehlungsfeature ähnliche Ideen liefert.

Für die Ähnlichkeitsmessungen, um ähnliche Ideen zu empfehlen, wendet das Empfehlungsfeature die Informationen von Knowledge Graphs (KGs) an, die Ressourcen beschreiben und Beziehungen und Fakten von Konzepten kodieren. Diese Arbeit umfasst Experimente zu fünf Konzeptähnlichkeitsmethoden und einer Ideenähnlichkeitsmethode unter Verwendung des Wikidata KG, um die Genauigkeit der verwendeten Ähnlichkeitsmaße zu verifizieren. Die Ähnlichkeitsmethode *path* verwendet die Pfadlänge zwischen den Konzepten im KG, die Methoden *res, lin, jcn* wenden auf unterschiedliche Art und Weise eine grafbasierte Messung des Information-Contents an und *wpath* kombiniert beide.

Die Evaluation zeigt das Potenzial von Wikidata als Wissensquelle für semantische Ähnlichkeit auf. Der semantische Ähnlichkeitsansatz bewies seine Eignung durch eine hohe Korrelation zur menschlichen Wahrnehmung von Wortähnlichkeit für fünf gut untersuchte Datensätze, R&G, M&C, WS353, WS353-Sim und SimLex. Die Methode *wpath* schnitt in drei von fünf Datensätzen am besten ab und wurde daher im Empfehlungsfeature angewandt. Die Ideenähnlichkeitsmethode ist eine Variante der Word Mover's Distance und bewies ihre Eignung in weiteren Experimenten.

Eine für diese Arbeit durchgeführte Think-Aloud-Userstudie mit fünf Teilnehmer*innen zeigt, dass das Empfehlungsfeature für das Clustering-Tool Orchard eine zielgerichtete und effizientere Iteration durch die Ideen ermöglicht. Die Teilnehmer*innen berichteten, dass sie mit dem neuartigen Feature gezielt Ideen für Cluster ihres aktuellen Interesses untersuchen konnten, was eine vertiefung zu verschiedenen Themen ermöglichte. Diese Beobachtungen könnten auf eine Verbesserung der Ideensynthese hindeuten, was in weiterer Forschung untersucht werden könnte.

Contents

List of Figures

List of Tables

1 Introduction

Enhancing human abilities through software support and intelligent tools has witnessed diverse approaches. In highly creative contexts, such as the process of ideation, intelligent tools have a risk of taking over tasks important to humans, which might lead to a decrease in creativity. Hence, it is especially valuable in such contexts to center human needs when developing intelligent tools. This thesis is about intelligent human-centered software-support in a creativity context with the use of semantic similarity measures.

1.1 Motivation

The research project Ideas to Market 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 people 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 those on the market. Still, finding valuable idea sparks has proven to be challenging. Due to the large number of idea sparks, it becomes unfeasible to manually check all of them and derive their benefits for advanced ideas. Ideas to Market aims to solve these problems with software support and by researching the human needs in creative processes. This thesis extends the existing software, Orchard, with a new feature. The software-supported collaborative-ideation process can be described in three phases, as illustrated below in Figure 1.1.

First, many idea sparks are collected in the Divergent Phase. Then for the Clustering Phase, experts organize and categorize idea sparks, by placing them in spacial relation on a whiteboard. Last, ideas are synthesized from the clustering. When clustering, the categories of the emerging clusters and the connections between idea sparks are not always clear. The decision to create a cluster is based on feeling and intuition and can be reversed any time. During the process, an order develops, and the relationships between idea sparks become more visible [Tassoul and Buijs, 2007]. Clustering is beneficial as an activity in acquiring a more profound understanding of the idea-space and producing more valuable ideas in the Convergent Phase [Tassoul and Buijs, 2007]. However, for growing numbers of idea sparks, it becomes more challenging to

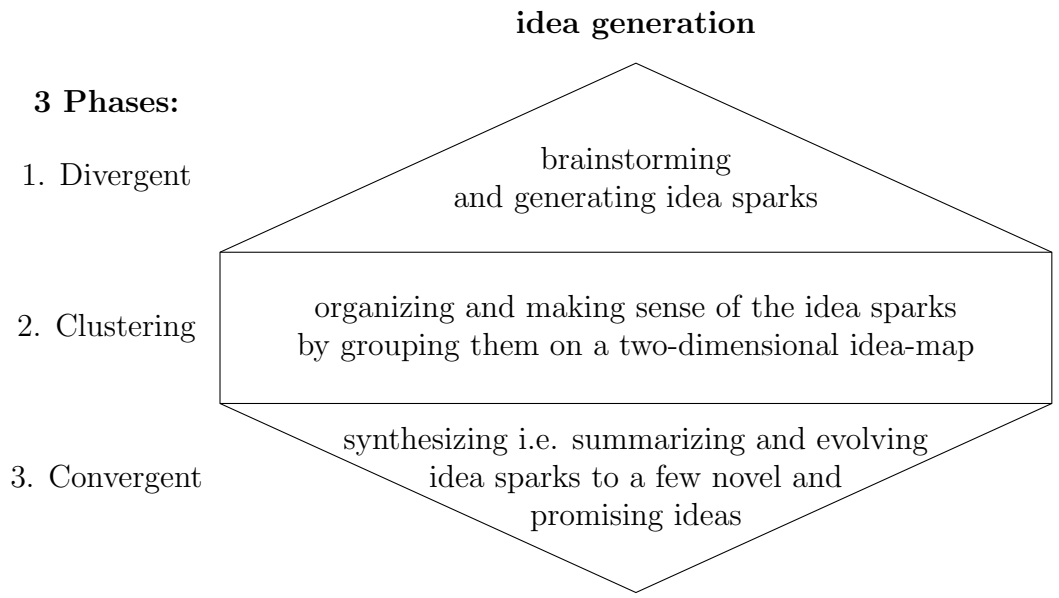


Figure 1.1: Three Phase Diamond of the Innovation Process based on Tassoul and Buijs [2007]

organize the idea-space and to take into account all potential idea sparks for one cluster. This task can then be monotonous and time-consuming, which can decrease the quality of the idea syntheses [Siangliulue, 2017]. This 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 1.2 on the left (1). Thereby, the user can walk through the idea sparks led by their changing interest of categories, clusters, and topics.

1.2 Orchard Clustering Application

In the research project Ideas to Market, the clustering Web-Application Orchard has been developed to support the Clustering Phase of the ideation process. Figure 1.2 displays the graphical user interface. Orchard is a tool for creative ideation to synthesize ideas from numerous idea sparks effectively. It is inspired by the *IdeaHound* project [Siangliulue et al., 2016]. For the Clustering Phase, the user can drag and drop ideas from the *Spark Stack* to the whiteboard. The user creates clusters and moves idea sparks to the whiteboard. They 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 1.2 (5). For the Convergent Phase, experts can write and archive their idea syntheses in the application.

The recommendation feature contains the following functionalities: As Figure 1.3 illustrates, the user can click on an idea spark or a highlighted terms of its description to select the current target. Based on that, idea sparks are recommended. Hence, the user specifies their interest for a particular topic and

Figure 1.2: 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 clicking on it and its caption is set in bold, see *SPARK 2* (4).

related idea sparks are listed in the recommendation frame sorted by highest similarity. The user can scroll through and drag them to the whiteboard, as shown in Figure 1.2 (1). Thus, to measure the similarity between spark ideas, this work implements and evaluates a semantic similarity approach.

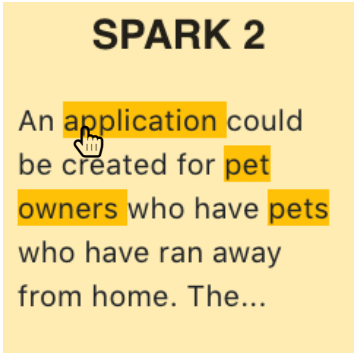


Figure 1.3: Spark with highlighted concepts

1.3 Semantic Similarity

There are several methods to measure the semantic similarity of terms, concepts, and instances. Semantic similarity is a metric that defines the commonality of concepts for their hierarchical relations. Semantic relatedness widens

this definition by including other relations [Zhu and Iglesias, 2017]. Measuring semantic similarity divides up into mainly corpus-based and knowledge-based approaches. Corpus-based semantic similarity metrics use statistical relations of words in extensive text collections. Two words are similar when their surrounding context is similar. Therefore, it relies on the occurrence of words and ignores the different meanings a word can have [Zhu and Iglesias, 2017]. In contrast, knowledge-based semantic similarity metrics measure similarities between defined concepts in a Knowledge Graph (KG). A Knowledge Graph records the relations between labeled entities through connected descriptive edges. Hence, similarities can be measured through edges between concepts in the KG. Different graph-based strategies that emerged in related works have shown considerable success, such as random walk, spreading-activation, information content, and path distances between concepts Zhu and Iglesias [2017]. These strategies have been combined into various methods. This thesis evaluates an idea similarity method and five concept similarity methods using the Wikidata KG[1].

1.4 Collaborative Ideation at Scale

In "Supporting Effective Collective Ideation at Scale", Siangliulue [2017] evaluated solutions to increase efficiency in synthesizing numerous amounts of ideas. One possibility they explored was to introduce a predefined idea-map. The idea sparks are then organized in clusters sorted by similarity score so that related and similar idea sparks are positioned near to each other [Siangliulue, 2017, 124]. Further, it is easier for the user to internalize the idea-space and thus interact more frequently with rare ideas [Siangliulue, 2017]. That is beneficial for the user because ideas can often be mundane or repetitive [Siangliulue et al., 2016]. With this, the user is then 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 approach to support and accelerate the clustering process without the mentioned drawback of fixation on given categories.

1.5 Goal of Thesis

This work aims to improve the synthesis of ideas by supporting the user in the clustering process using Orchard with the following goals:

First, the novel recommendation feature, with the functionality described in Section 1.2, ought to increase efficiency when clustering and enabling a targeted and human-controlled iteration through the idea sparks. This aids the user to interact more with valuable idea sparks that concern their current interest, without the fixation on predefined categories, as mentioned in Section 1.4. In

[1] `https://www.wikidata.org`

order to prove this point, a Think-Aloud user study with five participants has been conducted.

The semantic similarity approach of this work provides the necessary similarity measures. Hence, the second goal of this thesis is to prove the accuracy of the semantic similarity measures. Therefore, as described in Section 1.3, popular semantic similarity methods are implemented using the Wikidata KG. The performance evaluation of the similarity approach allows a well-founded decision to apply the similarity measures in the recommendation feature and elaborate on its limitations.

1.6 Structure of thesis

This thesis is organized in five chapters including the Introduction 1 and the final Conclusion chapter 5 as illustrated in Figure 1.4. Chapter 2 describes the theoretical foundation of this work by discussing prominent approaches of semantic similarity. This is done by introducing the Wikidata knowledge graph as a resource for the similarity measures. Section 2.3 presents a variant of the Word Mover's Distance to derive sentence similarity from the concept similarity. The last section 2.4 introduces aspects of human-centered artificial intelligence that are considered for interface design and the implementation of the recommendation feature.

Chapter 3 describes the implementation of semantic similarity methods in Section 3.1. In Section 3.2, experiments for word- and idea similarity are conducted with accessible datasets of human similarity assessments for pairs of English nouns, sentences, and ideas. In chapter 4, the best performing similarity method, *wpath* published by Zhu and Iglesias [2017] finds application in the recommendation feature for Orchard. With a user-study of 5 participants, the feature is evaluated regarding its purpose.

Chapter 5 reflects upon the goal of the recommendation feature and gives an outlook on possible follow-up research and further improvements.

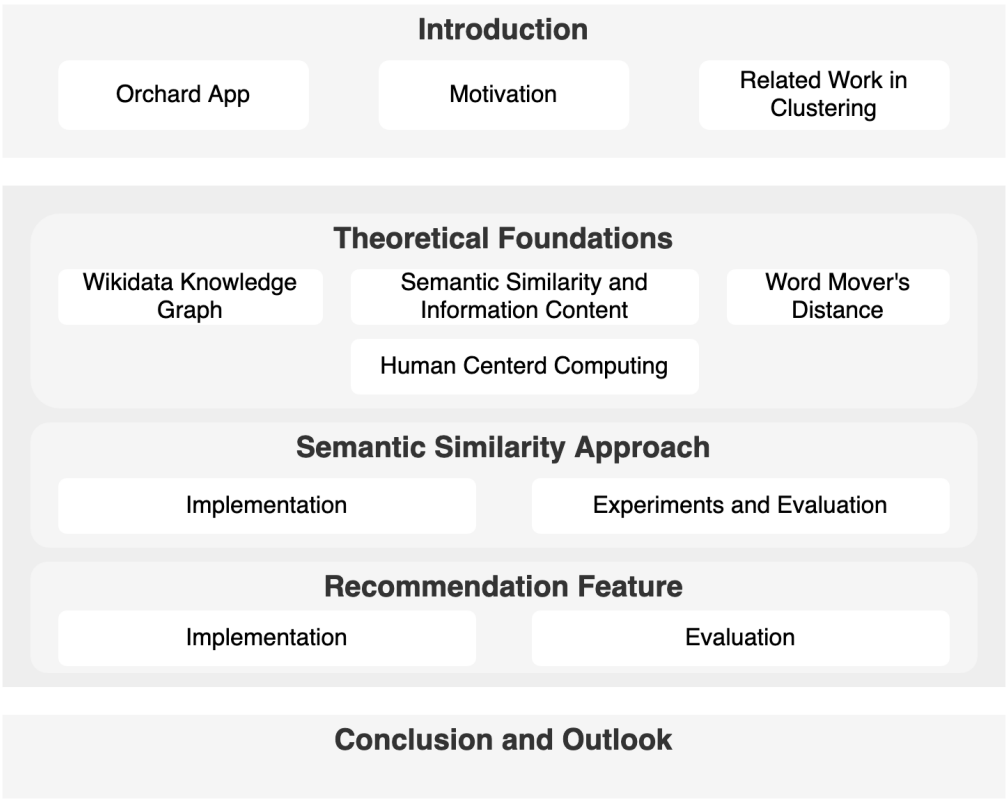


Figure 1.4: Structure of the thesis as a visualized overview.

2 Theoretical Foundations

This chapter explains the foundation of the semantic similarity methods implemented for this thesis. Section 2.1 provides definitions for knowledge graphs and introduces technologies of Semantic Web such as RDF and SPARQL. Section 2.2 defines the concept similarity methods and Section 2.3 the Word Mover's Distance for idea similarity. Furthermore, the concept of Information Content and Least Common Ancestor are explained in the context of KGs. Section 2.4 presents the aspects of human-centered artificial intelligence considered in the recommendation feature.

2.1 Knowledge Graphs

In the context of semantic web and linked data, many knowledge graphs like DBpedia[1], WordNet[2], and Wikidata[3], to name a few, are freely accessible and have gained increasing popularity. Such information networks find application in different tasks in the field of Natural Language Processing and Information Retrieval such as Word Sense Disambiguation, Topic Modeling, and Question Answering [Nastase, 2008]. Knowledge graphs record general knowledge in a semantic network that can be formally defined as follows:

Definition 2.1.1 (Knowledge Graph)**.** Let V be a set of vertices, E a set of Edges, and L a set of property labels, then a knowledge graph G can be defined as a directed labeled graph $G = (V, E, L)$. Edges in E consist of triples such as (v_1, p, v_2), where $v1, v2 \in V$ and $p \in L$ is a property label.

The recommendation feature depends on semantic similarity measures between concepts in a Knowledge Graph (KG). The occurring terms in the idea spark's descriptions are assigned to such concepts of the KG. A concept can be described by various terms and applies to a group of instances. For example, synonyms such as *car* and *automobile* are referencing the same concept. If the label of an instance is *Peugeot 104* then *car* would be one concept that applies to that instance. A KG makes this information accessible.

The advantage of KGs is that the similarity between two concepts becomes interpretable and comprehensible when looking up the path that connects them. Additionally, retrieving information about a common ancestor concept in the directed graph can provide more understanding. Such as in the KG of Figure 2.1, where the edges are directed from top to bottom with the

[1] `https://wiki.dbpedia.org`
[2] `http://wordnet-rdf.princeton.edu`
[3] `https://www.wikidata.org`

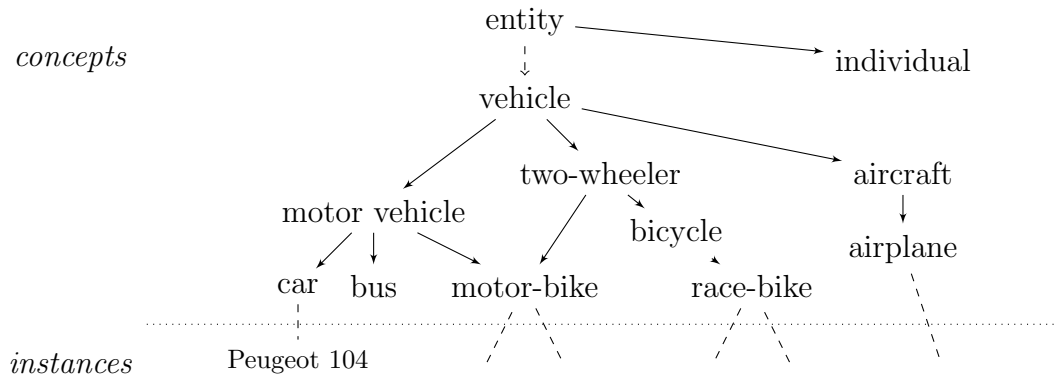


Figure 2.1: Part of a Knowledge Graph

concept *vehicle* as a common ancestor. Also, tracing back presumably false results to the origin by identifying incorrect relations in the KG is feasible. In statistical approaches, this information is more challenging to extract because the semantic relationships of concepts are not accessible as statements, like in a KG, but instead as distances in high-dimensional spaces.

2.1.1 Wikidata

Wikidata is a general domain knowledge-base that is openly and freely accessible and records more than 80 million items[4]. It covers most of the real-world entities and is considered useful as a KG for this approach. In contrast to most Knowledge Graphs, that reason knowledge from other sources such as Wikipedia, Wikidata is also curated by humans and, therefore, a continuously growing database. The relations between entities are described by property labels that can be categorized into two types: transversal and taxonomic relations [Paul et al., 2016]. Taxonomic properties are hierarchical relations between concepts. In Wikidata these properties are for example *subclass of* and *instance of*. Taxonomic relations are transitive, accordingly, when A subsumes B and B subsumes C, then also does A subsume C. Transversal properties describe the horizontal relation between concepts that do not imply any partial order, though these relations mostly are directed. The property labels are for example *part of* or *used by*.

2.1.2 RDF: Resource Description Framework

RDF is a standard data model that describes relations and statements of resources in triples *subject-predicate-object*. It extends the linked structure of the World Wide Web and is a component of the Semantic Web as a standard for self-descriptive and machine-readable resources. A collection of RDF statements are a labeled directed graph. The following example is a statement from

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

Wikidata to illustrate an RDF-triple in Figure 2.2:

```
    PREFIX wd:https://www.wikidata.org/wiki/
    PREFIX p:https://www.wikidata.org/wiki/Property:

    wd:Q11442   p:P279          wd:Q63247926
#   bicycle     subclass of     human-powered transport
#   subject     predicate       object
```

Figure 2.2: RDF-triple statement

2.1.3 SPARQL

SPARQL is the standard graph query-language to retrieve and manipulate data in RDF databases. A query uses Prefix declarations that shorten prefixes of URLs to improve readability. There are different query-types to specify the operation, such as SELECT, ASK, and CONSTRUCT. Explaining the SELECT query satisfies to understand how to retrieve statements from Wikidata for this work. SELECT defines a mapping between the statements in the database and the variables to return. The following example in Figure 2.3 describes a query that returns, for the variable *subItem*, all objects that fulfill the statement in line 5, thus, are a subclass of the concept bicycle. For execution, the query is

```
1       PREFIX wd:https://www.wikidata.org/wiki/
2       PREFIX p:https://www.wikidata.org/wiki/Property:
3       SELECT ?subItem
4       WHERE {
5           wd:Q11442    p:P279         ?subItem
6       #   bicycle    subclass of    variable subItem
7       }
```

Figure 2.3: Example SPARQL-query

send to the corresponding $SPARQL$-endpoint. The Wikidata KG is accessible at `https://query.wikidata.org/`.

2.2 Semantic Similarity Methods

The most simple similarity methods 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 a knowledge

graph $G = (V, E, L)$ and two concepts $c_i, c_j \in V$ let $length(c_i, c_i)$ denotes the length of the shortest path between c_i, c_j, then the similarity is calculated as

$$sim_{path}(c_i, c_j) = \frac{1}{1 + length(c_i, c_j)}. \tag{2.1}$$

Another widely used measurement is the Information Content (IC). The IC of a concept indicates how abstract or specific a concept is and how much information the entities of a concept have 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:

Definition 2.2.1 (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 2.1, 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 of $Prob(c_i)$ [Resnik, 1995]. Hence,

$$IC_{corpus}(c_i) = -log_e Prob(c_i).$$

This shows that the IC of concept c_i increases when the probability decreases and if there would be one concept subsuming all other concepts its IC would be 0, as illustrated by the diagram in Figure 2.4.

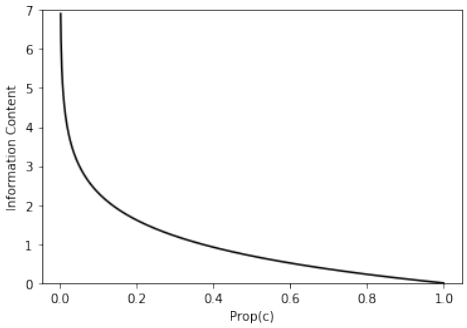


Figure 2.4: Decrease of the $IC(c)$ for increasing probability of the occurrence of concept c and its associated concepts.

Definition 2.2.2 (Information Content graph-based)**.** Let c_i be a concept, then

$$IC_{graph}(c_i) = -log_e Prob(c_i),$$

where

$$Prob(c_i) = \frac{\|entities(c_i)\|}{N},$$

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. For example, *entities(two-wheeler)* resolves to {*two-wheeler, bicycle, motor-bike, race-bike*}, as shown in Figure 2.1.

The IC characteristic is used in similarity metrics and often applied for the Least Common Ancestor (LCA) of the two concepts, which is defined for trees and directed acyclic graphs (DAG). In a tree, the LCA is the unique ancestor that is not ancestral to any other common ancestor of two concepts. For DAG the same definition yields a set of Least Common Ancestors (LCAs) because there could be multiple common ancestors that are not ancestral to any other common ancestor [Kowaluk and Lingas, 2005], such as the LCAs c_2 and c_3 of c_i and c_j in the example DAG in Figure 2.5.

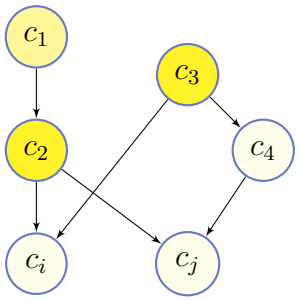


Figure 2.5: Directed Acyclic Graph

Assuming the KG is a DAG for which the LCA is further specified then the following two definitions apply.

Definition 2.2.3. The Least Common Subsumer is a common term in the context of ontologies since one concept subsumes their descendants in meaning for taxonomic relations. For this thesis, the Least Common Subsumer of two concepts c_i, c_j is denoted by $LCS(c_i, c_j)$, which is the LCA with the highest IC. Hence, the LCS is the most specific ancestral concept shared by both concepts c_i, c_j.

Alternatively, the selection between the common ancestors is defined as follows:

Definition 2.2.4. The Nearest Common Subsumer (NCS) is the common ancestor with the shortest distance for the two concepts. Let CA be the set of common ancestors for the concepts c_i and c_j, then the $NCS(c_i, c_j) = ncs \in CA$, where

$$length(ncs, c_i) + length(ncs, c_j) = \min_{ca \in CA} length(ca, c_i) + length(ca, c_j).$$

For example, in the KG shown in Figure 2.1, both, the LCS and the NCS of *motor-bike* and *bicycle* is the concept *two-wheeler*, and not *vehicle*, because it is less specific, ancestral to *two-wheeler* and connects the two for the shortest path.

The following concept similarity methods are defined for the LCS. However, they are as well valid with the use of NCS instead. When the NCS is used it is denoted by $method_{NCS}$ for any of the following *method*s.

Based on that, the *res* method [Resnik, 1995], measures the similarity of two concepts c_i, c_j as follows:

$$sim_{res}(c_i, c_j) = IC(LCS(c_i, c_j)) \tag{2.2}$$

Extending *res*, the *lin* method [Lin, 1998] considers the ratio between the IC of the two concepts and the IC of their LCS.

$$sim_{lin}(c_i, c_j) = \frac{2IC(LCS(c_i, c_j))}{IC(c_i) + IC(C_j)} \tag{2.3}$$

The *jcn* method [Jiang and Conrath, 1997] is similar to the *lin* method, besides it uses a distance function to capture the ratio between the IC of the LCS and its concepts.

$$dist_{jcn}(c_i, c_j) = IC(c_i) + IC(C_j) - 2IC(LCS(c_i, c_j))$$

$$sim_{jcn}(c_i, c_j) = \frac{1}{1 + dist_{jcn}(c_i, c_j)} \tag{2.4}$$

Zhu and Iglesias [2017] discuss different metrics of concept similarity in Knowledge Graphs and compare them to their approach *wpath* with gold standard data sets of human judgements of similarity. Their metric *wpath* for semantic similarity measures is slightly outperforming other widely used metrics. It considers shortest-path length between two concepts in the Knowledge Graph and the Information Content (IC) of their LCS. [Zhu and Iglesias, 2017] define their semantic similarity method as

$$sim_{wpath(c_i,c_j)} = \frac{1}{1 + length(c_i, c_j) \cdot k^{IC(lcs)}}, \tag{2.5}$$

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. For example, *car* and *bus* are more similar than *two-wheeler* and *aircraft*. Though for both pairs, the path length equals two, as shown in Figure 2.1. But the IC of *motor-vehicle* as their LCS is greater than the IC of *vehicle*, because *motor-vehicle* is more specific.

The graph-based method has proven excellent results by Zhu and Iglesias [2017]. Hence, it is of further interest for this thesis to explore the use of Wikidata to calculate the IC. Thus, for the semantic similarity measures of this work, different IC_{graph} methods are calculated and analyzed toward their performance.

2.3 Word Mover's Similarity

To provide recommendations of idea sparks in Orchard, they need to become comparable.Then the most similar idea sparks to the user's selected idea spark can be recommended. The idea sparks are not yet comparable to each other, with sheer similarities between the idea spark's concepts. Thus, a metric that derives an idea spark similarity from the underlying concept similarities needs to be defined. For this, a variant of the Word Movers Distance (WMD) [Kusner et al., 2015], namely the Word Mover's Similarity (WMS), can be used to compute similarities based on the previously calculated concept similarities. The Word Mover's Distance is a distance function between text documents that initially applies to vector word embeddings (*word2vec*), where the Euclidean distance of vectors describes the relations of words. For the approach in this thesis, the Euclidean distance of vector word embeddings is replaced with the similarity measures of concepts, see equation 2.2. The WMD is an instance of a well-studied transportation problem, namely Earth Movers Distance [Kusner et al., 2015]. The WMD is a linear minimization problem to transport, for this case, all words from one document to the other for minimal cost. The words are represented in a vector space and the transportation cost between words in different documents is the Euclidean distance.

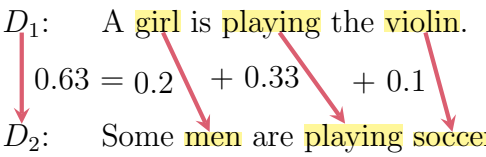


Figure 2.6: Shows the maximum flow between the components of D_1 and D_2, where the $nBOC$ are of equal length.

By replacing the vector representations with concept similarities it becomes a maximization problem, that maximizes the similarity between two documents. Idea sparks are viewed as documents that contain a finite number of concepts. A document D is represented as a normalized bag of concepts (nBOC) distribu-

tion $d \in \mathbb{R}^n$, where n is the number of unique concepts $c_i \in D$. The similarity of two documents D and D' is provided by the solution of the following linear program. Therefore, $T_{i,j}$ denotes the transport flow for each concept $c_i \in D$ to $c_j \in D'$ and $s(c_i, c_j)$ is the corresponding similarity:

$$\underset{T \leq 1}{max} \sum_{i,j=1}^{n} T_{i,j} s(c_i, c_j) \tag{2.6}$$

subject to:

$$\sum_{j=1}^{n} T_{i,j} = d_i \forall i \in \{1, \ldots, n\}$$

$$\sum_{i=1}^{n} T_{i,j} = d'_j \forall j \in \{1, \ldots, n\}$$

In example Figure 2.6, the documents D_1 and D_2 hold the same $nBOC$ distribution, because all concepts appear exactly once and the number of unique concepts for each one is three. The distribution value $d_i \in d$ for each concepts therefore amounts to 0.33. The similarity in Figure 2.6 maximizes when all flow from $d1_i \in d1$ goes to $d2_j \in d2$, where $s(c_i, c_j)$ is maximal. In case of unequal distributions of the compared documents, the transport flow splits in T to maximize the equation (2.3). The equation is constrained so that all outgoing flow of a concept $c_i \in D$ is limited in $d_i \in d$ and all incoming flow for a concepts $c_j \in D'$ is limited in $d'_j \in d'$.

This approach ignores the sequential order of words, incorporates the similarity of concepts pairs into the document similarity, and is stable for different lengths of the compared documents, which makes it applicable to the idea sparks. For a comparison with large numbers of documents and large nBOC, the linear problem becomes prohibitive, since the problem scales an average complexity of $O(p^3\ log\ p)$, where p denotes the number of unique words in the two documents [Pele and Werman, 2009]. Because of the limited number of idea sparks, this is of no concern in this thesis. However, there are efficient lower bounds to the WMD, which allow efficient computation of the problem for a k-nearest neighbor (kNN) search. The work of Kusner et al. [2015] has shown that the WMD, applied onto $word2vec$ embeddings, leads to very low error rates for the kNN task on eight supervised document classification datasets. In experiments carried out for this work described in 3.2.2 for document similarity with the underlying concept similarities, also show proportionate results.

2.4 Human-Centered Approach

In the field of machine learning and beyond, human-centered approaches have gained extensive attention. As machine learning discovers 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. The perspective of human-centered artificial intelligence considers the socio-cultural aspects of humans and helps them to understand the interaction with an intelligent system [Riedl, 2019]. There is a risk for intelligent tools of taking over tasks important to humans [Holbrook, 2018], which could lead to a decrease in creativity. Clustering is considered a highly creative task that follows various strategies [Tassoul and Buijs, 2007]. Hence human-centered concepts of interpretability are especially valuable in such contexts. Thus, the user is not tempted to trust intelligent systems blindly and to perceive them as a "black box" [Gillies et al., 2016]. Viewing the problem through the human lens and centering human needs in the evaluation ensures that the problem stays grounded and the intelligent system tries not to solve tasks that are best performed by humans.

The mentioned concerns also apply to this thesis because the idea spark recommendations are based on knowledge graph statements 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. For example, users might experience recommendations as unrelated to some selected concepts, so that they will try to identify and then avoid using these concepts to receive recommendations. 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 are performed best by humans, makes the system more adaptive and gives the user the feeling of being in control. This means the user will be more likely to interact with the system [Abdul et al., 2018]. For example, in the Orchard application, the user selects a concept of an idea spark to be more specific about their current interest. For more interpretability and a better understanding of the recommendations, the idea sparks in the recommendation frame are visualized with highlighted concepts as well. This is illustrated in Figure 2.7 in which the color saturation depends on the similarity to the selected concept *car*. The visualization of similarity measures ought to support the user in creating a mental model of the recommendation feature and to understand why a particular idea spark ranks as most similar.

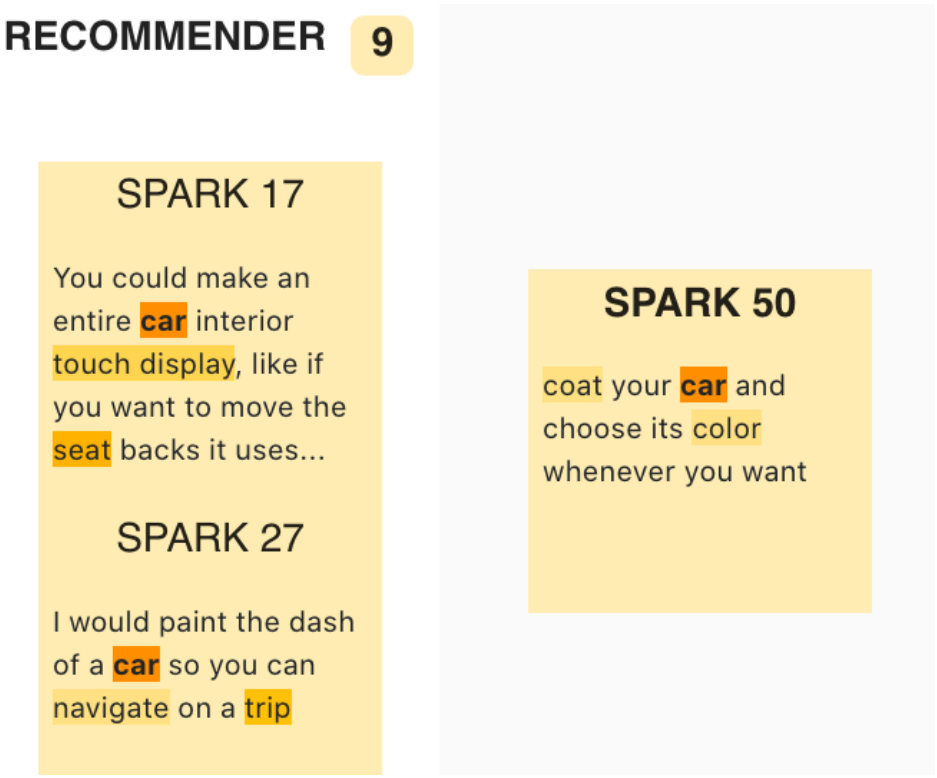


Figure 2.7: Recommendation Feature, displaying the top two recommended sparks with highlighted concepts for the selected concept *car*.

3 Semantic Similarity Approach

In this chapter, the implementation of concept and idea similarity methods presented in Section 2.2 using the Wikidata knowledge graph are illustrated in Section 3.1 . The following Section 3.2 describes the experiments and datasets to evaluate and measure the performance of the implemented similarity methods. To sum up this chapter, a brief discussion about the obtained results is presented in Section 3.3.

3.1 Implementation

The semantic similarity methods are implemented with the Python programming language, using the modules *networkx* [Hagberg, 2008] and *igraph-python* [Nepusz, 2006] for network algorithms and data structure. To compute the WMS the LP-modeler from *PuLP* [Mitchell, 2020] is applied. The UML-diagram in Figure 3.1 illustrates the different classes of the implementation and how they interact. The following section explains the basic functionalities of classes and algorithmic decisions.

3.1.1 Graph Construction

To evaluate the recommendation feature, the gold standard idea dataset[1] created by Mackeprang et al. [2019] is used. This contains verified annotations of DBpedia concepts of the occurring words of idea sparks, which are predominantly nouns. The concepts of the DBpedia database record references with the *owl:sameAs* predicate to the corresponding Wikidata concepts. In the class *Prepossessing* as illustrated in 3.1, the concept-annotations from DBpedia are mapped to Wikidata. The set of Wikidata concepts of the idea dataset is then filtered for stop-words, such as *I, my, You*. The stop-word list used is the `ENGLISH_STOP_WORDS` list of *sklearn* [Pedregosa et al., 2011].

A representation of the knowledge graph in a *DiGraph* class is implemented, which is a directed graph defined by a set of concepts. To distinguish those concepts for which we seek to measure similarities, they are referred to as *idea concepts*, as they appear in the idea sparks. The idea concepts are the last child layer in the directed graph, and their ancestral relations are recursively queried from the knowledge graph's *SPARQL*-endpoint.

[1] For the gold standard idea data set, see `https://osf.io/k2ey7/`

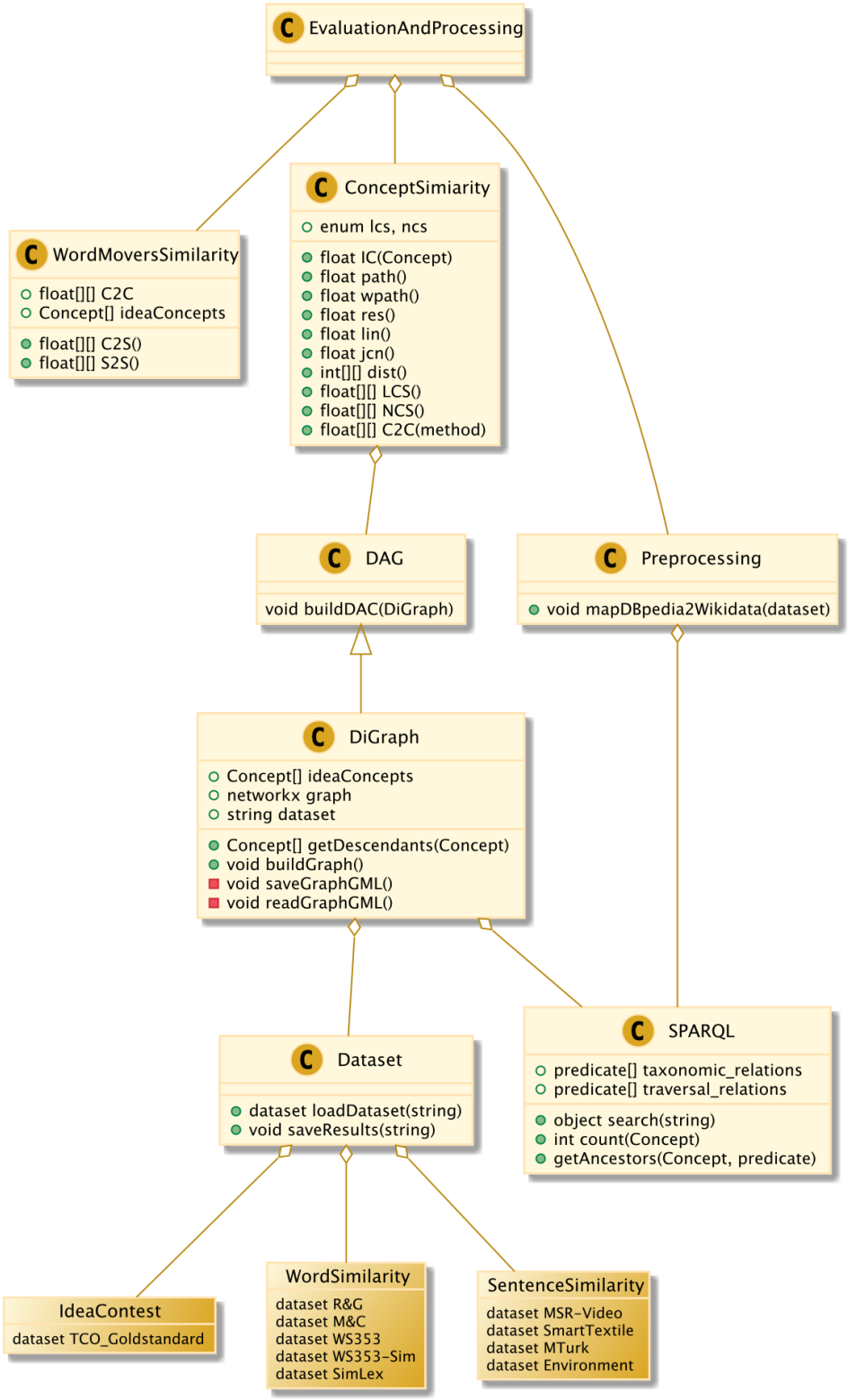


Figure 3.1: UML Class Diagram of the semantic similarity approach

In the implementation of DiGraph, a distinction is made between taxonomic and traversal relations. The taxonomic relations used from the Wikidata KG are

```
p:P279      # subclass of,
p:P31       # instance of,
p:P171      # parent taxon,
p:P460      # said to be the same as.
```

For these relations, all connected ancestors are recorded in the graph because they imply a partial order, as described in Section 2.1. The relations considered to be traversal in this work are

```
p:P361      # part of,
p:P527      # has part,
p:P1542     # has effect,
p:P1889     # different from,
p:P366      # use,
p:P2283     # uses,
p:P1535     # used by.
```

These relations are only retrieved for the idea concepts and enhance the concept similarity. In some cases, concepts are only connected by traversal relations to the graph. Hence, ignoring traversal relations, the similarity measures for such concepts would amount to zero.

The directed graph retrieved from Wikidata contains cycles. This prohibits a strictly taxonomic order and makes it unfeasible to calculate the IC of a concept, as described in the following section 3.1.2. Therefore, a directed acyclic graph (DAG) from the directed graph is generated in a bottom-up fashion by adding edges as long as no cycle emerges. This is beneficial because relations close to the bottom layered idea concepts are more significant for semantic similarity methods. With the reduction to a DAG, the graph loses about 10% of its edge relations. As an example, Figure 3.2 illustrates a subgraph of the DAG with all concepts subsumed by the concept *transport*, where the yellow concepts are the ones occurring in the idea sparks.

The $DAG = (E, V)$ enables a faster computation of all shortest paths, lowering the complexity to $\Theta(E + V)$ with a topological sorting algorithm [Cormen, 2009], and enables simple calculations of the LCSs between the idea concepts. The procedure is the following: In a first step, all shortest directed paths in the DAG are calculated. These distances between two concepts $c_i, c_j \in V$ along the topological order are denoted as $dist(c_i, c_j)$. With this distance measurement, only distances for concepts that have an ancestral relation are encountered. Hence, the *dist* measurement to define the shortest-path method between all concepts is used by deriving the distance between two concepts from their distance to a common ancestral concept. Two variants of the shortest-path measurement are implemented. One that minimizes the general path length

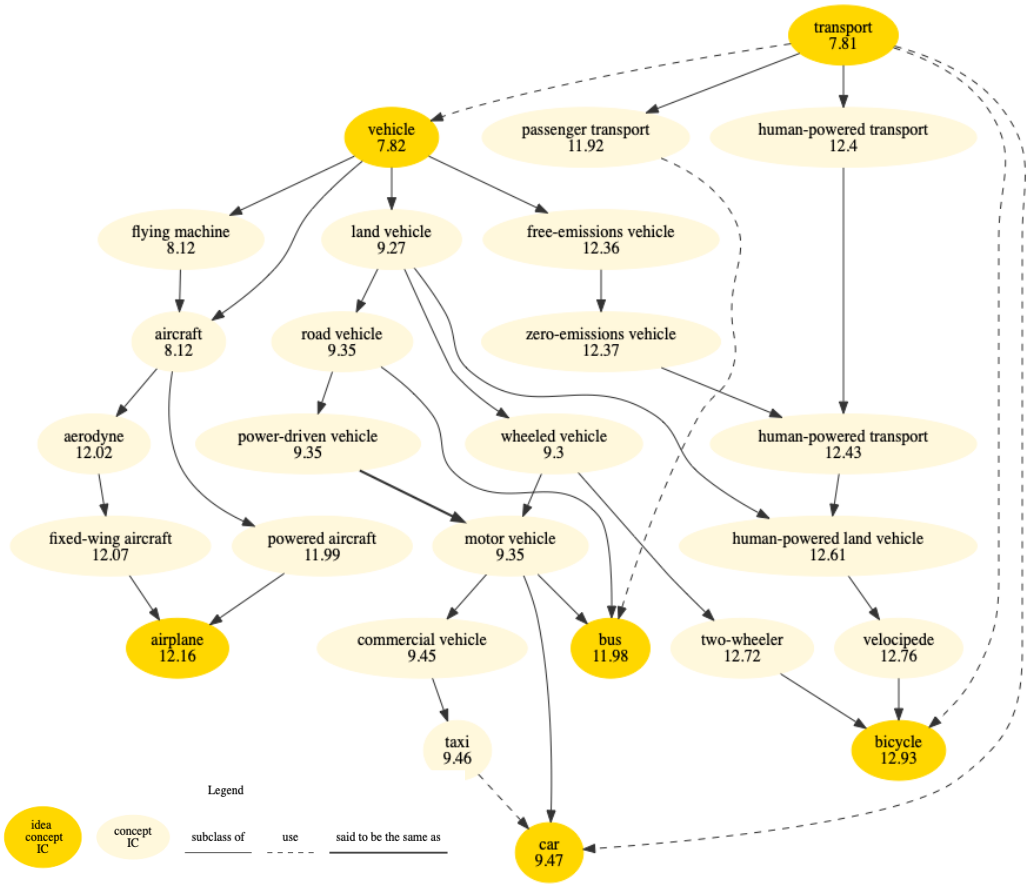


Figure 3.2: Sub-graph of the local knowledge graph with the concept *transport* as root-concept, and the calculations of the IC.

from the common subsumers to the concepts. And the other maximizes among the ICs of their common subsumers and calculates the path length over their LCS. The *distance* methods are defined as

$$distance_{ncs}(c_i, c_j) = \min_{ncs \in CS_{c_i,c_j}} dist(c_i, ncs) + dist(ncs, c_j) \tag{3.1}$$

$$distance_{lcs}(c_i, c_j) = dist(c_i, lcs) + dist(lcs, c_j), \tag{3.2}$$

where CS are the common subsumers and lcs is the unique LCS of c_i, c_j according to Definition 2.2.3. In a good structure taxonomy, the above functions should yield the same result for a given concept pair, because their common subsumer with maximal IC should also be the one with the shortest connecting path. For tree structures, this is always true, whereas, in a DAG, it is not [Dash et al., 2013]. Wikidata, for example, contains several meta-classes that connect highly different concepts for a relatively short path length, which are mostly not the common subsumers with maximal IC. In Figure 3.2, only the concepts related to *transport* are displayed. To illustrate the impact of the

two distance methods, measuring the distance between *car* and *bicycle* over its NCS, namely *transport* results in a distance of 2, whereas the distance over its LCS, *wheeledvehicle*, is at 4 with an IC of 9.3. The effects of the two variants will be discussed in the following Experiments in Section 3.2.

3.1.2 Graph Based Information Content

The calculation of the graph-based Information Content is not as straight forward for the Wikidata KG as proposed in Definition 2.2.2. It becomes difficult to count all *entities* that are descendants of a concept, because of the graph's size and its complex structure. The WordNet taxonomy and the DBpedia-Ontology are smaller. Thus, it is feasible to count all descendent concepts as proposed. Besides, with cycles in the graph, the IC is less meaningful because all concepts that are part of the same cycle have the same IC since they are descendants from one another. A topological ordering is then not possible. To counteract this problem, the graph is reduced to a DAG, as described above. Thereby the IC becomes locally measurable for concepts in the DAG knowledge graph representation. Meaningful information of the Wikidata KG, which is not included in the local DAG, will be ignored. Consequently, all the relations to concepts that are not part of the idea concepts and their ancestors are not considered. In particular, the number of descendants for any idea concept that is not ancestral to any other idea concept in the dataset equals one, e.g., in Figure 3.2, this would apply i.a. for the concepts *bike, bus* and *car*. Hence, they have the same IC. To fill the IC of concepts with more meaning and benefit from the richness of Wikidata, a *SPARQL*-query was implemented, as displayed in Figure 3.3, that counts the number of direct descendants of taxonomic relations for a concept in Wikidata. The IC is then calculated by summarizing these values for all descendent concepts as its number of descendent *entities*.

Let c be a concept in the DAG, $count(c)$ denotes the value for the *SPARQL*-query in 3.3, then

$$\|entities(c_i)\| = count(c_i) + \sum_{c_d \in descendants(c_i)} count(c_d)$$

The IC of c is defined as

$$IC(c) = -log_e \frac{\|entities(c)\|}{N}, \tag{3.3}$$

where N the sum over all $count(c)$ of concepts c in the DAG,

$$N = \sum_{c_i \in DAG} count(c_i).$$

The measure of $\|entities(c)\|$ is not an exact measure of the number of descendant concepts in the DAG and their direct descendants in Wikidata but

rather an approximation, because any concept c_j that is part of the DAG and also appears in the $count(c_i)$ of another concept c_i is counted multiple times. This vagueness can be ignored because of the sparseness of the DAG compare to the Wikidata KG. For the gold-standard idea dataset, the DAG contains 2417 concepts, and the total number N considers over 23 million concepts in the Wikidata KG. This number shows that the concepts of the DAG are just a fraction of the considered concepts with the *count*-query of Figure 3.3.

```
SELECT (COUNT(*) AS ?count)
WHERE   {
    ?item ( p:P279 | p:P31 | p:P171 ) wd:concept.
}
```

Figure 3.3: *SPARQL*-Query that counts the number of items that are *instance of*, *subclass of* or *parent taxon* of a concept in the Wikidata KG. The responds contains the queried number for its *count* keyword.

Figure 3.2 displays the sub-graph of the DAG with the calculated IC-values showing how the IC develops over concepts in the DAG. Besides, the IC's strong dependence on the bottom-layered concepts can be observed, because most concepts that are higher in the graph have few direct descendants. For example, the concept *motor vehicle*, which subsumes *car* and *bus* in the DAG, has an IC value that is just slightly less than the one of *car*, since Wikidata contains especially many instances of car.

3.1.3 Concept Similarity

With the collected information, the concepts similarity methods described in section 2.2 can be implemented. This is done in a class called *ConceptSimilarity* as shown in the UML-diagram 3.1. For each pair of idea concepts, their distance and LCSs is stored in a matrix. For the *path* similarity method all shortest paths between idea concepts with the distance function $distance_{lcs}$ 3.1 or $distance_{ncs}$ 3.2 are computed following its definition 2.1. The method *wpath* weights the similarity of *path* with the IC, which is approximated as described above. The methods res, lin, jcn only use the IC of the two concepts and their NCS. All methods can also use the distance function 3.2 and the selection of the LCS function for maximal IC. The variants are denoted with a keyword flag in the following Experiments, e.g., $path(lcs)$.

3.1.4 Word Mover's Similarity

Computing the similarities between idea sparks, the *PuLP-python* module [Mitchell, 2020] is used to solve the linear maximization problem of the WMS, given the similarities between concepts. For every idea spark a bag of concepts

(BOC) and its distribution d is computed to add all the constraints of the equation 2.3 and solve the linear problem. For the computation of similarities between concepts and idea sparks, the number of concepts representing the spark is limited to the five most similar concepts. Splitting the flow of a single concept for too many concepts might make the measure meaningless because the average similarity of one concept to many others could be rather comparable. Using only the most similar concept advances ideas containing more concepts and makes the similarity less differential. For this thesis a limit of the five most similar concepts is used. However, no quantitative evaluation is provided here.

3.2 Experiments

The following two experiments are the evaluation of the presented similarity methods and their implementation. The goal is to assess the validity of the similarity measures used for the recommendation feature. Therefore experiments on the performance of the presented methods are conducted. The performance is measured in correlation to the human perception of similarity. This criterion is meaningful because the recommendations ought to match the user's expectations. Furthermore, it should be more intuitive for the user to understand the feature when there is a common perception of similarity and create a mental model of an intelligent system, as stated in Section 2.4. Besides, the experiments evaluate the usefulness of the Wikidata KG for a semantic similarity approach. With the results, the fit of the semantic similarity measures for the recommendation feature in the Orchard application can be assessed. The first experiments measure word similarity, using standard datasets of semantic similarity assessments. The second focuses on sentence similarity, evaluated with one dataset of short sentences and three datasets of idea similarity, for which the Idea2Market Project conducted similarity assessments [Hadler et al., 2019].

3.2.1 Word Similarity Task

The word similarity task is evaluated with the Spearman's rank correlation, which is commonly used in related work and is a standard method to evaluate semantic similarity measures [Zhu and Iglesias, 2017]. Thus, we can compare our results to a strong baseline of knowledge and corpus-based methods.

Datasets

To evaluate the perceived concept similarity and presented methods, the following datasets are used.

- R&G [Rubenstein and Goodenough, 1965] is a widely used dataset of human word similarity assessments, containing 65 pairs of words. For

these pairs of common English nouns, the participants were asked to judge their *similarity in meaning* on a scale from 0 (not similar) to 4 (synonymous). Other relations of the word pairs are ignored.

- M&C [Miller and Charles, 1991] replicated the study of R&G with a subset of 30 word-pairs.

- WS353 [Finkelstein et al., 2001] consists of 353 word-pairs and as well considers other relations then the similarity in meaning and therefore rather measures the relatedness of words.

- WS353-Sim [Agirre et al., 2009] contains 203 word-pairs and is a selection of the WS353 that focuses on semantic similarity excluding pairs where the similarity is due to high relatedness.

- SimLex [Hill et al., 2015] consists of 999 pairs of words to evaluate semantic similarity. The dataset contains 111 adjective-pairs, 222 verb-pairs, and 666 noun-pairs for which the experiments only use the 666 noun-pairs.

Implementation of Word Similarity Method

For the word similarity task, the words of the datasets are mapped to a maximal number of ten possible concepts of the Wikidata KG to consider the ambiguity of a word. Figure 3.4 shows the $SPARQL$-query with a limit of 10 concepts with the word *bank* occurring in its *label* or *alternative label* that is recorded in Wikidata. The statement in line 13 of Figure 3.4 ensures that the queried concept is connected to the graph with at least one of three predicates. In a second step, the results are further filtered, preferring exact matches and words beginning with a lowercase character, because the label of nouns in Wikidata usually begin with lowercase, whereas names start with an uppercase.

When humans consider the similarity of two words in the similarity assessments, they consider the two corresponding concepts that are most similar to each other [Zhu and Iglesias, 2017]. Analogically for two words, the most promising concepts are queried and filtered from Wikidata. Then the similarity of two words is the maximal similarity between their corresponding concepts. For a word w, let $c(w)$ denote a set of concepts that are senses of the word w, then the similarity measure between words can be defined as

$$sim_{word}(w_i, w_j) = \max_{c_i \in c(w_i), c_j \in c(w_j)} sim_{concept}(c_i, c_j) \tag{3.4}$$

,

where $sim_{concept}$ is any of the presented concept similarity methods [Zhu and Iglesias, 2017].

```
1  SELECT DISTINCT ?item ?itemLabel ?itemAltLabel WHERE {
2      SERVICE wikibase:mwapi {
3          bd:serviceParam wikibase:api "EntitySearch" .
4          bd:serviceParam wikibase:endpoint "www.wikidata.org" .
5          bd:serviceParam mwapi:search "bank" .
6          bd:serviceParam mwapi:language "en" .
7          ?item wikibase:apiOutputItem mwapi:item .
8          ?num wikibase:apiOrdinal true .
9      }
10     SERVICE wikibase:label {
11         bd:serviceParam wikibase:language "en".
12     }
13     ?item ( p:P279 | p:P171 | p:P31 ) ?superItem.
14 } ORDER BY ASC(?num) LIMIT 10
```

Figure 3.4: A Wikidata full text search query for a concept including the word *bank*. The predicates in line 13 are *subclass of, instance of, part of.*

Evaluation

Evaluating the presented concept similarity methods of chapter 2 with the described datasets in Section 3.2.1 follows an established methodology for semantic similarity measures. This consists of measuring Spearman's correlation between the similarity scores for word-pairs, calculated with the $word_{sim}$ method 3.4 combined with a concept similarity method and the means of the human similarity assessments [Zhu and Iglesias, 2017]. The Spearman's correlation score $\rho \in [-1, 1]$ indicates the performance of the compared measures. Scores closer to 1 show a higher correlation with human judgments and score of 0 indicates that the measures are completely unrelated to human ratings. The performance of the IC is evaluated based on the performance of methods that are using the IC. For the *wpath* method, the parameter k weights the additional impact of the IC. Smaller $k \in [0, 1]$ translates into stronger increases of similarity by the IC-value compared to the *path* baseline. For both tables, 5.1 and 3.2, the bold values in each column denote the highest correlation score for each dataset. The values after the doubled line are the best correlation scores found in related work of the knowledge obtained for the different datasets. In Table 3.1 the best results of Zhu and Iglesias [2017] of *wpath* where they used corpus-based IC are considered as a baseline.

Table 3.1 displays the results for the *wpath* method for different values of k from Zhu and Iglesias [2017], which was outperforming other baseline methods in their publication including those presented in this thesis. The results in Table 3.1 show that the impact of the IC improves the correlation score, except for the M&C dataset. However, when the IC becomes more dominant, the correlation lowers. It shows that the IC measures often yield extreme values,

Table 3.1: Spearman's Correlation of word similarity with *wpath*(*ncs*)-method for different values of k.

wpath(ncs) k	R&G (65)	M&C (30)	WS353 (348)	WS353-Sim (203)	SimLex (666)
k=0.1	0.670	0.768	0.322	0.524	0.325
k=0.2	0.721	0.827	0.334	0.553	0.339
k=0.3	0.730	0.841	0.356	0.593	0.356
k=0.4	0.743	0.830	0.359	0.611	0.358
k=0.5	0.781	0.844	0.358	0.617	0.370
k=0.6	0.796	0.862	0.357	0.627	0.382
k=0.7	0.799	0.859	0.355	0.636	0.393
k=0.8	0.804	0.876	0.359	0.647	0.405
k=0.9	**0.811**	0.905	**0.370**	**0.663**	**0.425**
k=1.0	0.805	**0.908**	0.361	0.641	0.419
Baseline	0.795	0.740	0.349	0.652	0.603

which strongly increases the similarity score.

Overall in all tested datasets, *wpath* performs best for $k = 0.9$. Compared to the implementation of [Zhu and Iglesias, 2017] using WordNet[2] and a corpus-based IC, the optimal k is more obvious, as their value of k yielded highest scores between 0.4 and 0.9. On average their implementation of *wpath* weighted best with $k = 0.8$. The implementation in this thesis has its average optima for 0.9, which matches the assumption that the graph-based IC of this work leads to more extreme values because of the higher number of considered concepts as stated in 3.1. Interestingly, the implementation of *wpath* outperforms the baseline by only a small margin except for the SimLex dataset, where a lower correlation was observed.

In Table 3.2 the other similarity methods introduced in Section 2.2 are compared to the optimal *wpath* with $k = 0.9$, as well as the two implementations of path *distance* using either the NCS or LCS. The IC-based methods res, lin, and jcn in general score a bit lower than the *path* method, nevertheless they yield comparable results. Using a ratio of the IC between the concepts and the NCS slightly improves the result, which is indicated by higher scores of lin compared to res in all datasets.

The alternative $distance_{lcs}$ method to compute the path-length, using the LCS indicated by the *lcs*-argument, additionally shows interesting results. The *path*(*lcs*) method improves its score compared to *path*(*ncs*) for $WS353$ and $SimLex$ but not for the other datasets. Besides, the *wpath*(*lcs*) does not benefit from the weighted IC, except for *WS353-Sim*, otherwise, the correlation scores drop with the increasing impact of the IC. The assumption using

[2] `https://wordnet.princeton.edu`

Table 3.2: Spearman's Correlation of word similarity for different methods

Method	R&G (65)	M&C (30)	WS353 (353)	WS353-Sim (201)	SimLex (666)
path(ncs)	0.805	**0.908**	0.361	0.641	0.419
wpath(ncs,k=0.9)	**0.811**	0.905	**0.370**	**0.663**	0.425
res(ncs)	0.739	0.823	0.364	0.525	0.293
lin(ncs)	0.790	0.871	0.368	0.586	0.358
jcn(ncs)	0.803	0.873	0.280	0.540	0.367
path(lcs)	0.794	0.901	0.362	0.605	**0.461**
wpath(lcs,k=0.9)	0.781	0.876	0.362	0.648	0.425
res(lcs)	0.591	0.702	0.283	0.481	0.184
Baseline					
Knowledge-based	0.920	0.910	0.415	0.652	0.603
Corpus-based	0.833	0.853	0.810	–	–
Hybrid	0.910	0.920	0.828	–	0.760

the computation of the LCS and its path-length between concepts was that abstract concepts such as the *Wikidata metaclass*[3] connects many highly different concepts for a relative short path-length. Such concepts have a low IC value in the graph-based implementation of this work. Therefore, the *wpath* would barely increase the similarity score. On the other hand, for example, the word-pair *bird, crane* (see Table 5.1 in the Appendix) has the LCS *taxon* which is a rather abstract concept. Using the LCS concept that maximizes the IC of common subsumers yields *bird*, see Table 5.2, and therefore yields a higher similarity for $wpath(lcs)$ with $k = 0.9$. Hence, dissimilar concept might have a longer path-length using $distance_{lcs}$ method, but a greater IC. The improvement for the *path* measure using the LCS shows for some datasets such as $WS353$ and $SimLex$, but for others, the correlation score drops, e.g., $M\&C$. An explanation would be that relatively similar concepts also have common subsumers with large IC for a greater path-length. Besides, using the IC of the subsumer becomes less accurate, because of the higher IC of the LCS for dissimilar concepts, as found in the results, see Appendix Table 5.2. This observation reflects the $res(lcs)$ method in Table 3.2, which scores are strongly below any other method across datasets.

Comparing the results to the collected baseline[4] shows that the correlation scores for the $R\&G$ dataset and its subset $M\&C$ are competitive. The $WS353$ dataset, containing word pairs that consider relatedness, generally shows a lower correlation score for knowledge-based methods in related work. Hybrid and corpus-based methods seem to perform better in measuring relatedness, as

[3] `https://www.wikidata.org/wiki/Q19361238`

[4] `https://aclweb.org/aclwiki/Similarity_(State_of_the_art)`

the collected baseline in Table 3.3 indicates, which explains the big difference to knowledge-based methods for $WS353$. The performance of the implementations in this work for the $SimLex$ dataset is well below the baseline.

3.2.2 Sentence Similarity Task

The sentence similarity measures of the WMS described in Section 2.3, are based on the concepts similarity measures of $wpath(ncs, k = 0.9)$. The same method is used to measure idea similarity. The correlation of our measures for sentence similarity is evaluated using Pearson's correlation score, which measures linear correlation and is, just as Spearman's correlation, commonly used in related literature [Traverso et al., 2016]. Pearson's correlation is used in this experiment because of the comparison to the benchmark idea similarity of Hadler et al. [2019], who also chose this method.

Datasets Idea and Sentence Similarity

For the Evaluation of the Idea Similarity measures the following three Challenges of idea generation are used: Improve the Environment [Nijstad and Stroebe, 2006], Mturk Mobile Features [Girotto et al., 2017], Fabric Display [Siangliulue et al., 2015]. Members of Ideas to Market project performed an idea similarity study with a selection of 253 idea pairs for each dataset. It then compared different algorithms to compute idea similarity with the previously created gold standard [Hadler et al., 2019]. Additionally, the MSR-Video dataset (Microsoft Research Video Description Corpus)[5] with 750 similarities of sentence pairs is used to evaluate the approach.

The four mentioned datasets contain no concept-annotations. The concept-annotation of the idea and sentence similarity datasets are generated with the babelfy API [6]. Babelfy is the state-of-the-art approach to solve entity linking and word sense disambiguation [Navigli and Ponzetto, 2012]. However, not all linked entities from the Babelfy-result hold mappings to Wikidata items. Accordingly, for the linked entities, only 42% remain as Wikidata items and only those concepts are used in the similarity measures. As the measures of idea similarity include uncertainties, judgment about the performance is limited to the described circumstances above.

Evaluation

For the *MSR-Video* dataset, which contains short sentences, an average amount of 2.1 concepts per sentence were linked, which is about 49% of concepts that were initially linked. However, Paul et al. [2016] mention that when there is at least one concept-annotation, a meaningful similarity can be measured. They claimed to capture only between 1 to 3 concept-annotations per sentence in

[5] `https://www.cs.york.ac.uk/semeval-2012/task6/index.html`
[6] `http://babelfy.org/`

their experiment, which reached a Pearson's correlation of 0.673 [Paul et al., 2016]. The WMS similarity measures show a reasonable correlation score for the *MSR-Video*[7] dataset, see Table 3.3. Still, it is below the knowledge-based benchmark of 0.71 from Traverso et al. [2016]. The idea similarity datasets consist of longer sentences. Beginning with the *SmartTextile* dataset, which contains an average of 4 concepts-annotations per idea, the WMS outperforms the baseline of stochastic and corpus-based methods of Hadler et al. [2019]. Correlation scores drop for the *MTruk* and *Environment* dataset. Both contain on average about 3.5 concepts-annotations. In general, the baseline correlation score of all three idea similarity datasets is rather low, which indicates a low accuracy of the assessed idea similarity. Feasible explanations might be the small number of five human assessments per idea pair, the missing research of idea similarity [Hadler et al., 2019], and its higher complexity compared to sentence similarity. However, the correlation of the approach of this work shows that it is competitive to the baseline and provides consistent results for a real-world application such as the recommendation feature.

Table 3.3: Pearson's Correlation of the Word Mover's Similarity method based on concepts similarities measured of wpath(ncs,k=0.9)

	MSR-Video (750)	SmartTextile (253)	MTurk (253)	Environment (253)
WMS	0.5849	0.4689	0.1571	0.1573
Benchmark	0.7100	0.4399	0.2675	0.4442

3.3 Discussion

In this chapter, the implementation of semantic similarity methods is described and compared to a strong baseline of related work. The performance of the word similarity task has shown that the approach of concept similarity, using Wikidata, is competitive to state-of-the-art knowledge-based methods. It can be concluded that Wikidata is, despite its complexity and crowd-source origin, a valuable source for knowledge-based methods and the computation of Information Content. Outperforming the baseline of [Zhu and Iglesias, 2017] in four out of five datasets shows that Wikidata is competitive to taxonomic dictionaries such as WordNet and smaller ontologies for common English nouns. It was shown that a graph-based approximation of the IC method proposed by Zhu and Iglesias [2017] is feasible with Wikidata as the good performance of the $res(ncs)$ method in Table 3.2 indicates.

[7] `https://www.cs.york.ac.uk/semeval-2012/task6/data/uploads/datasets/test-gold.tgz`

However, for the largest dataset *SimLex*, the correlation score is below the baseline, which indicates the approach's limitation and a need for further research of the Wikidata KG. The limits of the IC's computation of this work becomes visible when maximizing the IC for common subsumers, denoted by the *lcs*-argument in Table 3.2. This is why parts of the Wikidata KG are ignored depending on the input of bottom layered idea concepts to construct the graph and the set of considered predicates. Besides these drawbacks, Wikidata contains biases for different domains that are not equally represented depending on the interest of its users and curators. For example, Wikidata contains more than 6.5 million instances of *human* (Wikidata item: $Q5$). Therefore, the IC of the concept *human* is rather low. When applying the Wikidata KG in any context, these biases should be considered.

Another observation for Table 3.2 is that the correlation scores for most methods stay in a similar range, which indicates that the computation of the IC and the shortest path length measure implicitly similar characteristics of the graph. Most relevant for both measures is the hierarchical structure and partial order. Hence, the crucial part of the similarity measures is the generation of the DAG. Therefore, further improvements should concentrate on a more precise analysis of the Wikidata KG, exploring other predicates, and removing distracting concepts.

In Section 3.2.2, the implementation of the WMS are verified based on concept similarities by showing reasonable correlation scores for the sentence and idea similarity tasks, despite the described limitation through the missing validation on concepts-annotation. For a more meaningful assessment on the WMS, verified concept-annotations of the ideas with a higher annotation rate would be necessary.

4 Recommendation Feature

This chapter consists of three Sections. Section 4.1 briefly introduces the conducted implementation to integrate the recommendation feature into Orchard with its functionalities described in Section 2.4. The following Section 4.2 describes the conducted user test and summarizes its crucial observations. In the discussion 4.3, the results regarding the described goals of Section 1.5 and the human-centered concepts of Section 2.4 are debated.

4.1 Implementation

The Orchard clustering application[1] uses *React* to render the HTML and *redux* to store the state of the whiteboard, and the position of clusters and idea sparks. For a persistent state across browser restarts, it uses the browsers local storage. The gold standard idea sparks are queried from a SPARQL-server.

The calculated similarities for the gold standard dataset are added to the client files. The client accesses the similarities directly once the app loads without further server interaction during the clustering. The three similarity matrices are from concept to concept $C2C \in \mathbb{R}^{n \times n}$, from concepts to spark $C2S \in \mathbb{R}^{n \times m}$, and from spark to spark $S2S \in \mathbb{R}^{m \times m}$, where n is the number of unique concepts and m the number of idea sparks contained in the dataset.

In Figure 4.1, the Graphical User Interface in a state with two clusters is displayed, each containing two sparks, where one cluster is named *Glasses*. The Recommender displays the nine top-ranked out of the sparks left on the *Spark Stack* based on similarity measures to a single concept or idea spark. The recommendations are limited to nine because it is assumed to be unlikely to find good recommendations below the ninth rank when there was nothing useful before. As a result, the user is encouraged to use the *Spark Stack* to look for sparks concerning different topics where the user obtains all idea sparks by scrolling through them.

The Recommender in Figure 4.1 provides sparks to the user which are the most similar to their selected spark, namely SPARK 8, highlighted in bold. In this case, the matrix $S2S$ contains the similarities. Alternatively, the user can select a concept highlighted by a stronger background saturation color. Then the recommendation is based on the similarities in matrix $C2S$. The different saturation depends on the current selection of the user. The color spectrum to highlight concepts ranges from the pale goldenrod of the spark's squares to a strong orange. This range indicates the similarity of a concept to the current selection. Colors near the pale goldenrod of the spark visualize dissimilarity,

[1] `github.com/FUB-HCC/Innovonto-Orchard`

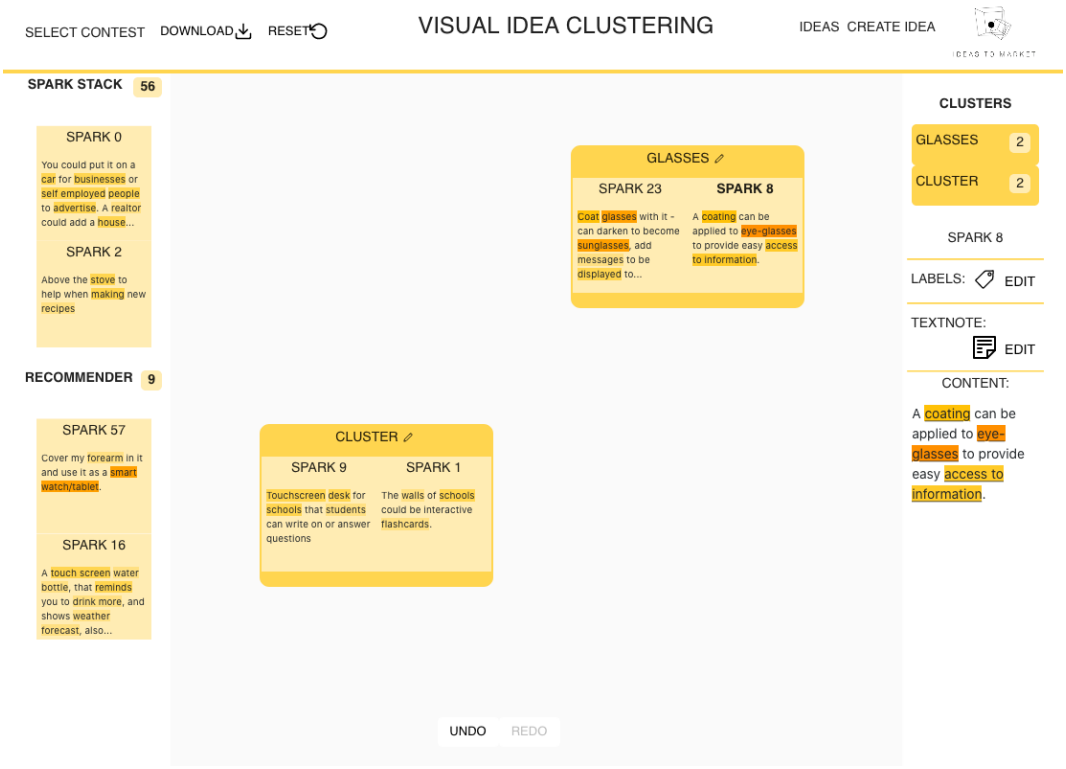


Figure 4.1: Orchard with selected Spark 8

and colors closer to a strong orange indicate high similarity. For example, the identity is displayed by the maximum of the color range, see concept *glasses* in Figure 4.2. That the concept is the current selection, is again visualized by its bold characters. When the selection is a concept, the color saturation is based on the similarity contained in $C2C$. For example, in Figure 4.2 the color saturation visualizes the similarity data that yields high similarity of *eye-glasses* and *sunglasses* to *glasses*. When the current selection is a spark, the $C2S$ matrix provides the similarities for the concept's color saturation.

4.2 Evaluation

Five user tests with experts were conducted showing qualitative evaluation of the recommendation feature. The reason to consider five participants is grounded in the findings of [Nielsen and Landauer, 1993] who argue that choosing an amount of five optimizes the cost/benefit ratios in usability evaluations. The used format is a Think-Aloud interview, where the expert clusters and creates ideas while speaking about their occurring thoughts and actions. The exact implementation of the user test is described in Appendix 5. For the questions of the evaluation concerning the performance of the recommendation feature, see Appendix 5. The following claims support the assumption that the recommendation feature improves the user's synthesis of idea sparks.

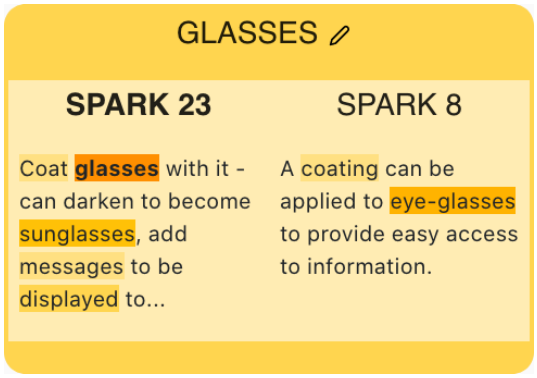


Figure 4.2: Cluster named Glasses with current selection of the concept *glasses*.

However, support for the claim would need a baseline user-study which is beyond the scope of this work. Hence, proving the following three claims does not conclude an improvement in the user's idea synthesis.

Claim 1. The recommendation feature enables a targeted iteration through the idea sparks, following the current interest.

Claim 2. The recommendations are matching the user's expectations for they interactive requests.

Claim 3. The visualization of similarities between concepts and spark helps the user to understand and create a mental model of the recommendation feature.

Furthermore, the experts evaluate the human-centered interface design choices. Examples include discussing the color saturation of concepts, as shown in Figure 4.2, and if it suffices their purpose, as described in Section 2.4.

4.2.1 Participants

Five participants, that all have a background in computer science, took part in the user study. Three of them were familiar with the clustering process beforehand. The others engaged in a short introduction about the purpose of clustering in the context of creativity. The five participants will be referenced as P1-P5.

4.2.2 Procedure

The participants familiarized themselves with the Orchard clustering application and its functionalities beforehand. Since the 60 idea sparks of the user-test concern the application case of the TCO-technology, the users read the information sheet about TCO, see Appendix 5.1. TCO stands for transparent conductive oxide and is a thin coating that makes materials and surfaces

touch-sensitive. Next, they started clustering with the instruction to synthesize three ideas from the idea sparks. The time clustering was limited to 30 minutes, followed by 10 minutes of synthesizing three ideas and 10 to discuss the four evaluation questions about the recommendation feature, see Appendix 5.1.

4.2.3 Findings

The participants used the first phase of clustering for exploring 3 to 10 idea sparks from the *Spark Stack*. P1 described that as "getting familiar with the ideas" until they were able to order and categorize them. Then, participants started using the recommendation feature but in different ways. Three participants were selecting ideas and concepts equally to perceive recommendations while firstly tending to select a spark and then later, in an advanced state, more frequently selecting singular concepts. P1 explained this observation with their aim to specify the selection with increasing knowledge of the idea sparks. Most participants used the Recommender to find more sparks concerning a specific subject. P2 described their aim towards the Recommender as "filling up existing clusters" and exploring similar sparks with the Recommender by repeatedly selecting sparks and concepts of one cluster. This aim reflects the observation that most participants (P1, P2, P4) stuck to one cluster and subject for a while until they find an idea spark of their interest that fits a different cluster, or they randomly start looking for something else. These observations support the claim 1 that the recommendation feature enables the user in a targeted iteration through the idea sparks lead by their current interest. Besides, it shows the possible increase of efficiency in the aim to rapidly reach a state of a sorted idea-space in the form of a complete clustering.

For the five user tests, two fundamentally different approaches towards the tool were observed. The approach of P1, P2, and P4 saw a strong focus on clustering and appending sparks to categories. This resulted in many clusters with different amounts of sparks. The approach of P3 and P5 showed a more critical concern for the quality of the sparks. When a spark was perceived as not helpful, it was put aside. Because more than half of the clustered sparks were viewed as not helpful, only a few clusters with one or two sparks emerged. At this point, it is worth mentioning that indeed many sparks missed the initial application of the technology and can be perceived as unfeasible. However, both strategies could benefit from utilizing the Recommender.

P5 exclusively selected sparks as target to receive recommendations due to a rather unspecific iteration through the sparks. Sparks were selected unconsciously. Most often, this meant that the last spark added to the whiteboard was selected. As a result, the sparks in the Recommender were similar to the last spark, and a similar iteration evolved going into one subject and then jumping the next one. Hence, it was easier placing the spark on the whiteboard when it was related to the same cluster that had been used previously.

Participants reported different observations about the order of recommended

sparks and the general quality of the displayed recommendations for their requests. P1 and P2 reported that they were mostly confident finding a spark that matches their expectations in the top three recommendations. However, P2 interjected that for recommendations based on a single concept, sparks below the third rank were frequently perceived as the most related. Hence P2 partially pulled sparks from the lower position to the whiteboard, which made it necessary to scroll through the recommendations.

P3 and P4 reported that they would have preferred an illustration of the similarity of recommendations arguing that when there are only dissimilar sparks to the target, extra time was consumed by identifying that there are probably no similar sparks left. In general, the user-tests provided support for the claim 2 that the recommendations were often reasonable to the user's expectations.

The visualization of similarities with the color saturation yields different approaches of using that information. Four participants mainly used the strong orange colors, which indicate a high similarity, to scan the keywords and sparks that might be similar rapidly. P3 reported some satisfaction for the recommended sparks when the highlighted concepts were close to orange on the color spectrum. Thus, it indicated a match of a similar spark to the target. However, the color highlighting was considered "a bit distracting" since a new selection for recommendation resulted in color changes for all words (P1).

4.3 Discussion

This chapter has illustrated the implementation of this work and its findings. Different strategies were able to benefit from the recommendation feature and the observations of the user tests provide support for the three claims.

The user tests point out that there is a demand to find specific sparks on the whiteboard, which was often not feasible with the similarity visualization of concepts. With increasing numbers of idea sparks, this might become even more important. One possible solution would be to add an option to the Recommender so that the user selects if they want to include sparks in the Recommender, that are already placed on the whiteboard. Additionally, the option of full-text-search to find sparks for any input was requested.

5 Conclusion and Outlook

The semantic similarity approach has shown its value for the recommendation feature. The recommendations were mostly perceived as accurate by the participants, and the experiments on concept similarity have shown a high correlation to human perceptions.

Follow-up research on clustering support with recommendations could compare different similarity measures, including statistical metrics, on their impact on the user's clustering strategy. However, the applications of the Wikidata KG in this context is not exhausted. Similarity measures could be further optimized with an analysis of the graph's structure and extensive refactoring, considering additional relations. Generating a directed acyclic graph from over 80 million Wikidata items could improve the measures of Information Content.

The evaluation has confirmed that the recommendation feature supports the user in a targeted iteration through the idea sparks. This aspect seems to gain relevance with an increasing number of idea sparks since it becomes more difficult for the user to keep in mind all sparks on the whiteboard. Considering that, the efficiency might be further improved.

The visualization of concept similarity to aid the user in understanding the semantic similarity measures and more easily detect similar idea sparks has only been partially achieved. The user study points out that some participants used the visualization to detect and scan related idea sparks. It also played a role in understanding the result of recommendations. However, when selecting an idea sparks, the visualizations were often not intuitive and described as distracting. Furthermore, to properly understand the impact of the recommendation feature on the clustering process and the idea synthesis, a baseline user study without the feature is necessary, to compare two conditions: with and without the recommendation feature.

Bibliography

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`.

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics on - NAACL '09*, page 19, Boulder, Colorado, 2009. Association for Computational Linguistics. ISBN 978-1-932432-41-1. doi: 10.3115/1620754.1620758. URL `http://portal.acm.org/citation.cfm?doid=1620754.1620758`.

Thomas H. Cormen, editor. *Introduction to algorithms*. MIT Press, Cambridge, Mass, 3rd ed edition, 2009. ISBN 978-0-262-03384-8 978-0-262-53305-8. OCLC: ocn311310321.

Santanu Dash, Sven-Bodo Scholz, Stephan Herhut, and B. Christianson. A scalable approach to computing representative lowest common ancestor in directed acyclic graphs. *Theoretical Computer Science*, November 2013. ISSN 0304-3975. doi: http://dx.doi.org/10.1016/j.tcs.2013.09.030. URL `http://uhra.herts.ac.uk/handle/2299/12152`. Accepted: 2013-11-21T12:22:17Z.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. volume 20, pages 406–414, January 2001. doi: 10.1145/503104.503110.

Marco Gillies, Bongshin Lee, Nicolas d'Alessandro, Joëlle Tilmanne, Todd Kulesza, Baptiste Caramiaux, Rebecca Fiebrink, Atau Tanaka, Jérémie Garcia, Frédéric Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, and Saleema Amershi. Human-Centred Machine Learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '16*, pages 3558–3565, San Jose, California, USA, 2016. ACM Press. ISBN 978-1-4503-4082-3. doi: 10.1145/2851581.2856492. URL `http://dl.acm.org/citation.cfm?doid=2851581.2856492`.

Victor Girotto, Erin Walker, and Winslow Burleson. The effect of peripheral micro-tasks on crowd ideation. In *CHI 2017 - Proceedings of the 2017 ACM SIGCHI Conference on Human Factors in Computing Systems: Explore, Innovate, Inspire*, pages 1843–1854. Association for Computing Machinery, May 2017. doi: 10.1145/3025453.3025464. URL https://nyuscholars.nyu.edu/en/publications/the-effect-of-peripheral-micro-tasks-on-crowd-ideation.

Thomas Hadler, Maximilian Mackeprang, and Claudia Müller-Birn. Benchmarking Sentence Similarity Algorithms in Collaborative Ideation. 2019.

Aric Hagberg. networkx: Python package for creating and manipulating graphs and networks, 2008. URL http://networkx.github.io/.

Felix Hill, Roi Reichart, and Anna Korhonen. SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. *Computational Linguistics*, 41(4):665–695, September 2015. ISSN 0891-2017. doi: 10.1162/COLI_a_00237. URL https://doi.org/10.1162/COLI_a_00237. Publisher: MIT Press.

Jess Holbrook. Human-Centered Machine Learning, June 2018. URL https://medium.com/google-design/human-centered-machine-learning-a770d10562cd.

Jay J. Jiang and David W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *arXiv:cmp-lg/9709008*, September 1997. URL http://arxiv.org/abs/cmp-lg/9709008. arXiv: cmp-lg/9709008.

Miroslaw Kowaluk and Andrzej Lingas. LCA Queries in Directed Acyclic Graphs. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *Automata, Languages and Programming*, volume 3580, pages 241–248. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-27580-0 978-3-540-31691-6. doi: 10.1007/11523468_20. URL http://link.springer.com/10.1007/11523468_20.

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

Dekang Lin. An Information-Theoretic Definition of Similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, July 1998. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-556-5.

Maximilian Mackeprang, Claudia Müller-Birn, and Maximilian Timo Stauss. Discovering the Sweet Spot of Human-Computer Configurations: A Case Study in Information Extraction. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):195:1–195:30, November 2019. doi: 10.1145/3359297. URL https://doi.org/10.1145/3359297.

George A. Miller and Walter G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, January 1991. ISSN 0169-0965. doi: 10.1080/01690969108406936. URL https://doi.org/10.1080/01690969108406936. Publisher: Routledge _eprint: https://doi.org/10.1080/01690969108406936.

J. S. Roy and S. A. Mitchell. PuLP: PuLP is an LP modeler written in python. PuLP can generate MPS or LP files and call GLPK, COIN CLP/CBC, CPLEX, and GUROBI to solve linear problems., 2020. URL https://github.com/coin-or/pulp.

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.

Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, December 2012. ISSN 00043702. doi: 10.1016/j.artint.2012.07.001. URL https://linkinghub.elsevier.com/retrieve/pii/S0004370212000793.

Tamas Nepusz. python-igraph: High performance graph data structures and algorithms, 2006. URL https://igraph.org/python.

Jakob Nielsen and Thomas K. Landauer. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 206–213, New York, NY, USA, May 1993. Association for Computing Machinery. ISBN 978-0-89791-575-5. doi: 10.1145/169059.169166. URL https://doi.org/10.1145/169059.169166.

Bernard Nijstad and Wolfgang Stroebe. How the Group Affects the Mind: A Cognitive Model of Idea Generation in Groups. *Personality and social psychology review : an official journal of the Society for Personality and Social Psychology, Inc*, 10:186–213, February 2006. doi: 10.1207/s15327957pspr1003_1.

Christian Paul, Achim Rettinger, Aditya Mogadala, Craig A. Knoblock, and Pedro Szekely. Efficient Graph-Based Document Similarity. In Harald Sack, Eva Blomqvist, Mathieu d'Aquin, Chiara Ghidini, Simone Paolo Ponzetto, and Christoph Lange, editors, *The Semantic Web. Latest Advances and New Domains*, volume 9678, pages 334–349. Springer International Publishing, Cham, 2016. ISBN 978-3-319-34128-6 978-3-319-34129-3. doi: 10.1007/978-3-319-34129-3_21. URL `http://link.springer.com/10.1007/978-3-319-34129-3_21`. Series Title: Lecture Notes in Computer Science.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. URL `http://jmlr.org/papers/v12/pedregosa11a.html`.

Ofir Pele and Michael Werman. Fast and robust Earth Mover's Distances. *2009 IEEE 12th International Conference on Computer Vision*, 2009. doi: 10.1109/ICCV.2009.5459199.

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

Mark O. Riedl. Human-Centered Artificial Intelligence and Machine Learning. *arXiv:1901.11184 [cs]*, January 2019. URL `http://arxiv.org/abs/1901.11184`. arXiv: 1901.11184.

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*. PhD thesis, Harvard University, May 2017. URL `https://dash.harvard.edu/handle/1/40046559`.

Pao Siangliulue, Joel Chan, Krzysztof Z. Gajos, and Steven P. Dow. Providing Timely Examples Improves the Quantity and Quality of Generated Ideas. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*, C&C '15, pages 83–92, Glasgow, United Kingdom, June 2015. Association for Computing Machinery. ISBN 978-1-4503-3598-0. doi: 10.1145/2757226.2757230. URL `https://doi.org/10.1145/2757226.2757230`.

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`.

Ignacio Traverso, Maria-Esther Vidal, Benedikt Kämpgen, and York Sure-Vetter. GADES: A Graph-based Semantic Similarity Measure. In *Proceedings of the 12th International Conference on Semantic Systems - SEMANTiCS 2016*, pages 101–104, Leipzig, Germany, 2016. ACM Press. ISBN 978-1-4503-4752-5. doi: 10.1145/2993318.2993343. URL `http://dl.acm.org/citation.cfm?doid=2993318.2993343`.

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.

Appendix

5.1 Recommendation Feature Evaluation Material

Git-Repositories

Semantic Similarity Approach:
`https://github.com/luka1220/SemanticSimilarityApproach`
Orchard Apllication:
`https://github.com/FUB-HCC/Innovonto-Orchard`

Think-Aloud and Interview Recommendation Feature

The Orchard Application[1] is an idea clustering tool to support experts in the creative process of synthesizing idea sparks and structuring the idea-space to evolve and select ideas concerning possible application of a given novel technology. The concern of the evaluation is the recommendation feature, which ought to support the expert during the process of clustering. The idea sparks used in the evaluation concern the application of the Transparent Conductive Oxides (TCO), which is a thin transparent nano-coating that makes materials and surfaces interactive and touch-sensitive. Your task is to organize 60 idea sparks into clusters that collect idea sparks concerning a common application of the TCO technology and possibly your are able to summarize them in an enhanced idea, combining the different aspects of the idea sparks as well as adding you own mindful contribution to a promising application of the technology.

1. We start with 5 minutes of reading on the TCO technology, followed by a short introduction to Orchard.
2. 30 minutes to cluster the idea sparks,
3. 10 minutes to create 2-3 ideas.
4. Lastly 10 minutes to evaluate the recommendation feature based on four questions.

Consent Form:

I agree to participate in the study conducted by the Human-Centered Computing Lab of the Freie Universität Berlin as part of a bachelor thesis. I understand that participation in this study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator. I understand that all data gathered in this test, including video and sound records, will be anonymized. I am aware that results from these tests might be published. Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date:

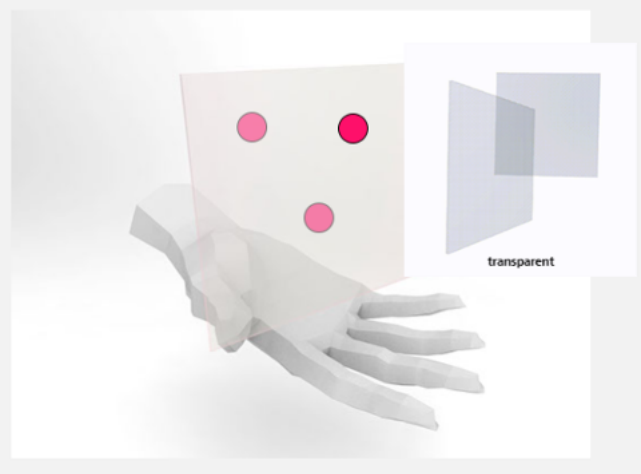
Please print your name:

Please sign your name:

[1] The application is accessible from the Network of FU-Berlin at `http://aghcc-srv02.imp.fu-berlin.de/apps/orchard/`

Information on Transparent Conductive Oxides[2]

- What is TCO?
  Transparent conductive oxides are materials that can be used as thin coatings to make materials and surfaces intelligent. They are transparent, conductive and flexible.



- What is its functionality?

  TCO coatings transform surfaces from objects and rooms into conductive and therefore interactive and touch-sensitive surfaces. Due to its transparency the original look and texture of surfaces is not changed.

- How can it be used?

  With TCO ink widely differing materials and objects can get coated, such as:

  - Temperature-sensitive materials such as plastic foils
  - Small and thin objects
  - large surfaces
  - Three-dimensional molded geometries
  - Flexible materials

- Where is it already used?

  The TCO coatings have versatile use cases such as smart labels, smart watches, intelligent textiles and instruments.

[2]descriptions from the Ideas2Market Clustering Workshop

Questions to elaborate on the Recommendation Feature

1. How useful are the recommendations for the interactive requests you toggle by selecting a idea spark or concepts of your interest?
2. To what extend are recommendations matching your expectations?
3. To what extend are you able to identify the relation between the recommended idea sparks and your selected idea spark or concept?
4. Comparing the two options of pulling an idea sparks: *SPARK STACK* and *RECOMMENDER*, see 5.1. To what extent do you prefer one option to the other, which option provides more sparks you considered valuable?

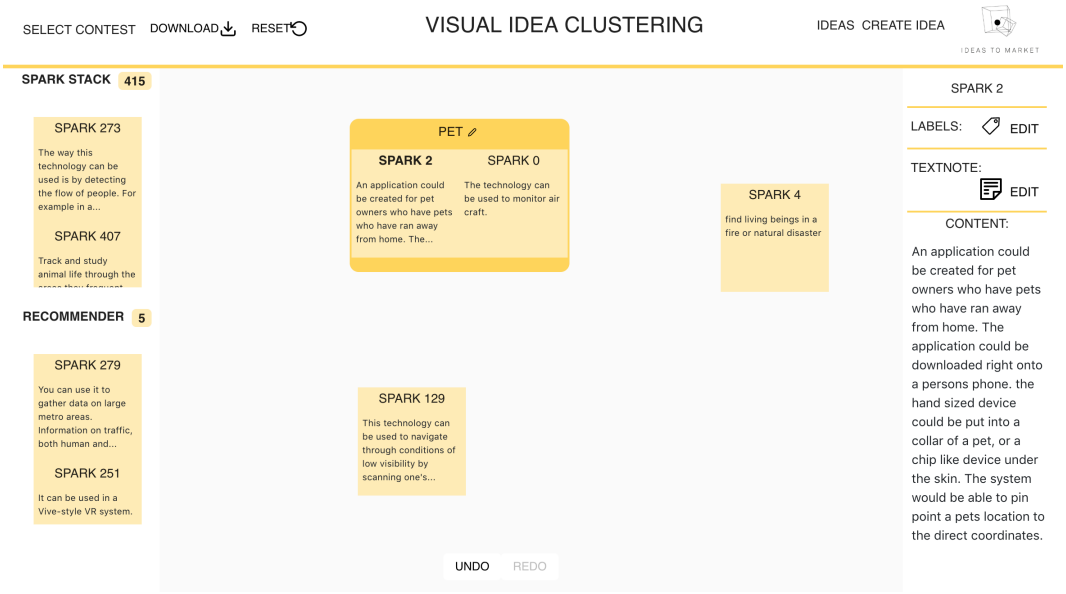


Figure 5.1: Orchard

5.2 Word Similarity Results

Table 5.1: M&C datasets human assessments means and wpath(ncs) similarity measurements with k=0.9. Yields a Spearman Correlation of 0.904

Word Pairs		mean	wpath	LCS
car	automobile	3.92	4.0	car
gem	jewel	3.84	4.0	gemstone
journey	voyage	3.84	2.86	travel
boy	lad	3.76	4.0	boy
coast	shore	3.7	4.0	bank
asylum	madhouse	3.61	4.0	psychiatric hospital
magician	wizard	3.5	4.0	magician in fantasy
midday	noon	3.42	4.0	noon
furnace	stove	3.11	3.08	stove
food	fruit	3.08	2.81	food
bird	cock	3.05	3.15	heraldic bird
bird	crane	2.97	1.39	taxon
tool	implement	2.95	4.0	tool
brother	monk	2.82	3.14	monk
lad	brother	1.66	2.5	male human
crane	implement	1.68	1.86	tool
journey	car	1.16	2.86	travel
monk	oracle	1.1	1.39	occupation
cemetery	woodland	0.95	0.77	geographical object
food	rooster	0.89	0.86	organism
coast	hill	0.87	1.62	landform
forest	graveyard	0.84	0.66	geographical object
shore	woodland	0.63	0.77	geographical object
monk	slave	0.55	1.19	position
coast	forest	0.42	0.71	natural geographic object
lad	wizard	0.42	0.72	person
chord	smile	0.13	0.59	economic concept
glass	magician	0.11	0.57	first-order metaclass
rooster	voyage	0.08	0.58	Wikidata metaclass
noon	string	0.01	0.75	goods

Table 5.2: M&C datasets human assessments means and wpath(lcs) similarity measurements with shortest path over LCS and k=0.9. Yields a Spearman Correlation of 0.882

Word Pairs		mean	wpath(lcs)	LCS
car	automobile	3.92	4.0	car
gem	jewel	3.84	4.0	gemstone
journey	voyage	3.84	2.86	travel
boy	lad	3.76	4.0	boy
coast	shore	3.7	4.0	bank
asylum	madhouse	3.61	4.0	psychiatric hospital
magician	wizard	3.5	4.0	magician in fantasy
midday	noon	3.42	4.0	noon
furnace	stove	3.11	3.08	stove
food	fruit	3.08	2.81	food
bird	cock	3.05	3.15	heraldic bird
bird	crane	2.97	2.1	bird
tool	implement	2.95	4.0	tool
brother	monk	2.82	3.14	monk
lad	brother	1.66	2.5	male human
crane	implement	1.68	1.86	tool
journey	car	1.16	2.86	travel
monk	oracle	1.1	1.39	occupation
cemetery	woodland	0.95	0.66	territorial entity
food	rooster	0.89	0.91	fodder
coast	hill	0.87	1.62	landform
forest	graveyard	0.84	0.58	territorial entity
shore	woodland	0.63	0.77	geographical object
monk	slave	0.55	1.34	social class
coast	forest	0.42	0.64	human settlement
lad	wizard	0.42	0.72	person
chord	smile	0.13	0.7	art form
glass	magician	0.11	0.55	human activity
rooster	voyage	0.08	0.79	motion
noon	string	0.01	0.75	object of group