Freie Universität Berlin

Master Thesis at the Institute for Computer Science at the Freie Universität Berlin

Human-Centered Computing (HCC)

Conceptualization and Evaluation of Idea Similarities based on Semantic Enrichment & Knowledge Graphs

Maximilian Timo Stauss

Reviewer: Prof. Dr. C. Müller-Birn

Second reviewer: Prof. Dr. M. Margraf

Advisor: M.Sc. M. Mackeprang

Berlin, 04.01.2021

Freie Universität Berlin
Institut für Informatik
Takustr. 9
14195 Berlin

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Änderungen entnommen wurde.

Berlin, 04.01.2021

(Maximilian Timo Stauss)

Abstract

This master thesis gives an overview of different semantic measures on general knowledge graphs like Wikidata. The focus was on the comparison of short idea texts. For these pairs of ideas, semantic measures were implemented which should represent human intuition as well as possible. To test the quality of the implemented measures, they were compared with a state-of-the-art algorithm from the field of machine learning on the one hand and with the results of a user study on the other hand. The basis of this work is the book 'Semantic Similarity from Natural Language and Ontology Analysis'. [20], which provides an overview of different semantic measures on Knowledge Graphs. By using this approach to compare short idea texts, the author hopes to show alternatives to commonly used machine learning approaches.

Zusammenfassung

Die vorliegende Masterarbeit gibt einen Überblick über verschiedene semantische Maße auf generellen Knowledge Graphs wie Wikidata. Dabei lag der Fokus auf dem Vergleich kurzer Ideentexte. Für diese Ideenpaare wurden semantische Maße implementiert, welche die menschliche Intuition möglichst gut abbilden sollen. Um die Güte der implementierten Maße zu testen, wurden diese einerseits mit einem state-of-the-art Algorithmus aus dem Bereich des Machine Learnings und andererseits mit den Ergebnissen einer Nutzer*innenstudie verglichen. Die Grundlage dieser Arbeit bildet das Buch 'Semantic Similarity from Natural Language and Ontology Analysis' [20], welches einen Überblick über verschiedene semantische Maße auf Knowledge Graphs bietet. Durch die Nutzung dieses Ansatzes zum Vergleich kurzer Ideentexte erhofft sich der Autor Alternativen zu häufig genutzten Maschine Learning Ansätzen aufzuzeigen.

Contents

Abstract i

Zusammenfassung iii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Research Goal . . . 2
- 1.3 Research Approach & Methodology . . . 3

2 Thematic Classification 5
- 2.1 Ideation Context . . . 5
  - 2.1.1 Large Scale Ideation . . . 5
  - 2.1.2 User Studies & Crowd Workers . . . 5
  - 2.1.3 Idea Challenges . . . 6
- 2.2 Linked Data & Knowledge Graphs . . . 6
  - 2.2.1 RDF . . . 7
  - 2.2.2 Knowledge Graph Definitions . . . 7
  - 2.2.3 SPARQL . . . 8
- 2.3 Semantic Enrichment . . . 8
- 2.4 Idea Similarity & Semantic Measures . . . 9
- 2.5 Machine Learning . . . 9
- 2.6 Summary . . . 10

3 Semantic Measures 11
- 3.1 Direct & Indirect Group-wise Measures . . . 11
- 3.2 Algorithms for Knowledge Graphs . . . 12
- 3.3 Measures Based on Graph Structure Analysis . . . 13
- 3.4 Measures Based on Concept Feature Analysis . . . 14
- 3.5 Measures Based on Information Theoretical Analysis . . . 15
  - 3.5.1 Definitions of Information Content & MICA . . . 15
  - 3.5.2 Information Content Algorithms . . . 15
  - 3.5.3 Information Based Algorithms . . . 17
- 3.6 Measures based on machine learned embedding . . . 17
- 3.7 Selection of Semantic Measures . . . 18
  - 3.7.1 Limitation for Comparability . . . 18
  - 3.7.2 SPARQL Endpoint Timeouts . . . 18
  - 3.7.3 Preliminary Selection Step . . . 19
- 3.8 Summary . . . 19

4 Study of Similarity between Ideas **21**
4.1 Idea Selection . . . 21
4.2 Idea Annotation . . . 22
4.3 Gather Intuitive Similarity (User Study) . . . 24
4.3.1 Conceptualization & User Interface . . . 24
4.4 Gather Semantic Measures . . . 26
4.4.1 Semantic Measures on Knowledge Graphs . . . 26
4.4.2 Semantic Measures based on USE-Embedding . . . 27
4.5 Calculations on Semantic Measures . . . 27
4.5.1 Correspondence versus Correlation . . . 27
4.5.2 Semantic Measures as Estimators . . . 27
4.6 Summary . . . 28

5 Implementation **31**
5.1 User Study: Manual Similarity . . . 31
5.1.1 Backend . . . 31
5.1.2 Frontend . . . 31
5.1.3 Results . . . 32
5.2 Wikidata Semantic Measure Toolkit . . . 32
5.2.1 Timeouts, AsyncIO & Caching . . . 33
5.2.2 Layout . . . 33
5.2.3 Dependencies . . . 35
5.2.4 Next Steps . . . 35
5.3 Universal Sentence Encoder . . . 35
5.4 Results . . . 35
5.5 Summary . . . 37

6 Results **39**
6.1 User Study Results . . . 39
6.1.1 Intuitive Similarities . . . 39
6.1.2 Feedback . . . 41
6.1.3 Clarity . . . 41
6.2 Preliminary Selection of Computed Semantic Similarities . . . 41
6.2.1 Direct Approach . . . 43
6.2.2 Indirect Approach . . . 44
6.2.3 Selected Semantic Algorithms on Knowledge Graphs . . . 46
6.3 Timings . . . 46
6.4 Summary . . . 46

7 Evaluation **47**
7.1 Evaluation of Hypotheses . . . 47
7.1.1 Evaluation of Hypothesis $\mathscr{H}_1$ . . . 47
7.1.2 Evaluation of Hypothesis $\mathscr{H}_2$ . . . 51
7.2 Limitations & Improvements . . . 52
7.3 Research Question . . . 52

7.4 Research Contribution 53
7.5 Summary 53

8 Conclusion **55**
8.1 Summary 55
8.2 Future Work 56

9 Appendix **57**
9.1 DBpedia & Wikidata Interface 57
9.2 Semantically Enriched Ideas 58
9.3 Study Results 59
9.4 Idea Similarity Mock-Up 61

Bibliography **65**

List of Figures

4.1 Task Flow Diagram of Semantic Enrichment as used in [28]. 23
4.2 Example DBpedia Spotlight annotation using the first idea. Showing the candidates and their confidence score for the term 'athletes'. . . . 23
4.3 Showing the disambiguation interface for searches on Wikidata for the terms 'athlete' and 'hurting' provided by DBpedia Spotlight. 23
4.4 Overview of the steps expected from every crowd worker. 24
4.5 Mock-up of a pairwise comparison for ideas 1 and 5 from Table 4.1. . 25

6.1 Heat-maps showing the similarities used for evaluation and quality checks in (a) and the averaged crowd worker results in (b). 40
6.2 Plot showing the mean square error of every semantic measure to the intuitive similarity measures. The measures are grouped by approach and aggregation function. A lower mean square error implies higher similarity. 43
6.3 Plot showing the cosine similarity to the intuitive similarity measures. The measures are grouped by approach and aggregation function. A higher cosine similarity implies higher similarity. 45

7.1 Heat maps of computed similarities. (a) Ground Truth is shown as a reference. 48
7.2 Plot comparing $\text{sim}_{\text{AVG-CMatch}}$ with $\text{sim}_{\text{USE,angular}}$ and the ground truth. 50
7.3 Plot comparing $\text{sim}_{\text{BMM-Lin,naive}}$ with $\text{sim}_{\text{USE,angular}}$ and the ground truth. 50

9.1 Showing the disambiguation interface for searches on Wikidata for the terms 'athlete' and 'hurting' provided by DBpedia Spotlight. 57
9.2 Wikidata SPARQL Query Interface 57
9.3 Plot comparing $\text{sim}_{\text{BMA-CMatch}}$ with $\text{sim}_{\text{USE,angular}}$ and the ground truth. 59
9.4 Plot comparing $\text{sim}_{\text{BMM-Resnik,log}}$ with $\text{sim}_{\text{USE,angular}}$ and the ground truth. 60
9.5 Plot comparing sim_{TO} with $\text{sim}_{\text{USE,angular}}$ and the ground truth. . . . 60
9.6 User Study Intro - Not Activated 61
9.7 User Study Intro - Activated 62
9.8 User Study Intro - Read Ideas 62
9.9 User Study Task - Idea Comparison 63
9.10 User Study Outro - Questionnaire 63

List of Tables

4.1 Ideas selected for Comparison Study. All ideas are extended so that the challenge is part of the idea text 29

6.1 Table showing the consequences of lowering the number of results by eliminating those results with the highest double check difference 'DC Diff'. 'Results' is the number of results in the remaining set, 'Variation' is the coefficient of variation, 'Variance' is the variance, 'Std Deviation' describes the standard deviation, and 'DC Diff' the highest double check difference still in the remaining set. 40

6.2 Collected Meta Data: Clarity rating together with the double check difference 'DC Diff' and quality check similarities 'QC Sim' for the idea pair (5,6). The 'Feedback' attributed to the corresponding crowd worker. 42

6.3 All direct semantic measures on knowledge graph compared by mean square error 'MSE', cosine similarity 'COS' and angular similarity 'ANG'. 43

6.4 All indirect feature based semantic measures on knowledge graph compared by mean square error 'MSE', cosine similarity 'COS' and angular similarity 'ANG'. 44

6.5 Indirect information based semantic measures on knowledge graph aggregated by the best match max algorithm (BMA) compared by mean square error 'MSE', cosine similarity 'COS' and angular similarity 'ANG'. 45

6.6 All different semantic algorithms selected for evaluation compared by 'Approach', aggregation algorithm 'Aggregator', mean square error 'MSE', cosine similarity 'COS' and angular similarity 'ANG'. 46

7.1 Statistics for the harmonized upper triangle matrices. With 'Mean', 'Min', and 'Max' the according values within each matrix. The standard deviation is described by 'STD'. 51

7.2 Comparison of the semantic similarity measures to the WTA Scores. '$>$' describes the number of similarities closer to the ground truth. Under 'WTA to NE' are the sums of the winning differences against $\text{sim}_{\text{naive}}$. Under 'WTA to USE' are the sums of the winning differences against $\text{sim}_{\text{USE,angular}}$. 51

1 Introduction

Merriam-Webster defines innovation as "the introduction of something new," yet coming up with novelties is not an easy task. Missing inspiration or fixation on a narrow area can hinder the ideation process. Research suggests that large-scale ideation platforms can be beneficial to the innovation process, in order to overcome those limitations. By leveraging online crowds and their diverse backgrounds, those platforms enable collaborative ideation: *Everybody has the potential to inspire anyone else.* One of the biggest challenges is the sheer amount of ideas collected. With ideation challenges containing hundreds of thousands of ideas it is not feasible for one person to read and evaluate each and every idea. Software and Tools that visualize ideas and set them in relation to each other can help providing organization and grouping. In order to provide such organization and visualization, an algorithmic understanding of the ideas is needed.

1.1 Motivation

The research-group Human-Centered Computing (HCC) is active in ideation contexts. In the ideation context an ideation task describes generally the search for an idea that solves a given problem. Such a task can range from simple brainstorming or ideation sessions in a conference room with five to ten participants to *large-scale ideation*, where ideation tasks are solved on a much larger scale. IBM for example held an 'Innovation Jam' with 46,000 ideas submitted from 150,000 participants [7]. Such an amount of ideas makes it difficult to find those non-redundant, novel ideas by hand [49] and therefore its not surprising that it took reviewers about six months to assess all ideas submitted [7]. It is a common goal for coordinators in large-scale ideation tasks to increase the novelty and creativity of each and every individual ideator. Research suggests that in the ideation task itself it can be beneficial to show external ideas to the ideator. It is possible to increase creativity by exposing the ideator to similar ideas and/or ones that are very different [17, 11, 48]. Studies show that those ideas shown should be diverse, non-repetitive, and heterogeneous to improve the innovation process [30, 50]. But how does one identify similar or different ideas to the one at hand?
Based in that context, the paper 'Discovering the Sweet Spot of Human-Computer Configurations: A Case Study in Information Extraction' [28] was written. In this paper we explored the effects of different levels of automation in the context of ideation. We developed a tool to extract information[1] from idea texts and link[1] that information

[1]Information extraction and linking are core techniques in the field of natural language processing (NLP). In our case is information extraction and linking are synonymous to semantic enrichment. It is further explained in section 2.3.

to a knowledge graph[2]. The goal was "to find the right level of algorithmic support, whereby the quality of the information extraction should be as high as possible, but, at the same time, the human effort should be low". While we were collecting the data for the different levels of automation [28, Chapter 4] we searched for ways to compare the quality of the results. We settled on the following solution:

> "An established way of measuring the quality in the area of information extraction is to measure precision, recall and the F-measure. Whereas precision defines the number of concepts annotated correctly, recall defines the number of concepts correctly disambiguated relative to the number of all concepts found. The F-measure is the harmonic mean of precision and recall." (Mackeprang et al. [28])

We were not completely satisfied by this solution. So to reflect our discussions, we added the following as a limitation and a proposal for future future work:

> "[We] experienced the limitation of a 'Boolean' approach to concept extraction during our creation of the gold standard: We decided to allow multiple correct concepts for one term, as multiple definitions were appropriate (especially for colloquial technical terms, such as 'screen'). Having a distance-based quality metric (how far is the 'correct' concept away from the annotated one) could be helpful in such instances." (Mackeprang et al. [28])

That limitation and its implications motivated this thesis. Especially the *binary approach* introduced by precision and recall inspired the search for alternatives.

1.2 Research Goal

To overcome that limitation I was looking for idea similarities of form *'Idea A' and 'Idea B' have an similarity of x%*.

$$\text{idea\_similarity} : \texttt{Idea} \times \texttt{Idea} \to [0,1] \subset \mathbb{R}$$

with `Idea` being the infinite set of all idea texts.

Because we were already using knowledge graphs[2] for disambiguation[3] in [28] I focused on solutions leveraging those. The first papers I found in that area introduced the terminology of semantic measure[4] on knowledge graphs [42, 54, 26, 45]. With that terminology I was able to find [20] in which different semantic measures on knowledge graphs are compared.

[2] A 'knowledge graph' is a graph containing and connecting knowledge and is further explained in section 2.2.

[3] Disambiguation is part of the semantic enrichment process. It is further explained in section 2.3.

[4] Semantic measures are measures leveraging the meaning of concepts and texts to compare them. They are further explained in chapter 3.

Another semantic measure up for consideration is Google's 'Universal Sentence Encoder' (USE) [10]. It uses machine learning to generate embedding vectors that are easily convertible into similarity scores[5].

Those initial findings brought me to the following research question:

Research Question

How well suited are 'Semantic Measures on Knowledge Graphs' to compare ideas regarding their similarity?

To answer this question, I decided to examine the following two auxiliary hypotheses:

Hypothesis $\mathscr{H}_1$

A semantic measure exists that corresponds with human intuition, the 'ground truth'.

Hypothesis $\mathscr{H}_2$

Knowledge-Graph based approaches correspond stronger than similarities based on the machine learning model Universal Sentence Encoder (USE).

1.3 Research Approach & Methodology

To test the hypotheses formulated and to answer the research question several steps need to be taken. After the terminology is explained in chapter 2 and the different semantic measures are introduced in chapter 3, the main study is described in chapter 4. This study follows five steps:

1. Ten ideas are selected for a pair-wise comparison.
2. Crowd workers are tasked to compare all ideas.
3. Semantic measures are selected and implemented.
4. The similarity scores are calculated using the measures implemented.
5. The semantic measures are compared with the results of the user study.

The implementation of those steps is then explained in chapter 5. After that, in chapter 7 the results of that study are discussed. The hypotheses $\mathscr{H}_1$ & $\mathscr{H}_2$ and the research question are evaluated here based on the results. In section 7.2 the limitations of this thesis will be discussed.

[5]Machine learning and Google's USE-Model are further explained in section 2.5.

2 Thematic Classification

This chapter describes the terminology needed for the following chapters. As mentioned in the introduction, this thesis is motivated by [28]. Therefore the ideation context is of great importance. After an overview of the ideation context, I describe knowledge graphs in order to illustrate the process of semantic enrichment. The final section covers machine learning and word embeddings needed for hypothesis $\mathscr{H}_2$.

2.1 Ideation Context

Ideation describes the journey from a blank piece of paper to an idea. Along that process, systems and strategies can be applied to support ideators in their creativity. It is possible to increase creativity by exposing the ideator to similar ideas and/or ones that are very different [17, 11, 48]. Studies suggest that those ideas shown should be diverse, non-repetitive, and heterogeneous to improve the innovation process [30, 50].

2.1.1 Large Scale Ideation

Large scale ideation describes ideation tasks on a immense scale with tens of thousands of ideas and hundreds of thousands of participants [7]. Usual problems for coordinators include handling the amount of ideas submitted [49, 7] and increasing the quality of the outcome of each and every individual ideator [17, 11, 48].

2.1.2 User Studies & Crowd Workers

User studies, especially those leveraging crowd-workers, play a significant role in research regarding large scale ideation.

User Studies
: A user study is an evaluation of something involving individuals directly. The places for such studies range from labs over users' natural environments [67]. User studies are an essential tool in "virtually any design endeavor". They "may include methods such as surveys, usability tests, rapid prototyping, cognitive walkthroughs, quantitative ratings, and performance measures" (Kittur et al. [24]).

Crowd-Workers
: The term crowd-workers describes "human users on the web" that are paid to "complete simple tasks that would otherwise be extremely difficult (if not impossible) for computers to perform" (Kittur et al. [24]).

Research suggests that "micro-task markets such as Amazon's Mechanical Turk are promising platforms for conducting a variety of user study tasks" (Kittur et al. [24]). All user studies conducted in this thesis are done via Amazon's Mechanical Turk.

2.1.3 Idea Challenges

In prior user studies the research group held challenges to generate ideas for applications of new (non-existent) products [27, 29]. In those challenges crowd workers were tasked to produce ideas leveraging one of new technologies listed. An example task for an ideation challenge was:

> "Imagine you could have a coating that could turn every surface into a touch display. Brainstorm cool products, systems, gadgets or services that could be built with it." (Mackeprang et al. [29])

The descriptions of all three challenges are the following:

Bionic Radar
: A technology can perceive the movement of objects such as people, living beings, and objects. By remembering the object's movement, it is capable of recognizing the same object later on. Furthermore, the technology can compare the object's movement with the movement of other objects, and thereby conclude comprehensive movement patterns. These movement patterns enable the detection of objects with the same movement profile.

Fabric Display
: A touch-sensitive "fabric display" that could render high resolution images and videos on any fabric through a penny-sized connector.

Transparent Conductive Oxides (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. 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.

In these challenges exist more than $1,600$ individual idea texts. A typical idea text is a couple of sentences long and therefore too short for most automatic natural-language processing tools.

> "Analyze athletes to see what part of them is either helping or hurting them in things such as running." (Ideator during Bionic Radar Challenge)

2.2 Linked Data & Knowledge Graphs

Linked data is a generalization for interconnections and publications of structured data on the web [6]. Such data could be:

- *Jim Morrison* was lead singer of the *Music Band The Doors*.
- *The Soft Parade* was the fourth *Music Album* of *The Doors*.

An often used representation of that information are ontologies or, specifically, knowledge graphs. "An ontology is a formal explicit specification of a shared conceptualization for a domain of interest." [18] Knowledge graphs are ontologies, but as mentioned in [15], a variety of definitions exists. As a first generalization a knowledge graph can be viewed as a 'directed multi-graph'[1] containing and connecting knowledge [20]. Those knowledge graphs are commonly implemented as RDF triplet stores.

2.2.1 RDF

The Resource Description Framework (RDF) enables the description of information. A resource of information can either be a *URI*[2], a *Blank Node*[2] or a *Literal*[2]. A *URI*, uniform resource identifier, is a "string of characters that identifies a resource that can be accessed over the Internet" (Wiktionary). So `https://www.wikidata.org/wiki/Q1`, or `wd:Q1` for short[3], is the URI for the Wikidata entry 'universe' [66, 63]. The description of information is represented by a RDF triple of the form *(subject, predicate, object)* [6, 32]. A RDF triple corresponds to a simple statement in the form of:

```
JimMorrison   playsInBand   TheDoors.
```

Listing 2.1: Example RDF Triple

With that, an RDF-based definition of knowledge graphs can be introduced.

> "We define a Knowledge Graph as an RDF graph. An RDF graph consists of a set of RDF triples where each RDF triple (s, p, o) is an ordered set of the following RDF terms: a subject $s \in U \cup B$, a predicate $p \in U$, and an object $U \cup B \cup L$. An RDF term is either a URI $u \in U$, a blank node $b \in B$, or a literal $l \in L$." (Färber et al. [16])

2.2.2 Knowledge Graph Definitions

Based on that RDF knowledge graph definition, I can introduce the following definitions used throughout this thesis. The definitions are derived from [20]. Some definitions are simplified or omitted to better reflect the knowledge graphs considered.

Knowledge Graph $\mathbb{KG} = \langle \mathcal{C}, \mathcal{P}, \mathcal{R} \rangle$
The Knowledge Graph $\mathbb{KG}$ consists of Concepts $\mathcal{C}$, Predicates $\mathcal{E}$ and Relationships $\mathcal{R}$. Therefore it is defined by $\langle \mathcal{C}, \mathcal{P}, \mathcal{R} \rangle$.

Concepts $\mathcal{C}$ describes the set of 'things' sharing common properties.

[1] For basic definitions of graphs see [12, Appendix B.4].
[2] For definitions of *URI*, *Blank Node* and *Literal* see [32, Section 3.2].
[3] The short-hands for URIs are further explained in subsection 2.2.3.

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
SELECT ?album
WHERE
{
  ?album wdt:P175  wd:Q45354;
         wdt:P31  wd:Q482994.
}
```

Listing 2.2: Wikidata SPARQL Query to list all albums of The Doors.

Predicates $\mathcal{P}$ describes the types of relationships which can established between two concepts $u, v \in \mathcal{C}$.

Relationships $\mathcal{R}$ describes the set of concrete relations between two concepts $u, v \in \mathcal{C}$ using a specific predicate $p \in \mathcal{P}$. Relations follow the RDF-inspired nomenclature: `subject`, `predicate`, `object` or $\langle s, p, o \rangle$.

Path describes an ordered set of relationships that start at a concept and end at (a different) one. Example path from a to d: $[\langle a, p_1, b \rangle, \langle b, p_2, c \rangle, \langle c, p_1, d \rangle] \in \texttt{paths}(a, d)$.

Root Element $\top$ describes the root of all elements within a knowledge graph $\mathbb{KG}$, meaning, that $\forall c \in \mathcal{C} \Rightarrow \texttt{paths}(\top, c) \neq \emptyset$.

Leaves are those concepts with no outgoing relations.

$$l \in \mathcal{C} \text{ is a leaf } \Leftrightarrow \forall p \in \mathcal{P}, c \in \mathcal{C} \setminus l \nexists \langle l, p, c \rangle \in \mathcal{R}$$

2.2.3 SPARQL

SPARQL, short for SPARQL Protocol and RDF Query Language, is the defacto query language to access RDF triplet stores such as knowledge graphs [22, 65]. It is as fundamental to the semantic web as SQL is to relational databases [58]. As the name SPARQL implies the syntax is comparable to SQL as well.
As seen in Listing 2.2 prefixes can be used to shorten URIs significantly. Common prefixes such as `rdf:`, `wd:`, or `dbpedia:` are often predefined. It is also possible to define own prefixes. Good introductions to SPARQL can be found on Wikidata [61, 62]. All SPARQL queries in this thesis are tested against the Wikidata Query Service [60]. An example of the Wikidata Query Service can be seen in Appendix 9.2.

2.3 Semantic Enrichment

As explained in subsection 2.2.2, knowledge graphs hold concepts. Accordingly, a step is necessary to get the underlying concepts from a given text. This step is called semantic enrichment. The combination of terms and concepts to identify and disambiguate terms is called 'semantic enrichment' and the process of retrieving terms

is 'information extraction'. In [28, Chapter 5.2] we used 20 idea texts from [27] for our information extraction tool. The goal of that tool was to disambiguate the terms[4] of an idea by selecting the closest concept of the DBpedia knowledge graph[5].

Semantic enrichment describes the extension of text with semantic meaning. Therefore, the text is split into terms and every relevant term is *annotated* with corresponding *concepts*.

The goal is to define and disambiguate words or terms by describing the meaning of them. So in a text about 'doors', for example, the term 'doors' could be defined as either the band *The Doors* or, as Wikidata says, "[a] movable structure used to open and close an entrance" ([59]).

2.4 Idea Similarity & Semantic Measures

Idea similarity describes how similar two ideas are. It is important that the similarity is not only at the word level, but includes the whole idea and its implications. The semantics should also be compared. All semantic measures should have the following signature:

$$\text{sim} : \texttt{Idea} \times \texttt{Idea} \to [0,1] \subset \mathbb{R}$$

with `Idea` being the infinite set of all idea texts.

The algorithmic approach to idea similarity using semantic measures is very comprehensive and is therefore explained in detail in chapter 3.

2.5 Machine Learning

Machine Learning describes the attempt to solve a problem computationally without explicitly providing an algorithm for said problem. With the availability of huge data sets, also known as 'big data', this new approach emerged. In big data it is extremely difficult or even impossible to build an explicit algorithm making sense of such a data set. A common example is the detection of spam in emails:

> "We know what the input is: an email document that in the simplest case is a file of characters. We know what the output should be: a yes/no output indicating whether the message is spam or not. But we do not know how to transform the input to the output. What is considered spam changes in time and from individual to individual.
> *What we lack in knowledge, we make up for in data.*" (Alpaydin [4])

[4] A term is the longest (compound) word that has a meaning by itself.

[5] The knowledge graph $\mathbb{KG}_{\text{DBpedia}}$ contains about 4.2 million items, including 685 classes and 2795 predicates, automatically extracted from Wikipedia entries [36].Further information can be obtained via `https://wiki.dbpedia.org/`.

Therefore, it is up to the computer to take lots of emails in, some of which are marked as spam, and derive a way to differentiate spam from legitimate emails. The 'machine' is supposed to 'learn' what spam is.

Applications of Machine Learning

Machine learning has many different applications. Some of them are 'Learning Associations', 'Classification', and 'Regression' [4, 9]. In this thesis the focus is on semantic measures. Therefore it is of interest to use machine learning capabilities to generate a semantic measure. 'Classification' and its specialization 'Knowledge Extraction' are a good fit for that task.

Universal Sentence Encoder (USE)

For this thesis machine learned models[6] resulting in embedding vectors are of interest [14, 2, 25]. Many different algorithms exist with Google's Universal Sentence Encoder being a current state-of-the-art approach to sentence embeddings for short text fragments [10, 37, 5]. It introduces models for encoding sentences into embedding vectors[7]. The main goal of those pre-trained models is to overcome the limited amounts of training data available in the area of natural language processing (NLP). The resulting embedding vectors are easily converted into a similarity score. That score corresponds positively with human similarity ratings [3]. The models are open source and easy to implement [53, 51].

2.6 Summary

This chapter gave the necessary description for ideation, knowledge graphs, idea similarity, and machine learning. It explained what user studies are and what the term crowd worker means. Prior idea challenges were introduced. Relevant terms related to knowledge graphs, idea similarity, and machine learning were explained. All further chapters build on the definitions and descriptions.

[6] For an introduction to machine learning see [4].
[7] For an introduction to embeddings see [25].

3 Semantic Measures

This chapter explains in detail the four different types of measures considered for this thesis. In sections 3.1 and 3.2 I will introduce definitions, that are needed to describe semantic measures within the context of knowledge graphs. With those definitions I will then list representative algorithms for those categories. The four categories of semantic measures considered are:

Measures based on graph structure analysis
: Similarity is based on the degree of interconnection between concepts. Shortest paths and graph distances play a vital role in this category.

Measures based on concept feature analysis
: Similarity is based on shared or distinct 'features'. Features are extracted from concepts and relations within the knowledge graph.

Measures based on information theoretical analysis
: Similarity is based on the amount of information two concepts have in common. This category shares core principles with the feature based approach.

Measures based on machine learned embedding
: Similarity is based on embedding vectors that are converted into a similarity score. This is an alternative approach to semantic similarity.

All semantic measures on knowledge graphs in this thesis are derived from [20]. After all semantic measures are introduced, I will briefly discuss the usage or avoidance of specific algorithms in section 3.7.

3.1 Direct & Indirect Group-wise Measures

As mentioned in section 2.3, the data processed by knowledge graphs are concepts. For the progression of these following chapters, it can be presumed that a set of annotations is available for each idea text, which assigns the idea to its concepts[1]. As in [20] I divide the measures for knowledge graphs presented in the following chapters into two categories. Pair-wise measures use concept similarity (see Algorithm 3.1) and group-wise measures (see Algorithm 3.2).

Concept Similarity
: The similarity of two concepts $u, v \in \mathcal{C}$ is represented by a real number.

$$\text{sim} : \mathcal{C} \times \mathcal{C} \to \mathbb{R} \tag{3.1}$$

[1] The creation of such annotations is explained in section 4.2.

Concept Similarity for Groups of Concepts
The similarity of two groups of concepts $U, V \subset \mathcal{C}$ is represented by a real number.

$$\text{sim}: \ 2^{\mathcal{C}} \times 2^{\mathcal{C}} \to \mathbb{R} \tag{3.2}$$

Where group-wise measures can be used directly to derive a similarity score between two groups of concepts, a further step is needed for pair-wise measures. Every pair-wise similarity introduced can be transformed into a indirect group-wise measure using one of the following aggregation algorithms as listed in [20]:

Average A naive average of the sum of all pair-wise similarities between two groups.

$$\text{sim}_{avg}(U, V) : \frac{\sum_{u \in U} \sum_{v \in v} \text{sim}(u, v)}{|U| \cdot |V|}$$

It is important to note, that $\text{sim}_{avg}(U, U) \leq 1$.

Max Average The averaged maximum of all pair-wise similarities between two groups.

$$\text{sim}_{max\text{-}avg}(U, V) : \frac{1}{|U|} \sum_{u \in U} \max_{v \in V} \{\text{sim}(u, v)\}$$

Best Match Max An extension of the *Max Average* algorithm that uses the maximum of the two possible arrangements.

$$\text{sim}_{BMM}(U, V) : \max\left(\text{sim}_{max\text{-}avg}(U, V), \text{sim}_{max\text{-}avg}(V, U)\right)$$

Best Match Average Another extension of the *Max Average* algorithm that averages the results of the two possible arrangements.

$$\text{sim}_{BMA}(U, V) : \frac{\text{sim}_{max\text{-}avg}(U, V) + \text{sim}_{max\text{-}avg}(V, U)}{2}$$

Group-wise similarities not based on aggregated pair-wise algorithms are called *direct group-wise measures.*

3.2 Algorithms for Knowledge Graphs

This section introduces algorithms that are vital for many measures based on knowledge graphs. It uses the nomenclature introduced in subsection 2.2.2.

Direct Descendants $D'(c) \subset \mathcal{C}$ describes the set of all concepts with $c \in \mathcal{C}$ as one its ancestor.

$$\begin{aligned} D' : \mathcal{C} &\to 2^{\mathcal{C}} \\ c &\mapsto D'(c) = \{d \mid \langle c, P, d\rangle \in \mathcal{R} : P \subset \mathcal{P}\} \end{aligned}$$

with $P \subset \mathcal{P}$ describing those predicates representing `is_descendent_of`.

Direct Ancestors $A'(c) \subset \mathcal{C}$ describes the set of all concepts with $c \in \mathcal{C}$ as its descendants.

$$A' : \mathcal{C} \to 2^{\mathcal{C}}$$
$$c \mapsto A'(c) = \{a \mid \langle a, P, c\rangle \in \mathcal{R} : P \subset \mathcal{P}\}$$

with $P \subset \mathcal{P}$ describing those predicates representing `is_descendent_of`.

Descendants $D(c) \subset \mathcal{C}$ describes the set of all concepts descending from $c \in \mathcal{C}$.

$$D : \mathcal{C} \to 2^{\mathcal{C}}$$
$$c \mapsto D(c) = \begin{cases} \varnothing, & \text{if } D'(c) = \varnothing. \\ \bigcup_{c' \in D'(c)} D(c'), & \text{otherwise.} \end{cases}$$

Ancestors $A(c) \subset \mathcal{C}$ describes the set of all concepts where $c \in \mathcal{C}$ is a descendant.

$$A : \mathcal{C} \to 2^{\mathcal{C}}$$
$$c \mapsto A(c) = \begin{cases} \varnothing, & \text{if } A'(c) = \varnothing. \\ \bigcup_{c' \in A'(c)} A(c'), & \text{otherwise.} \end{cases}$$

Depth $depth(c)$ describes the length of the shortest path[2] from $\top$ to c.

Leaves $leaves(c)$ describes the set of all leaves[2] belonging to $c \in \mathcal{C}$.

$$leaves : \mathcal{C} \to 2^{\mathcal{C}}$$
$$c \mapsto leaves(c) = \begin{cases} \{c\}, & \text{if } c \text{ is leaf node.} \\ \bigcup_{c' \in D(c)} leaves(c'), & \text{otherwise.} \end{cases}$$

3.3 Measures Based on Graph Structure Analysis

Graph structure analysis strips away the semantic nature of the knowledge graph and simply takes the graph properties into account. In this class of semantic measures graph-traversals like shortest paths to find similarities.
With the shortest path between two nodes u, v as explained in [12, 20] I will use $sp : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ as the length of the shortest path between two concepts.

Pair-Wise Measures

A pair-wise measure featuring the shortest path between two concepts as a similarity has been mentioned amongst others by Rada in 1989 [20, 40].

$$\mathrm{sim}_{\mathrm{Rada}}(u, v) = \frac{1}{\mathrm{sp}(u, v) + 1}$$

[2] See subsection 2.2.2 for definitions of 'path' and 'leaves'.

Direct Group-Wise Measures

In [20] the authors proposed the following algorithm: The similarity of two sets of concepts U, V is defined by to the length of the longest shortest path $\max\{sp(c, \top)\}$ which links a concept within the both sub-graphs $G^*(U) \cap G^*(V)$ to the root element $\top$.

$$\text{sim}_{\text{Gentleman}}(U, V) = \max_{c \in G^*(V) \cap G^*(V)} \{sp(c, \top)\}$$

with $G^*(C)$ being the sub-graph of the knowledge graph $\mathbb{KG}$ induced by $\bigcup_{c \in C} A(c)$, the union of the ancestors of all concepts in C.

3.4 Measures Based on Concept Feature Analysis

Concept feature analysis focuses on looking at concepts as a set of features. Such features can be the presence or absence of relations of a specific type. Another feature could be the number of different relations coming from or going to a concept.

CMatch is the naive implementation in that category. The concept match algorithm looks as the number of ancestors the two concepts u, v have in common.

$$sim_{\text{CMatch}}(u, v) = \frac{|A(u) \cap A(v)|}{|A(u) \cup A(v)|}$$

Bulskov introduced a weighted algorithm which combines the ancestors u and v, while allowing to weight one of them higher.

$$sim_{\text{Bulskov}}(u, v) = \alpha \frac{|A(u) \cup A(v)|}{|A(u)|} + (1 - \alpha) \frac{|A(u) \cup A(v)|}{|A(v)|}$$

RE is named after the two authors Rodrígues and Egenhofer [43]. It compares the ancestors u and v have in common with those they do not have in common.

$$sim_{\text{RE}}(u, v) = \frac{|A(u) \cap A(v)|}{\alpha \cdot |A(u) \setminus A(v)| + (1 - \alpha) \cdot |A(v) \setminus A(u)| + |A(u) \cap A(v)|}$$

Direct Group-Wise Measures

For the two measures introduced here, $A(C) = \bigcup_{c \in C} A(c)$ applies.

Term Overlap

The term overlap similarity describes the relationship of the ancestors both in U and V to all ancestors.

$$\text{sim}_{\text{TO}}(U, V) = \frac{|A(U) \cap A(V)|}{|A(U) \cup A(V)|}$$

Normalized Term Overlap

The normalized term overlap similarity describes the relationship of the ancestors both in U and V to the smaller set of ancestors of U and V.

$$\text{sim}_{\text{NTO}}(U, V) = \frac{|A(U) \cap A(V)|}{\min(|A(U)|, |A(V)|)}$$

3.5 Measures Based on Information Theoretical Analysis

For the information theoretical approach, it is considered that each and every concept has a level of specificity. All algorithms of this category use the information content to measure a score for a concepts importance.

3.5.1 Definitions of Information Content & MICA

Information Content (IC) describes the degree of abstraction of a concept. The more specific a concept is the higher the information content should be.

$$\mathrm{IC} : \mathcal{C} \to \mathbb{R}^+$$

So for example, 'House Cat' is expected to be more specific than 'Animal' or 'Living Thing', but less specific than 'Sphynx', a specific type of house cat. Therefore 'House Cat' should have a higher score for information content than 'Animal' but a lower score than 'Sphynx'.

With a score like that it becomes possible to find the 'Most Informative Common Ancestor', or MICA for short, of two concepts.

Most Informative Common Ancestor (MICA) describes the ancestor of u and v with the highest Information Content for two concepts $u, v \in \mathcal{C}_{KG}$

$$\begin{aligned} \mathrm{MICA} : \mathcal{C} \times \mathcal{C} &\to \mathcal{C} \\ u, v &\mapsto c \in \{c \mid c \in A(u) \cap A(v) : \max(\mathrm{IC}(c))\} \end{aligned}$$

3.5.2 Information Content Algorithms

In [20] the author divides information content in three general categories:

- Basic Estimators of Information Content
- Extrinsic Information Content
- Intrinsic Information Content

3.5.2.1 Basic Estimators of Information Content

To show an example for an estimator of concept specificity I wanted to showcase the a-preriori-score (APS) mentioned in [46]. The assumption is, that a concept is more specific if it has only few decendents.

$$IC_{\mathrm{APS}}(c) = \frac{1}{|D(c)| + 2}$$

Derived Information Content Algorithms

Considering the shortcomings explained in section 3.7.2, I decided to derive a naive algorithm from $IC_{\text{APS}}(c)$ using both ancestors and decendents.

$$IC_{\text{naive}}(c) = \alpha \cdot \left(1 - \frac{1}{|A(c)| + 1}\right) + (1 - \alpha) \cdot \frac{1}{|D(c)| + 1} \tag{3.3}$$

with $A(c)$ the set of ancestors of c, with $D(c)$ the set of decendents of c and α a weight to favour ancestors or decendent, defaulting to $\alpha = 0.5$.

To better accommodate the tree-like structure of a knowledge graph, I added the logarithm to better approximate the 'level' in which a given concept is.

$$IC_{\log}(c) = \alpha \cdot \left(1 - \frac{1}{\log(|A(c)| + 1) + 1}\right) + (1 - \alpha) \cdot \frac{1}{\log(|D(c)| + 1) + 1} \tag{3.4}$$

with $A(c)$ the set of ancestors of c, with $D(c)$ the set of decendents of c and α a weight to favor ancestors or decendent, defaulting to $\alpha = 0.5$.

3.5.2.2 Extrinsic Information Content

In [41] the author describes the following information content algorithm.

$$IC_{\text{Resnik}}(c) = \log(|D(\top)|) - \log(|\mathcal{I}(c)|)$$

with $D(\top)$ the set of all concepts and $\mathcal{I}(c) = \{a | \langle a, P, c\rangle \in R : P \subset P\}$ the concepts in the knowledge graph pointing at c.

3.5.2.3 Intrinsic Information Content

In [70] the authors describe an algorithm considering the relative number of decendents and the relative depth.

$$IC_{\text{Zhou}}(c) = k \left(1 - \frac{\log(|D(c)|)}{\log(|C|)}\right) + (1 - k)\frac{\log(\text{depth}(c))}{\log(\text{depth}(G_T))}$$

with $D(c)$ the set of decendents of c, C the set of all concepts, $\text{depth}(c)$ the depth of c in the knowledge graph and $\text{depth}(G_T)$ the total depth of the knowledge graph.
Another interesting algorithm was proposed by [44] by incorporating the leaves of the tree.

$$IC_{\text{Sanchez}}(c) = -\log\left(\frac{\frac{|\text{leaves}'(c)|}{|A(c)|} + 1}{|\text{leaves}| + 1}\right)$$

with $A(c)$ the set of ancestors of c, $\text{leaves}'(c)$ the exclusive set of leaves of c (if c itself is a leaf $\Rightarrow \text{leaves}'(c) = \emptyset$) and $|\text{leaves}|$ the number of all leaves of the knowledge graph.

3.5.3 Information Based Algorithms

All measures based on information theoretical analysis depend on the function of information content (IC) and its respective function for the most informative common ancestor (MICA). It is a convention that MICA uses the same IC-function as the similarity algorithm.

Pair-Wise Measures

The pair-wise measures all leverage MICA- and IC-functions to derive a similarity score. The difference is in how those functions are combined

Resnik is the naive measure using information content of the most informative common ancestor.

$$\mathrm{sim}_{\mathrm{Resnik}}(u, v) = \mathrm{IC}(\mathrm{MICA}(u, v))$$

Faith additionally uses the information content of the two concepts in question.

$$\mathrm{sim}_{\mathrm{Faith}}(u, v) = \frac{\mathrm{IC}(\mathrm{MICA}(u, v))}{\mathrm{IC}(u) + \mathrm{IC}(v) - \mathrm{IC}(\mathrm{MICA}(u, v))}$$

Lin published a different approach.

$$\mathrm{sim}_{\mathrm{Lin}}(u, v) = \frac{2 \cdot \mathrm{IC}(\mathrm{MICA}(u, v))}{\mathrm{IC}(u) + \mathrm{IC}(v)}$$

NUnivers is close to $\mathrm{sim}_{\mathrm{Lin}}$ but uses max instead of the average.

$$\mathrm{sim}_{\mathrm{NUnivers}}(u, v) = \frac{\mathrm{IC}(\mathrm{MICA}(u, v))}{\max(\mathrm{IC}(u), \mathrm{IC}(v))}$$

Direct Group-Wise Measures

In this category only the group information content was listed.

Group Information Content computes the relative groupwise information content.

$$\mathrm{sim}_{\mathrm{GIC}}(U, V) = \frac{\sum_{c \in A(U) \cap A(V)} IC(c)}{\sum_{c \in A(U) \cup A(V)} IC(c)}$$

3.6 Measures based on machine learned embedding

As mentioned in section 2.5, the Universal Sentence Encoder, or USE, is well suited for the task of sentence embedding. It results in embedding vectors of 512 numbers. To compare two (embedding) vectors to each other the cosine similarity is often used [69, 37, 64].

$$\begin{aligned} \mathrm{sim}_{\cos} :\ & V \times V \to [-1, 1] \subset \mathbb{R} \\ & \vec{u}, \vec{v} \mapsto \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \cdot \|\vec{v}\|} \end{aligned}$$

To bind the cosine similarity between $[0,1] \subset \mathcal{R}$ a common practice is to calculate its angular similarity [64]. The authors of the Universal Sentence Encoder propose this as well.

> "We find that using a similarity based on angular distance performs better on average than raw cosine similarity." (Cer et al. [10])

$$\begin{aligned} \text{sim}_{\text{angular}} : \; & V \times V \to [0,1] \subset \mathbb{R} \\ & \vec{u}, \vec{v} \mapsto \left(1 - \arccos\left(\frac{\text{sim}_{\cos}(\vec{u}, \vec{v})}{\pi}\right)\right) \end{aligned}$$

3.7 Selection of Semantic Measures

In this section I will briefly explain which of the measures introduced were considered for implementation.

3.7.1 Limitation for Comparability

In order to achieve a good comparability, I have limited myself to algorithms with their image being in the set of destination $[0,1] \subset \mathbb{R}$. Also the feature based similarity $\text{sim}_{\text{Bulskov}}$ was therefore ruled out. Another algorithm that does not satisfy this constraint is $\text{sim}_{\cos}$. So for the comparisons in chapter 7 I just looked at $\text{sim}_{\text{angular}}$ which is derived from $\text{sim}_{\cos}$.

3.7.2 SPARQL Endpoint Timeouts

As discussed further in section 7.2 not all algorithms presented could be implemented on the public SPARQL endpoint to the wikidata knowledge graph. That SPARQL endpoint drops every request exceeding 30 seconds of computing time. This rendered various algorithms not usable for this thesis.

Issues with Functions for Information Content

As foreshadowed in subsection 3.5.1 the algorithms for information content exceed the capabilities of the public SPARQL endpoint.

- For IC_{Zhou} the depth(G_T) cannot be calculated.
- For IC_{Sanchez} the $|\text{leaves}|$ cannot be calculated.
- For IC_{Resnik} the $D(\top)$ cannot be calculated.

The only information content algorithm that is calculable with the public SPARQL-Endpoint is $IC_{\text{APS}}(c)$. This is why I introduced the two IC functions to not exclude another category of similarity measures.

Issues with Path Functions

The SPARQL Query language does not offer a direct way to retrieve the shortest path between to concepts. It is possible to request all paths. However, the number of paths quickly becomes very large and again led to timeouts. This, together with the fact, that graph structure based approaches do not use the semantics of the underlying graph caused me to not further investigate this approach. This excludes both measures based on graph structure analysis: sim_{Rada} and $\text{sim}_{\text{Gentleman}}$.

3.7.3 Preliminary Selection Step

After the exclusion of three similarity measures and three IC functions, I had to determine which remaining functions to use. I decided to keep $IC_{\text{APS}}(c)$, $IC_{\text{naive}}(c)$, $IC_{\log}(c)$ together with all aggregation algorithms. I will compare all functions in a preliminary step to further evaluate the best performing algorithm among those. The best performing algorithms of each category will then be evaluated.

3.8 Summary

This chapter introduced four different categories of semantic measures: graph structure based, graph feature based, information based, and machine learned approaches. Nine pair-wise semantic measures on knowledge graphs and four aggregation methods to calculate group-wise similarities from pair-wise measures were introduced in the non-machine learned categories. Additionally, four direct group-wise measures were presented for the structure, feature, and information based approaches. For the machine learned approach, two measures were introduced. Finally, I explained which algorithms are considered for further investigation and how I will narrow down the number of measures for evaluation.

4 Study of Similarity between Ideas

In this chapter I explain the different steps taken for the study of idea similarities. I describe the proceedings followed and reasoning behind them without diving too deep into implementation details. Those details are further discussed in chapter 5.
The goal of this study is to quantify intuitive similarities between the given ideas and compare them against the measures based on knowledge graphs and the machine learning algorithm USE. It is observable that some ideas are not as easily comparable and therefore a similarity score can only be an approximation. This is why I employ multiple crowd workers and average the results. The study follows the following general steps:

1. Ten ideas are selected for a pair-wise comparison.
2. The ideas are annotated semantically.
3. Crowd-workers are tasked to compare all ideas.
4. Semantic measures are selected and implemented.
5. The similarity scores are calculated using the measures implemented.
6. The semantic measures are compared with the results of the user-study.

4.1 Idea Selection

My advisor and I selected ten ideas from prior studies as mentioned in section 2.1. We settled on ten as a compromise between sample size and comparison effort. The number of comparisons equals $\frac{n \cdot (n-1)}{2}$ with n being the number of ideas compared and intuitive similarities being considered symmetric. In the user-study an average task time of 20 minutes is expected for all 45 comparisons. Initially, we tested 12 ideas. The 66 comparisons took 50% longer than the 45 and we were concerned it would increase fatigue within the task itself[1].
To select the ideas, my advisor and I first considered randomly picking the ideas. After some conceptualizing we decided against it. Instead, we wanted ideas that are explicitly similar or explicitly different to others. We first picked an idea randomly from the 'bionic radar' challenge[2].

> "Analyze athletes to see what part of them is either helping or hurting them in things such as running." (Anonymous Crowd Worker)

[1] As seen in subsection 6.1.2 the ten ideas showed signs of user fatigue already.
[2] 'Bionic Radar' is further explained in section 2.1

We identified *sport* as a domain we could classify this idea in. Based on that we looked for other clear domains we could find within the set of all ideas. We found *family*, *fashion*, *health*, *security*, *sports* & *transport* as often occurring domains within all ideas and decided to look for:

1. Ideas in those domains within the three different challenges
2. Ideas within those domains that achieve their goals with different mechanisms
3. Somewhat similar ideas in the different challenges

I then revised the ideas selected to explicitly mention the challenge they are from.

> "Analyze athletes *with bionic radar* to see what part of them is either helping or hurting them in things such as running."

This was done to lower the barrier for understanding the idea by making implicit information explicit. The resulting ideas and the reasoning behind their selection can be seen in Table 4.1.

4.2 Idea Annotation

As mentioned in section 2.3 all ideas selected must be annotated for the semantic measures on knowledge graphs. Initially I planned on doing the semantic enrichment as a user study as well. In [28] we designed a user study for crowd worker to annotate text. A crowd worker task using that method would yield semantic annotations I could use for comparison, but I could not find a way to average the results of multiple results from crowd workers. For that a semantic measure would be needed to find something comparable to an average of two concepts. So I chose to annotate the ten ideas myself. I used DBpedia Spotlight [13] and the search functionality on Wikidata.

Annotation Process

The annotation process followed the 'manual approach' done in [28]. As seen in Figure 4.1 for every of the ten ideas I first searched for relevant terms using DBpedia Spotlight and my own experience with knowledge graphs. As seen in Figure 4.2 DBpedia Spotlight offers possible annotations and a initial ranking. DBpedia Spotlight is, as the name implies, connected to the DBpedia knowledge graph connects its entries to Wikidata via `owl:sameAs`. Therefore it was possible to find the same concepts in the Wikidata knowledge graph. I then used the best matching concepts and searched for them within Wikidata, as seen in Figure 4.3. For those concepts found I checked for linked concepts that might better describe the term. After I found the best matching annotation I added it to the YAML-File, as exemplary shown in Listing 4.1. The final annotation file can be found in the Appendix section 9.2.
As seen in lines 5 & 6 of Listing 4.1 I decided to include a concept for the challenge itself. I found the following concepts with their Wikidata definitions as the closest for the challenges[3]:

[3]See subsection 2.1.3 for the definitions of the idea challenges.

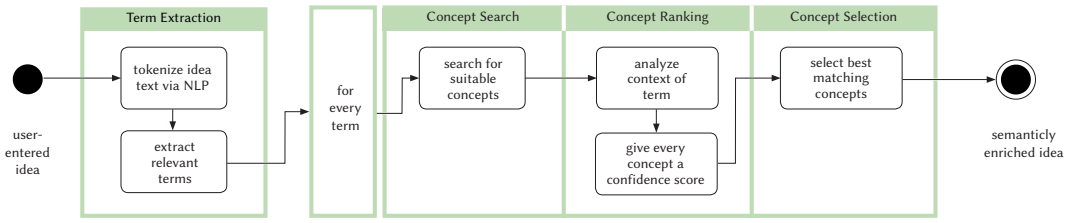
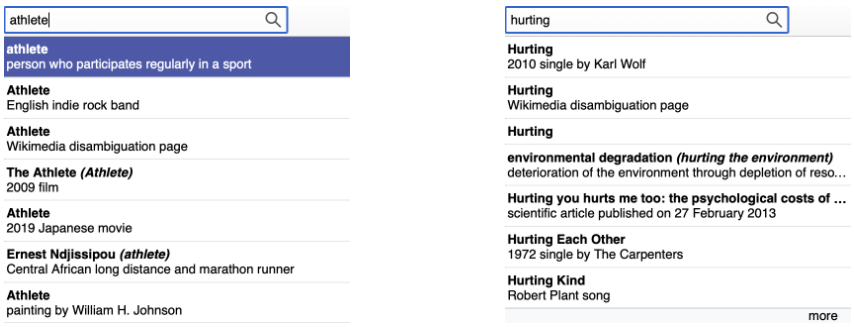


Figure 4.1: Task Flow Diagram of Semantic Enrichment as used in [28].

Figure 4.2: Example DBpedia Spotlight annotation using the first idea. Showing the candidates and their confidence score for the term 'athletes'.



(a) Successful Search (b) Unsuccessful Search

Figure 4.3: Showing the disambiguation interface for searches on Wikidata for the terms 'athlete' and 'hurting' provided by DBpedia Spotlight.

```
content: "Analyze athletes with bionic radar to see what [...]"
concepts:
- concept: "wd:Q2066131"  # Athlete
  token: [1,1]            # athletes
- concept: "wd:Q47528"    # Radar
  token: [3,4]            # bionic radar
[...]
```

Listing 4.1: Idea 01 with two annotated terms (simplified).

Bionic Radar - wd:Q47528 - Radar
Object detection system based on radio waves.

Fabric Display - wd:Q54006339 - Display Technology
Type of technology used for display of text or graphics on a screen.

Transparent Conductive Oxides (TCO) - wd:Q23808 - Interface
Point of interaction between two things.

It could also be feasible to add own entries for those challenges to the knowledge graph. Since I focused on the one general knowledge graph, I leave the linking of different knowledge graphs to future work.

4.3 Gather Intuitive Similarity (User Study)

With those ten ideas selected the plan is to task crowd worker in a user study to quantify the similarity between the ideas. The goal of this user study is to enable statements like the following for every pair out of the ten ideas: *'Idea A' and 'Idea B' have an intuitive similarity of x%.*

$$\text{sim}_{intuitive} : \texttt{Idea}' \times \texttt{Idea}' \to [0,1] \subset \mathbb{R}$$

with $\texttt{Idea}' \subset \texttt{Idea}$ being the finite set of all idea texts from Table 4.1.

4.3.1 Conceptualization & User Interface

Based on previous studies [28] this user study consists of three steps as seen in Figure 4.4. An introduction of the task, the task itself and an epilogue. I decided to

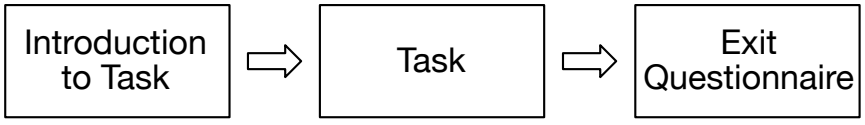


Figure 4.4: Overview of the steps expected from every crowd worker.

follow the Material Design Guidelines [31] as they are well established at the HCC work group.

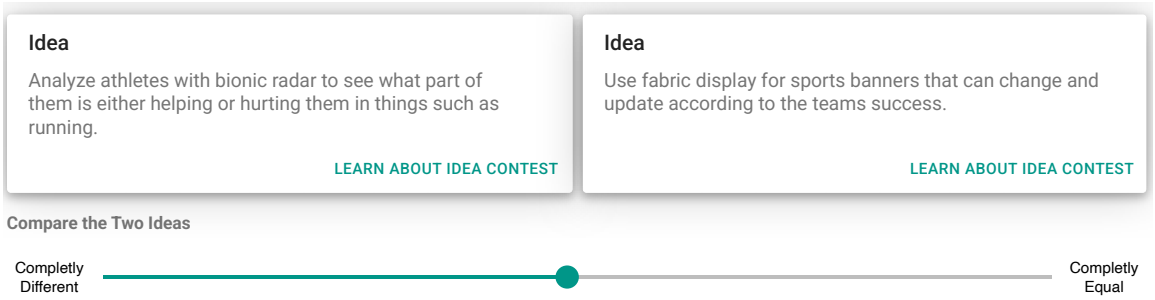


Figure 4.5: Mock-up of a pairwise comparison for ideas 1 and 5 from Table 4.1.

> "Material Design is guided by print design methods — typography, grids, space, scale, color, and imagery — to create hierarchy, meaning, and focus that immerse viewers in the experience." (Material Design Introduction [31])

Screenshots for the introduction, task, and questionnaire can be found in Apendix 9.4.

4.3.1.1 Introduction & Outro

The introduction consists of a general description of the task itself, together with an estimated time and the offered payment[4]. The introduction ended with an example pairwise comparison. It said in the description, that those two ideas are considered more similar and therefore asked the crowd worker to rate it high. If the rating was above 75% the rating bar would turn green and a button to start the task would appear. The rating interface can be seen in Figure 4.5.
After the task was completed a minimal survey was shown. It asked for a clarity rating and optional full-text feedback. My advisor and I decided against an extensive exit questionnaire, like TLX [21], because no different interfaces were planned. We agreed that problems with the interface could be deduced from the feedback form and clarity rating.
The results as shown in chapter 6 and discussed in chapter 7. The feedback and other limitations will be discussed in subsection 6.1.2.

4.3.1.2 Task

The task itself had three mayor components. First, the crowd worker was shown all ten ideas with the explanation of the challenges they are from. After all ideas were read the comparison task started. As seen in Figure 4.5 every idea pair was shown as two cards next to each other, or above on another for small screens. Every card consists of the idea text and a button leading to the definition of the idea contest the idea was from. Some feedback suggested, that the interface was not as clear as expected. This will be further discussed in subsection 6.1.2 and section 7.2.

[4]The estimated time for that task was 20 minutes, conducted by a dry run of the task. Participants on Amazon's Mechanical Turk received $4 per task, resulting in a payment rate of $12/h.

4.3.1.3 Quality Checks

I employed three different quality checks. Two of them were explicitly coded into the task and one was determined by the selection of the ten ideas. The quality checks were not hidden but also had no visual clues to differentiate them from the other comparisons.

Attention Check
The attention check occurred in the middle of the task. The goal was to check whether a crowd worker was reading the ideas or just moving the slider. After the crowd worker finished the 22nd comparison an extra comparison was shown with the same idea on each side. From all 41 tasks submitted 11 did not pass this check.

Consistency Check
The goal was to check if a crowd worker would be coherent over the course of the task. For the consistency check the 3rd comparison was shown again at the end of the task. As research suggest the importance of this consistency check is rather low [68].

> "If the same people are asked the same question in repeated interviews, only about half give the same answers." (Zaller and Stanley [68])

Similarity Check of Idea 5 & Idea 6
As seen in Table 4.1 ideas 6 was specifically chosen to be similar to idea 5. This allowed for $sim(Idea\ 5, Idea\ 6)$ to be somewhat predictable.

Those quality checks enabled me to look at a task and get a first assessment. So if the attention check is not passed I can assume that the other idea pairs are not read thoroughly either. If the similarity check was under 50% or the consistency check answers had a difference of more than 40 percentage points I used that as an indicator to check a submission more in-depth for irregularities or patterns. The application of the quality checks can be found in subsection 6.1.1.

4.4 Gather Semantic Measures

Semantic measures from the categories explained in chapter 3 are gathered for evaluation and comparison with the intuitive similarity.

4.4.1 Semantic Measures on Knowledge Graphs

For semantic measures on knowledge graphs I use the library developed by me. It allows to compare two sets of concepts with each other using the semantic measures introduced in chapter 3. The implementation of the semantic measures is explained in section 4.4. To use the library and gather the results I used the semantic annotations from section 4.2.

4.4.2 Semantic Measures based on USE-Embedding

The TensorFlow models for the universal sentence encoder are freely available at TensorFlow hub [1, 52]. As explained in further detail in section 5.3, the idea texts are converted into vectors using the embed function from the TensorFlow model. As explained in 3.6 a similarity score can than be calculated with $\text{sim}_{\text{angular}}$.

4.5 Calculations on Semantic Measures

The evaluation of the hypotheses mentioned in section 1.2 proved to be more complicated than expected. Many statistical tools are not applicable here. Accordingly, the focus was on correspondence rather than correlation. The calculations done are explained in chapter 6 and discussed in chapter 7.

4.5.1 Correspondence versus Correlation

The initial plan was to test for correlation between the measures and the intuitive similarities. This would have yielded numbers to compare in a very structured way. But looking closer at the data I realized, that a correlation test was not applicable. When looking at a similarity, whether generated by a computer or humans, it results in a matrix of similarity values for every pair. This matrix can also be seen as a vector of similarity pairs. It is important to note that one vector represents only one sample and not more. Accordingly, the sample size would be 2 for each measure. I therefore decided to use other ways to determine correspondence rather than correlation.
As explained in section 3.6 two embedding vectors can be compared using the cosine and angular similarity. As explained there are two vectors for every computed measure. One containing the intuitive similarities per pair, or the ground truth, and one containing the computed similarities per pair. So the naive metric was to apply cosine and angular similarity to the ground truth and the computed similarities [47, 19].

4.5.2 Semantic Measures as Estimators

An alternative view of the data is to consider a similarity measure as an estimator. If again the intuitive similarities are taken as the ground truth, a search for the best estimator becomes possible. A common way to to find the best estimator is by finding the minimal mean squared error [23]. The mean square error, MSE, can be defined as:

$$\begin{aligned} \text{MSE}: \ & V \times V \to \mathbb{R} \\ & \vec{u}, \vec{v} \mapsto \frac{1}{n} \sum_{i=1}^{n} (\vec{u}_i - \vec{u}_i)^2 \end{aligned}$$

with n being the dimension of the vectors and $\vec{v}_i$ the i-th element of $\vec{v}$.

That means, the measure with the minimal $\text{MSE}(\text{GT}, *)$ is the measure best approximating the intuitive similarities. Considering semantic measures as estimators

gave me the opportunity to use a simple estimator as a lower bound. This estimator simply predicts that each idea pair (i, j) has similarity 0.5 with the special case $(i, i) = 1$.

$$\text{sim}_{\text{naive}}(u, v) = \begin{cases} 1.0, & \text{if } u = v \\ 0.5, & \text{otherwise.} \end{cases}$$

This estimator allows to identify bad performing estimators. If a measure generally calculates worse similarities than this naive estimator it provides no added value.

4.6 Summary

In this chapter I explained the six steps needed to study the similarities between the idea set. I illustrated why only ten different ideas were used and stated the reasoning behind the selection of the single ideas. Then, the annotation process needed for the implementation was explained and I described how the semantic measures were gathered. The user study with its quality checks was laid out in detail. An overview of what has to be implemented was given for the semantic measures. Finally I addressed possible measures to evaluate the quality of the different semantic measures.

	Idea Text	Domain	Reason
1	Analyze athletes with bionic radar to see what part of them is either helping or hurting them in things such as running.	Sports	Randomly Picked
2	Use bionic radar to identify fouls or false starts in sporting events.	Sports	Similar to Idea 1, but different mechanism
3	With bionic radar I will know which family member is entering the home.	Security	Mechanism comparable to Idea 1
4	Shippers, drivers and captains lack technical support. Bionic radar could help to achieve an efficient weight distribution on the respective means of transport, to avoid damage to goods or the safe and possibly to save fuel.	Transport	Randomly picked from other domain
5	Use fabric display for sports banners that can change and update according to the teams success.	Fashion, Sports	Same domain as idea 1 but different application
6	A fabric display hat that switches between the colors and logos of all your favorite sports teams.	Fashion, Sports	Similar to idea 5
7	Imagine making flexible masks out of that fabric display and displaying someone else's face on it. Or anything else, any image could be displayed to express emotions or ideas.	Fashion	Same domain and technology as idea 6 but somewhat different
8	With TCO playground equipment may have interactive buttons that allow you to play together.	Family	Somewhat different idea to rest of ideas
9	TCO stickers that can monitor a persons heartbeat, body temperature and other health factors.	Health, Sports	Different Technology with similarities to idea 1
10	Many burglars break into retail stores, offices, or houses by crashing a window. This could be prevented by connecting TCO treated windows to the alarm system.	Security	Different Technology with similarities to idea 3

Table 4.1: Ideas selected for Comparison Study. All ideas are extended so that the challenge is part of the idea text

5 Implementation

This chapter describes the implementation details of the different parts of the study described in chapter 4. The main sections are the user study to gather the manual similarity, the implementation of the semantic measures on knowledge graphs, the gathering of the universal sentence encoder vector embeddings and the gathering of the raw algorithmic similarities.

5.1 User Study: Manual Similarity

In this section I give an overview over the different techniques employed for the user study. Software developed at the HCC will be discussed briefly.

5.1.1 Backend

The backend consists of two parts. To handle the data I adopted batch-manager my advisor used for the ICV process in [28]. This tool offers a REST-API that allows to store the data necessary for a user study. This means the idea pairs as well as the similarities submitted. My advisor helped me in providing the updates necessary to this tool in order to deliver the idea-pairs to the frontend. The batch-manager also offers a Swagger User Interface that allowed for easy access to the different API calls via web.
For the interaction with the crowd workers I used my-turk, a tool developed at the HCC [35]. This tool allowed me to start a user study for a fixed batch of crowd workers and accept or reject the similarities submitted.

5.1.2 Frontend

With the backend in place I was able to program my user study. Because the study frontend was supposed to be embedded as an iframe into the Mechanical Turk interface I had to use web technologies [34]. I decided to use TypeScript because of the optional type system and the higher debuggability that comes with it. For faster development and a higher chance of compatibility I decided to use a frontend framework. Out of preference I chose Vue.js together with the material design framework Vuetify [55, 56, 57].

Layout

The Vue.js frontend consists of four views. The intro, the view showing all ten ideas, the rating task, and the outro. Screenshots of all rendered views are in Apendix 9.4. The source code is available at `https://git.imp.fu-berlin.de/mx-masterarbeit/idea-pair-rating-framework`.

Intro The intro follows the guidelines for crowd tasks my advisor derived from previous studies and [33]. We followed those guidelines before in [28]. A task description explains what assignment the crowd worker has to expect, how much time is estimated for the task, and what the compensation will be. The information for the description comes from the backend. Below the description is a tutorial with associated example. A crowd worker could only progress to the next view by rating the example idea pair between 80% and 90%.

Read All Ideas This view displays all ideas in this assignment. The crowd worker had to check a box saying "I have carefully read all Ideas listed above." to progress to the next view.

Rating Task This view shows one rating pair at a time. Analogous to the example the crowd worker is supposed to use the slider to rate the two ideas presented. After the slider is moved at least once the button to progress to the next task becomes active. If the the last idea pair is rated the button leads to the next view. A bar at the top of the view shows the progress within the task. The quality checks employed are further explained in subsubsection 4.3.1.3. For the implementation I simply inserted the quality check pairs at the necessary position in the array of ideas.

Outro The outro has two purposes. On the one hand it collects meta information about the task and the crowd workers. On the other hand it transforms the data according to the backend so it can all be transmitted via a POST request by clicking the finish button.

5.1.3 Results

The results of this study were accessible via the swagger interface. After I downloaded the results I had do the quality checks and approve or reject each individual worker. I used a jupyter notebook to evaluate the given results and approved or rejected them using my-turk. for

5.2 Wikidata Semantic Measure Toolkit

The Wikidata Semantic Measure Toolkit (wdsmt) offers tools to access entries on wikidata. It was written in Python 3.9 and set up using object oriented paradigms where useful. At this time it only supplies the libraries necessary to use it as a library to interact with the Wikidata knowledge graph. The use with other knowledge graphs is not tested but was thought of at time of writing this. It should be possible to use the library with other knowledge bases such as DBpedia. An extension of this toolkit to support usage as a CLI to calculate the similarities is planned.

```
SELECT DISTINCT ?decendent
WHERE {
VALUES ?pre {
  wdt:P31
  wdt:P279
}
?item (wdt:P31 | wdt:P279)* wd:Q35120.
?decendent ?pre ?item.
}
```

Listing 5.1: SPARQL Query: Decendents of Entity

5.2.1 Timeouts, AsyncIO & Caching

During initial testing I found that many SPARQL-Requests result in a timeout after 30 seconds. Queries asking for to many ancestors or to many descendants will not finish. This happens mostly when looking for decendants of concepts close to the root $\top$. Such a query can be seen in Listing 5.1 Especially when calculating most informative common ancestor[1]this became an issue. Calculating the MICA for one concept pair sequentially could take several minutes. With that in mind I looked into parallelization. I found the async await pattern [39] as the most useful. In theory this could limit the maximum execution time of a request to 30 seconds. In practice this was not the case. The public interface to Wikidata is rate limited and only allows for about 5 parallel connections from one IP-address at a time. So I had to use semaphores and the execution time increased drastically again. The next solution I found was caching. Due to the fact that, for MICA, I look at all concepts in the ancestor list of both ideas of the pair it is to be expected, that the same concepts at the top of the tree are looked at the most. As mentioned before those are the concepts that result in requests that time out. So I decided to cache every request and start my similarity collection with a warm up phase in which the information content for every concept was loaded into a Redis cache.

5.2.2 Layout

The library consists of seven classes in the package `wdsmt.classes`. Matching the best practices every class has its own file where the filename corresponds with the class name. Future work includes the plan of adding aggregating classes or command-line interfaces to the top level of that package.

Concepts provides the internal data type for all similarities. To handle timeouts from the endpoint an 'infinity' concept was added. If a request times out it returns said infinity concept. Looking at the information content a concepts can be considered less informative than concepts that do not time out. Therefore the infinity concept sets them apart but still keeps a mathematical value for calculations. Unions, intersections and set differences are handled accordingly.

[1]For definitions of Information Content & MICA and other algorithms see chapter 3.

```
async with GraphAccessor(predicates) as graph:
    ancestors = await graph.ancestors(concept)
    ancestor_count = await graph.ancestor_count()

    decendents = await graph.decendents(concept)
    decendent_count = await graph.decendent_count()
```

Listing 5.2: Example Usage of GraphAccessor

A set containing the infinity concept has a length of ∞. Because of that the *magic function* `__len__()` cannot be defined and `length()` has to be used.

GraphAccessor is the foundation of this library. Within that class are all functions implemented that are needed to access the knowledge graph. All SPARQL-Queries are handled throughout this class. A GraphAccesseor needs to be instantiated with the predicates considered for vertical movement and the endpoint. The default predicates are 'instance of (wdt:P31)' and 'subclass of (P279)' together with the default endpoint `wikidata`. An exemplary use of the main functions provided can be seen in Listing 5.2. Only this class does any caching.

FeatureBasedSimilarity implements the pair-wise similarities discussed in section 3.4. This class combines the GraphAccessor with the functions provided by Concepts to calculate its similarities. The function signatures follow all: `async def sim( u: Concept, v: Concept)-> float`

InformationContent implements functions to calculate the information content as described in subsection 3.5.2. Due to limitations introduced by the public wikidata endpoint only the functions discussed in section 3.5.2.1 are implemented. An information content function has the form: `async def ic(c: Concept)-> float`.

InformationBasedSimilarity implements the pair-wise similarities discussed in section 3.5. This class relies on information content to calculate the most informative common ancestors. Because of the rate limit constraints of the public wikidata endpoint a semaphore is implemented to default to a maximum of four parallel requests. After instantiation the function signatures follow: `async def sim(u: Concept, v: Concept)-> float`

DirectGroupwiseSimilarity implements the direct group-wise measures discussed in section 3.1. For the information based direct group-wise measure a function for the information content is expected as third parameter.

IndirectGroupwiseSimilarity implements the indirect group-wise measures discussed in section 3.1. In contrast to other similarity classes this class does not need to be initiated. It just aggregates to groups of concepts as two lists and a pair-wise similarity function to apply. All functions in this class have the same basic signature: `async def group_sim(U: [Concept], V: [Concept], sim: function)-> [float]`

In further refactoring I would combine both group-wise classes into a single class.

5.2.3 Dependencies

The library needs at least Python 3.9 to be installed. For caching redis 6 is running locally. The python packages installed are:

- `numpy` for scientific computing and array handling
- `aiosparql` as an asynchronous SPARQL Wrapper
- `aiocache[redis]` for asynchronous caching via redis

According to best practices the python dependencies are listed in the *requirements.txt*.

5.2.4 Next Steps

The next steps for this library are first of all the unification of the interfaces. A mayor goal for consistency is to make all similarity classes usable without instantiation as 'IndirectGroupwiseSimilarity' is already. This would make the code cleaner, easier to understand and easier to use. Another area for improvement is the caching. An option do load without caching and an interface to bust specific parts of the cache are missing. The caching as it is also hinders test coverage. Another step could be to elevate this library into a commandline interface.

5.3 Universal Sentence Encoder

The universal sentence encoder is employed using tensorflow. Similar to the data collection with the Wikidata Semantic Measure Toolkit I uses jupyter notebooks to collect the word embeddings and calculate the similarities. A minimal working example of similarities derived from embeddings from universal sentence encoder can be seen in Listing 5.3.

5.4 Results

The results collected for this thesis were mainly done with python. When possible I used Jupyter notebooks [38]. However, since Jupyter notebooks have limited compatibility with `async/await`, I collected semantic measures on knowledge graphs using simple python scripts. For the similarities I chose a matrix for storage, where the indices correspond to the idea number[2]. The similarities together with its metadata was stored in a json file for future usage. The simplified JSON-File belonging to $\text{sim}_{\text{BMA-CMatch}}$ is shown in Listing 5.4.

[2]Python is zero indexed, but the idea numbering starts at one so the index of idea i is $i - 1$.

```
import tensorflow as tf
import tensorflow_hub as hub
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity

def angular_similarity(embedding_a, embedding_b):
    cos_sim = cosine_similarity(embedding_a, embedding_b)
    return 1 - (np.arccos(cos_sim) / np.pi)

use = "https://tfhub.dev/google/universal-sentence-encoder-large/5"
embed = hub.load(use)
embeddings = embed(idea_texts)
print(cosine_similarity(embeddings, embeddings))
print(angular_similarity(embeddings))
```

Listing 5.3: Minimal Working Example for Similarities of Embedding Vectors

```
{
    "_meta": {
        "aggregator": "best_match_average",
        "category": "feature-based",
        "name": "CMatch"
    },
    'similarity': [
        [1.   0.7  0.66 0.75 0.69 0.52 0.66 0.7  0.78 0.67]
        [0.7  1.   0.8  0.71 0.79 0.54 0.6  0.68 0.75 0.74]
        [0.66 0.8  1.   0.7  0.66 0.58 0.62 0.66 0.76 0.71]
        [0.75 0.71 0.7  1.   0.67 0.51 0.57 0.58 0.69 0.6 ]
        [0.69 0.79 0.66 0.67 1.   0.53 0.59 0.67 0.72 0.68]
        [0.52 0.54 0.58 0.51 0.53 1.   0.74 0.76 0.49 0.56]
        [0.66 0.6  0.62 0.57 0.59 0.74 1.   0.76 0.67 0.62]
        [0.7  0.68 0.66 0.58 0.67 0.76 0.76 1.   0.72 0.71]
        [0.78 0.75 0.76 0.69 0.72 0.49 0.67 0.72 1.   0.74]
        [0.67 0.74 0.71 0.6  0.68 0.56 0.62 0.71 0.74 1.  ]
    ]
}
```

Listing 5.4: Simplified JSON File Containing a Similarity Measure

5.5 Summary

In this chapter I explained in detail the different implementation tasks done. The main focus was on the Wikidata Semantic Measure Toolkit which in its current form is a library for interacting with the public Sparql endpoint. Furthermore, the user study was described in detail. The source code for all implementations and evaluations can be found at https://git.imp.fu-berlin.de/mx-masterarbeit.

6 Results

In this chapter I will show the results of the user study and its quality checks, describe the preliminary selection process of the computed semantic similarities and create the graphs used for evaluation. For the following chapters I prepend the name of the aggregation method to the indirect measure. So the maximum average aggregation of $\text{sim}_{\text{CMatch}}$ is written as $\text{sim}_{\text{maxAVG-CMatch}}$.

6.1 User Study Results

The user study yielded as a result the different similarities of the individual users and consequently the average similarity. I first describe the results of the quality checks and further calculations on the results. Then discuss the feedback and clarity ratings.

6.1.1 Intuitive Similarities

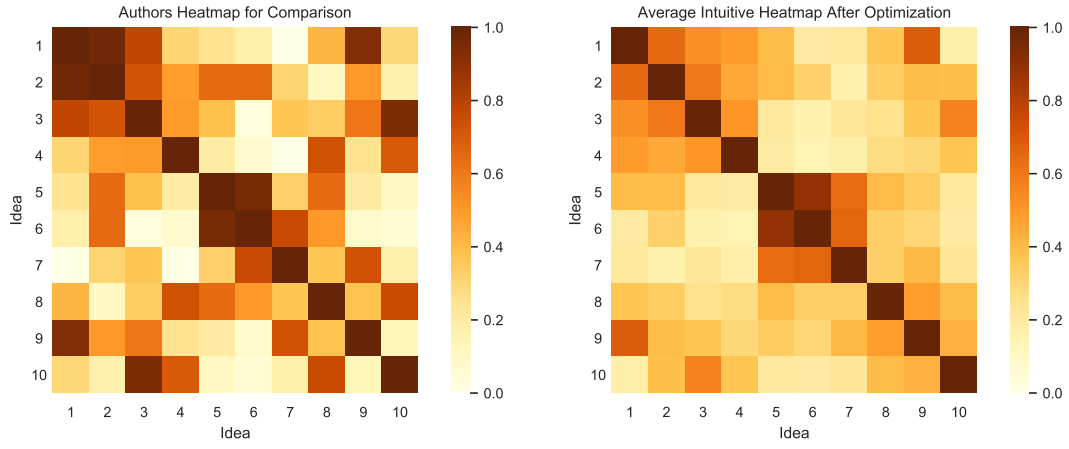
The user study consisted of one initial batch with five challenges and four batches with nine challenges each. For the resulting 41 similarity maps I looked at the 'attention check' and 'quality check'
Initially I found 24 responses as acceptable and usable. For six the attention check passed but the quality check diverged from what I was expecting. I looked at the heat-maps of those and decided to lower the threshold for the quality check. Out of those seven responses two were usable then. The other four scored the similarity of the quality check under 30% and were therefore rejected. Eight responses did not pass the attention check. For two responses was the the data somewhat corrupted and I could not even apply the quality checks.

Minimizing Variation

This resulted in 26 similarities. For the now 26 similarities I looked at the double check. I decided to optimize for coefficient of variation [8]. As seen in Table 6.1 excluding those responses from the results that changed in the double check test more than 50% minimizes coefficient of variation and variance. I therefore settled on using in total 20 similarities by crowd workers were used in this study. As seen in Figure 6.1 the averaged results differ from my evaluation result. It is important to note that the heat map is symmetrical along the diagonal. Some similarities remain, for example, the pair (1,9) and the pair (3,8) are elaborated in both heat maps. The similarity for the pair (1,2) is significantly lower in the averaged perception than in mine.

Results	Variation	Variance	Std Deviation	DC Diff
26	0.692	0.057	0.222	84.75%
25	0.670	0.054	0.217	80.25%
24	0.664	0.054	0.217	78.25%
23	0.640	0.052	0.211	69.25%
22	0.635	0.051	0.210	54.50%
21	0.645	0.052	0.213	51.25%
20	0.630	0.041	0.212	43.75%
19	0.630	0.049	0.206	37.25%
18	0.633	0.051	0.209	35.00%
17	0.656	0.050	0.207	28.50%
16	0.661	0.050	0.206	28.00%

Table 6.1: Table showing the consequences of lowering the number of results by eliminating those results with the highest double check difference 'DC Diff'. 'Results' is the number of results in the remaining set, 'Variation' is the coefficient of variation, 'Variance' is the variance, 'Std Deviation' describes the standard deviation, and 'DC Diff' the highest double check difference still in the remaining set.



(a) Evaluation Heat-map

(b) Crowd Worker Average

Figure 6.1: Heat-maps showing the similarities used for evaluation and quality checks in (a) and the averaged crowd worker results in (b).

6.1.2 Feedback

From the 20 results used 8 were submitted with a feedback. Five of those were crowd worker saying thank you. The remaining six might give some insight for improvement in further studies.

Usability The responses regarding usability might show some drawbacks with the interface. Especially feedback

F1. An explanation of TCO would have helped.

F2. It would be nice to see the actual number of our rating when we drag the bar on either side.

Monotony The responses regarding the monotony of this task was somewhat expected. As explained in section 4.1, the ten ideas were a compromise between data amount and task load. It is noteworthy that only one crowd worker has made a comment in that direction.

F3. I enjoyed doing this task, but it might be nice to have more ideas to rate.

Insufficient Explanation Three answers described the insufficient amount of examples in the intro section. Particularly noteworthy is the reference to the lack of an example of less similar ideas.

F4. Hard to determine many of these. But thanks for the opportunity!

F5. I think there should be more practice examples to fully understand what possible pair ratings could be

F6. Interesting task, should have example of less connected ideas

Table 6.2 shows the meta data received ordered by the clarity rating. As a side note, it should be noted that a striking number of tossed out responses praised the study. Perhaps a further quality check can be derived from this.

6.1.3 Clarity

The clarity ratings range from 1 to 4 with 3 being the most given score (60%). As seen in Table 6.2 the clarity is decoupled from the feedback with 'F5.' being the only exception. A possible explanation might be that 3 was the option in the middle.

6.2 Preliminary Selection of Computed Semantic Similarities

As mentioned in section 3.7 I initially computed all semantic measures. In a preliminary step I compared all computed semantic measures with the average intuitive measure using the mean square error and the cosine similarity. As explained in section 4.5 I looked for measures where the MSE was minimal and the cosine similarity was greatest. In Figure 6.2 the MSE of all semantic measures on knowledge graphs with the average intuitive measure is shown. In Figure 6.3 the cosine similarity of all semantic measures on knowledge graphs with the average intuitive measure is shown.

Clarity	DC Diff	QC Sim	Feedback
1	13.75%	100.00%	I think there should be more practice examples..
1	26.25%	74.50%	
2	43.75%	99.75%	
3	03.50%	53.50%	Hard to determine many of these. But thanks for..
3	04.00%	85.75%	Interesting task, should have example of less..
3	05.25%	79.00%	
3	07.00%	68.00%	
3	01.00%	93.25%	It would be nice to see the actual number of..
3	13.75%	97.00%	
3	16.50%	92.75%	An explanation of TCO would have helped.
3	25.25%	94.00%	
3	28.00%	98.25%	
3	28.50%	100.00%	
3	35.00%	97.25%	
3	37.25%	95.25%	
4	08.50%	69.50%	
4	13.75%	94.25%	I enjoyed doing this task, but it might be nice..
4	20.75%	93.25%	
4	21.00%	95.50%	
4	25.50%	100.00%	

Table 6.2: Collected Meta Data: Clarity rating together with the double check difference 'DC Diff' and quality check similarities 'QC Sim' for the idea pair (5,6). The 'Feedback' attributed to the corresponding crowd worker.

The machine learned approach using the universal sentence encoder is present as a reference.

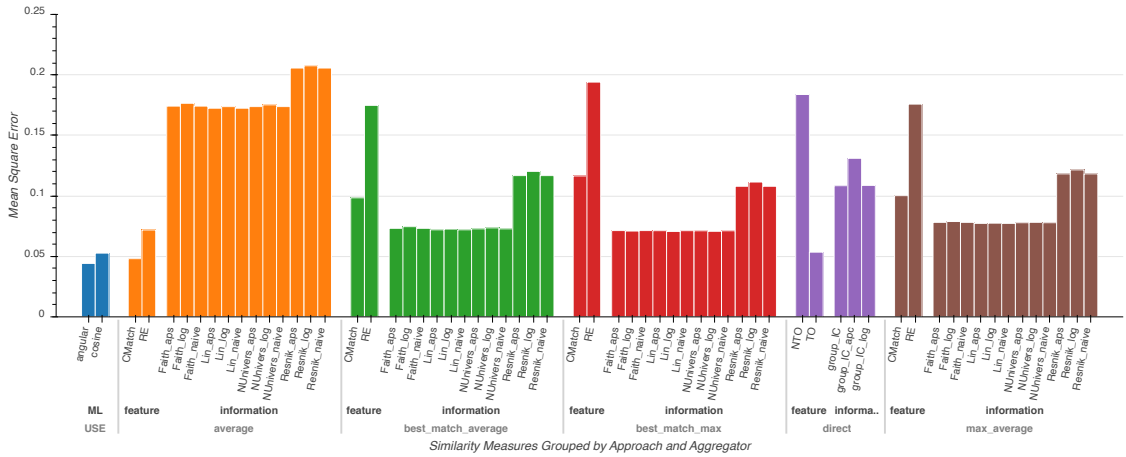


Figure 6.2: Plot showing the mean square error of every semantic measure to the intuitive similarity measures. The measures are grouped by approach and aggregation function. A lower mean square error implies higher similarity.

6.2.1 Direct Approach

For the five different direct approaches only one is worth considering. Looking at Table 6.3 the term overlap sim_{TO} outperforms the others in mean square error as well as in cosine and angular similarity. The normalized term overlap sim_{NTO} has the worst mean square error but performs better in cosine and angular similarity than the direct information based approaches. Within the information based approaches IC_{naive} and $\text{IC}_{\log}$ both perform slightly better than IC_{APS} for sim_{GIC}. Only sim_{TO} will be further evaluated.

Semantic Measure	MSE	COS	ANG
sim_{TO}	0.053	0.931	0.881
sim_{NTO}	0.184	0.903	0.859
$\text{sim}_{\text{GIC,naive}}$	0.108	0.755	0.772
$\text{sim}_{\text{GIC,log}}$	0.109	0.755	0.772
$\text{sim}_{\text{GIC,apc}}$	0.131	0.694	0.744

Table 6.3: All direct semantic measures on knowledge graph compared by mean square error ‘MSE’, cosine similarity ‘COS’ and angular similarity ‘ANG’.

6.2.2 Indirect Approach

In the indirect approach, as explained in section 3.1, an aggregation algorithm was applied to pair-wise semantic measures. With two feature based pair-wise measures, four information based pair-wise measures, three different IC functions, and four aggregation algorithms I ended up with $(2+3\cdot 4)\cdot 4 = 56$ different similarity measures. A cursory glance at the Figures 6.2 & 6.3 show their overall relatedness.

6.2.2.1 Feature Based

For the indirect feature based approach stands out that the average aggregation minimizes the mean square error but does not yield the highest cosine similarity within that category. As seen in Table 6.4, the $\text{sim}_{\text{CMatch}}$ provides the best results in that category across all aggregation algorithms. The minimal mean square error is 0.048 for the AVG aggregation, $\text{sim}_{\text{AVG-CMatch}}$, and the maximum cosine similarity is 0.924 for the BMA aggregation, $\text{sim}_{\text{BMA-CMatch}}$. I kept both for further evaluation.

Semantic Measure	MSE	COS	ANG
$\text{sim}_{\text{AVG-CMatch}}$	0.048	0.899	0.856
$\text{sim}_{\text{maxAVG-CMatch}}$	0.100	0.922	0.874
$\text{sim}_{\text{BMA-CMatch}}$	0.098	0.924	0.875
$\text{sim}_{\text{BMM-CMatch}}$	0.116	0.921	0.873
$\text{sim}_{\text{AVG-RE}}$	0.072	0.888	0.848
$\text{sim}_{\text{maxAVG-RE}}$	0.176	0.904	0.859
$\text{sim}_{\text{BMA-RE}}$	0.175	0.905	0.860
$\text{sim}_{\text{BMM-RE}}$	0.194	0.903	0.858

Table 6.4: All indirect feature based semantic measures on knowledge graph compared by mean square error 'MSE', cosine similarity 'COS' and angular similarity 'ANG'.

6.2.2.2 Information Based

The indirect approaches differ very little within each category, as seen in Figure 6.2 and 6.3. Therefore, I only examined the best performing aggregation algorithm, the average algorithm sim_{avg}. With looking at Table 6.5, it became apparent that $\text{sim}_{\text{BMM-Lin,log}}$ provided the minimal mean squared error results across all information based aggregation algorithms. $\text{sim}_{\text{BMM-Resnik,log}}$, while having the highest mean squared error, resulted in the best cosine and angular similarity. It should be noted that the differences are very marginal. I kept $\text{sim}_{\text{BMM-Lin,log}}$ and $\text{sim}_{\text{Resnik,log}}$ for further evaluation.

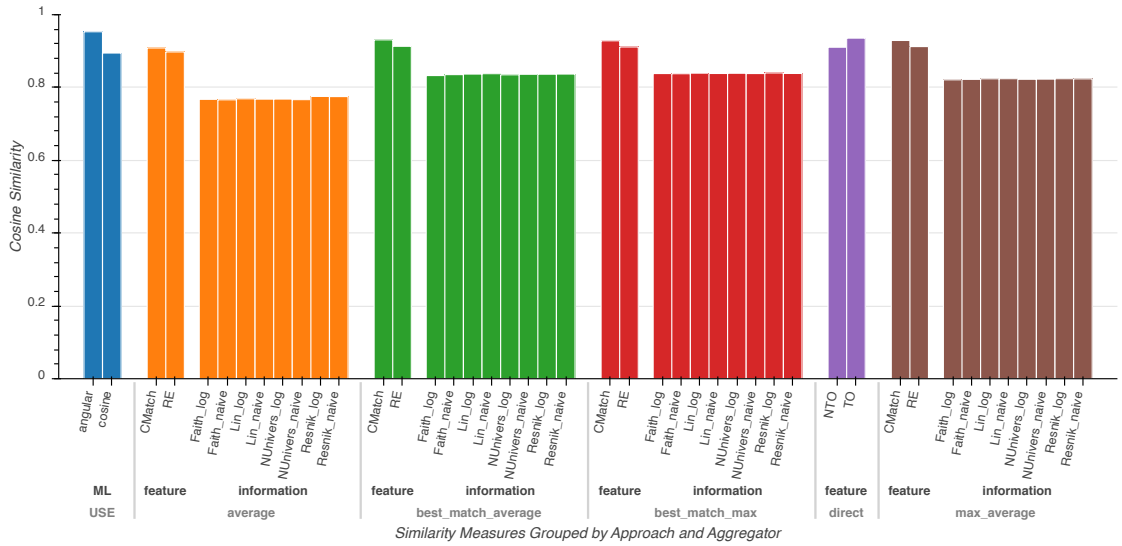


Figure 6.3: Plot showing the cosine similarity to the intuitive similarity measures. The measures are grouped by approach and aggregation function. A higher cosine similarity implies higher similarity.

Semantic Measure	MSE	COS	ANG
$sim_{\text{BMM-Faith,APS}}$	0.071	0.846	0.821
$sim_{\text{BMM-Faith,naive}}$	0.071	0.846	0.821
$sim_{\text{BMM-Faith,log}}$	0.071	0.847	0.822
$sim_{\text{BMM-Lin,APS}}$	0.071	0.847	0.821
$sim_{\text{BMM-Lin,naive}}$	0.071	0.847	0.821
$sim_{\text{BMM-Lin,log}}$	0.071	0.848	0.822
$sim_{\text{BMM-NUnivers,APS}}$	0.071	0.847	0.821
$sim_{\text{BMM-NUnivers,naive}}$	0.071	0.847	0.821
$sim_{\text{BMM-NUnivers,log}}$	0.071	0.848	0.822
$sim_{\text{BMM-Resnik,APS}}$	0.108	0.847	0.821
$sim_{\text{BMM-Resnik,naive}}$	0.108	0.847	0.821
$sim_{\text{BMM-Resnik,log}}$	0.111	0.848	0.822

Table 6.5: Indirect information based semantic measures on knowledge graph aggregated by the best match max algorithm (BMA) compared by mean square error ‘MSE’, cosine similarity ‘COS’ and angular similarity ‘ANG’.

6.2.3 Selected Semantic Algorithms on Knowledge Graphs

The algorithms selected for further evaluation are sim_{TO}, $\text{sim}_{\text{AVG-CMatch}}$, $\text{sim}_{\text{BMA-CMatch}}$, $\text{sim}_{\text{Lin,log}}$, $\text{sim}_{\text{Resnik,log}}$. They are listed together with $\text{sim}_{\text{USE,angular}}$ in Table 6.6.

Name	Approach	Aggregator	MSE	COS	ANG
sim_{TO}	FB	-	0.053	0.931	0.881
$\text{sim}_{\text{AVG-CMatch}}$	FB	AVG	0.048	0.899	0.856
$\text{sim}_{\text{BMA-CMatch}}$	FB	BMA	0.098	0.924	0.875
$\text{sim}_{\text{Lin,log}}$	IB	BMM	0.071	0.848	0.822
$\text{sim}_{\text{Resnik,log}}$	IB	BMM	0.111	0.848	0.822
$\text{sim}_{\text{USE,angular}}$	ML	USE	0.044	0.948	0.897

Table 6.6: All different semantic algorithms selected for evaluation compared by 'Approach', aggregation algorithm 'Aggregator', mean square error 'MSE', cosine similarity 'COS' and angular similarity 'ANG'.

6.3 Timings

Considering the time that is required to go from a set of idea texts to a map of similarities, major differences are apparent. Especially algorithms based on the $D(c)$ have the potential to generate timeouts. Therefore, these algorithms require orders of magnitude more time to compute. I had requests for information based similarities that took several hours including cache warming, where feature based similarities took about ten to twenty minutes. For the universal sentence encoder the timings are around five to ten minutes, not considering the download of the embedding. Because of hardware issues it was not possible to provide exact timings.

6.4 Summary

This chapter presented the results of the user study, including intuitive similarities, quality checks, and user feedback. It also explained the process of finding the best average of the individual intuitive similarities. In the second section the preliminary selection step was explained. Therefore, all generated semantic measures were compared and the best performing were selected for evaluation. The next chapter will evaluate the semantic measures in relation to the research question.

7 Evaluation

This chapter presents the evaluation of the results obtained in the previous chapter. First, the research hypotheses are evaluated and, based on this, the research question is discussed. Limitations and research contributions are also featured in this chapter.

7.1 Evaluation of Hypotheses

To evaluate the Hypotheses and finally the research question I used the metrics as discussed in section 4.5. As already explained, correlations cannot be determined. To derive a sense of similarity I used the euclidean distance, mean square error, cosine and angular similarities between the ground truth and the calculated similarities for every pair: sim_{TO}, $\text{sim}_{\text{AVG-CMatch}}$, $\text{sim}_{\text{BMA-CMatch}}$, $\text{sim}_{\text{Lin,log}}$, $\text{sim}_{\text{Resnik,log}}$, $\text{sim}_{\text{USE,angular}}$, and the sim_{naive}, the naive estimator described in subsection 4.5.2.
I also used a winner takes all metric to compare two measures to a third one. This metric is sort of a signum function for the distance of two vectors and a ground truth.

$$\text{WTA}_{\vec{GT}} : \; V \times V \to \times V$$

$$\vec{u}, \vec{v} \mapsto \text{WTA}_{\vec{GT}}(u,v)_i = \begin{cases} \quad 1, & \text{if } \vec{GT}_i - \vec{u}_i > \vec{GT}_i - \vec{v}_i \\ \quad 0, & \text{if } \vec{GT}_i - \vec{u}_i = \vec{GT}_i - \vec{v}_i \\ - \quad 1, & \text{if } \vec{GT}_i - \vec{u}_i < \vec{GT}_i - \vec{v}_i. \end{cases}$$

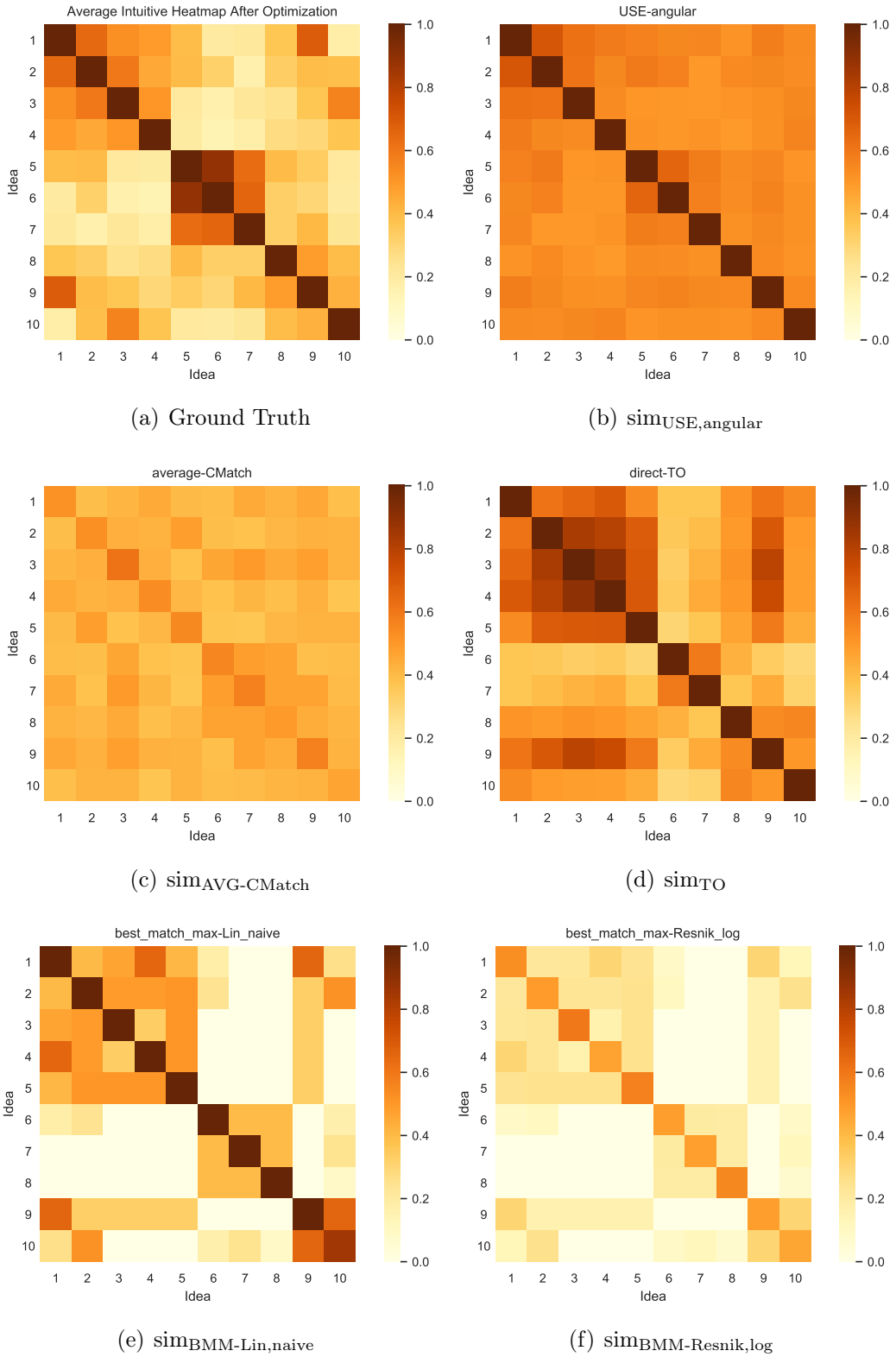
with $\vec{GT}$ being the ground truth.

This allows me to compare the number of times an algorithm is closer to the ground truth than another algorithm.

7.1.1 Evaluation of Hypothesis $\mathscr{H}_1$

A semantic measure exists that corresponds with human intuition, the 'ground truth'.

As explained in section 2.5 embeddings using the universal sentence encoder are well suited for comparison of texts. With the lack of clear goal values for correspondence I decided to compare the measures to USE. I used a visual approach together with the distances explained in subsection 4.5.1 and 2.5. First I created a heat map for every measure to test. As seen in Figure 7.1 all heat maps show high values on the diagonal. An exception is $\text{sim}_{\text{AVG-CMatch}}$. As mentioned in section 3.1, the aggregation algorithm 'average' does not guarantee $\text{sim}_{\text{AVG}}(U,U) = 1$. All other measures have very high reflexive similarities. That led me to the creation of another graph. It plots the similarity values of the ground truth, $\text{sim}_{\text{USE,angular}}$, and one measure on a graph. I used the vector form of the similarity measures. To increase understandability I



(a) Ground Truth

(b) $\text{sim}_{\text{USE,angular}}$

(c) $\text{sim}_{\text{AVG-CMatch}}$

(d) sim_{TO}

(e) $\text{sim}_{\text{BMM-Lin,naive}}$

(f) $\text{sim}_{\text{BMM-Resnik,log}}$

Figure 7.1: Heat maps of computed similarities. (a) Ground Truth is shown as a reference.

omitted all reflexive pairs (i, i) and and used the averaged similarity score for every symmetric pair $\frac{sim(i,j)+sim(j,i)}{2}$. This resulted in the five graphs showing 45 similarity pairs. Exemplary shown in Figure 7.2 and Figure 7.3 are the graphs for $\text{sim}_{\text{AVG-CMatch}}$ and $\text{sim}_{\text{BMM-Lin,naive}}$. The other three graphs are shown in Appendix 9.3.

Information Based Algorithms

The first observation I made was that both information based algorithms had many outliers. For $\text{sim}_{\text{BMM-Resnik,log}}$ and $\text{sim}_{\text{BMM-Lin,log}}$ 19 similarities were 0%. Looking at the differences $\text{sim}_{\text{BMM-Resnik,log}}$ was closer to the ground truth for 20 pairs, 13 if the 0% similarities are omitted. $\text{sim}_{\text{BMM-Lin,log}}$ was closer to the ground truth for 25, 17 if the 0% similarities are omitted. The fact that such a high percentage of unusable results were provided by both information based measures[1] rendered them inadequate for verify hypothesis $\mathscr{H}_1$. Since both algorithm sill performed relatively well they hold potential for future investigations

Feature Based Algorithms

None of the three algorithms in that category had such an amount of extremes as the information based ones. sim_{TO} had one similarity value at 90% and two additional values higher than 80%[2]. As seen in Table 7.1, both $\text{sim}_{\text{AVG-CMatch}}$ and $\text{sim}_{\text{BMA-CMatch}}$ had no extremes. For $\text{sim}_{\text{AVG-CMatch}}$ the highest value was under 50% similarity and the lowest score was over 35% similarity. For $\text{sim}_{\text{AVG-CMatch}}$ the highest value was under 90% similarity and the lowest score was over 50% similarity. In comparison $\text{sim}_{\text{USE-angular}}$ had a highest similarity of 71% and a lowest similarity of 30%. The averaged intuitive similarities had a highest similarity of 89% and a lowest similarity of 15%.
In direct comparison based on figures xy the difference of sim_{TO} to the ground truth was closer in 26 occurrences than the difference of $\text{sim}_{\text{USE-angular}}$ to the ground truth. The difference of $\text{sim}_{\text{AVG-CMatch}}$ to the ground truth was closer in 36 occurrences than the difference of $\text{sim}_{\text{USE-angular}}$ to the ground truth. The difference of $\text{sim}_{\text{BMA-CMatch}}$ to the ground truth was closer in 5 occurrences than the difference of $\text{sim}_{\text{USE-angular}}$ to the ground truth. For all idea pairs the winner-takes-all scores are listed in Table 7.2. Those winner-takes-all results against $\text{sim}_{\text{USE,angular}}$ suggest that $\text{sim}_{\text{AVG-CMatch}}$ and sim_{TO} might verify hypothesis $\mathscr{H}_1$. To further test them I looked at the winner-takes-all results against the naive estimator, $\text{sim}_{\text{naive}}$. As seen in Table 7.2 the $\text{sim}_{\text{AVG-CMatch}}$ is the only semantic measure on knowledge graph being better than the naive estimator. Therefore, $\text{sim}_{\text{AVG-CMatch}}$ is the only semantic measure on knowledge graphs that verifies hypothesis $\mathscr{H}_1$. One drawback is, that the diagonal of the averaged CMatch measure is always $\text{sim}_{\text{AVG-CMatch}}(i, i) < 0$. For further evaluation I added a naive solution as $\text{sim}^*_{\text{AVG-CMatch}}$ where per definition reflexivity holds.

[1] The problem in too many zero values was found in all information based algorithms.
[2] While excluding the 100% similarities for reflexive pairs.

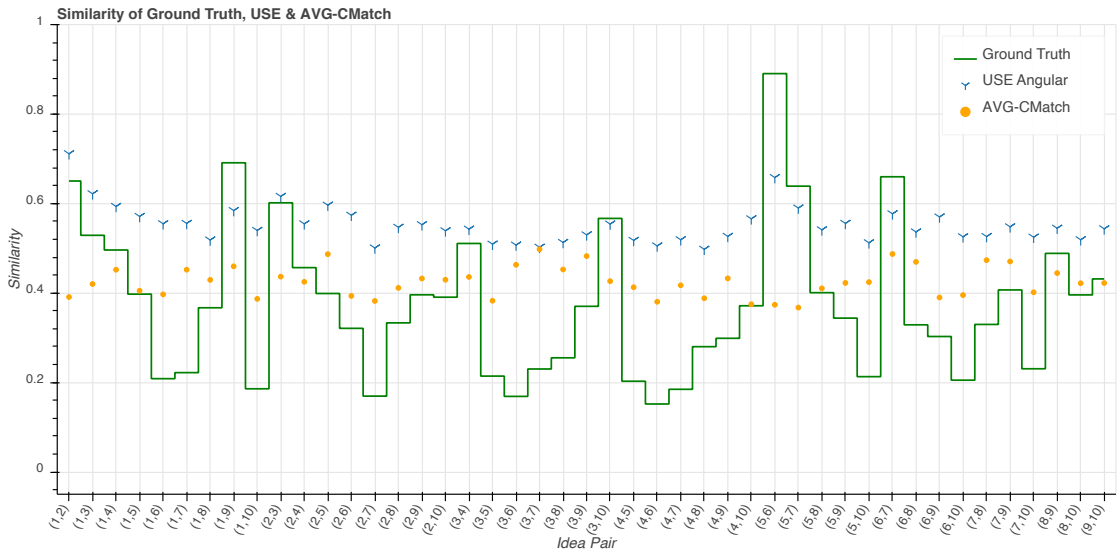


Figure 7.2: Plot comparing $\text{sim}_{\text{AVG-CMatch}}$ with $\text{sim}_{\text{USE,angular}}$ and the ground truth.

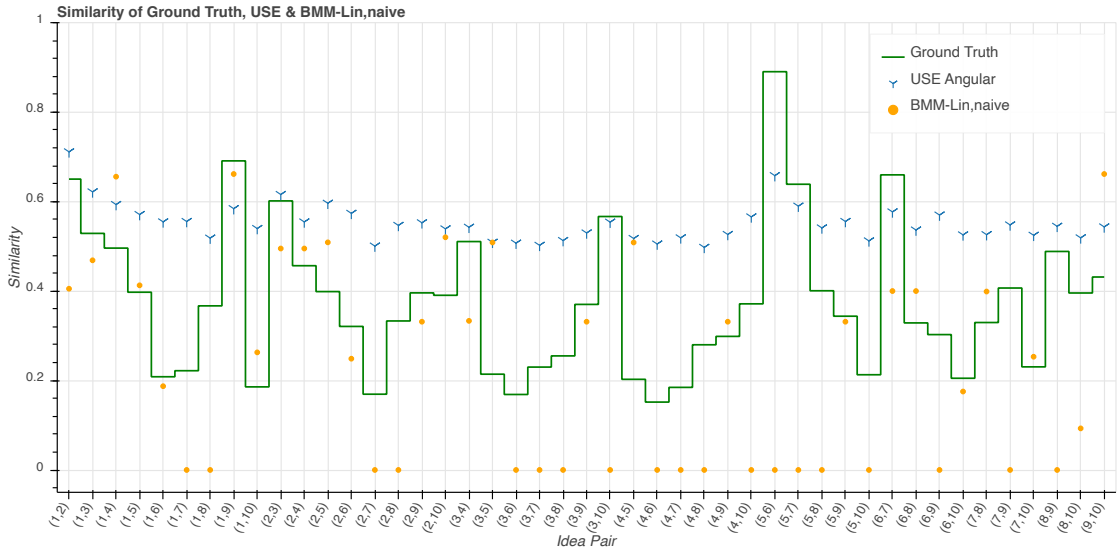


Figure 7.3: Plot comparing $\text{sim}_{\text{BMM-Lin,naive}}$ with $\text{sim}_{\text{USE,angular}}$ and the ground truth.

Similarity Measure	Mean	STD	Min	Max
Ground Truth	0.375	0.166	0.152	0.890
sim_{TO}	0.526	0.156	0.297	0.900
$\text{sim}_{\text{AVG-CMatch}}$	0.425	0.034	0.367	0.498
$\text{sim}_{\text{BMA-CMatch}}$	0.667	0.080	0.492	0.796
$\text{sim}_{\text{BMM-Lin,naive}}$	0.231	0.229	0.000	0.662
$\text{sim}_{\text{BMM-Resnik,log}}$	0.114	0.112	0.000	0.308
$\text{sim}_{\text{USE,angular}}$	0.551	0.043	0.498	0.711
$\text{sim}_{\text{naive}}$	0.500	0.000	0.500	0.500

Table 7.1: Statistics for the harmonized upper triangle matrices. With 'Mean', 'Min', and 'Max' the according values within each matrix. The standard deviation is described by 'STD'.

	WTA to NE			WTA to USE		
Name	$<$	$>$	$=$	$<$	$>$	$=$
sim_{TO}	44	46	10	50	32	18
$\text{sim}_{\text{AVG-CMatch}}$	68	32	0	70	28	2
$\text{sim}^{*}_{\text{AVG-CMatch}}$	68	22	10	70	18	12
$\text{sim}_{\text{BMA-CMatch}}$	12	78	10	10	76	14
$\text{sim}_{\text{BMM-Lin,naive}}$	42	49	9	48	37	15
$\text{sim}_{\text{BMM-Resnik,log}}$	32	68	0	42	56	2
$\text{sim}_{\text{USE,angular}}$	16	79	5	-	-	-
$\text{sim}_{\text{naive}}$	-	-	-	66	14	20

Table 7.2: Comparison of the semantic similarity measures to the WTA Scores. '$>$' describes the number of similarities closer to the ground truth. Under 'WTA to NE' are the sums of the winning differences against $\text{sim}_{\text{naive}}$. Under 'WTA to USE' are the sums of the winning differences against $\text{sim}_{\text{USE,angular}}$.

7.1.2 Evaluation of Hypothesis $\mathscr{H}_2$

Knowledge-Graph based approaches correspond stronger than similarities based on the machine learning model Universal Sentence Encoder (USE).

As seen in subsection 7.1.1 only $\text{sim}_{\text{AVG-CMatch}}$ needs to be considered to evaluate hypothesis $\mathscr{H}_2$. Looking at Table 7.2 shows, that for the given ground truth $\text{sim}_{\text{USE,angular}}$ results in worse similarities than the naive estimator. Therefore, as $\text{sim}_{\text{AVG-CMatch}}$ performs better than $\text{sim}_{\text{USE,angular}}$ and better than $\text{sim}_{\text{naive}}$ it verifies hypothesis $\mathscr{H}_1$ for the given ground truth. Potential shortcomings of this evaluation will be discussed in section 7.2.

7.2 Limitations & Improvements

This thesis had three major limitations. The most significant was the restriction of Wikidata's public SPARQL endpoint. Especially the timeout after 30 seconds rendered many algorithms not usable. For a long time I had tried to set up a graph database server with university resources. But the unclear and outdated documentation together with the opaque hardware requirements prevented this. The lengthy attempts to set up a own Wikidata or DBpedia instance also prevented further investigations. So variations with the relations, as well as path analysis remain a task for future work. It might be even possible to circumvent the timeouts of the IC functions all together.
The two other limitations relate to the intuitive similarities obtained in the user study. The size of the similarity matrix is with 10×10 rather small. As it was the intention to gather complete pair-wise view on a sample set, further investigation is necessary. As seen in subsection 6.1.1, the intuitive average had a local minimum for variation and variance but the standard deviation did not yet balance. So a bigger sample size, at least five to ten times bigger, might be of interest.
The design of the user study in particular offers room for improvement. As seen in subsection 6.1.2 it would have been helpful to offer a lower bound of what a low similarity looks like.

7.3 Research Question

How well suited are 'Semantic Measures on Knowledge Graphs' to compare ideas regarding their similarity?

As described in section 7.1, semantic measures on knowledge graphs show potential for comparisons in the idea space. Even the Universal Sentence Encoder could not show a clear correspondence to the tested pairs of ideas. The results of the information based similarities are heavily tainted by the outliers, but the results independent of them seem to be heading in the right general direction. Together with the other limitations, no clear statement can be made. The fact that the naive estimator made such good predictions raises doubts about the quality of the ground truth.
Also the increased effort needs to be considered. The machine learning approach just needs the publicly available embedding model. In contrast, there is significantly more effort required for semantic measures. The required semantic enrichment together with the increased time needed to compute the similarity scores might prevent or at least hinder the use in some cases.
And then again, the benefit of using semantically enriched data together with knowledge graphs is the understandability and interpretability of the results. I can track what caused or hindered a similarity. Based on that comprehensibility further variations of similarity across different dimensions are imaginable.
So in conclusion: While the results of this thesis are very promising, further investigations are necessary to give a definite answer how well suited 'Semantic Measures on Knowledge Graphs' are.

7.4 Research Contribution

The main contribution of this thesis is the combination of a public knowledge graph and available ontology based semantic measures. Even though semantic measures on ontologies and knowledge graphs are not new, there is, to my knowledge, no similarity toolkit leveraging publicly available general knowledge graphs. Toolkits I could find were supposed to be used with smaller more specified graphs, such as gene databases. The second contribution was using a (small) set of ideas with all connections tested for similarity. None of the publicly available data sets met this condition of inter-connectivity, not even outside the realm of ideation.
As explained in detail in section 7.3, this thesis can only be a first step for taking advantage of publicly available general knowledge graphs and the semantic connections they hold.

7.5 Summary

In this chapter I evaluated and discussed the results presented in chapter 6. I evaluated the hypotheses introduced in section 1.2 and discussed the research question based on those evaluations. The limitations and research contributions of this theses were also reviewed.

8 Conclusion

In this paper I investigated the added value of semantic measures based on general knowledge graphs for similarity detection between ideas. As mentioned, semantic measures on knowledge graphs are an interesting alternative to the often used machine-learned approaches. Although they require more time for implementation, annotation and execution, they are more comprehensible. They might even offer the possibility of a multidimensional view of the meaning of the semantic. As explained in chapter 7 the measures discussed in this thesis do not yield perfect results. But they offer a glimpse into an alternative approach to semantic comparison of short texts.
The following sections summarize the work done in this thesis and give an outlook into future work.

8.1 Summary

In chapter 1 I explained the motivation based on the paper 'Discovering the Sweet Spot of Human-Computer Configurations: A Case Study in Information Extraction' [28]. The objective was to find measures that can compare idea texts semantically. This search led to two different approaches, one based in machine learning and one based in the semantic web. So the research goal became to compare both approaches.
Chapter 2 laid the theoretical foundation by introducing the ideation context, basic vocabulary from the semantic web needed for the thesis, introducing the terminology for idea similarity, and gave a small overview of machine learning. Due to their importance semantic measures got a separate chapter. In chapter 3 I focused on semantic measures on knowledge graphs. I explained in detail the different types of measures, starting at aggregated pair-wise algorithms and group-wise algorithms. With this distinction at hand I introduced pair- and group-wise algorithms in the three mayor categories: graph structure based, concept feature based, and information based measures. Finally I introduced one state-of-the-art algorithm for measures based on machine learned embeddings: Google's Universal Sentence Encoder. The chapter ended with the foreshadowing of some implementation problems and limitations.
To be able to compare all measures I needed a idea texts, a ground truth of similarities between those, semantic annotated of the idea texts, and the semantic measures to test. The planned process was described in chapter 4 and the implementation was explained in chapter 5.
With all implementations done, I looked at the resulting similarity scores. In chapter 6 I evaluated the results of the user study and described the preliminary reduction of the 61 different semantic measures on knowledge graphs down to five. This led to the comparison of said measures with a machine learned measure derived from Google's Universal Sentence Encoder and a naive estimator. The evaluation of those comparisons can be found in chapter 7. The results of that evaluation are very

promising. Semantic measures on knowledge graphs exist that perform well on the gathered ground truth. As seen in section 7.2, no general validity can be derived from that observation. Particularly the small size of the ground truth that entails its pitfalls. As seen a naive estimator performed better than many semantic measures reviewed. However, it is worth noting that the results of this thesis are encouraging to continue along this path.

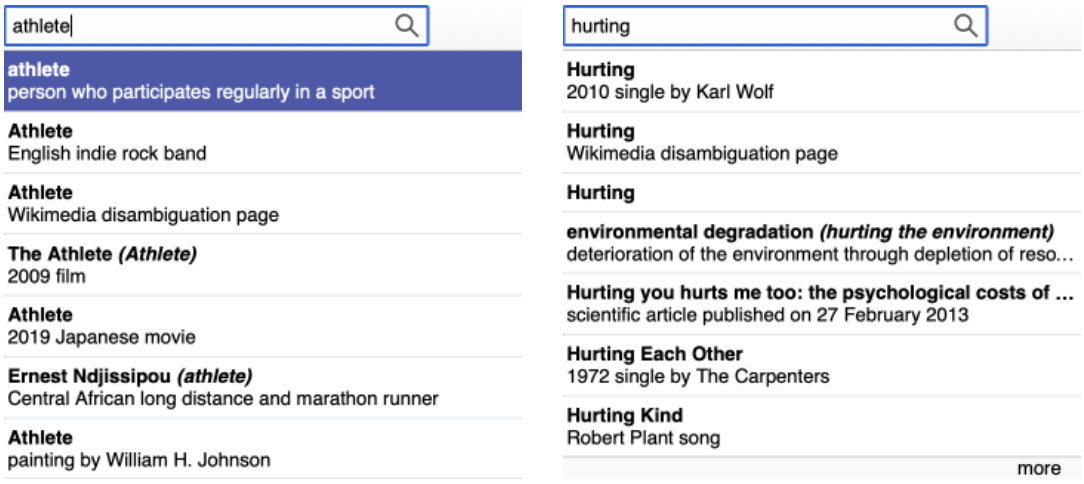
8.2 Future Work

The ultimate effect of pursuing this approach could be the multidimensional consideration of semantic similarity. I experienced it myself, while collection my personal similarities for evaluation. I have considered one pair of ideas to be similar in a different way than another. However, both showed a high degree of similarity to each other, just in different ways.
Along this way I see a lot of potential for future work. As mentioned in section 7.2, it might be of interest to set up a own knowledge base server. It would then be possible that, on the one hand, the algorithms that could not currently be implemented can be implemented. On the other hand the speed of the requests can be increased, since the depth of an element can become part of the element. So the limitations of this thesis could be overcome. Also the possible combination of general and specialized knowledge graphs might be of interest.
Methods could be developed to understand semantic enrichment as part of the ideation process itself. Immediately, this would increase the classification and discoverability of the collected ideas and reduce sources of error, since the ideators know what they mean.
Further space for future research is provided by ground truth. A further user study is conceivable, which turns the validation of the algorithms around. It would be conceivable to show crowd workers the result of different semantic measures and let them decide which algorithm maps similarities better.

As this thesis has shown semantic measures on knowledge graphs offer much opportunity for further research, especially in the ideation context.

9 Appendix

9.1 DBpedia & Wikidata Interface



(a) Successful Search (b) Unsuccessful Search

Figure 9.1: Showing the disambiguation interface for searches on Wikidata for the terms 'athlete' and 'hurting' provided by DBpedia Spotlight.

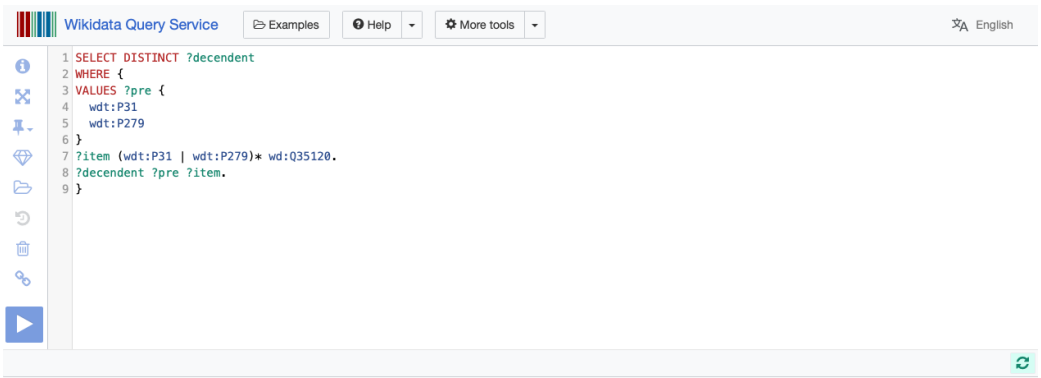


Figure 9.2: Wikidata SPARQL Query Interface

9.2 Semantically Enriched Ideas

```
- content: "Analyze athletes with bionic radar to see what part of them
    is either helping or hurting them in things such as running."
  id: "idea_01"
  innovonto_uuid: "38aa2640-6efb-49ee-afd7-fb7a786cb406"
  metadata:
    concepts:
    - concept: "wd:Q2066131"
      title: "Athlete"  # optional
      token: [1,1]      # would be 'some'
    - concept: "wd:Q47528"
      title: "Radar"  # optional
      token: [3,4]    # bionic radar
    - concept: "wd:Q14944328"
      title: "Example"
      token: [18,18]
    - concept: "wd:Q105674"
      title: "Running"
      token: [20,20]
    topics:  # topics are not connected to tokens but the whole idea
    - concept: "wd:Q349"
      title: "Sports"
    - concept: "wd:Q12147"
      title: "Health"
-
  content: "Use bionic radar to identify fouls or false starts in
     sporting events."
  id: "idea_02"
  innovonto_uuid: "54764cfe-83f4-4a16-b9a0-2294019522dc"
  metadata:
    concepts:
    - concept: "wd:Q47528"
      title: "Radar"  # optional
      token: [1,2]    # bionic radar
    - concept: "wd:Q15839293"
      title: "Foul"
      token: [5,5]
    - concept: "wd:Q162991"
      title: "False Start"
      token: [7,8]
    - concept: "wd:Q16510064"
      title: "Sporting Event"
      token: [10,11]
    topics:  # topics are not connected to tokens but the whole idea
    - concept: "wd:Q349"
      title: "Sports"
    - concept: "wd:Q47528"
      title: "Radar"
-
  content: "With bionic radar I will know which family member is
     entering the home."
  id: "idea_03"
```

```
innovonto_uuid: "e29368ec-4a13-417a-8f04-b8e5bd13fdff"
metadata:
  concepts:
  - concept: "wd:Q47528"
    title: "Radar"
    token: [1,2]  # bionic radar
  - concept: "wd:Q171318"
    title: "Kinship"
    token: [7,8]
  - concept: "wd:Q7743"
    title: "Home"
    token: [12,12]
  topics:  # topics are not connected to tokens but the whole idea
  - concept: "wd:Q47528"
    title: "Radar"
  - concept: "wd:Q8436"
    title: "Family"
  - concept: "wd:Q2526135"
    title: "Security"
```

Listing 9.1: YAML File of Enriched Ideas (Shortened)

9.3 Study Results

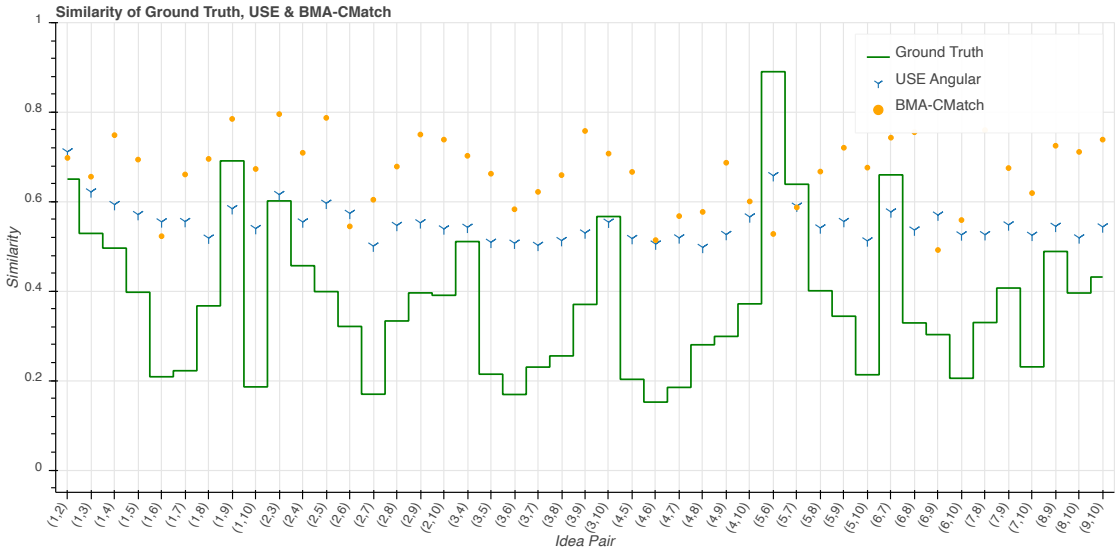


Figure 9.3: Plot comparing $\text{sim}_{\text{BMA-CMatch}}$ with $\text{sim}_{\text{USE,angular}}$ and the ground truth.

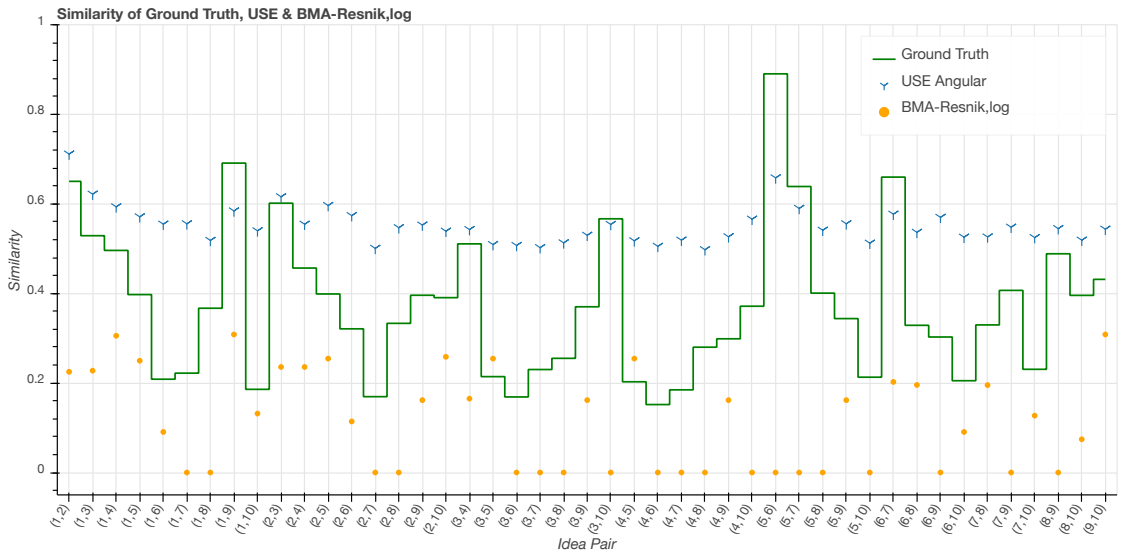


Figure 9.4: Plot comparing $\text{sim}_{\text{BMM-Resnik,log}}$ with $\text{sim}_{\text{USE,angular}}$ and the ground truth.

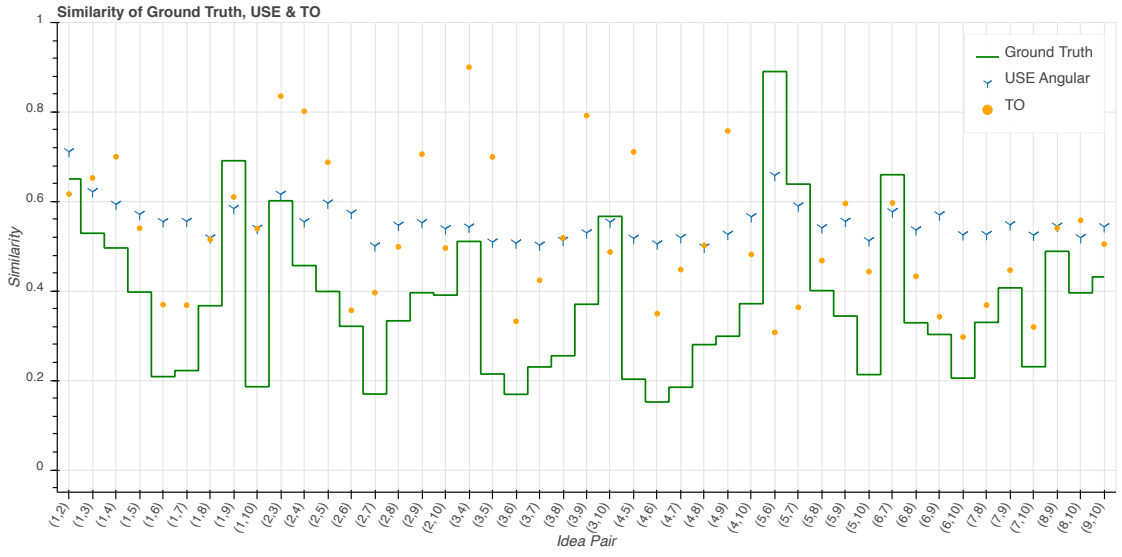


Figure 9.5: Plot comparing sim_{TO} with $\text{sim}_{\text{USE,angular}}$ and the ground truth.

9.4 Idea Similarity Mock-Up

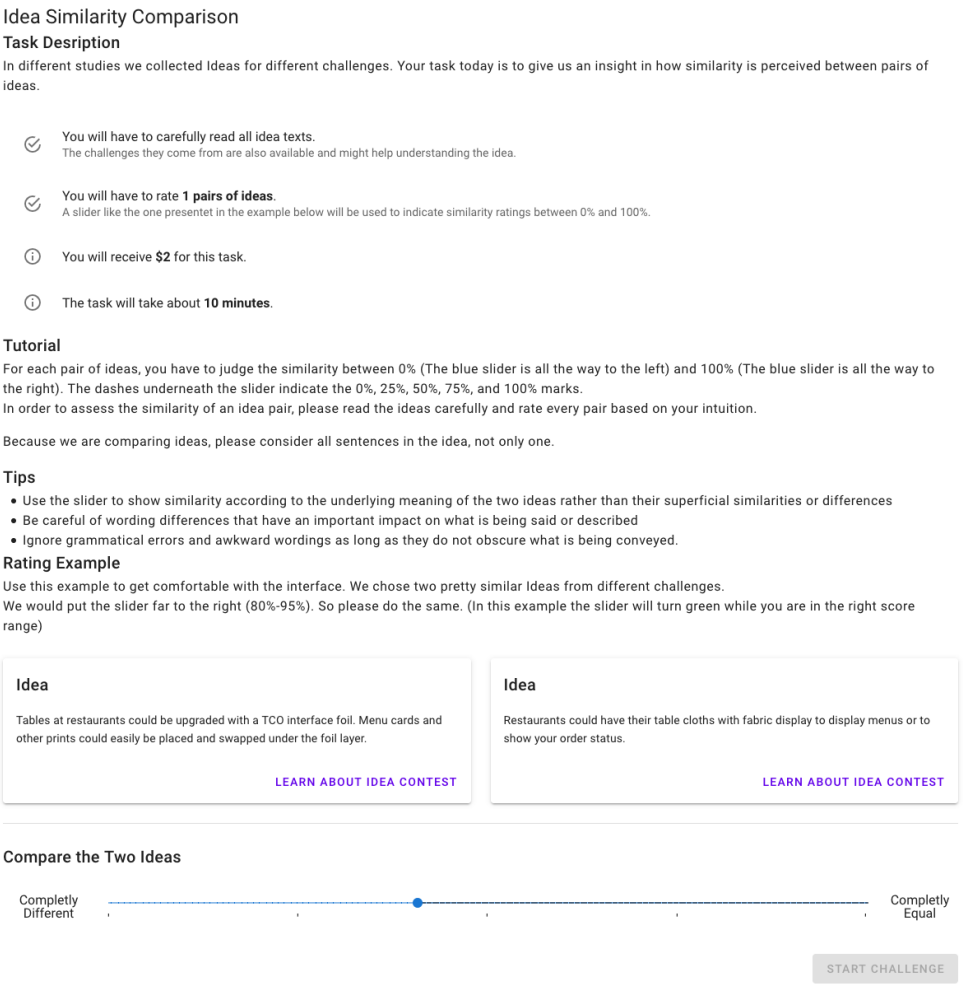


Figure 9.6: User Study Intro - Not Activated

Figure 9.7: User Study Intro - Activated

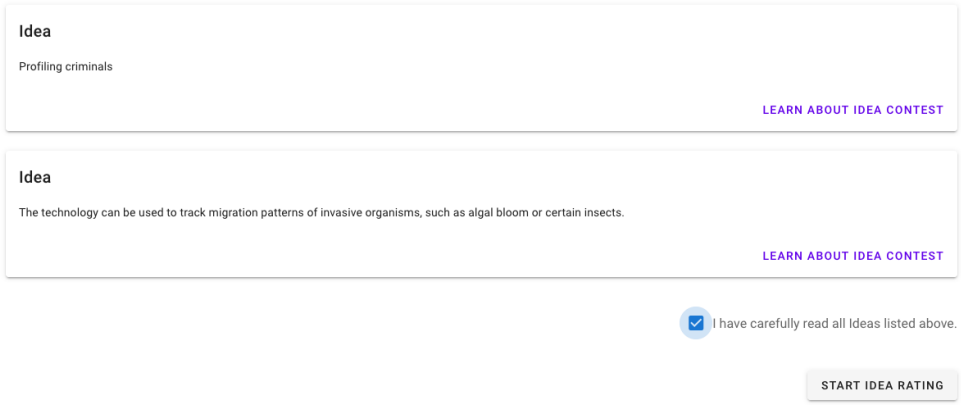


Figure 9.8: User Study Intro - Read Ideas

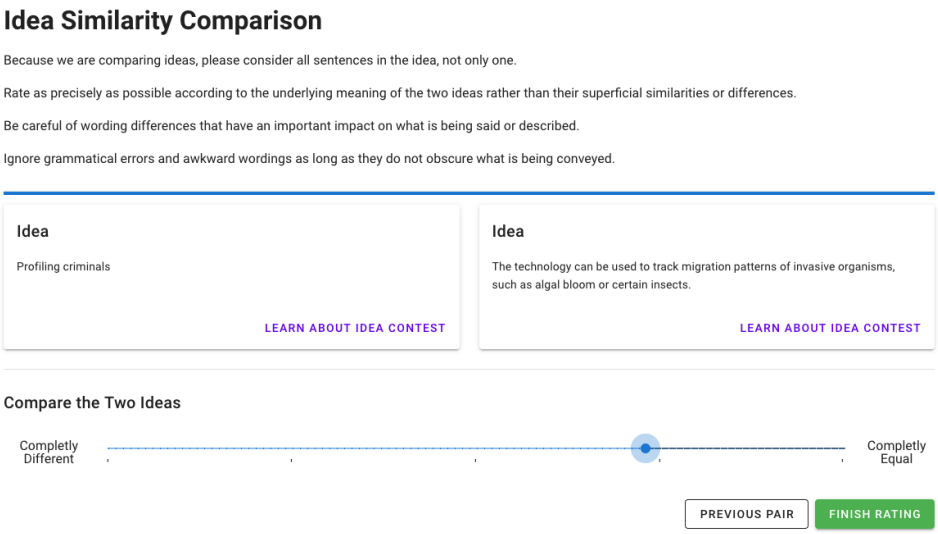


Figure 9.9: User Study Task - Idea Comparison

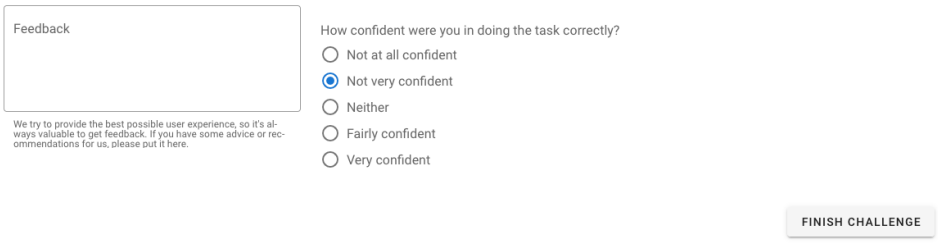


Figure 9.10: User Study Outro - Questionnaire

Bibliography

[1] Martín Abadi et al. "TensorFlow: A System for Large-Scale Machine Learning". In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, Nov. 2016, pp. 265–283. ISBN: 978-1-931971-33-1. URL: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

[2] Palakorn Achananuparp, Xiaohua Hu, and Xiajiong Shen. "The evaluation of sentence similarity measures". In: *International Conference on data warehousing and knowledge discovery*. Springer. 2008, pp. 305–316.

[3] Faez Ahmed et al. "Structuring Online Dyads: Explanations Improve Creativity, Chats Lead to Convergence". In: *Proceedings of the 2019 on Creativity and Cognition*. 2019, pp. 306–318.

[4] Ethem Alpaydin. *Introduction to Machine Learning*. 2014. ISBN: 9780262325752. URL: https://mitpress.ublish.com/ereader/26/?preview#page/1.

[5] Meysam Asgari-Chenaghlu, Narjes Nikzad-Khasmakhi, and Shervin Minaee. "Covid-transformer: Detecting trending topics on twitter using universal sentence encoder". In: *arXiv preprint arXiv:2009.03947* (2020).

[6] Christian Bizer, Tom Heath, and Tim Berners-Lee. "Linked data: The story so far". In: *Semantic services, interoperability and web applications: emerging concepts*. IGI Global, 2011, pp. 205–227.

[7] Osvald M Bjelland and Robert Chapman Wood. "An inside view of IBM's 'Innovation Jam'". In: *MIT Sloan management review* 50.1 (2008), pp. 32–40.

[8] Charles E. Brown. "Coefficient of Variation". In: *Applied Multivariate Statistics in Geohydrology and Related Sciences*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 155–157. ISBN: 978-3-642-80328-4. DOI: 10.1007/978-3-642-80328-4_13. URL: https://doi.org/10.1007/978-3-642-80328-4_13.

[9] Jason Brownlee. *Machine Learning Mastery*. Jan. 13, 2020. URL: https://machinelearningmastery.com.

[10] Daniel Cer et al. "Universal sentence encoder". In: *arXiv preprint arXiv:1803.11175* (2018). eprint: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/46808.pdf.

[11] Joel Chan et al. "Semantically Far Inspirations Considered Harmful?: Accounting for Cognitive States in Collaborative Ideation". In: *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*. ACM. 2017, pp. 93–105.

[12] Thomas H Cormen et al. *Introduction to algorithms*. MIT press, 2009.

[13] Joachim Daiber et al. "Improving Efficiency and Accuracy in Multilingual Entity Extraction". In: *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*. 2013.

[14] C. De Boom et al. "Learning Semantic Similarity for Very Short Texts". In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. 2015, pp. 1229–1234. DOI: 10.1109/ICDMW.2015.86.

[15] Lisa Ehrlinger and Wolfram Wöß. "Towards a Definition of Knowledge Graphs." In: *SEMANTiCS (Posters, Demos, SuCCESS)* 48 (2016), pp. 1–4.

[16] Michael Färber et al. "A comparative survey of dbpedia, freebase, opencyc, wikidata, and yago". In: *Semantic Web Journal* 1.1 (2015), pp. 1–5.

[17] Victor Girotto, Erin Walker, and Winslow Burleson. "The effect of peripheral micro-tasks on crowd ideation". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM. 2017, pp. 1843–1854.

[18] Thomas R Gruber. "A translation approach to portable ontology specifications". In: *Knowledge acquisition* 5.2 (1993), pp. 199–220.

[19] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[20] Sébastien Harispe et al. "Semantic similarity from natural language and ontology analysis". In: *Synthesis Lectures on Human Language Technologies* 8.1 (2015), pp. 1–254.

[21] Sandra G Hart and Lowell E Staveland. "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research". In: *Advances in psychology*. Vol. 52. Elsevier, 1988, pp. 139–183.

[22] Olaf Hartig. "An Introduction to SPARQL and Queries over Linked Data". In: *Web Engineering*. Ed. by Marco Brambilla, Takehiro Tokuda, and Robert Tolksdorf. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 506–507. ISBN: 978-3-642-31753-8.

[23] *Introduction to Probability: MSE*. URL: https://www.probabilitycourse.com/chapter9/9_1_5_mean_squared_error_MSE.php (visited on 01/03/2021).

[24] Aniket Kittur, Ed H Chi, and Bongwon Suh. "Crowdsourcing user studies with Mechanical Turk". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2008, pp. 453–456.

[25] Will Koehrsen. *Neural Network Embeddings Explained - Towards Data Science*. Oct. 2, 2018. URL: https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526 (visited on 01/14/2020).

[26] Juan J Lastra-Díaz et al. "HESML: A scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset". In: *Information Systems* 66 (2017), pp. 97–118.

[27] Maximilian Mackeprang, Abderrahmane Khiat, and Claudia Müller-Birn. "Innovonto: An Enhanced Crowd Ideation Platform with Semantic Annotation (Hallway Test)". Englisch. In: vol. TR-B-18-02. FU Technical Reports Serie B. Berlin, 2018.

[28] Maximilian Mackeprang, Claudia Müller-Birn, and Maximilian Stauss. "Discovering the Sweet Spot of Human-Computer Configurations: A Case Study in Information Extraction". Englisch. In: *Proceedings of the ACM on Human-Computer Interaction*. Vol. 3. CSCW'19 195. ACM, Nov. 9, 2019. DOI: 10.1145/3359297.

[29] Maximilian Mackeprang et al. "The Impact of Concept Representation in Interactive Concept Validation (ICV)". http://dx.doi.org/10.17169/refubium-3971. 2019.

[30] Richard L. Marsh, Joshua D. Landau, and Jason L. Hicks. "How examples may (and may not) constrain creativity". In: *Memory & Cognition* 24.5 (Sept. 1996), pp. 669–680. ISSN: 1532-5946. DOI: 10.3758/BF03201091. URL: https://doi.org/10.3758/BF03201091.

[31] *Material Design*. URL: https://material.io/design/ (visited on 12/22/2020).

[32] Brian McBride. "The resource description framework (RDF) and its vocabulary description language RDFS". In: *Handbook on ontologies*. Springer, 2004, pp. 51–65.

[33] *Mturk Api Reference: Developer Guide*. URL: https://docs.aws.amazon.com/AWSMechTurk/latest/AWSMechanicalTurkRequester/Welcome.html (visited on 01/03/2021).

[34] *Mturk Api Reference: External Question*. URL: https://docs.aws.amazon.com/AWSMechTurk/latest/AWSMturkAPI/ApiReference_ExternalQuestionAr
html (visited on 01/03/2021).

[35] Gobie Nanthakumar. "A tool driven approach to Mechanical Turk user experiments". Freie Universität Berlin, 2019. eprint: https://www.mi.fu-berlin.de/en/inf/groups/hcc/theses/finished/2019-Theses/tool-driven-approach-to-mechanical-turk-user-experiments/2019_BA_Nanthakumar.pdf.

[36] *Ontology | DBpedia*. Nov. 17, 2020. URL: https://wiki.dbpedia.org/services-resources/ontology (visited on 11/17/2020).

[37] Christian S Perone, Roberto Silveira, and Thomas S Paula. "Evaluation of sentence embeddings in downstream and linguistic probing tasks". In: *arXiv preprint arXiv:1806.06259* (2018).

[38] *Project Jupyter: Homepage*. URL: https://jupyter.org (visited on 01/03/2021).

[39] *Python 3 - asyncio - Asynchronous I/O*. URL: https://docs.python.org/3/library/asyncio.html (visited on 12/19/2020).

[40] Roy Rada et al. "Development and application of a metric on semantic nets". In: *IEEE transactions on systems, man, and cybernetics* 19.1 (1989), pp. 17–30.

[41] Philip Resnik. "Using Information Content to Evaluate Semantic Similarity in a Taxonomy". In: *arXiv preprint cmp-lg/9511007* (1995).

[42] Ignacio Traverso Ribón. "A Framework for Semantic Similarity Measures to Enhance Knowledge Graph Quality". PhD thesis. Karlsruher Institut für Technologie (KIT), Aug. 10, 2017.

[43] M Andrea Rodríguez and Max J. Egenhofer. "Determining semantic similarity among entity classes from different ontologies". In: *IEEE transactions on knowledge and data engineering* 15.2 (2003), pp. 442–456.

[44] David Sánchez and Montserrat Batet. "Semantic similarity estimation in the biomedical domain: An ontology-based information-theoretic perspective". In: *Journal of biomedical informatics* 44.5 (2011), pp. 749–759.

[45] David Sánchez et al. "Ontology-based semantic similarity: A new feature-based approach". In: *Expert systems with applications* 39.9 (2012), pp. 7718–7728.

[46] Vincent Schickel-Zuber and Boi Faltings. "OSS : A Semantic Similarity Function based on Hierarchical Ontologies". In: *Artificial Intelligence* (2007), pp. 551–556.

[47] Ali Seyed Shirkhorshidi, Saeed Aghabozorgi, and Teh Ying Wah. "A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data". In: *PLOS ONE* 10.12 (Dec. 2015), pp. 1–20. DOI: 10.1371/journal.pone.0144059. URL: https://doi.org/10.1371/journal.pone.0144059.

[48] Pao Siangliulue et al. "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. Tokyo, Japan: ACM, 2016, pp. 609–624. ISBN: 978-1-4503-4189-9. DOI: 10.1145/2984511.2984578. URL: http://doi.acm.org/10.1145/2984511.2984578.

[49] Pao Siangliulue et al. "Large-Scale Collaborative Innovation: Challenges, Visions and Approaches". In: *2016 AAAI Spring Symposium Series*. Mar. 5, 2016. eprint: https://www.aaai.org/ocs/index.php/SSS/SSS16/paper/view/12761/11969.

[50] Pao Siangliulue et al. "Toward Collaborative Ideation at Scale: Leveraging Ideas from Others to Generate More Creative and Diverse Ideas". In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. CSCW '15. Vancouver, BC, Canada: ACM, 2015, pp. 937–945. ISBN: 978-1-4503-2922-4. DOI: 10.1145/2675133.2675239. URL: http://doi.acm.org/10.1145/2675133.2675239.

[51] *TensorFlow Tutorial: Word embeddings*. URL: https://www.tensorflow.org/tutorials/text/word_embeddings (visited on 12/26/2020).

[52] *TensorFlow: Hub*. URL: https://www.tensorflow.org/hub (visited on 12/31/2020).

[53] *TensorFlow: Universal Sentence Encoder*. URL: https://tfhub.dev/google/universal-sentence-encoder/4 (visited on 12/26/2020).

[54] Ignacio Traverso et al. “GADES: a graph-based semantic similarity measure”. In: *Proceedings of the 12th International Conference on Semantic Systems*. ACM. 2016, pp. 101–104.

[55] *TypeScript: Typed JavaScript*. URL: https://www.typescriptlang.org (visited on 01/03/2021).

[56] *Vue.js*. URL: https://vuejs.org (visited on 01/03/2021).

[57] *Vuetify*. URL: https://vuetifyjs.com/ (visited on 01/03/2021).

[58] *W3C Opens Data on the Web with SPARQL*. Jan. 15, 2008. URL: https://www.w3.org/2007/12/sparql-pressrelease (visited on 12/31/2020).

[59] *Wikidata Entry: Door*. Nov. 17, 2020. URL: https://www.wikidata.org/wiki/Q36794 (visited on 11/17/2020).

[60] *Wikidata Query Service*. URL: https://query.wikidata.org/ (visited on 12/26/2020).

[61] *Wikidata Query Service tutorial*. URL: https://wdqs-tutorial.toolforge.org/ (visited on 12/26/2020).

[62] *Wikidata: SPARQL tutorial*. URL: https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial (visited on 12/26/2020).

[63] *Wikidata:Data access*. URL: https://www.wikidata.org/wiki/Wikidata:Data_access (visited on 12/26/2020).

[64] *Wikipedia: Cosine similarity*. URL: https://en.wikipedia.org/wiki/Cosine_similarity (visited on 12/31/2020).

[65] *Wikipedia: SPARQL*. URL: https://en.wikipedia.org/wiki/SPARQL (visited on 12/26/2020).

[66] *Wiktionary: uniform resource identifier*. URL: https://en.wiktionary.org/wiki/uniform_resource_identifier (visited on 12/26/2020).

[67] Xiaojun Yuan and Catherine Dumas. “Cyber Behaviors in Seeking Information”. In: *Encyclopedia of Cyber Behavior*. IGI Global, 2012, pp. 365–382. DOI: 10.4018/978-1-4666-0315-8.ch031.

[68] John Zaller and Stanley Feldman. “A Simple Theory of the Survey Response: Answering Questions versus Revealing Preferences”. In: *American Journal of Political Science* 36.3 (1992), pp. 579–616. URL: http://www.jstor.org/stable/2111583.

[69] Vitalii Zhelezniak et al. “Correlation coefficients and semantic textual similarity”. In: *arXiv preprint arXiv:1905.07790* (2019).

[70] Zili Zhou, Yanna Wang, and Junzhong Gu. “A new model of information content for semantic similarity in WordNet”. In: *2008 Second International Conference on Future Generation Communication and Networking Symposia*. Vol. 3. IEEE. 2008, pp. 85–89.