Pontifícia Universidade Católica
do Rio de Janeiro

Felipe Poggi de Aragão Fraga

On Automatic Generation of Knowledge Connections

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática.

Advisor: Prof. Marcus Vinicius Soledade Poggi de Aragão

Rio de Janeiro
September 2022

Pontifícia Universidade Católica
do Rio de Janeiro

Felipe Poggi de Aragão Fraga

On Automatic Generation of Knowledge Connections

Dissertation presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática. Approved by the Examination Committee:

Prof. Marcus Vinicius Soledade Poggi de Aragão
Advisor
Departamento de Informática – PUC-Rio

Profa. Simone Diniz Junqueira Barbosa
Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio

Prof. Marco Antonio Casanova
Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio

Rio de Janeiro, September 23rd, 2022

Felipe Poggi de Aragão Fraga

Felipe is a Knowledge Management enthusiast, who has dedicated considerable time and effort to studying, implementing, and teaching knowledge systems and workflows. He wishes to deploy the system presented in this dissertation, together with other ones, to enhance his knowledge and share it with the world. Felipe is finishing an MSc in Artificial Intelligence, focused on Natural Language Processing (NLP) at the Pontifical Catholic University of Rio de Janeiro (PUC-Rio), in Brazil. He comes from a bachelor's degree in Mechanical Engineering, also at PUC-Rio.

Bibliographic Data

Poggi de Aragão Fraga, Felipe

On Automatic Generation of Knowledge Connections / Felipe Poggi de Aragão Fraga; advisor: Marcus Vinicius Soledade Poggi de Aragão. – 2022.

162 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2022.

Inclui bibliografia

1. keywordpre – Teses. 2. keywordpre – Teses. 3. Gestão de Conhecimento Pessoal. 4. Processamento de Linguagem Natural. 5. Extração de Conceitos. 6. Similaridade de Textos. 7. Geração de Conexões. 8. Aplicativos de Anotação. 9. Links Bidirecionais. I. Soledade Poggi de Aragão, Marcus Vinicius. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

To the Personal Knowledge Management community, for the inspiration to pursue this topic and resources to build this system. To my parents, who have provided me with everything I needed to work on this project. To my advisor, who guided me beautifully through this process of research and discovery.

Acknowledgments

I would like to thank my parents above all, for the opportunity and incentive of pursuing my Master's degree, and for the unconditional love and support in life.

I would like to thank my girlfriend, Maria Clara, for supporting me, pushing me, loving me, and helping me in several ways during this process.

I would like to thank my advisor, who guided me through important decisions, gave me the freedom to trail my own path, and was very important to bring this work to a conclusion.

I would like to thank my friends, who inspire me, support me, cherish me and make my life better.

I am extremely grateful to the PKM Community on Twitter, for inspiring me, showing me important resources, and providing me with ideas and pathways for making this work come to fruition.

Within this community, I would like to especially thank Tiago Forte and RJ Nestor for teaching me about workflows and mentalities of digital productivity in Personal Knowledge Management.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Poggi de Aragão Fraga, Felipe; Soledade Poggi de Aragão, Marcus Vinicius (Advisor). **On Automatic Generation of Knowledge Connections**. Rio de Janeiro, 2022. 162p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Recently, the topic of Personal Knowledge Management (PKM) has seen a surge in popularity. This is illustrated by the accelerated growth of apps such as *Notion*, *Obsidian*, and *Roam Research*, and the appearance of books like "How to Take Smart Notes" and "Building a Second Brain".

However, the area of PKM has not seen much integration with the field of Natural Language Processing (NLP). This opens up an interesting opportunity to apply NLP techniques to knowledge operations tasks.

Our objective is the development of a Software System that uses NLP and note-taking apps to transform a siloed text collection into an interconnected and inter-navigable text collection. The system uses navigation mechanisms based on shared concepts and semantic relatedness between texts.

In this study, we present a methodology to build this system, the research context, demonstrations using examples, and an evaluation to determine if the system functions properly and if the proposed connections are coherent.

Keywords

Personal Knowledge Management; Natural Language Processing; Concept Extraction; Text Relatedness; Connections Generation; Note-taking apps; Bi-directional Hyperlink.

Resumo

Poggi de Aragão Fraga, Felipe; Soledade Poggi de Aragão, Marcus Vinicius. **Geração Automática de Conexões para Gestão de Conhecimento**. Rio de Janeiro, 2022. 162p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Recentemente, o tópico de Gestão de Conhecimento Pessoal vem ganhando muita popularidade. Ilustrado pelo rápido crescimento de aplicativos como *Notion*, *Obsidian*, e *Roam Research* e da aparição de livros como "*How to Take Smart Notes*" e "*Building a Second Brain*".

Contudo, ainda é uma área que não foi fortemente envolvida pelo Processamento de Linguagem Natural (NLP). Isso abre uma bela oportunidade para a aplicação de NLP em operações com conhecimento.

Nosso objetivo é o desenvolvimento de um sistema de software que utiliza NLP e aplicatovps de anotação para transformar uma coleção de textos isolados em uma coleção de textos interconectada e inter-navegável. Isso é feito usando mecanismos de navegação baseados em conceitos mencionados e recomendações semânticas.

Neste trabalho apresentamos a metodologia para construir o sistema, demonstrações com exemplos palpáveis, assim como uma avaliação para determinar a coerência dos resultados.

Palavras-chave

Gestão de Conhecimento Pessoal; Processamento de Linguagem Natural; Extração de Conceitos; Similaridade de Textos; Geração de Conexões; Aplicativos de Anotação; Links Bidirecionais.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI – Artificial Intelligence

NLP – Natural Language Processing

PKM – Personal Knowledge Management

TfT – Tools for Thought

KB – Knowledge Base

NER – Named Entity Recognition

RE – Relation Extraction

IE – Information Extraction

OpenIE – Open Information Extraction

RDF – Resource Description Framework

SPARQL – SPARQL Protocol and RDF Query Language

RQ – Research Question

1
Introduction

The recent surge in popularity of the Personal Knowledge Management (PKM) field has led to the golden age of note-taking tools. Several new note-taking apps have been released since 2016, revolutionized how knowledge operations take place, by providing the general public with brand-new functionalities that were not previously available. The most important of those being that of bidirectional hyperlinks.

Even though advancements in note-taking tools have coincided perfectly with the accelerated development of Natural Language Processing (NLP), these two fields still interact in a very superficial way. This presents a huge opportunity for combining the fields of Personal Knowledge Management and Natural Language Processing, by providing modern note-taking tools with features based on Artificial Intelligence, while using the already available functionalities as a starting point.

In a more structured way, the main motivation for this work comes from the intersection of three elements:

1. The field of Personal Knowledge Management
2. A new generation of Note-Taking Tools
3. Natural Language Processing research

The field of Personal Knowledge Management is based on systems to **create an external and persistent collection of a person's knowledge**. This idea is originally attributed to Vannevar Bush's reflections from 1945, As We May Think, (Bush, 1945), and has been influenced by several other thinkers over the years.

Recently, PKM has become increasingly popular, as evidenced by the recent success of the book "*How to Take Smart Notes*", (Ahrens, 2017), which explains in detail the note-taking process of Sociologist Niklas Luhmann, a prolific Sociologist who used a peculiar note-taking methodology called the Zettelkasten.

Another piece of evidence for this increase in popularity is the appearance of a new category of tools for note-taking. These tools provide new ways of organizing knowledge, including the integration of multiple knowledge sources.

Over the last 5-10 years, several apps have been launched, with some notable examples of these tools being:

- Roam Research[1]
- Obsidian[2]
- Readwise[3]
- Notion[4]
- Tana[5]
- Mem[6]

These apps share a common theme of adding new features and possibilities for knowledge organization and connection. In special, when compared to the hierarchical, folder-document system used by traditional note-taking apps such as Evernote[7] and One-Note[8].

One of these features is essential for this work, **bidirectional hyperlinks** used to organize knowledge. This feature is closely related to how Ted Nelson originally visualized hyperlinks, representing both directions of the link, Front-links (outgoing) and Back-links (incoming), (Nelson, 1965).

Compared to traditional web hyperlinks, incoming links to a page provide a fundamentally different type of knowledge, by showing where this specific page was mentioned. In this dissertation, we apply NLP to leverage this knowledge by elaborating on top of current functionalities presented by hyperlink-based (networked) note-taking tools.

For an interactive experience reading this dissertation, access:

`https://github.com/fisfraga/Knowledge-Connector/`

The dissertation's text was used as input for the system hereby proposed. There you may find the resulting interconnected version of this dissertation hosted in at least one of the note-taking apps mentioned in this section. Using these apps should provide the reader with a sample of the new functionalities they provide.

[1] https://roamresearch.com/
[2] https://obsidian.md/
[3] https://readwise.io/ – Not a note-taking app, but rather a knowledge integration tool
[4] https://www.notion.so/product
[5] https://tana.inc/
[6] https://get.mem.ai/
[7] https://evernote.com/
[8] https://www.onenote.com/

1.1 Problem Statement

The new set of features presented by the new generation of Note-Taking software hugely expands the possibilities for users working with knowledge. Being able to use features such as Backlinks and Transclusion adds an entirely new dimension to operating with knowledge for the users' toolkits. Transclusion is when content from one hypertext document is included inside another document, without having to leave the original location.

With huge possibilities, however, there are potentially huge amounts of work and effort. When users face dozens of pieces of text (notes, articles, highlights, etc.) they wish to link and organize, there certainly is some effort involved in doing so, but it surely is manageable. The problem arises when users find themselves with *several dozens*, *hundreds*, or even *thousands* of different pieces of text they wish to explore, connect and organize.

Users who wish to use these features are seldom faced with the cumbersome task of looking through huge amounts of notes in order to be able to connect two related texts. This opens up the perfect opportunity to use Artificial Intelligence to aid the process of finding connections between pieces of knowledge, more specifically, of course, Natural Language Processing.

The main general objective of this work is to deploy Natural Language Processing functionalities that are capable of leveraging and enhancing current note-taking apps functionalities. The intended use of NLP is to facilitate, enhance, or replace human effort in the pursuit of operating with knowledge inside modern note-taking software apps.

Operating with knowledge would usually include visualizing, understanding, manipulating, and creating, among other tasks. This will be further discussed in subsequent chapters.

In order to arrive at a more specific problem definition, this broad objective is broken down into a specific use case, where it would be possible to approach solving this problem, in a practical way.

Suppose a Text Collection is presented, and an individual wants to access what is inside to recall, explore, or study the content. The Text Collection here can contain a wide range of texts, from a collection of internet articles to an entire book, multiple books, academic papers, personal notes, a selection of highlights, Twitter Threads from a given author, a Science Textbook, anything.

The chosen path by this dissertation to aid the user in the exploration of a given text collection is to **automatically propose connections between texts while using note-taking software to visualize and navigate the text collection**.

The goal of this system is to propose connections that may be accessed by human users, not by computers. The obvious choice for a way of "*using*" these connections is through human navigation of the text collection through the proposed connections.

Connections, or hyperlinks, are an important part of Knowledge Management, they bridge different ideas and may connect two or more texts into a trail of thought. With this in mind, connections for navigating the text collection will be proposed along two distinct paths:

1. **Concepts Connections:** Navigation between texts through shared concepts. i.e. Navigate from a text to a concept mentioned in the text, and from such concept to other texts where it appears.

2. **Text Relatedness Connections:** Navigation through a recommendation system where texts are suggested based on Semantic Textual Relatedness.

These two principles for connecting texts will be used as means of connecting conceptual ideas and are very important in the context of this dissertation.

1.1.1 Formal Problem Definition

The problem is formally defined as:

Problem: How to automatically generate connections to transform a siloed text collection into an interconnected and inter-navigable text collection, represented by a graph?

Specific Details: How to propose knowledge connections between texts using shared concepts and semantically related texts? How to leverage modern note-taking software tools to enable navigation using the generated connections?

1.1.2 Mathematical Problem Definition

This problem is also described mathematically:

Given a text collection T, create an equivalent, yet connected knowledge graph G_T that can be navigated by users. By adding new node and edge types to the graph.

The graph, G_T, is defined as:

$$G_T = (V, E) \tag{1-1}$$

$$V \subseteq (T, C, A) \tag{1-2}$$

1. T: Text nodes
2. C: Concept nodes
3. A: Author nodes

Each node represents a page, which contains text that may be accessed, consumed, and edited.

From these three node types, the goal is to generate edges that represent all combinations of connections between the nodes, which are bidirectional:

$$E \subseteq (\overleftrightarrow{TT}, \overleftrightarrow{TC}, \overleftrightarrow{CC}, \overleftrightarrow{AA}, \overleftrightarrow{AC}, \overleftrightarrow{AT}) \tag{1-3}$$

1. $\overleftrightarrow{TT}$: Text $\leftrightarrow$ Text edges
2. $\overleftrightarrow{TC}$: Text $\leftrightarrow$ Concept edges
3. $\overleftrightarrow{CC}$: Concept $\leftrightarrow$ Concept edges
4. $\overleftrightarrow{AA}$: Author $\leftrightarrow$ Author edges
5. $\overleftrightarrow{AC}$: Author $\leftrightarrow$ Concept edges
6. $\overleftrightarrow{AT}$: Author $\leftrightarrow$ Text edges

The underlying problem to solve is the creation of the graph G_T. There are 9 components of the graph (3 node types, and 6 edge types) that need to be calculated in order to successfully transform the text collection into an interconnected graph.

Initially, 2 node types and 1 edge type are already known, ($\{T, A\} \in V$; $\overleftrightarrow{AT} \in E$). The remaining components are calculated by further dividing the problem into two sub-problems.

These two sub-problems are equivalent to the two different navigation standpoints. The sub-problems, and their corresponding graph components, are defined as follows:

1. Extract Concepts mentioned in the text and their Relationships.
    a. Extract Concepts mentioned in the text. ($C \in V$; $\{\overleftrightarrow{TC}, \overleftrightarrow{AC}\} \in E$)
    b. Identify Semantic Relations between Concepts. ($\overleftrightarrow{CC} \in E$)
2. Compute Semantic Relatedness between Texts. ($\{\overleftrightarrow{TT}, \overleftrightarrow{AA}\} \in E$)

1.2 Research Questions

The following Research Questions were chosen as means of representing and leading the exploration of the Research Problem.

Research Question 1: Can the combination of NLP with Networked note-taking tools improve the Knowledge Management functions of Recall, Elaboration, and New Insight?

This first Research Question is focused on addressing the demand for such a solution, by looking to understand the demand in the first place. Is it possible to combine these two components, Natural Language Processing and Networked Note-Taking Tools, to generate valuable results?

This dissertation looks to answer this first question by proposing a system that combines these two components and applies them to an originally siloed text collection, looking to enhance the user experience of consuming and exploring its content. The three tasks chosen to represent this exploration are of **Recall, Elaboration, and New Insight**.

These tasks are actually Knowledge Management functions, presented in Burkhard (2005) as a part of a framework for Knowledge Visualization, and will be discussed further in chapter 3. It is just important to note that these three tasks are used as reference points.

Research Question 2: How to propose connections between any two given texts present in a text collection?

The second Research Question looks to pave the way for how the connections will actually be generated. This question is open-ended since there are several different ways to propose connections between two texts. The idea, though, is to use this question as a means of convergence. Since there are so many different possibilities for connecting texts, which ones will actually be chosen?

This question addresses the fundamental functionalities of the proposed system. The essence of this dissertation is generating connections, ideally by more than just a single path, this question represents this essence.

Research Question 3: Are Concept Nodes a useful mechanism for navigating a text collection?

The last Research Question is intended to be more reflexive, the idea here is to evaluate in some way the mechanisms proposed for navigating a text collection. The recommendation system based on semantic relatedness between

texts is a pretty straightforward idea, but using concepts as a mechanism for navigation between texts is more subjective and open to interpretation.

Assuming that the connections are successfully generated, do they represent a valid source of information? Do concepts actually work as a mechanism for navigating between texts? These are the types of reflections that this Research Question represents, to look at the path chosen to propose a solution that solved the identified demand.

1.3 Methodology

The Methodology for carrying out this research is divided into 3 main parts, each one represented by a different chapter in the dissertation. The first two parts correspond to the two sub-problems in the section 1.1, and are considered as data collection for the last part. First, the Generation of Concepts Connections, chapter 5, second, the Generation of Text Semantic Relatedness Connections, chapter 6. The third part is the most important one, represents the essence of this study, and is considered the actual methodology, the Creation of an Interconnected Knowledge Collection, chapter 7. This last part encompasses the first two and is where everything falls into place.

1.3.1 Generating Concepts Connections

The first part of the methodology is presented in chapter 5, it is focused on the first sub-problem of Extracting Concepts mentioned in the text, as well as the Relationships between them. This part of the methodology outlines the NLP tasks performed in order to obtain the necessary information to generate all the Connections that involve concepts, namely the connections between Texts and Concepts, Authors and Concepts, and also between Concepts and other Concepts.

This section is further divided into two main sub-sections, Extracting Concepts Mentioned in the Text Collection and Identifying Relationships between Concepts.

Extracting Concepts Mentioned in the Text Collection

Conceptual Entities, or simply Concepts, are considered the most important element of this research, this part of the methodology details the necessary tasks to identify the concepts that are present in the texts and determine which ones are considered to be relevant. This is done following the tasks of Entity Recognition, Entity Linking, and Concepts Filtering, which will be explained in detail in chapter 5

Identifying Relationships between Concepts

This task is based on collecting knowledge from external Knowledge Bases for each of the relevant concepts identified in the text, the idea is to identify relationships and relatedness between concepts. The relationships will determine the connections between concepts and are usually either a simple *relatedTo* connection or more elaborate relationships based on commonsense knowledge, such as Rain "*HasPrerequisite*" of Clouds.

1.3.2 Text Semantic Relatedness Connections

The second part of the methodology is presented in chapter 6, it is directed to finding the Semantic Relatedness between any two given texts as a source of direct connections between two different texts. The idea here is to use Corpus-based Semantic Textual Relatedness methods to compute the relatedness (analogous to the distance) between any two texts.

The main objective for this part of the methodology is to compute a numerical score for the relatedness which **captures the semantic meaning and surrounding context** of each word. This means that the actual meaning of the texts is captured and not only a lexical similarity between words, for example, assigning a high relatedness between texts that mention airports and airplanes.

1.3.3 Creating Interconnected Knowledge Collection

This part is referred to as the Methodology itself, chapter 7, while the two previous parts are simply data collection in preparation for this one. The idea here is to use the information collected on the connections using concepts and the text relatedness connections and actually build the Interconnected Knowledge Collection which will be used for navigation.

This section is divided into three main parts, Building the Interconnected Graph, Content Selection for Node Pages and Adding Navigation with Obsidian.

Building the Interconnected Graph is directed at creating connections, by using the information from the two previous parts of the methodology to select the nodes and edges that will solve the problem according to the mathematical definition, subsection 1.1.2. This section details the design decisions for filtering the nodes and edges, looking to create an interconnected text collection that is coherent, relevant, and useful.

Content Selection for Node Pages is simply directed at choosing

what information is going to be portrayed in the pages for each Node type. This task basically organizes the original content of the Text Collection together with additionally collected knowledge, combining them into pages of content.

Adding Navigation with Obsidian focuses on combining the structure of the Interconnected Graph G_T with the Node Pages. This is done by leveraging the note-taking software called Obsidian to integrate both of them together and create an interconnected Text Collection that may be navigated.

1.4 Expected Contributions

The research and implementation carried out in this Master's dissertation expects to result in three main contributions.
The main expected contributions of this work are:

1. To automatically generate connections between texts in a way that users can navigate, as opposed to machine-readable connections.

Most works on the topic of creating knowledge repositories from a text dataset focus on creating machine-readable information. In this work, we chose to focus on creating a user-oriented output, focused on the user experience of consuming, navigating, and understanding the content, as opposed to creating machine-readable connections.

2. To combine Natural Language Processing with modern Note-Taking Tools to generate automatic connections that enhance human understanding and navigation of knowledge.

The combination of these two fields is central to the opportunities explored in this dissertation. This combination is already being explored in industry tools, but the structure provided by an academic research project may be a great contribution for further developments and for better integration of NLP with PKM and note-taking tools.

3. To present the idea of automatically using shared concepts as a navigation device and an integral part of a text navigation system.

Concepts play a central part in learning, communication, elaboration, and all aspects involved in Knowledge Management, yet it is **not** a common practice to use concepts as a means for navigation and exploration.

1.5 Dissertation Structure and Organization

The remaining of this dissertation is organized as follows: chapter 2 presents the NLP tasks that are foundational to this work. The chapter 3 presents the Context for the Research carried out in this dissertation, diving

deep into the inspirations and theoretical foundations for the proposed system. The chapter 4 discusses the Related Works to this dissertation, passing through Concept Extraction, Knowledge Bases, and Semantic Relatedness.

The Methodology is spread throughout three different chapters, chapter 5 presents the methodology for extracting concepts from text and finding relationships between concepts, while chapter 6 explains the procedures for calculating the Semantic Relatedness between texts. Finally, chapter 7 details the proposed methodology for combining the information extracted from previous chapters to build the system for generating connections using NLP in combination with modern Note-taking tools.

To end the dissertation, chapter 8 presents the Evaluation, together with discussions on potential use cases, and chapter 9 presents the main contributions and future works.

> "We shape our tools, and thereafter our tools shape us." – Marshall McLuhan

2
NLP Foundations

This chapter presents the theoretical foundations for this dissertation, focusing on the NLP tasks used and mentioned throughout the text. This chapter has the objective of providing the necessary background to understand the proposed methodology and the intended uses for the system this dissertation presents.

The chapter includes basic definitions and use-cases that will be presented for the relevant sub-fields of Natural Language Processing, as well as detailed explanations of specific tasks, and concepts belonging to this theme.

Natural Language Processing (NLP) is an area of research and application that explores how computers can be used to understand and manipulate natural language text to perform useful actions.

Although formally, NLP may be considered a branch of Artificial Intelligence, it is not totally contained within any given field because of its strong multidisciplinary aspects. The foundations of NLP lie in between a number of disciplines, it is an overlap between computer and information sciences, linguistics, mathematics, psychology, and artificial intelligence.

Below, the sub-fields and tasks of NLP that are considered relevant and essential for the understanding of this dissertation are presented.

2.1
Information Extraction

First and foremost, we present the sub-field of NLP called Information Extraction (IE). This is a somewhat broad field, with a couple of tasks belonging to it, some of which will be mentioned below.

Information Extraction is defined by (Sarawagi, 2008) as the automatic extraction of structured information such as entities, relationships between entities, and attributes describing entities from unstructured sources.

As this definition suggests, IE revolves around manipulating data regarding entities and adding structure to them, the most common structure is that of relationships between entities.

2.2 (Named) Entity Recognition

Named Entity Recognition and Classification (NERC) was one of the first tasks formally defined by Information Extraction and is comprehended as seeking to recognize and classify named entities mentioned in unstructured text.

The task is called "Named" Entity Recognition (NER) because it was originally directed at restricting the scope of the task to entities that were considered to be rigid designators, which include proper names, as well as some entities that are easily distinguishable, such as biological species and substances, (Nadeau and Sekine, 2007).

The original types for classifying Named Entities were those of Organization (ORG), Localization (LOC), Person (PER), and Miscellaneous (MISC). Since the first problem definitions of NER, additional types were added to the task, usually by the name of Fine-grained Entity Recognition, which splits the broader types into fine-grained types.

Though many fine-grained types were introduced, there were not as many new types of entities added to the task of Entity Recognition, usually being a further division of the original named entities types.

A common mathematical definition for the task of Entity Recognition is treating it as a Sequence Labelling problem. Of assigning a tag to each word in the text(sequence) with their corresponding entities.

This labeling usually follows the BILOU tagging scheme, which corresponds to the Beginning, the Inside, and the Last tokens of multi-token entities as well as Outside tokens (not an entity) and Unit-length entities.

Lastly, it is worth mentioning the task of Concept Extraction, as a sub-task of Entity Recognition. Concept Extraction is defined as classifying words in phrases into sequences of concepts and non-concepts. According to (Parameswaran et al., 2010), concepts are useful by providing standalone information, while random non-concepts are not.

2.3 Mentions and Entities

Before jumping ahead into other sub-tasks of IE, it is important to outline some important relationships and distinctions between Entities and Mentions.

Entity (e): An entity is any abstract or concrete object of fiction or reality. Entities are "usually" linked to a URI (Unique Resource Identifier), which represents that entity in a reliable and unique way.

Mention (m): A mention of an entity in a data source (usually text) is

a string intended to denote an entity. (Weikum et al., 2021)

When an entity label appears in a text, that appearance is called a mention of that entity. The text represented in the mention is called the surface form of the entity. (Weikum et al., 2021)

It is worth noting that two different entities can be represented by the same surface form. This is the case for hypernyms and for proper names like "Washington", which may be referring to many different entities.

While a mention may initially be associated with more than one entity, this is not the intended end result. To solve this issue, the task of Entity Linking is used.

2.4 Entity Linking

Entity Linking is a sub-task of Entity Canonicalization. Canonicalization can be defined as the process of converting data that involves more than one representation into a unique, standard approved format.

When we apply this to Entities, the task becomes that of creating one single representation for all observations of the same entity, regardless of name variants.

A simplified way to explain this, using the terminology of "mentions" and "entities" is to map every mention into an entity.

The specific sub-task of Entity Linking has an additional restriction that makes the process easier. Entity Linking happens only when entities belong to an existing KB that already has canonicalized entities.
The task of Entity Linking can be represented by two steps:

- Observe a new set of mentions
- Link the mentions into known entities.

The Figure 2.1 shows an example of this task being executed, by comparing the mentions in the text with possible entities they may map to, (Weikum et al., 2021).

2.5 RDF → Resource Description Framework

RDF is a standard model for data interchange on the Web. RDF extends the linking structure of the Web to use URIs to identify not only the relationship between resources but also to identify the two resources at the end of a link.

The basic structure of RDF is Triples, following the format of Subject, Predicate, and Object (SPO).

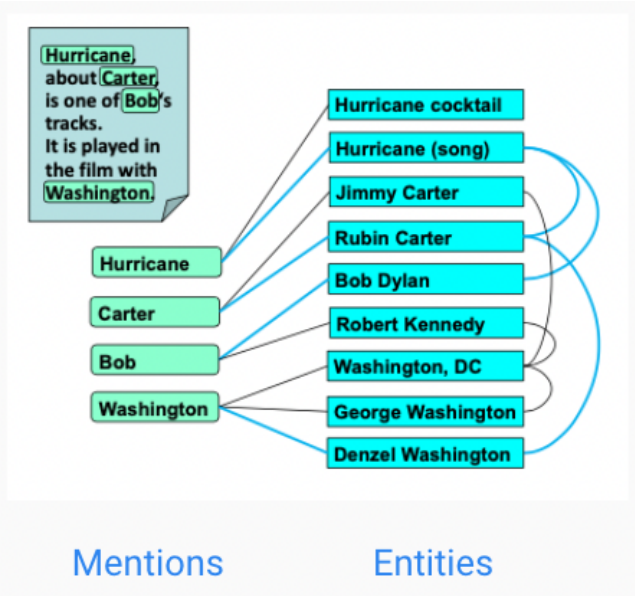

Figure 2.1: Explanation of the Entity Linking task

RDF is a great tool to map relationships because of its simplicity, flexibility, and underlying structure. The linking structure forms a directed, labeled graph, where the edges represent the named link between two resources (entities), represented by the graph nodes.

When applied to entities and relations between entities, the RDF model restricts the three roles in a subject-predicate-object triple as follows:

- S must be a URI identifying an entity.
- P must be a URI identifying a relation.
- O must be a URI identifying an entity (or a literal denoting an attribute value).

In the context of this dissertation, the RDF triples format will be used to represent the relations between concepts, and also to represent the edges between text and concepts.

2.6 Relation Extraction

The task of Relation Extraction is an important complement to Named Entity Recognition within the field of Information Extraction, and in some way, Relation Extraction depends (or includes) on the task of Entity Recognition (or Concept Extraction) to be able to work.

The formal definition for Relation Extraction is very much aligned with its name: Extracting the semantic relationships, between two or more entities from natural text, (Bach and Badaskar, 2007).

The resulting output for relation extraction is in the format of RDF triples. Where the Subject and Object positions represent entities (or mentions) and the Predicate position represents the relations extracted between them. An example of an extracted relation is presented in Figure 2.2, Where the Orange and Blue highlights represent entities, and the green one represents the relationship between them.

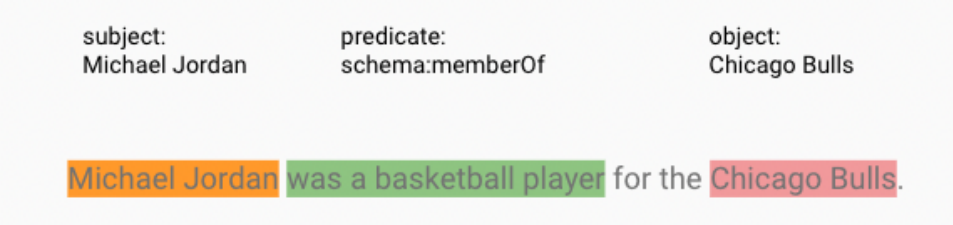

Figure 2.2: Example of RDF triple displaying an Extracted Relation

Relation Extraction methods rely on Part-of-Speech (POS) information to identify linguistic and lexical patterns which provide information on potential relations between entities, (Asghar, 2016).

When talking about Relation Extraction, it is important to mention the task of Open Information Extraction (OpenIE), which was proposed by "Open Information Extraction from the Web", (Etzioni et al., 2008) and describes the Relation Extraction task in open domain contexts, where there are no pre-defined entities (as it would be the case in the Entity Linking task).

2.7 Knowledge Bases

A knowledge base (KB) is a collection of structured data about entities and relations. Knowledge bases usually follow the RDF Schema for representing data on entities and relationships.

Knowledge bases may be of encyclopedic nature, looking to present a broad representation of knowledge, and can also be domain-specific, looking to represent specific information regarding a specific use case.

Some of the expected characteristics of a reliable knowledge base are the presence of data on relevant entities and relations, as well as their respective types, a high-quality and high precision of this information, and the feasibility of scaling to higher volumes while maintaining a flexible and structured schema.

Knowledge Bases are an important component of the Linked Open Data (LOD) initiative, by being a way of portraying large amounts of data that can be accessed publicly. The Linked Open Data is defined by Tim Berners-Lee as

"Linked Data which is released under an open license, which does not impede its reuse for free.".

It is worth mentioning the SPARQL query language, which is the query language for working with RDF. SPARQL can be used to express queries across diverse RDF data sources, and are especially relevant to accessing the information contained inside Knowledge Bases.

Some of the most important (and relevant to this dissertation) Knowledge Bases at the time of writing are DBpedia (Lehmann et al., 2015), Freebase (Bollacker et al., 2007), YAGO (Rebele et al., 2016), WordNet (Miller, 1995), BabelNet (Navigli and Ponzetto, 2010), and ConceptNet (Speer et al., 2018).

2.8 Semantic Similarity

The task of Semantic Similarity is defined by identifying the degree of similarity between the meaning of various text components like words, sentences, or documents.

When applied to text, the task is called Semantic Textual Similarity (STS) and is defined as the measure of semantic equivalence between two blocks of text, (Chandrasekaran and Mago, 2021).

Semantic Similarity is a huge advance when compared to Syntactic Similarity, since it captures the meaning contained inside a given text, instead of capturing only the lexical similarity between texts, i.e. string level similarity.

2.9 Semantic Relatedness

Semantic Relatedness is a more general notion than Semantic Similarity between concepts. Semantic Relatedness refers to human judgments of the degree to which a given pair of concepts are related, not similar or equivalent, (Pedersen et al., 2007).

The main difference here is based on the meaning. Semantic relatedness is a measure of how the meanings are related, while semantic similarity is a special case of semantic relatedness, which is tied to the likeness (in the shape or form) of the concepts, (Pedersen et al., 2007).

A measure of semantic similarity returns a numeric score that quantifies how much two concepts are alike, usually based on is-a relations (Resnik, 1995), which are closely attached to the actual meaning of each concept, not as much to the semantic context.

Relatedness is based on the context, on the semantic field, whereas similarity is based on the semantic definition.

An example to illustrate this distinction is presented in Figure 2.3.

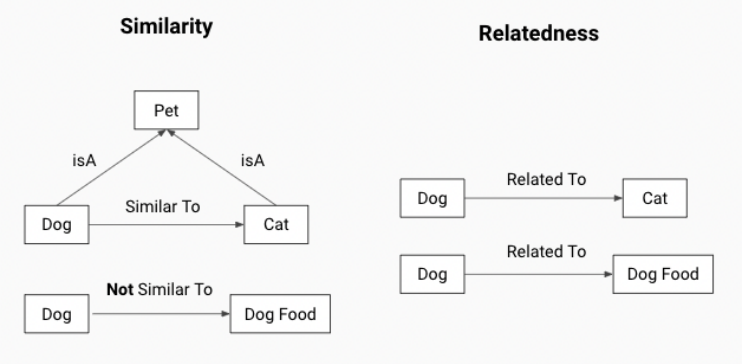

Figure 2.3: Illustration of the difference between Semantic Similarity and Semantic Relatedness

In this dissertation, the major interest is in Semantic Relatedness, which is broader and captures the most possibilities for connecting pieces of text.

2.10 Word Embeddings

An important definition to close out this section is word embeddings. An embedding is an instance of some mathematical structure contained within another instance. A word embedding is the representation of a word within a mathematical structure of a vector.

Word embeddings are vectors, defined as a numeric representation of natural language text which encodes the semantic meaning and context of such text. In simpler terms, word embeddings are vectorial representations of words.

Word embeddings are considered to be especially useful for the tasks of Semantic Relatedness and Semantic Similarity because they enable a mathematical similarity to be calculated between any two given texts.

Recent word embeddings are able to capture the context around the word that they are representing, as is the case for the BERT architecture, which utilizes Bidirectional encoding to capture the words before and after any given word while representing their semantic meaning, (Reimers and Gurevych, 2019), (Devlin et al., 2018).

3
Research Context

This chapter will outline a rationale to explain in detail the inspiration for applying Natural Language Processing to generate connections within a text collection. The main influences for this task are twofold. First, the emergence of modern note-taking software tools, and second, the idea of enhancing human intellect, on individual and collective scales by means of knowledge management.

Both of these influences will be expanded in further detail, followed by an argument outlining the opportunity and positive impacts of generating connections between knowledge in a text collection, as opposed to following the traditional convention of keeping documents strictly separate from each other inside folders (dungeons), which makes it extremely hard to explore connections between ideas.

This chapter is further divided into the following sections, section 3.1 on Knowledge Management and Human Intelligence presents the main bodies of work and technological advancements that motivate this dissertation. Then, section 3.2 explains important definitions of Knowledge Management, Knowledge Representations, and Knowledge Visualizations, while section 3.3 explains the difference between Hierarchical and Networked Organizations of Knowledge, and details opportunities and tools for using Networked Organization.

3.1
Knowledge Management and Human Intelligence

3.1.1
New Generation of Note-Taking Software Tools

The last 5 years have been of extreme importance to the field of note-taking software apps, also commonly denominated as "*Tools for Thought*". A huge part is due to the recent emergence of "*Networked Note-Taking*", best

exemplified by apps such as Roam Research[1], Obsidian[2], Tana[3], and Mem[4]. This category of apps presents a new set of tools for humans to capture, synthesize and create knowledge, which is based on features such as bidirectional hyperlinks and transclusion, which increase the ease of communication between separate notes. This enhances the note-taking and sense-making capabilities that were previously accessible to the public.

The major distinction these apps present is that of non-linear thought, that is, thinking in multiple directions while still being able to trace back thoughts to their original starting place. This is possible due to a structured way of storing and handling information which is well known to information science: Graphs.

By treating information using a graph knowledge structure, it is possible to connect notes (or any text) in a non-linear and non-hierarchical way, in contrast to the purely hierarchical standard operating procedures for note-taking apps not so long ago. This evidently opens up new possibilities for storing, manipulating, presenting, and visualizing knowledge, which will be explored in further detail in this chapter.

The appearance of Networked Note-Taking tools is relevant because they are closely related to the field of Knowledge Management and the benefits that the area provides.

3.1.2 The Benefits and Opportunities of Knowledge Management

While the term "Knowledge Management" was originally used to portray Knowledge Management within organizations and enterprises, here in this dissertation, Knowledge Management simply refers to the larger encompassing field of dealing with Knowledge, which includes the sub-fields of Personal Knowledge Management, Collective Knowledge Management, Corporate Knowledge Management, and any subsequent variations.

That said, on the surface level, Knowledge Management can be defined as the explicit and systematic management of processes that enable knowledge resources to be identified, created, stored, organized, shared, and used. (Girard and Girard, 2015) (Serrat, 2017)

On a deeper level, though, Knowledge Management is about much more than organizing knowledge. It is about extending the human mind's cognitive abilities. The Extended Mind Thesis, presented in (Clark and Chalmers, 1998)

[1] https://roamresearch.com/
[2] https://obsidian.md/
[3] https://tana.inc/
[4] https://get.mem.ai/

argues that both human cognition and mind, are a result of internal AND external entities that hold two-way interactions between one another. This creates a coupled system that can be seen as a cognitive system in its own right.

This idea is very well represented by the quote presented in the chapter 1, "We shape our tools, and thereafter our tools shape us." The emphasis here being in the fact that "**thereafter our tools shape us**", this shaping is happening on a cognitive and psychological level, by enhancing or degrading mental capabilities.

This means that by following Clark and Chalmers's interpretation of the human cognition and mind, Knowledge Management is about extending human's mental capacities by improving the external entities with which our internal cognition interacts with.

One possible mechanism that fits into this description is the creation of an external and trusted collection of knowledge, that can be accessed at any moment to recollect several types of knowledge.

This idea has been a topic of interest for, at least, as long as 70 years. Originally surfaced by engineer and scientist Vannevar Bush's (1945) reflection: As We May Think, right after WWII, (Bush, 1945). In his reflection, Bush proposes a curious mechanism he called the "Memex", which "would closely mimic the associative processes of the human mind, but it would be gifted with permanent recollection". Since "permanent recollection" is not guaranteed, the word "persistent" was chosen to describe such knowledge collection.

3.1.3 A Persistent Collection of Individual Knowledge

Bush's work is one of the first known inspirations for the topic of extending human intellect with a Persistent Collection of Knowledge, as it was also one of the first inspirations for the Hyperlink system we use today. Bush was certainly ahead of his time while portraying a device in which individuals would compress and store all of their books, records, and communications as far back as 1945. Even more impressive, is the fact that he anticipated mechanization (automation) of such a system, so that it "may be consulted with exceeding speed and flexibility".

Bush also referred to the idea of associative trails, he uses this term for the action of connecting different items through trails of thought, which can be saved and accessed as a representation of a new piece of more complex knowledge. He even proposes a new profession, of "trail blazers, those who find delight in the task of establishing useful trails through the enormous mass

of the common record".

This is the perfect moment to express the clear convergence and overlap of ideas behind the functionalities of modern note-taking tools and Bush's ideas of navigation, association, and connections. Bush, however, was not the only person thinking about an external collection of knowledge.

Another important influence and precursor of the idea of having a persistent collection of Knowledge is German sociologist and philosopher, Niklas Luhmann. He utilized a physical index card system for thinking and connecting notes, explained in (Ahrens, 2017) as the Zettelkasten, in German, or the Slip-Box in an English translation. This framework for thinking and working with his own collection of knowledge is linked to his astounding literary production. (Schmidt, 2014).

Niklas Luhmann was a prolific contributor to academia, Ahrens, in 30 years, he published 58 books and hundreds of articles, translations not included. He wrote on law, politics, economy, communication, art, education, and epistemology. After extensive research on Luhmann's workflow and contributions, the German sociologist Johannes F.K. Schmidt concluded that his productivity could only be explained by his unique working technique, of using an external collection of his knowledge (Schmidt, 2014).

This is a strong argument in favor of the Extended Mind Thesis, where Luhmann was a better contributor than his colleagues, not by what he had inside his mind, but by the external entities with which his mind interacted with.

Fast-forward to 2022, contemporary thinker and digital productivity expert, Tiago Forte has recently released his book titled Building a Second Brain, (Forte, 2022). Forte's book was written after teaching thousands of students the skill of, in his words, building a "Second Brain", i.e. creating an external, Persistent Collection of Knowledge to act as an extension and ally to our biological human brain.

Some of the proposed benefits of building, and engaging with, a Persistent Collection of External Knowledge are:

- **The rediscovery of one's knowledge**. As humans, we are prone to being victims of the Recency bias, or the Serial Position Effect, (Ebbinghaus, 1964), where we unconsciously recall recent discoveries or memories better, (Murre and Dros, 2015). Having one's "past" knowledge safely stored in an external collection presents a solution to this problem by allowing for rediscovery.
- **Revisiting and digesting one's ideas**. (Ahrens, 2017) suggests that taking further notes on what one has learned is a form of deliberate

practice, as it gives feedback on one's understanding of a subject or lack of it.

- **Connecting different ideas (items of knowledge) through associative trails**. By having a storage place for knowledge, one can connect any new (or old) idea to any existing piece of knowledge. This is much easier with modern note-taking tools.
- **Lead more fulfilling lives**. In addition to remembering more and thinking better, integrating technology with our biological brains, and expanding our extended mind can result in a life of more fulfillment, (Forte, 2022).

With regard to the last and most debatable item of leading a more fulfilling life, Bush presented a very similar opinion.

> "Presumably man's spirit should be elevated if he can better review his shady past and analyze more completely and objectively his present problems. He has built a civilization so complex that he needs to mechanize his records more fully if he is to push his experiment to its logical conclusion and not merely become bogged down part way there, by overtaxing his limited memory. His excursions may be more enjoyable if he can reacquire the privilege of forgetting the manifold things he does not need to have immediately at hand, with some assurance that he can find them again if they prove important."

Since the idea of a Persistent Collection of Knowledge is proposed and revisited by a number of thinkers, and allegedly brings several potential benefits, a follow-up question emerges. **"How can one represent their knowledge externally in a persistent and comprehensive way?"**.

This question will be the subject of section 3.2 on Representing and Visualizing knowledge.

The idea of creating and using an external collection of knowledge for thinking is not exclusive to personal use, such as the "Memex", proposed by Bush, the "Zettelkasten" by Luhmann, and the "Second Brain", by Forte. Engineer and inventor, Doug Engelbart was a proponent for using external representations to enhance collective knowledge management.

3.1.4 Collective Knowledge Management and Bootstrapping Paradigm

Another branch of Knowledge Management that has been a source of inspiration for the system proposed in this dissertation is that of Collective Knowledge Management.

Proposing a way to connect knowledge originating from different people is a challenging task, but one which has the potential to generate positive

returns. This subsection outlines the ideas by Doug Engelbart as a source of inspiration for the creation of a system that can connect different pieces of knowledge inside a text collection.

Doug Engelbart describes a system for managing collective knowledge in (Engelbart, 1992) called the CODIAK, The "COncurrent Development, Integration and Application of Knowledge", for which the central component was the Dynamic Knowledge Repository (DKR), an external collection of knowledge.

An interesting aspect regarding the CODIAK and the DKR is that they are a centerpiece for Engelbart's idea of focusing on managing collective intelligence, what he would refer himself to as the "Collective IQ".

Engelbart describes the Collective IQ as being a special set of collective capabilities, similar to individual ability to solve problems, Collective IQ is built upon the collective perceptual, motor, and cognitive abilities applied to solving problems. Part of Engelbart's theory is that significant collective capability is only reached by "augmenting" the basic human capabilities, (Engelbart, 2004).

Engelbart goes further into the idea of augmenting and explains a central concept of his work, which is Bootstrapping. This is the term he chose to describe the idea of improving the improvement process, (Engelbart, 1992). In other words, Engelbart envisioned that the better a collective group gets, the greater the improvement rate will be.

In order to augment the collective capabilities, Engelbart identified two separate systems which come together to compose a larger Augmentation System:

1. The Tool System: Appropriately coordinated systems of artifacts and tools.

2. The Human System: Vocabulary, conventions, roles, organizational structures, rules of conduct, methods of cooperation and education, etc.

The most fundamental metric for Collective IQ is measuring the collective ability to handle knowledge. As described in (Engelbart, 1992), the aspects of Collective IQ that Engelbart considered most important are the process and the assets produced by that process, both are strictly related to knowledge.

1. **Process**: How well did a group develop, integrate, and apply its knowledge? Was the process smooth, collaborative, and collective?

2. **Assets Produced by that Process**: How effective was the group's shared repository of knowledge? How easily could information be syn-

thesized, stored, retrieved, and updated? How coherent was the group's shared vision of the problem and its potential solutions?

The Tool System is a fundamental part of improving human capabilities and the improvement itself. It is no surprise to see Engelbart and Lehtman dedicate a large amount of his works describing Software Sytems, such as the Open Hyperdocument System (OHS), Engelbart (1990), the On-Line System (NLS) Engelbart and Lehtman (1988), the CODIAK, Engelbart (1992) and Dynamic Knowledge Repository (DKR), Engelbart (2004).

It may be interpreted that a crucial aspect of the Collective IQ Bootstrapping is improving the systems for collective thinking. The question that arises from this observation is: **"How can Knowledge from multiple people be connected and visualized for collective understanding?"**

Another interesting aspect of the Bootstrapping Paradigm is that Engelbart suggests that teams should be interdisciplinary, with stakeholders from various domains.

3.1.5 Interdisciplinary Pursuits

Although Engelbart clearly understands the importance of connecting people from various domains, he does not dive very deep into Interdisciplinary possibilities. Fortunately, there are several authors that have been outlining the importance of Interdisciplinary pursuits, (Frodeman, 2017), (Klein, 2015) (Brew, 2008) (de Souza, 2021);

One of the most relevant works on this topic is the Transdisciplinary Manifesto, (Nicolescu, 2002). A strong connection point between the ideas of Engelbart and Nicolescu is presented in Article 3 from (Nicolescu, 2002), which states that:

> Transdisciplinarity complements disciplinary approaches. It occasions the emergence of new data and new interactions from out of the encounter between disciplines. It offers us a new vision of nature and reality. Transdisciplinarity does not strive for mastery of several disciplines but aims to open all disciplines to that which they share and to that which lies beyond them.

This quote provides strong arguments supporting the usefulness of Knowledge Management and the generation of connections between texts from different disciplines. The "encounter between disciplines" is a huge source of data for Transdisciplinary purposes, and is one of the intriguing opportunities that this dissertation seeks to explore, to connect ideas from different areas,

and take an interdisciplinary approach when proposing connections between knowledge.

3.2 Knowledge Representation and Visualization

At this point, the benefits of and reasons for diving into the field of Knowledge Management are very much clear. This opens up the question of "How" may knowledge be presented, organized, represented, and visualized in the best possible way?

This section presents fundamental definitions of knowledge management, knowledge representation, and knowledge visualization in preparation for the final section, where we make an argument for interconnected and networked knowledge representations and against hierarchical and "separating" knowledge representations, which would be the traditional functioning of computer folders which separate documents and files from other documents and files.

Before carrying out the comparison between the two categories of Hierarchical and Networked organization for representing and visualizing knowledge, we will explain the processes involved in Knowledge Management and also outline the fundamental models to represent and visualize knowledge. These are important steps to solve the challenges of presenting effective tools to support the process of Knowledge Management, be it Personal or Collective.

3.2.1 Definitions of Knowledge Management and Visualization

Before diving into the different ways to represent knowledge, we first present a more detailed definition of Knowledge Management and understand what are the processes involved:

Building on top of the previous definition by (Serrat, 2017): Knowledge management is the explicit and systematic management of processes that enables knowledge resources to be identified, created, stored, organized, shared, and used.

In order to better understand this definition, it is possible to differentiate between four main processes in Knowledge Management, according to (Alavi and Leidner, 2001):

1. Knowledge Creation
2. Knowledge Storage and Retrieval
3. Knowledge Transfer
4. Application of Knowledge

These distinctions between different processes of knowledge management are extremely important as they will lead to different forms of representation, organization, and visualization for each type of process.

This dissertation will focus on two processes, 1. creation, and 2. storage and retrieval. Different methods to represent, organize, and visualize Knowledge will be considered from the perspective of storing and retrieving knowledge with the objective of creating new knowledge.

The created or retrieved knowledge could be further transferred to somebody else or applied in any given way. Again, the ideas of Collective Intelligence and Collective Knowledge representations, and Bootstrapping human capabilities are only an inspiration to this dissertation and not the direct scope of what will be presented.

An important observation to be made when talking about these three tasks of representing, organizing, and visualizing is the fact that Knowledge Representation and Visualization are very connected in one way or another. There are individual definitions and taxonomies for each of these specific actions, yet they have many common aspects and are present in one another.

There is also an interesting distinction to outline, the difference between Information Visualization and Knowledge Visualization.

Information visualization is defined by (Card et al., 1999), as the use of computer-supported, interactive, visual representations of abstract data to amplify cognition. In other words, Information visualization is usually concerned with presenting visually a set of data.

Knowledge Visualization, however, "examines the use of visual representations to improve the transfer and creation of knowledge between at least two persons" (Burkhard, 2005). In simpler words, Knowledge Visualization refers to techniques that do not primarily visualize existing data, but knowledge, which a priori resides in one person's mind and is being restructured or resurfaced in another person's mind. Note that the intended recipient of a visual knowledge artifact may be the author himself, at a later moment in time, not necessarily needing at least two, but one person.

The important distinction here is the fact that knowledge visualization is reliant on human interpretation, and ultimately on human psychology. This means that how humans perceive information is important, but there is special interest in how it will be interpreted, and the thought process behind it.

3.2.2 Basic Knowledge Representation Models

The most basic representation of knowledge, as described in (Van Harmelen et al., 2008), is Classical Logic, where the most common choice is First-Order Logic, using predicates. Mathematicians and philosophers use representation through Logic for centuries, this is the foundation for representing thoughts and ideas, as well as the building block for other representations which extend logical statements.

Other than the Logical representation, which may even be considered to be a language in which to represent knowledge, there are three basic ways to represent knowledge, Spatially, with Features, and using Networks, as explained in (Markman, 2013).

Spatial Models of representation are those which rely on visual or spatial representation of information, where all of the encoded knowledge is visually accessible and within reach. This makes Spatial Models effective for small amounts of knowledge, but not as suited for larger amounts of knowledge, since it depends solely on space to encode information, which is not enough for either large amounts of depth (detail) or breadth (reach).

Feature Models of representation use symbols to represent different types of information. Features act much like a sign in Semiotics theory, they communicate a meaning that is not the sign (feature) itself, to an interpreter. A feature is a symbol, an entity, or an object in the representing world, which corresponds to a "real object", or simply another concept being represented by it. Features may be visual, conceptual, and mathematical.

Network Models of representation use a graph structure of nodes and links to convey knowledge. The critical aspects for representing networks are the structure of the networks (nodes and links), together with the labels on the nodes and links. A very important representation based on Network Models is that of Semantic Networks, which are composed of nodes representing concepts and links representing relations between concepts.

An important aspect of Semantic Networks is that it allows for the propagation of relations, which is a generalization of inheritance. What this means is that Semantic Networks are a proposal for the structure of long-term memory. Memory is searched automatically by passing activation across the links, and analogously, Semantic Networks allow the activation of one concept to activate many other concepts as well.

There is also an additional layer that may be applied to extend each of the three basic representations, which is the Structured Representations. This usually combines representations, which end up being used alongside one

another, connecting them in some ways.

A Structured Representation basically adds more logic to the basic representations. Some examples would be providing more detailed features; combining different elements; organizing existing features; providing more structure for connections; as well as other more elaborate representations.

These three knowledge representations are fundamental. In pure essence, they are biological and psychological, but certainly end up being relevant for digital purposes. This goes back to the importance of how humans will interpret information, such interpretation happens on the biological and psychological levels.

3.2.3 Knowledge Visualization

For any representation of knowledge to be consumed and understood by a human, it must first be visualized, even if conceptually visualized only in the subject's mind.

Burkhard (2005) outlines a powerful framework for categorizing the transfer and creation of knowledge. Though the framework may also be used for the other two processes, of storage and retrieval, and of knowledge application.

The framework is based on four specific questions:

- Why should knowledge be visualized? (aim)
- What type of knowledge needs to be visualized? (content)
- Who is being addressed? (recipient)
- Which is the best method to visualize this knowledge? (medium)

From these questions, the Knowledge Visualization Framework is created, it presents four perspectives that need to be considered when creating visual representations: Function, Knowledge Type, Recipient, and Visualization Type.

Function perspective answers why a visualization should be used, a knowledge type perspective clarifies the nature of the content, a recipient type perspective points to the different backgrounds of the recipient/audience, and finally, the visualization type perspective structures the main visualization types according to their individual characteristics. The elements of the Knowledge Visualization Framework are presented in Figure 3.1.

As stated in subsection 3.2.1, this study will focus on the Creation and on Storage and Retrieval processes of Knowledge Management. This corresponds to the Function perspective, which would be best represented by Recall, Elaboration, and New Insight.

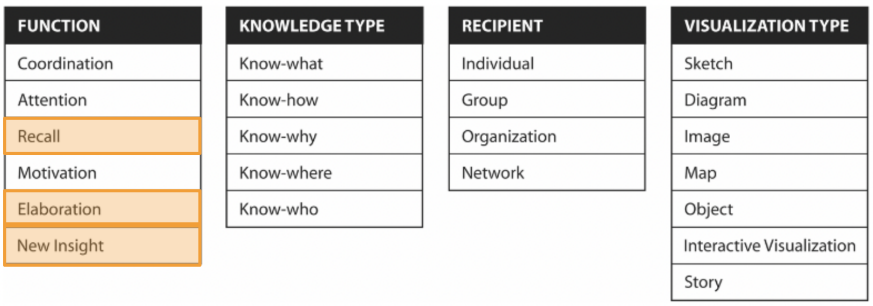

Figure 3.1: Knowledge Visualization Framework, from (Burkhard, 2005)

The selected Knowledge Visualization Functions of Recall, Elaboration, and New Insight are important guiding principles for this dissertation and will be referred to frequently as means of directing the analysis and discussions presented.

The remaining perspectives are not as important as the Function. The Knowledge type is mainly **Know-what**. The recipient is an **Individual**, which is frequently the author himself. The visualization type is closely related to the knowledge representation models, specific possibilities and combinations will be outlined in the next subsection.

3.3 Hierarchical and Networked Organization

This section will outline two existing paradigms for organizing knowledge digitally, as means to Represent and Visualize Knowledge. First, a purely Hierarchical Knowledge Organization, while also detailing some limitations of this approach. Second, in explaining the methods for Networked Organization of Knowledge, by listing a few existing examples, their strengths according to the knowledge visualization functions, and also potential opportunities for applying automatic generation of knowledge connections within each method.

3.3.1 Hierarchical Organization - Folders and Documents

The hierarchical organization of information and knowledge is pretty intuitive, it closely resembles a spatial physical organization of organizing items into boxes, which is very effective for separating items into specific containers. Physically, this works well and this idea also represents an important principle behind the idea of Taxonomies, which literally means to “classify”. Classification, however, is fundamentally different from organization, yet the principles applied to organizing knowledge are in huge part equivalent to the principles used to classify organisms or any other category of entities.

This leads to a misleading principle, that knowledge should be organized by its existing characteristics, instead of by its potential uses. This would be analogous to organizing knowledge on bacteria as being unrelated to human beings, which taxonomically is true, but in practical terms is false, since bacteria play an important role in the functioning of the human body (Costello et al., 2009).

Following this principle, the most common way of storing and organizing knowledge is that of using folders containing documents, which in turn contain linear prose. The use of prose to represent knowledge is much related to the concept of using Classical Logic, by describing knowledge using words and predicates that seek to form an argument.

Prose is defined by Oxford Dictionary "as a written or spoken language in its ordinary form". If applied to knowledge representation, prose would be a descriptive report of a piece of knowledge. A linear sequence of words, which come together to convey ideas and form arguments. In the case of narrative prose, words are used to compose stories and to suggest analogies between ideas.

It is undeniable that the act of writing prose to convey knowledge requires a profound understanding of the knowledge in order to successfully represent the information by composing the knowledge in a linear prose format. In other words, putting ideas into linear, written arguments requires more skill than drawing a simple visual diagram with keywords.

The use of words to describe and store knowledge is very natural and is related to a high degree of understanding, yet with regard to actually representing the knowledge, there are some limitations, especially when dealing with Recall, Elaboration, and New Insight.

Firstly, knowledge represented using prose usually requires a larger time investment to be consumed. Following the 3 main knowledge representation models, Spatial, Feature, and Network, prose can be defined as a network of words, spatially organized in a way that indicates each word is directly connected to two other words, one behind and one ahead.

The problem with this is that the meaning a "sequence of words" conveys is directly dependent on the words around it, which would eventually correspond to all words inside the document needing to be accessed linearly in order to fully visualize the knowledge thereby represented. This means that unless a summary is provided, the time and effort required to retrieve knowledge stored inside a single document is considerably large.

Another major limitation is defined by the lack of easy interaction between knowledge from different documents. When knowledge is stored inside

documents, they are usually inaccessible from outside the document, one needs to open a document and access the knowledge (in a costly manner), and only then, start searching other documents for another piece of knowledge to interact with.

At this stage the situation tends to get worse, because potentially worthy connections to a given piece of knowledge are probably living inside seemingly unrelated folders, separated by year, by "the subject they belong to", or by any other separation metric used for hierarchical division.

As (Ahrens, 2017) explains, this is a huge challenge of using folders, one of the most popular information organization system people use. Folders are keeping things "in modular form, sorted by topic, separated by disciplines, and generally isolated from other information". There are very few means of easily connecting knowledge from different documents and different folders.

The two previous limitations are accentuated by a final limitation of scalability. When dealing with small amounts of data, these two limitations may be worked around, by employing the effort of going through all of the documents individually in order to study and compare the relevant information, but when the amount of data increases, or when the task is executed several times, the outlined limitations become huge challenges.

3.3.2 Networked Organization

The main principle behind a Networked Organization of knowledge can be boiled down to the basic elements presented in (Markman, 2013), the structure and labels of the nodes and links in the network.

That implies that Networked Organization is a generalization, and there is no unique method for representing knowledge using networks that stand out as being the most popular and undisputed reference for this category. That said, there are some popular methods, and also a couple of alternative solutions that follow the paradigm of organizing knowledge using networks.

An intuitive example of a Networked Organization is that of a Knowledge Graph, which closely resembles the idea of a Semantic Network. Knowledge graphs are probably the most popular method in this category, but curiously, its most common use-case is not human visualization of knowledge but representing knowledge in a machine-readable format.

When directed at human consumption and interpretation, two common formats of networked organization would be those of Mind Maps (Buzan and Buzan, 2006) and Concept Maps (Haller, 2011), which actually share a common limitation when compared to Hierarchical organization, that of scalability.

Mind Maps and Concept Maps both struggle to portray large amounts of knowledge in a limited space.

The next solutions were built or adapted specifically to be used together with modern Note-Taking Tools. Three solutions are presented, they are Zettelkasten, Maps of Content, and Discourse Graph. These are the state-of-the-art solutions for Personal Knowledge Management using Networked Organization methods.

Zettelkasten - The Slip-Box

The Zettelkasten is a principle for taking and organizing notes that was already mentioned in this chapter, here we discuss it in terms of practical functionalities and what functions of knowledge visualization are present in it.

The basic premises of the Zettelkasten, the Slip-box, are that the notes are written with the intent of being permanently stored, after the user had time to think about what is being written, and most importantly, with the possibility of being connected to any other note in the slip-box.

The slip-box works by creating a set of notes while reading or thinking about a subject, these notes are considered to be Literature Notes, they serve the function of capturing initial ideas based on a literary source.

The Literature Notes are further processed and condensed into Permanent Notes, which portray the opinions of the user regarding the content of the Literature Notes among the user's personal interests and interpretations. The Permanent Notes are the final format of knowledge, which are then connected to other permanent notes by means of hyperlinks.

The process of revisiting ideas, thinking about them, and connecting them with other notes is what makes the Zettelkasten unique; this forces users to elaborate, understand, connect, and therefore, learn seriously. (Ahrens, 2017)

There are two main types of connections between notes in a Zettelkasten:

1. Connecting a note to a merely related note, in order to be able to resurface this connection later. This may be done even if these two notes are not about the same topic. The idea is to be exposed to divergent thinking, being able to gather insights even from seemingly unrelated topics, which contain a similarity in the way of thinking.

2. Connect notes that are intimately related and compose a specific train of thought, similar to what (Bush, 1945) refers to as trails of thought. This is usually the case when a note is written with an intention of being attached to another note. An example would be that Note B contains an explanation for Note A, and Note C elaborates on a specific detail of Note B. The train of thought would be composed of A $\rightarrow$ B $\rightarrow$ C.

An interesting opportunity for generating connections with this style of organizing knowledge is regarding the first way of connecting notes. When a new note is inserted into the Zettelkasten, it is possible to suggest recommendations for connecting it to any of the other existing notes.

With regard to the Knowledge Visualization functions, the Zettelkasten presents a unique way of navigating through knowledge by leveraging and incentivizing hyperlinks between notes. The organization structure is perfectly designed for Elaboration and New Insight since one of the major principles is to contrast new entries to the system with other existing notes.

The function of Recall would seem to be a challenge initially, but there is a workaround for this challenge, which are entry notes, notes that act as a contents page, or entry points to a specific topic, where relevant notes are mapped out for easier recall, which leads perfectly to the next knowledge organization method.

Maps of Content

Maps of Content is an idea originally presented in (Kimbro, 2003) that was recently revived for use with modern note-taking tools in (Milo, 2022). It serves as a means of mapping out all of the different topics that are covered inside an overarching corpus of several documents.

This method works exclusively with modern note-taking tools, where it is possible to create hyperlinks to other documents, and access their content by simply navigating to the desired document.

The functioning is exactly like a table of contents works, but instead of referring to elements inside one single document, the Map of Contents may have references to all of the documents in the corpus, which are easily accessible, as a means of organizing knowledge.

This works similarly to a hierarchical organization by classifying documents into a determined section of the map, but rather than belonging to one single “folder”, documents may belong to many different trails of thought or topics.

This way of organizing knowledge is amazing for the task of Recall, due to having the documents mapped out in an expanded view. This method does not stand out as being extremely exciting for Elaboration and New Insight, but it opens up opportunities for these tasks, by having other documents easily accessible.

This type of organization surely could benefit from the automatic generation of connections between documents. By adding connections, an additional layer of navigation could be built. This would represent adding machine-generated interconnections to a human-made Map, bringing diversity to possi-

ble Elaborations and New Insights, while still being possible to maintain both versions of the map independently.

Discourse Graph - Knowledge Synthesis

The last method for organizing knowledge using networked organization is the Discourse Graph, which is actually a method focused on Knowledge Synthesis. The Discourse Graph is proposed in (Chan, 2020) and is accompanied by an official software extension by the author for the Roam Research app.

The idea behind the Discourse Graph is to facilitate Knowledge Synthesis for researchers, by laying out a data model for organizing knowledge.

The Data Model is composed of 4 types of notes:

- **Question** notes
- **Synthesis** notes
- **Observation** notes
- **Context snippet** notes

Additionally, these note (node) types have a system of relationships between them, as illustrated by Figure 3.2

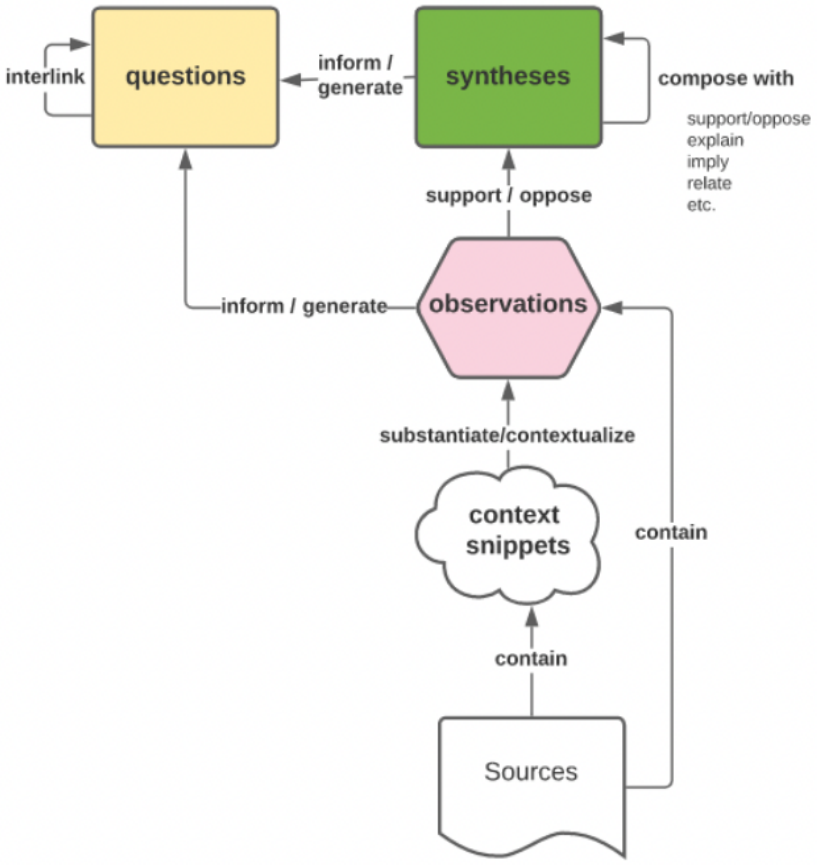

Figure 3.2: Relationships between nodes for the Discourse Graph, (Chan, 2020)

Together, these 4 types of notes and the relationships between them provide a framework for thinking externally about a given topic, while organizing knowledge in a format that allows for rich layers of context and aids synthesis.

While analyzing this method by looking at the Knowledge Visualization functions, it is possible to say that this method was designed with the specific functions of Elaboration and New Insight in mind! The proposed Data Model is perfectly suited for these two tasks.

The Recall does not seem to be the focus of the data structure, but given that it is oriented towards elaboration and understanding of a topic, and that a higher recall tends to be a byproduct of elevated understanding, the recall also is enhanced when compared to many other knowledge organization methods.

Automatic suggestions of knowledge connections in a context related to knowledge synthesis is a delicate issue, but there still seems to be space for machine-generated connections to be useful, especially when a given Discourse Graph reaches a larger size. The suggestions should be more on the exploratory side, and less on precise recommendations following the relationships taxonomy proposed in (Chan, 2020).

After discussing different opportunities for organizing knowledge, it may be stated that there are huge opportunities available when dealing with applying automatic generation of knowledge connections to existing Networked Organization methods for representing knowledge.

4
Related Works

This chapter contains a detailed explanation of the related works for this dissertation. section 4.1 starts by mentioning studies that combine external structured knowledge with semantic relatedness to perform somewhat similar tasks to the one described in this dissertation. section 4.2 outlines the current solutions for the tasks related to Information Extraction, specifically those of Entity Recognition, Concept Recognition, and Relation Extraction.

The rest of this chapter is organized in the following way, section 4.3 is focused on knowledge graphs and is the most related one to the overall workflow of this dissertation. It is separated into two sub-sections, the first, subsection 4.3.1, portraying the methods for transforming natural text into knowledge graphs, and the second, subsection 4.3.2 focusing on how to actually use knowledge graphs, or knowledge bases to produce relevant results.

Then, section 4.4 discusses the different possibilities to calculate the semantic relatedness between texts, also an important aspect of this dissertation. While section 4.5 concludes this chapter.

4.1
Similar Works in Literature

There are a couple of specific recent works that have been proposing the generation of connections between texts using external knowledge bases and semantic relatedness metrics between texts. This section briefly outlines the works that propose methodologies that are considered to be related to this dissertation.

Becker et al. (2021b) Proposes CO-NNECT, a framework that proposes connection paths between sentences using concepts mentioned and the relations between them. This work uses ConceptNet concepts, combined with language models trained on knowledge relations from ConceptNet to identify implicit paths between sentences. Maria Becker suggests this framework could be used for enriching texts and even enriching knowledge bases themselves, she also uses CO-NNECT to add knowledge path constraints to language models with an improvement in sentence generation representation of implicit knowledge (Becker et al., 2021c)

Dessì et al. (2021) Uses State-of-the-art NLP Tools and resources to create a Scientific Knowledge Graph representing knowledge contained inside academic productions in a natural language format.

Ilkou (2022) uses Entity Extraction and the DBpedia Knowledge graph to generate Personal Knowledge Graphs with specific e-learning users' personal information regarding learning profiles and activities, looking to enhance the learning experience.

Blanco-Fernández et al. (2020) presents a system that, given a question and a right answer, automatically generates wrong (yet related) answers to distract the user in multiple-choice questions. The system uses knowledge bases and semantic relatedness between texts.

Tomaszczyk and Matysek (2020) presents some of digital tools that may increase the efficiency of scientific research and facilitate conceptual work, information retrieval, note-taking and writing up of research. This paper is not focused on generating connections, but rather on digital (note-taking) tools and their functionalities, which are very important in the scope of this dissertation.

4.2 Information Extraction

This section presents existing solutions for extracting information on entities and the relationships between them. More importantly, this section sheds a light on extracting concepts from text, a task that represents a central piece of this dissertation.

4.2.1 Entity Recognition

The focus of this section is more on the problem definition of Entity Recognition rather than on available solutions. The field actually started by solving the Entity Recognition problem rigorously only for Named Entities. As the central survey in the field, (Nadeau and Sekine, 2007) explains, Named Entity Recognition and Classification (NERC) was aimed at restricting the task to entities that were rigid designators, another word for proper nouns and entities that are easily distinguishable (highly unambiguous). The original problem of Named Entity Recognition, as defined in (Sang and De Meulder, 2003) aimed at identifying and classifying 3 types of proper entities, PER (person), ORG (organization), and LOC (location), as well as a MISC (miscellaneous) category for other proper entities.

Some notable datasets that portray new categories in them are Ontonotes, (Weischedel et al., 2013), which expands the total number of cat-

egories to 18 types. WNUT (Workshop on Noisy User-generated Text), 2016 focused on Twitter Entities (Strauss et al., 2016) and 2017 focused on rare entities (Derczynski et al., 2017).

Recent studies on Fine-Grained Entity Recognition also rarely depict conceptual or abstract entity types as a part of their taxonomy. The cases for (Ringland et al., 2019), (Ling and Weld, 2021), (Choi et al., 2018), (Ding et al., 2021), and (Gillick et al., 2016).

Named Entity Recognition is not suitable for dealing with knowledge in a broad sense. Maybe for corporate or other context-specific knowledge, but not when dealing with scientific knowledge or any broader sense of learning and understanding. Fortunately, there are two other promising areas of study that address the situation, namely Concept Extraction, and Relation Extraction.

4.2.2 Concept Recognition

The task of concept recognition has attracted attention for some time, notable tools were presented for this task over the years. (Mendes et al., 2011) introduces DBpedia Spotlight, which approaches concept extraction as a text annotation task, and is able to annotate mentions with DBpedia resources (which include abstract concepts).

Parameswaran et al. (2010) extracts concepts from corpora using a market-basket problem approach, it extracts the concepts by defining them as being "a k-gram that represents a real or imaginary entity, event or idea that many users may be interested in". While (Dalvi et al., 2009) extracts concepts from the web, using information from HTML elements as sources of information. This extraction method gives more importance to the concepts that are usually searched for in search engines.

There are a handful of recent studies that present important developments for the task of Concept Extraction.

Becker et al. (2021a) extracts ConceptNet concepts from natural text by applying a series of semantic manipulations to form candidate phrases, which are further matched and mapped to the ConceptNet concepts.

Chabchoub et al. (2018) presents an improvement to DBpedia Spotlight, by adding a Stanford NER component, (Finkel et al., 2005), and performing the disambiguation of the extracted concepts.

Waldis et al. (2018) extracts n-gram concepts in a given text using Convolutional Neural Networks. Finally, (Fang et al., 2021) proposes GACEN (Guided Attention Concept Extraction Network), which is a technique of attention networks feeding a CRF to extract concepts using the title, topic,

and clue words.

There is yet another alternative, a very important one for this dissertation. Recent Surveys on Named Entity Recognition techniques (Li et al., 2020) and (Canales and Murillo, 2017) both point to a set of industry-based tools. One of the tools mentioned in both surveys has the functionality of identifying conceptual entities.

The industry-based service that is capable of identifying conceptual entities is provided by Dandelion API (SpazioDati, 2012), which performs a high-quality identification of entities that are linked to datasets in the Linking Open Data (LOD) cloud, in this case, the DBpedia knowledge base.

4.2.3 Relation Extraction

The connecting link between Entity Recognition and Knowledge Graphs is identifying the Relations between them, this is done by the task of Relation Extraction, some of the most notable papers on this task are outlined in this subsection.

The task of relation extraction for the context of this dissertation is best represented by the paper titled "Open Information Extraction from the Web", (Etzioni et al., 2008), where the task of Open Information Extraction (OpenIE) is introduced together with the first solution to the problem, as defined in chapter 2.

Two surveys on this subject, (Pawar et al., 2017) and (Asghar, 2016) outline several ways of obtaining relations between entities or concepts. The major divisions are between categories of purely linguistic or syntactic patterns (Fader et al., 2011), and Machine Learning methods, which can be divided into supervised ML methods, (Kambhatla, 2004) and unsupervised ML methods (Wu and Weld, 2010) (Yan et al., 2009).

There is also the possibility of applying joint modeling of entity recognition and relation extraction, such as (Choi et al., 2006), (Yu et al., 2019), (Tran et al., 2021), (Roth and Yih, 2004), and (Dai et al., 2019).

4.3 Knowledge Bases

The task of generating knowledge bases from natural text is the most similar task to the intended purpose of this dissertation, this sub-section outlines two distinct tasks for generating knowledge representations from natural text, knowledge bases, and concept maps, as well as a few specific contexts in which knowledge graphs are used as input for other tasks.

4.3.1 Generating Knowledge Representations

The first representation that will be covered is generating concept maps from text. Concept Maps, as mentioned in section 3.2, are supposed to be a visual representation (for humans), different from knowledge graphs, which are machine-readable representations.

Concept Maps

Falke et al. (2017) presents a technique based on relation extraction, as seen in section 4.2, where the binary propositions are extracted from text, then filtered and processed to represent knowledge contained in multiple documents into a single concept map.

Yang et al. (2020) presents a Neural Network model to generate concept maps as a middle product of a technique for representing documents as graphs, Doc2graph.

Knowledge Graphs

Regarding the task of generating Knowledge Graphs, there are numerous approaches that create knowledge graphs from natural text. Here, we'll mention two main use cases, first is creating knowledge graphs from structured data with the intent of creating a publicly accessible knowledge graph, following the inspiration of Linked Open Data (LOD), that can be used by anyone for a wide range of tasks. The second case is that of creating a knowledge graph on a private corpus, looking to extract relevant information for a specific context.

The most important publicly available resources of commonsense knowledge for the context of this dissertation are DBpedia, ConceptNet, and WordNet. There are many other publicly available knowledge graphs, which focus on more specific use cases, usually either a specific topic (Ernst et al., 2015) or specific relations (Lin et al., 2019).

WordNet was one of the first public initiatives of building a database containing word-level knowledge. The first release of WordNet, (Miller et al., 1990), was aimed at proposing an online lexical reference system, an equivalent of an online dictionary.

In addition to word definitions, WordNet also defined synsets, which are groups of words with the same meaning. From the synsets, WordNet is capable of identifying a set of semantic relations, (Miller, 1995), such as synonyms, antonyms, hyponyms (and hypernym), meronyms (and holonyms), and entailment. WordNet was the first tool for obtaining relations between words, through publicly accessible queries.

The DBpedia is a public knowledge base created from Wikipedia pages in more than 100 languages, where each resource is mapped to a unique identifier,

corresponding to Wikipedia pages. DBpedia maps Wikipedia Infoboxes into an ontology covering 320 classes and 1650 properties, which represent the different relations between DBpedia resources, which are all represented in RDF triples pointing to other resources, which may be queried using the SPARQL query language.

DBpedia is a central piece in the Linked Open Data (LOD) cloud, being used as a reference point for several hundred data sets and other knowledge bases which use DBpedia's unique identifiers, one of which is ConceptNet, (Speer et al., 2018).

ConceptNet (Speer et al., 2018), (Liu and Singh, 2004) is a publicly available commonsense knowledge base aimed at supporting modern NLP techniques by means of a knowledge graph that connects words and phrases using labeled edges.

ConceptNet is oriented toward multiple relations between concepts with a total of 36 relations between concepts, such as *IsA*, *UsedFor*, and *CapableOf*, which are extracted from several different resources, including DBpedia, WordNet, OpenCyc (Lenat et al., 1990) and Open Mind Common Sense (OMCS) (Singh et al., 2002), turning it into a unique piece for Linked Open Data and makes it fundamentally different from WordNet and from DBpedia.

Aside from building knowledge bases to be publicly available, there are also important works regarding the construction of knowledge graphs for personal use. (Martinez-Rodriguez et al., 2018) implements an approach for constructing Knowledge graphs from an input corpus of documents, using Relation Extraction following OpenIE and Entity Linking to DBpedia resources.

Dessì et al. (2021) proposes a methodology using several state-of-the-art NLP tools to transform Scholarly Domain knowledge contained inside academic productions into a scientific knowledge graph, which represents detailed knowledge of the scientific literature.

Zhang et al. (2022) proposes Deep Knowledge Extraction (DeepKE), an open-source knowledge extraction toolkit focused on low-resource scenarios for knowledge base population in Chinese and in English. DeepKE provides Named Entity Recognition, Relation Extraction, and in turn, Knowledge Base Construction functionalities using Deep Learning architectures.

4.3.2 Using Knowledge Graphs

This section looks to briefly outline a couple of works that actually use knowledge bases, instead of building them. One of the main uses for knowledge graphs is Text Similarity, which will be detailed in section 4.4, its subtask

of finding the similarity between concepts, however, will be explored in this section.

Resnik (1995) presents a commonly used semantic similarity measure using the is-A taxonomy from WordNet to provide Information Content. The algorithm measures the semantic similarity between two concepts by measuring how much information they share in common, specifically defined by superclasses of the WordNet is-A relation.

Piao and Breslin (2015) presents Resim, a Resource Similarity metric between DBpedia Resources based on Linked Data Semantic Distance (LSDS). (Leal et al., 2012) proposes a Semantic Relatedness approach between concepts using the paths on an ontological graph extracted from DBpedia. Finally, (Speer et al., 2018) describes a relatedness measure between concepts from ConceptNet, namely the ConceptNet Numberbatch.

Another use for knowledge bases is query expansions, (Anand and Kotov, 2015) utilizes DBpedia and ConceptNet to perform query expansions and retrieve additional results to a given query.

With regard to Knowledge Graph Embeddings (KGE), (Wang et al., 2017) presents a survey on several techniques that create Knowledge Graph Embeddings, the objective of a KGE is to embed entities and relations into continuous vector space, looking to simplify the manipulation while preserving the inherent structure of the KG.

This technique can be used for a number of tasks, including a couple of applications on the actual Knowledge Graph and external applications such as Relation Extraction (Weston et al., 2013), Question Answering (Bordes et al., 2014), and Recommender Systems (Zhang et al., 2016).

4.4 Text Semantic Relatedness

This section presents works that perform the tasks of Text Semantic Relatedness and Text Semantic Similarity. As explained in chapter 2, Semantic Similarity is a subtask of Semantic Relatedness, and it is actually the task of Semantic Textual Similarity that has a larger body of research. The good thing is, Similarity is still a metric for Relatedness, so Semantic Similarity may be used as a valid source of information. Even though the focus of this dissertation is on Semantic Relatedness, works on Semantic Similarity will be explored as being equally relevant.

This task is usually divided into two main categories of algorithms: Knowledge-based methods and Corpus-based methods (Gomaa and Fahmy, 2013), (Chandrasekaran and Mago, 2021). Within the Corpus-based methods,

which take as input a collection of texts, a corpus, there are two main sub-categories of algorithm types, those that use Lexical, string-based, information, and others that use Semantic, meaning-based, information.

This section will focus on Corpus-based approaches that use Semantic information, which will be used in this dissertation, while first mentioning some knowledge-based approaches that are used to evaluate the work.

4.4.1 Knowledge-based methods

The knowledge-based approaches carry this name because they are dependent on external knowledge bases to collect information that will help to assign a Relatedness score between two given texts. This usually builds upon similarity metrics for the Concepts or words, seen in the previous section.

Yazdani and Popescu-Belis (2013) presents a relatedness metric according to the Visiting probability using Random Walks between two given texts. Visiting probability is calculated using relations matrixes between concepts, in the paper, Yazdani and Popescu-Belis use two different relation types, Wikipedia links $\{0, 1\}$, and a relatedness score between concepts, between 0 and 1.

Huang et al. (2012) uses a large combination of features to train a machine learning model that uses several features extracted from knowledge bases, such as the Wikipedia Link-based Measure, (Witten and Milne, 2008), concept importance to the text using concept relations, WordNet-based LCH (Leacock and Chodorow's path-length measure) (Leacock and Chodorow, 1998), the Strongest and Average connection between concepts in two documents, etc.

Budanitsky and Hirst (2006) looks to evaluate several methods that use WordNet relations between words, predominantly by using the graph structure of WordNet to find paths between concepts as a means to later calculate the distance between two documents.

Ni et al. (2016) builds a Concept2Vector representation of concepts and then computes a cosine similarity to determine the similarity between concepts and eventually between documents, by combining distances between concepts in each document.

Tutek et al. (2016) builds a combined graph with all the concepts present in a candidate pair of documents, and uses several metrics based on paths between concepts from one document to the other to determine the relatedness score for the resulting graph of the candidate pair.

4.4.2 Corpus-based methods

This section outlines the state-of-the-art Corpus-based Semantic Relatedness metrics, starting with the evolution and limitations of traditional methods.

Traditional approaches to Corpus-based Semantic Relatedness focused on using lexical information, comparing the words that appear in each text by the letters that composed it. However, since the introduction of Word Embeddings (Mikolov et al., 2013), similarity metrics started to use the Semantic meaning of the words. (Kusner et al., 2015) presents Word Mover's Distance, a metric that finds the distance that embedded words from one document need to travel to reach embedded words in another document. (Kenter and Rijke, 2015) proposes a technique to calculate Short Text Similarity using a supervised machine learning method that only takes as input word embeddings such as Word2vec and GloVe.

One limitation of pre-trained word embeddings is the presence of polysemic words, these are words that have multiple meanings, such as the word "bank", which can refer to a river bank or a financial bank. This limitation has recently been solved with the help of Language Representation Models.

One of the first models to solve this problem was BERT (Bidirectional Encoder Representations from Transformers), (Devlin et al., 2018), which uses a Transformers (encoder-decoder) architecture, (Vaswani et al., 2017) to create a vectorial representation of the text that captures the meaning of surrounding words. BERT pre-trains language models using the two directions of a text, this way, words that only occur later in the text still influence the vectorial representation of earlier words. BERT is used for multiple NLP tasks, one of the main applications is precisely that of Semantic Textual Similarity.

Several architectures have built on the work presented in (Devlin et al., 2018) and enhanced the original encoding of BERT, many of which are notably relevant for the Task of Semantic Textual Similarity. This is the case for models like ALBERT, (Lan et al., 2019); RoBERTa, (Liu et al., 2019); DistilBERT, (Sanh et al., 2019); and SpanBERT, (Joshi et al., 2019).

Another notable work that builds on the original BERT model is SBERT. (Reimers and Gurevych, 2019) presents Sentence BERT, which generates sentence embeddings of a text without having to run all of the texts through a BERT architecture, which significantly reduces computation time, from 65 hours to 5 seconds.

4.4.3 Hybrid methods

Another possibility is the combination of both knowledge-based and corpus-based methods, as an example, (Nguyen et al., 2019) details a methodology to combine word similarity based on pre-trained word vectors together with word similarity based on external sources of knowledge to compute short-text semantic similarity.

4.5 Related Works Summary

Several different works within the field of NLP are relevant for this dissertation. However, among all of the different tasks and works outlined in this chapter, the most important ones are the tasks of **Concept Extraction** using DBpedia Spotlight, (Mendes et al., 2011) and Dandelion API, (SpazioDati, 2012), the **Knowledge Bases** from DBpedia, (Lehmann et al., 2015) and ConceptNet, (Speer et al., 2018), as well as **Text Semantic Relatedness** using Language Models, with (Devlin et al., 2018) and (Reimers and Gurevych, 2019).

5
Concepts Connections

This chapter and chapter 6, on Text Semantic Relatedness, are written in support of the methodology, they build on specific technical implementations and NLP tasks that are too detailed and would otherwise bring unnecessary information to the methodology presented in chapter 7. Together, the two chapters will be outlining how the two sub-problems were solved individually, whereas chapter 7 will focus on how all of the collected information is united into one overarching workflow.

As explained in the problem definition in section 1.1, the two main types of paths for navigation the text collection are through Concepts Connections and through a Text Semantic Relatedness Recommendation System. These two paths are further represented by the two sub-problems of 1. Extracting Concepts mentioned in the text, and the Relationships between them, and 2. Compute Semantic Relatedness between Texts.

This chapter outlines all of the research and implementation that was carried out to solve the first of the two sub-problems defined in section 1.1. Here, we describe the local problem definition for creating concept connections and explain detailed procedures related to this task.

5.1
Introduction

The idea of using Concepts to connect texts was largely inspired by the idea of Trails of Thought, in (Bush, 1945). Trails of thought occur when numerous items have been joined to form a trail, so that they can be reviewed in a sequential manner, where items belonging to a same trail may share a common topic, or even be connected across different domains.

Concepts are seen as a very promising representation of the content inside a text, because it can be a source of connections across domains. Concepts represent what is inside the text without being limited to the specific topic being portrayed in the given text. This happens because a concept is a fundamental building block of knowledge, which means it is reused across domains.

The expected result is that concepts will represent paths between ideas,

providing a practical way to expand any given trail of thought by presenting other texts that mention the same concept or even related concepts, regardless if the texts share a common topic or not.

This chapter represents the foundation for obtaining all the necessary data regarding concepts in order to prepare for the proposed methodology in chapter 7, where the navigation mechanism is proposed using the data collected and processed in this chapter.

The remainder of this chapter is structured as follows: section 5.2 presents the goals and explains the task of identifying connections using concepts in greater detail. section 5.3 shows a visual example of what is expected from this part of the methodology, while section 5.4 details and formalizes the procedures carried out to attain the desired results for this chapter.

5.2 Goals and Problem Definition

This section outlines the Objectives and the Problem Definition for the task of proposing connections using Concepts.

The objective of this chapter is threefold. Firstly, to identify concepts mentioned in a given text dataset. Second, to compute any relationships between each of the concepts. Third, to leverage these relationships between concepts to determine a semantic relatedness score between any two given texts in a text collection.

The first two parts of the objective, identifying concepts and relationships between them, represent a central component to the generation of connections for the text collection, while the third part, of text relatedness, will be important for the text connections and for the evaluation of the proposed connections.

Given the objectives for the chapter, a Local Problem Definition is defined to illustrate the desired results, as well as to pave the way for a solution that handles each aspect of the problem at hand.

The problem definition for this chapter is divided into 4 parts. Given a text dataset:

1. Extract the entities and concepts mentioned in each text while performing entity linking to an external knowledge graph.

2. Filter the concepts considered relevant to the text collection.

3. Capture quantitative and qualitative relationships between concepts using external knowledge.

4. Use relationships between concepts to determine a relatedness score between texts, based on the concepts mentioned in them.

5.3 Motivating Example

This section presents an example text that will be used throughout the dissertation to demonstrate the executed procedures. The text passage is intended to provide a better understanding of the procedures and to serve as a guide and companion for the entire methodology. This includes the proposed methodology in chapter 7 and both support chapters, this chapter and the next on text semantic relatedness, chapter 6.

The text presented in Figure 5.1 is a passage from the book "Building a Second Brain", by Tiago Forte, (Forte, 2022), talking about the importance of Personal Knowledge Management, and the type of lessons shared in the book. This example text is part of a Text Collection composed of Book Highlights, extracted from the digital reading software Kindle, from Amazon.

> "In the same way that personal computers revolutionized our relationship with technology, personal finance changed how we manage our money, and personal productivity reshaped how we work, personal knowledge management helps us harness the full potential of what we know. While innovations in technology and a new generation of powerful apps have created new opportunities for our times, the lessons you will find within these pages are built on timeless and unchanging principles."

Figure 5.1: The Example Text used throughout the dissertation

Using the Example Text as a guide, this section shows a simple example of the procedures applied in this chapter, in order to capture the necessary and relevant information for the construction of the interconnected collection in the Methodology, chapter 7.

Figure 5.2 describes the example text in three different versions, the first without any concepts identified, the second one with all potential concept mentions identified, and the third and last version has the final selection of relevant concepts to the encompassing text collection. This is achieved by applying both a filtering (removing) process and an expansion (addition) process.

Table 5.1 details the mentions and corresponding entities extracted for the example text, this table represents the same mentions portrayed in

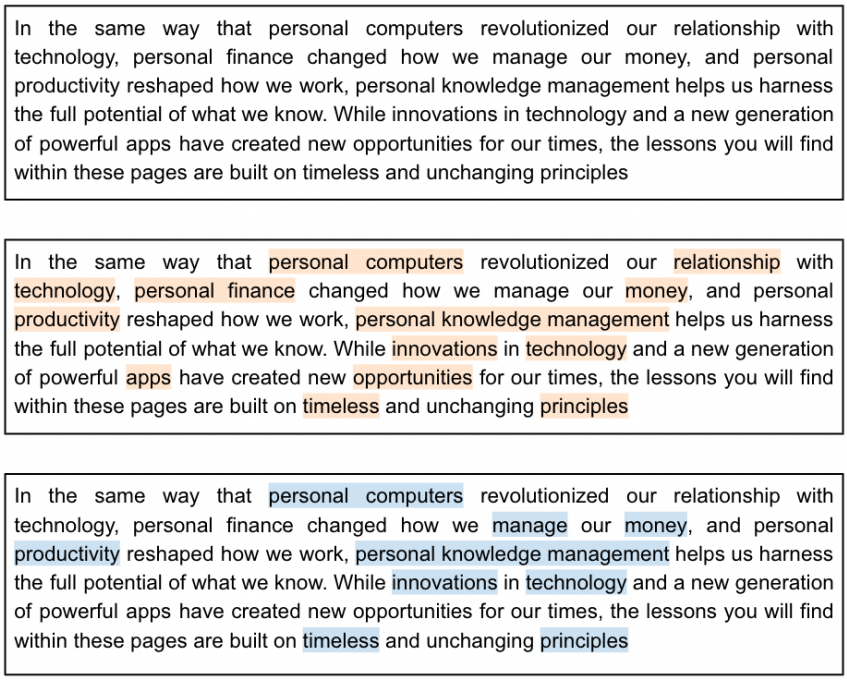

Figure 5.2: Three stages of Concept Extraction, before (top), initial extraction (middle), and final selection of concepts (bottom)

Table 5.1: Mentions Database with Concepts extracted from the Example Text

Entity Name	Mention Text (if different)	Position		Confidence
		Start	End	
Personal computer		21	39	0.612
Interpersonal relationship	relationship	59	71	0.503
Technology		77	87	0.636
Personal finance		89	105	0.704
Money		132	137	0.578
Productivity		152	164	0.681
Personal knowledge management		187	216	0.865
Innovation	innovations	276	287	0.589
Technology		291	301	0.636
Mobile app	apps	335	339	0.470
Equal opportunity	opportunities	357	370	0.406
Timeless (TV series)	timeless	444	452	0.667
Principle	principles	468	478	0.458

Figure 5.2, but in tabular form and with additional information, such as the Confidence and the corresponding Entity.

Finally, Table 5.2 presents all the desired information that this chapter seeks to obtain for all concepts in the final selection of concepts for the example text, as well as for any text that enters the proposed system.

Table 5.2: Description of desired information for each Concept

Category	Description
DBpedia URL	The UI for the concept
Entity Name	The name of the concept
Description	DBpedia Abstract describing the concept
Image	Link to image illustrating the concept
DBpedia Types	The associated Types to the concept, following OWL or DBpedia Ontology
Subject	The topics of the resource
Wiki Links OUT	Links from this concept's Wikipedia page to other Wikipedia pages
Wiki Links IN	Link from other Wikipedia pages to this concept's Wikipedia page
ConceptNet Relatedness	ConceptNet Quantitative Relatedness to other concepts
ConceptNet Relations	ConceptNet Qualitative Relationships with other concepts

5.4 Local Methodology (Procedures)

5.4.1 Entity Recognition and Entity Linking

The first task in the pipeline for this chapter is identifying the entities in the text. The term Entity Recognition is used for this first task, rather than Concept Extraction because the tools deployed for extracting concepts are not exclusive to concept recognition and return all kinds of entities, fortunately, abstract conceptual entities are included.

This task is actually the joint modeling of Entity Recognition and Entity Linking, which according to (Luo et al., 2015), increases precision, since performing entity recognition and disambiguation simultaneously provides better results because the information from each task improves the performance of the other task.

As explained in chapter 2, when dealing with entities, there is a distinction between mentions (surface form) and entities (identified resource).

The joint task of obtaining disambiguated entities from the text is formalized in three stages: 1. Spotting mentions in the text 2. Collecting candidate entities for each mention 3. Performing disambiguation to determine the most likely entity represented by the mentions

At the end of these three stages, the labels obtained for each mention correspond directly to an entity linked in a Knowledge Base.

To perform the task of joint Entity Recognition and Linking, a number of Entity Recognition tools were considered and two different tools were chosen to perform the task. They are Dandelion API, (SpazioDati, 2012) and DBpedia Spotlight, (Mendes et al., 2011).

The reasons behind the choice of such a combination of tools are:

- Both are capable of performing the task of Entity Linking jointly with Entity Recognition
- Both link entities to the same external knowledge base (DBpedia)
- Both provide (some) information on the confidence attributed to the result of Joint Entity Recognition and Linking.

  Dandelion provides precise confidence scores between 0 and 1, while DBpedia Spotlight accepts a confidence threshold as a parameter, which allows for the range of a confidence level to be intuited.

The tool used for initial Entity Recognition was the Dandelion API, which is tailored to identify concepts, and also provides an exact confidence level for each mention. The DBpedia Spotlight was later used to further enhance the concept selection.

In order to illustrate how Entity Recognition works with the Dandelion API, we present the results of Entity Recognition applied to the example text detailed earlier. Figure 5.3 shows the Dandelion Demo Interface for visualizing the extracted entities for the input text, while Table 5.1 and Table 5.4 show the same extracted entities for the input text, but in a table format, each with a different emphasis.

Figure 5.3: The mentions identified in the example text, provided by the Dandelion Demo Interface

A raw mentions database is built after running the Dandelion API Entity Extraction Endpoint. This mentions database contains information on all of the mentions identified and the corresponding entities. The focus of this database is to portray information about the original text where the mention occurs, this information includes but is not limited to: the mention

(surface form), entity name, entity DBpedia URL, location within the text, and confidence score.

In order to illustrate how the mentions database is structured, Table 5.3 presents all the extracted data for an example concept mention, of "Technology"

Table 5.3: Information extracted for the Entity "Technology" from the Dandelion API

Category	Value
Source	Dandelion
Mention text	technology
Start position	77
End position	87
Entity Name	Technology
DBpedia URL	http://dbpedia.org/resource/Technology
Confidence	0.6365
Semantic Categories	Main topic articles, Technology, Technology systems
DBpedia Types	[empty]

5.4.2 Concepts Selection (Filtering and Enhancing)

After the task of joint Entity Recognition and Linking, with the resulting dataset containing mentions, the next step is that of filtering the dataset according to the specific use case. This subsection marks the transition from dealing with entities to dealing with concepts, the term "concepts" will be used to refer to the set of entities after having filtered out the entities extracted in subsection 5.4.1, in order to prioritize conceptual entities.

This step depends on the list of undesired DBpedia types, which is a parameter for the system, as well as the threshold confidence level. The decision making process behind this filtering and enhancement process will be discussed in detail in the proposed methodology, section 7.3, nonetheless, this step is represented in the local methodology of concept connections, since it is a central piece for this section.

The idea is to filter unwanted concepts from the entire mentions database, by applying the filtering process, then enhancing the filtered mentions database by applying Entity Recognition from DBpedia Spotlight to identify concepts that "survived" the filtering process.

The first step in the concept selection is to apply the filter to remove the unwanted DBpedia types. The decision-making for this step will be discussed in chapter 7, for now, the important thing to understand is that a couple of

Table 5.4: Mentions Database with Concepts extracted from the Example Text, with special detail to certain Entities that will be **filtered out**

Entity Name	Position		Confidence	DBpedia Type
	Start	End		
Personal computer	21	39	0.612	
Interpersonal relationship	59	71	0.503	
Technology	77	87	0.636	
Personal finance	89	105	0.704	
Money	132	137	0.578	
Productivity	152	164	0.681	
Personal knowledge management	187	216	0.865	
Innovation	276	287	0.589	
Technology	291	301	0.636	
Mobile app	335	339	0.470	
Equal opportunity	357	370	0.406	
Timeless (TV series)	444	452	0.667	/TelevisionShow
Principle	468	478	0.458	

DBpedia Types will be filtered out of the mentions database. Some examples of types that will be excluded from posterior manipulations are /Film, /Band, /Magazine, and /TelevisionShow.

After removing selected Entity Types, the next step is to filter out pairs of mention and entity that may not be as relevant, using the Confidence level. The **baseline for the confidence level is set at 0.60**, but this may be easily changed at execution, and will also be deliberately varied to evaluate the system.

The cells in Table 5.4 signaled in red represent mentions that would be filtered out of the initial concept extraction, according to the Entity type and confidence filters.

Following this initial filtering, a **concepts list** is created with all the remaining entity URLs, and is then used for two important tasks, first, to filter the mentions database to show only the relevant concepts. Second, the concepts list is used to **enhance the Mentions Database by using DBpedia Spotlight**.

The DBpedia Spotlight, which has a broader reach and captures more entities, is then used to complement the mentions list by adding all mentions identified for concepts in the concepts list. This is done in order to find any mention of those concepts that may have been missing from the mentions database.

The last manipulation is executed to ensure that only concepts which appear in multiple texts are added to the mentions list, in order to ensure that

every concept page will serve as a navigation standpoint and not as a dead end (more on this in chapter 7).

The output for this section is a filtered database of mentions and a final concept list.

5.4.3 Additional Concepts Information using Knowledge Bases

Once the filters are applied to the concepts list, and the mentions database is enhanced, the next step is to collect more information on the concepts, which are not available through Entity Recognition. For this purpose, two different Knowledge Bases were consulted, DBpedia, (Lehmann et al., 2015) and ConceptNet, (Speer et al., 2018).

To organize the data regarding the concepts, a new database was created to store all the additional information on the concepts. While the mentions database is directed at information from the text where it came from, the concepts database is totally focused on holding additional information regarding each individual concept and the relations between them.

To build the concepts database, a list of unique entities' DBpedia URLs present in the mentions database was used. From the final list of relevant concepts, the two different Semantic Knowledge Graphs, DBpedia and ConceptNet were queried to obtain additional information.

DBpedia was accessed in order to obtain structural information on each individual concept, as well as capturing what are other concepts related to it through Wikipedia page links.

DBpedia was queried using the OpenLink Virtuoso SPARQL protocol endpoint, provided by DBpedia to access any type of information from its knowledge graph. The SPARQL queries are made following the RDF structure of SPO triples (subject-predicate-object). In this case, either the subject or the object must be a concept in the concepts database, while the predicate is a DBpedia relation type, usually denoted by the RDFS or DBpedia ontologies, but not limited to them.

When the concept was queried as the subject, the following relation types were used as predicates:

- rdfs:label
- rdfs:comment
- dbo:thumbnail
- dbo:WikiPageWikiLink (Wiki Links OUT)
- rdf:type

– dct:subject

When the concept was queried as the object, a single relation type was used as a predicate:

– dbo:WikiPageWikiLink (Wiki Links IN)

An example of the result obtained in the queries to DBpedia is presented in Table 5.5, which shows the additional information extracted from DBpedia for the concept of "Knowledge".

Table 5.5: Example of information obtained from DBpedia for the concept of "Knowledge"

Category	Extracted Values
DBpedia URL	http:/Knowledge
Entity Name	Knowledge
Description	Knowledge is a familiarity or awareness, of someone or something, such as facts (descriptive knowledge), skills (procedural knowledge), or objects (acquaintance knowledge) contributing to ones understanding. By most accounts, knowledge can be acquired in many different ways and from many sources, including but not limited to perception, reason, memory, testimony, scientific inquiry, education, and practice. The philosophical study of knowledge is called epistemology.
Image	http://commons.wikimedia.org/wiki/Special:FilePath/Knowledge-Reid-Highsmith.jpeg?width=300
DBpedia Types	['http://dbpedia.org/ontology/MusicGenre', 'http://www.w3.org/2002/07/owl#Thing']
Subject	'Categories: [Concepts_in_epistemology', 'Knowledge', 'Intelligence', 'Mental_content', 'Virtue', 'Main_topic_articles']
Wiki Links OUT	['Writing', 'Learning', 'Understanding', 'Technology', 'Belief', 'Mind', 'Personal_knowledge_management', 'Wisdom', 'Post-scarcity_economy', 'Fact', 'Decision-making', 'Peace']
Wiki Links IN	['Writing', 'Learning', 'Understanding', 'Technology', 'Belief', 'Mind', 'Personal_knowledge_management', 'Wisdom', 'Wealth', 'Risk', 'Government', 'Hierarchy', 'Image']

ConceptNet was used to obtain commonsense knowledge between concepts, which are represented by several different detailed relationship types between two given concepts. The goal with ConceptNet was twofold, to obtain these granular relation types between concepts to enrich the connections between concepts, and also to obtain a relatedness score between two concepts using the ConceptNet Numberbatch API endpoint.

ConceptNet was queried using the ConceptNet API, where queries are posted directly as API URL requests, describing the concept which is to be queried and the desired relationship types. ConceptNet queries also follow the structure of triples, but they define their own ontology to deal with the triples, namely: start, rel(ation), and end.

Table 5.6 presents the list of the used relation types and the result obtained for each query. The queries are made to ConceptNet for each concept in the concepts list, the table presents the results for a query using the concept of "Knowledge", showing the related concepts for each of the corresponding relation types.

Table 5.6: Example of Related concepts extracted from ConceptNet for the concept of "Knowledge"

Category	Related Concepts to "*Knowledge*"
/r/RelatedTo	[erudition, knowing, data, complete, know_how, information, prospective, information, intercourse, science, significant, know, study, known, wisdom, course, carnal_knowledge, perception, place, awareness, know, epistemology, information, intelligence, wisdom, brain, power, gathered, gathered_intelligence, education]
/r/IsA	[powerful_thing, good_thing, power, applied_information, power_if_used_correctly, information, understanding]
/r/HasContext	[philosophical, archaic, legal]
/r/Causes	
/r/CapableOf	[increase_value, change_people_greatly, advance_mankind, seed_ideas, open_mind, open_human_mind, make_person_happy, make_person_sad]
/r/MotivatedByGoal	
/r/Desires	
/r/HasProperty	[powerful, unlimited, power]
/r/HasPrerequisite	[thought]
/r/PartOf	[innovation, understanding]
/r/UsedFor	[cutting]
/r/Antonym	[ignorance]
/r/DistinctFrom	
/r/Synonym	[cognition, knowingness, ken, awareness, learning, cognizance, knowledge]
/r/SimilarTo	

The information from both DBpedia and ConceptNet was kept separate, with the help of a mapping function between the URIs from both Knowledge

Graphs, which are different. This was possible due to ConceptNet having an attribute that points to external DBpedia URLs, making the mapping function between URIs possible.

5.4.4 Compute Relatedness Matrices between Concepts

This subsection is directed at computing relatedness between concepts, as a preliminary task before calculating the relatedness between texts. The final result for this section is a set of relatedness matrices representing the relatedness between each concept in the text collection.

To determine the relatedness between concepts, two methods were used. First, a quantitative method that represents the semantic relatedness between concepts. Second, a binary value that signals the presence of any descriptive relation between the concepts. A different relatedness matrix was calculated for each type of relationship.

5.4.4.1 Numerical Relatedness between Concepts

With regard to numerical relatedness, the idea is to represent a relatedness score between two given concepts, ranging from 0 to 1. It is worth noting that a relatedness score is the inverse of a distance score between two concepts. This type of relatedness allows for an intermediary value between any two concepts, even if the concepts do not share an explicit relationship between them.

For example, the concepts “tea” and “coffee” may not share an explicit relationship between them, but they are definitely more related than the concepts “tea” and “giraffe”.

The chosen method to calculate the quantitative relatedness between concepts is the ConceptNet Numberbatch, which is represented by a dedicated endpoint in the ConceptNet API, (Speer et al., 2018), by a query of “/relatedness” followed by two concepts.

This query was used to obtain the distances between each pair of concepts. The score presented is a relatedness score, which has a format of a score between -0.1 and 1, with 1 being the most related, for words that are synonyms. This score was further normalized to represent relatedness scores between 0 and 1.

An example for the result obtained for the Relatedness between concepts from ConceptNet Numberbatch is presented in Table 5.7, which shows additional information extracted from ConceptNet for the relatedness between

selected concepts as compared to the concept of "Knowledge".

Table 5.7: Examples of the relatedness scores for the concept of "Knowledge". Numerical Relatedness are from ConceptNet Numberbatch and Shared Relationships from selected categories of relations.

Relatedness to: "**Knowledge**"		
Concept	Numerical Relatedness	Shared Relationship
Information	0.460	1
Wisdom	0.442	1
Understanding	0.434	1
Intelligence	0.332	1
Learning	0.328	1
Perception	0.281	1
Memory	0.181	1
Technology	0.177	1
Logic	0.161	1
Brain	0.145	1
Thought	0.120	1
Observation	0.118	1
Biology	0.063	0
Exercise	0.030	0
Innovation	0.017	1
Nutrient	-0.007	0
Marathon	-0.035	0
Pricing	-0.080	0

5.4.4.2 Shared Relationships between Concepts

In the other approach to calculating the relatedness between concepts, a connection matrix was built to show concepts that share relationships with one another. Different relation types were selected from the relations extracted from DBpedia and ConceptNet and used to create this connection matrix, similar to what is proposed in (Yazdani and Popescu-Belis, 2013).

A connection Matrix is a binary matrix, composed of only 0s and 1s, where the number 1 is used to represent that a connection between two given concepts exists, and 0 is used when there is no connection. Different relation types are used in order to obtain different metrics for capturing relationships between concepts.

The relationships considered between concepts were all applied in both directions, meaning that if concept A is related to concept B, then concept B is automatically related to concept A, even if the relationship was not identified in subsection 5.4.3. This is done because the idea is for the relatedness matrix

to represent a metric distance, which means it should satisfy four properties: non-negativity, the identity of indiscernible, symmetry, and triangle inequality.

All of the different ConceptNet relationships described in Table 5.6 were considered as connections between concepts. While for DBpedia, the presence of Wikipedia Links going OUT and coming IN to a concept was considered as a relationship between those concepts, described in Table 5.5.

This means that after the manipulations proposed in this section, there were two different relatedness metrics between concepts, 1. the quantitative relatedness and 2. the shared relationships. Table 5.7 presents an example of both relatedness metrics for the concept "Knowledge".

Both of these matrices are used in the next section to calculate the Knowledge-Based Text Relatedness.

5.4.5 Knowledge-Based Text Relatedness

This section details one of the two approaches for calculating the relatedness between texts. While chapter 6 will deal with the *Corpus-based* approach, this section details the procedures used to calculate a *Knowledge-based* approach for the relatedness between texts. This is done by using the concepts mentioned in each of them and the matrices representing the relatedness between concepts.

Since one of the Research Questions in this dissertation is directed at studying the usefulness of Concepts as a mechanism for navigating a text collection, the idea is to implement the Knowledge-based Text Relatedness using the concepts extracted from each text as a means to determine the relatedness between texts.

Relatedness between Two Texts The chosen method to calculate the relatedness between two given texts was based upon one of the features presented in (Huang et al., 2012), precisely the Average connection strength between the concepts belonging to each of the two texts. This metric depends on the relatedness between the concepts, which is where the Concept Relatedness matrices come into play.

As described in the previous section on Concepts Relatedness, subsection 5.4.4, two different metrics were considered to propose connections between concepts, the Quantitative Relatedness from ConceptNet Numberbatch, (Speer et al., 2018) and a set of Qualitative Shared Relationships, based on the idea of a connection matrix, (Yazdani and Popescu-Belis, 2013). These two metrics were initially used independently to calculate the two separate

relatedness metrics between the texts.

For each of the concept-relatedness metrics, the following procedures are performed:

First, the concepts mentioned in each of the texts are identified. Next, an average relatedness is calculated considering all the relationships between the concepts in text A with the concepts in text B, including concepts that are present in both texts (relatedness score = 1).

The final relatedness between the two texts is composed of the average of the two relatedness metrics, arriving at a unified Average connection strength between the concepts present in each of the two texts.

Relatedness Matrix for all Texts The procedure described to calculate the relatedness between two texts is replicated for all the combinations of text pairs in the text collection. Resulting in a Knowledge-Based Text Relatedness Matrix, which will be used to evaluate the Coherence between the different types of Knowledge Connections.

6
Text Semantic Relatedness Connections

This chapter is written to support the methodology, in order to build on specific technical implementations and NLP tasks that are too detailed and would otherwise bring unnecessary information to the methodology in chapter 7. This chapter will outline how to solve the sub-problem 2. Compute Semantic Relatedness between Texts.

This chapter describes the local problem definition for the task of creating text-relatedness connections and explains detailed procedures related to this task.

6.1
Introduction

The idea of having a direct recommendation of what texts are related to any given piece of text comes from the idea of connecting notes inside a Zettelkasten, specifically for the moment when a user is inserting a new piece of knowledge into the system and wants to have a quick overview of notes that are already inside which may be related to the present piece of knowledge.

In general terms, connections between texts can also be extremely useful for tasks of recall and exploration, as they create a direct bridge between texts. This type of connection provides a clear view of what other texts and ideas are related to "the current text", or any given text.

The expected result for using text recommendations to connect different texts is that direct paths created between texts can provide a useful navigation mechanism to other relevant resources within the text collection.

This chapter presents the procedures for calculating the relatedness between each text in the text collection using corpus-based methods associated with Word Embeddings, specifically methods that use Language Model Encoders in some way.

The procedures and information in this chapter are presented in preparation for the Proposed Methodology in chapter 7, where the navigation mechanism is proposed by uniting the connections proposed in this chapter with the Concept-based Connections in chapter 5.

The rest of this chapter is structured as follows: section 6.2 presents the

goals and explains the task of proposing connections using text-relatedness in greater detail. section 6.3 shows a visual example of what is expected from this part of the methodology, while section 6.4 details and formalizes the procedures carried out to attain the desired results for this chapter.

6.2 Goals and Problem Definition

This section outlines the Objectives and the Problem Definition for the task of creating a system that recommends semantically related texts.

The objective of this chapter is simple, **for every pair of texts in the text collection, find a numerical score for the relatedness between them**, which captures the semantic meaning of each word, as well as its surrounding context.

This objective is naturally an extension of the basic problem of finding the semantic relatedness between two texts. The idea is to propose a methodology to calculate the relatedness between two texts and to use it to create a relatedness matrix for all the combinations of texts.

Given the objectives for the chapter, a Local Problem Definition is defined to illustrate the desired results, as well as to pave the way for a solution that handles each aspect of the problem at hand.

The problem definition for this chapter is divided into 2 parts.
Given a text dataset:

1. Encode the content of each text as a Vector Representation, capturing the semantic meaning and context of each word in a text.

2. Compute the relatedness score between each text, to create a relatedness matrix for all texts.

The encoding of texts is done in order to represent the texts so that it is possible to capture the semantic meaning of the text. This is done by encoding the text into a vectorial space, where it is possible to mathematically calculate the relatedness (or distance) between two given texts in a way that represents their meaning.

It is possible to calculate the distance between two texts in their raw string format, however, any distance metric would only cover lexical similarities between texts, and not semantic similarity, which is the intended information for this chapter.

6.3 Motivating Example

This section will use the same example text from the previous chapter, Figure 5.1, looking to illustrate the procedures outlined in this chapter with visual examples to enhance understanding of the procedures, by providing a visual example of the intermediary steps between the input and the output.

In order to illustrate how the process of encoding a text looks like, Figure 6.1 shows the example text represented in the raw text format, after the tokenization process and also as a word embedding, which is a vectorial representation, in the format of a tensor.

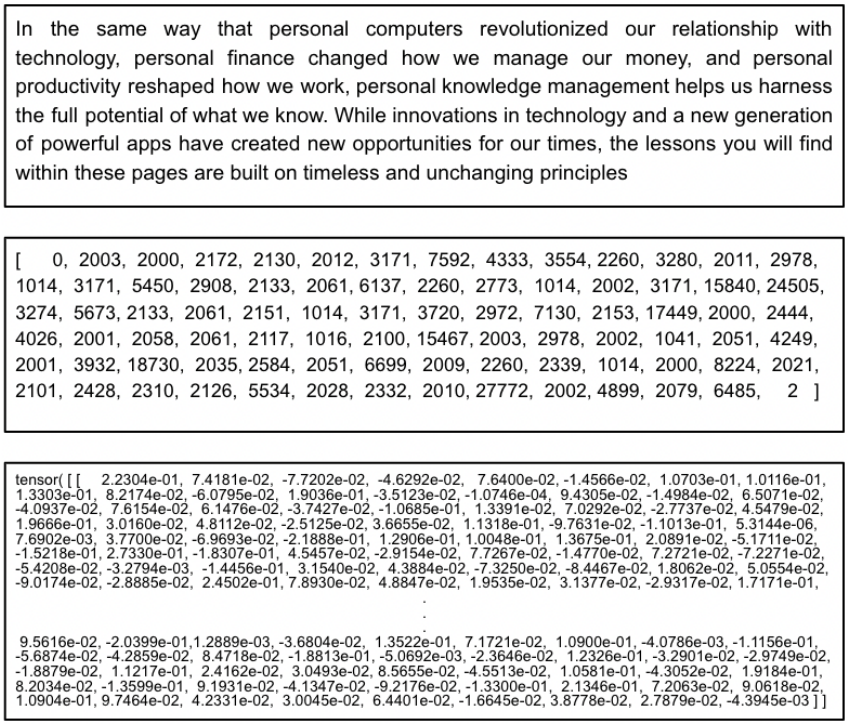

Figure 6.1: Three different Encodings of the example text, raw text (top), as tokens (middle), and as word embeddings (bottom)

Looking to illustrate how the output for this chapter looks like, Table 6.1 depicts the relatedness matrix for an example with 5 texts. The main diagonal is composed of 1s, which represent the maximum relatedness between texts, this is the case for the relatedness of each text with itself.

With the intent of describing the desired output for this chapter, Table 6.2 demonstrates an example of the relatedness between texts, presenting the comparison between the example text and 3 selected texts. The table presents the content of each text and the relatedness score between the original text and the corresponding text, obtained from the relatedness matrix.

Table 6.1: Example of the Relatedness Matrix format

Example Relatedness Matrix				
1.000	0.584	0.284	0.282	0.304
0.584	1.000	0.196	0.306	0.231
0.284	0.196	1.000	0.467	0.517
0.282	0.306	0.467	1.000	0.436
0.304	0.231	0.517	0.436	1.000

Table 6.2: Semantic Relatedness Score comparison for selected texts

Relatedness Score	Text
0.795	I've come to believe that personal knowledge management is one of the most fundamental challenges—as well as one of the most incredible opportunities—in the world today. Everyone is in desperate need of a system to manage the ever-increasing volume of information pouring into their brains. Those who learn how to leverage technology and master the flow of information through their lives will be empowered to accomplish anything they set their minds to.
0.390	Your summary says "Productize yourself"—what does that mean? "Productize" and "yourself." "Yourself" has uniqueness. "Productize" has leverage. "Yourself" has accountability. "Productize" has specific knowledge. "Yourself" also has specific knowledge in there. So all of these pieces, you can combine them into these two words.
0.116	Aristotle described virtue as a kind of craft, something to pursue just as one pursues the mastery of any profession or skill. "We become builders by building and we become harpists by playing the harp," he writes. "Similarly, then, we become just by doing just actions, temperate by doing temperate actions, brave by doing brave actions." Virtue is something we do. It's something we choose. Not once, for Hercules's crossroads was not a singular event. It's a daily challenge, one we face not once but constantly, repeatedly.

6.4 Local Methodology (Procedures)

This section presents the local procedures of the Methodology for the specific task of finding the semantic relatedness between texts. This section is divided into two parts, 1. The Encoding of Texts and 2. Computing Relatedness between Texts.

6.4.1 Encoding the Text

The first task in the pipeline for this chapter is to represent the text so that it may be possible to mathematically calculate the relatedness (or distance) between two given texts. This is done by **encoding the Text** into a vectorial space, more precisely, a tensor. This tensor represents the text numerically, in such a way that its multiple dimensions are able to capture differences in the semantic meaning of the texts.

One important requirement for this task is to capture not only the semantic meaning of words but also the distinction between different meanings of polysemic words. As explained in the Related Works section, section 4.4, dealing with words that have multiple meanings, such as "bank", is one of the major challenges faced when calculating text-relatedness using *pre-trained* word embeddings such as WordNet or GloVe.

The solution for generating a vectorial representation that distinguishes between different meanings of the same polysemic word is using **Encoders from BERT-like Language Models** to encode the text into embeddings. What this means is that words are encoded using Language Models that capture the context of each word, i.e. the surrounding words, and not only the definition of single words individually.

Two different methods are proposed for the generation of the embeddings, first using **the encoder from Transformer Language Models**, Wolf et al. (2020), and second, using a more computationally efficient **Sentence Transformers encoding**, Reimers and Gurevych (2019).

The idea is to implement two different ways of calculating Word Embeddings, so that in the Proposed Methodology, both may be discussed, looking to determine the most suited technique according to the specific context.

6.4.1.1 Language Models Encoders - BERT

Encoding using Language Models is the most powerful and robust approach, which encodes the vector representations, or word embeddings, according to all the specific words that are actually used in the corpus (the collection of all texts).

As demonstrated in the Motivating Example, section 6.3, the input for this methodology is simply the raw text, with the sequence of words that compose it, whereas the output is a tensor, a numerical representation that captures the semantic meaning of all the words in the text combined, across all of its dimensions.

The procedures for generating embeddings using Language Models Encoders may be summarized into 3 steps:

1. Tokenization of the words in the text
2. Encoding each word using the Language Model
3. Mean Pooling across all words to create Embedding for the entire text

Each of these steps will be explained in further detail.

The first step is to send the texts through a tokenization process. Tokenization is a type of encoding of the text, the principle behind it is to represent texts in a way that the Language Models can understand, this is done by mapping each word into what is called a "token".

A token is simply a unique identifier for a word, where each distinct word is mapped to a specific token, as illustrated in Figure 6.1. This identification is proposed by considering all the words present across the corpus and attributing identifiers for each distinct word. This is the major reason why the corpus needs to be inputted to the Language Model all at once for Language Model Encoding.

There are different tokenization algorithms, including words-based, character-based, and subword-based methods. These types of tokenization will determine the number of tokens generated.

The next step, of encoding each word using the Language Model, is where the semantic meaning is captured for each word. The tokens generated from the previous step are sent as inputs to the Language Model, which encodes the tokenized text according to the Language Model's architecture.

The Transformers architecture, (Vaswani et al., 2017), present in most of the modern Language Models, is an encoder-decoder architecture, meaning it has an Encoder component and a Decoder component, each with its own design. This architecture is designed to be used to first encode text into a numerical representation, where different manipulations may be performed, and then decode the numerical representation back to output in text format.

The important representation for the task of Text Semantic Relatedness is the numerical representation, right after the text is encoded. This means the decoder component is not used, only the encoder.

After the input tokenized text goes through the encoder of the Language Model, the output is a word embedding for each word in the text, i.e. a tensor representing each word in a multi-dimensional semantic space.

The last step of the process is to use all tensors that represent the text, one for each word, to compute a final numerical representation that captures

the semantic meaning of the entire text. This is done by applying **Mean Pooling** to the word embeddings of all the words in the text. A Mean Pooling layer basically returns a tensor with the mean value between all the word embeddings across all the positions of the tensor.

After the Mean Pooling layer is applied, the resulting tensor corresponds to the expected output for this subsection, a numerical representation that captures the semantic meaning of a given text.

One drawback of the Language Model Encoder method, though, is that the corpus is inputted to the Language Model all at once. This implies that generating compatible embeddings for a new text, one that was not previously in the corpus, is computationally very expensive, because the embeddings for all the already existing texts would need to be generated again, together with the embeddings for the new texts.

6.4.1.2 Sentence BERT - Sentence Transformers Encoders

Calculating Embeddings using Sentence Transformers is the most computationally efficient approach. Sentence Transformers are a generalization of the Sentence-BERT methodology, (Reimers and Gurevych, 2019), which is a workaround for generating Short Text Embeddings (Sentence Embeddings) using the BERT architecture.

The procedures in Sentence-BERT describe a model that is trained **ONLY to generate embeddings**, which means it only contains the Encoding architecture and **does not contain a decoder** component. The Sentence-BERT uses the BERT encoding and fine-tunes it to generate fixed-size vectors (embeddings) for the entire input text. The output of the Sentence-BERT is the actual Sentence Embeddings.

The Sentence-BERT is designed to perform one specific sub-task, of generating embeddings for an entire text, whereas the BERT architecture is designed for multiple NLP tasks, and to be used for the same sub-task, it has to be adapted for it, according to the procedures described in the previous subsection.

What Sentence-BERT does is automatically add a pooling operation to the output of BERT, equivalently to the Mean Pooling described above, while also fine-tuning a neural network architecture, composed of siamese and triplet networks, to produce sentence embeddings that are semantically meaningful and can be compared with cosine-similarity.

By focusing only on the encoding of the information, and being fine-tuned for this specific task, the Sentence Transformers derived from Sentence-

BERT are very computationally efficient, and can be used to perform semantic relatedness search and clustering in a fraction of the time that would be needed to run the entire BERT architecture.

The fixed-size embeddings generated as output means that all embeddings generated with the same model checkpoint will be compatible with one another. Compatibility for semantic similarity search does not depend on the texts being encoded together in the same batch, which is the case for the BERT Encoder, which computes embeddings while having access to all of the words in the corpus.

The Sentence-BERT model outputs fixed-sized Sentence Embeddings, or Short Text Embeddings, that can be easily explored to calculate the relatedness between them.

6.4.2 Computing Relatedness between Texts

Once a vectorial representation, or a Text Embedding, is available for all texts in the text collection, the next step is to find the relatedness between these vectors.

The relatedness between texts is a generalization of similarity and is the inverse of the distance between two texts. Computing the relatedness between texts becomes then a task of finding the similarity between the two embeddings representing each text.

In order to build a relatedness matrix, which is the intended output for this chapter, as described in section 6.3, it is simply a matter of calculating the relatedness between all of the pairs of texts in the collection.

There are a couple of different metrics that can be used to calculate the similarity between vectors, the most popular ones are the cosine-similarity, the negative Euclidean distance, and the negative Manhatten distance.

6.4.2.1 Distance Metrics between Vectors

When dealing with the Sentence Embeddings generated by using the Sentence-BERT Model, it is usually possible to run experiments using each of the three distance metrics mentioned above, the cosine-similarity, the negative Euclidean distance, and the negative Manhatten distance. However, experiments ran in (Reimers and Gurevych, 2019) show that the cosine-similarity has the best performance among the three alternatives.

Whenever dealing with the Language Model Encoders, however, the semantic similarity search can be quite costly. Luckily, there are workarounds

for this computationally expensive search, the most notable one being the use of a Faiss index, which stands for Facebook AI Similarity Search, and is a library designed for efficient similarity search and clustering of dense vectors by trading precision for memory and speed efficiency.

The distance metric used within Faiss is an efficient approximation of the minimum Euclidean Distance between vectors, where it can return any k-th nearest neighbor for a given vector.

6.4.2.2 Creating a Relatedness Matrix

It is possible to create several relatedness matrices by applying the desired distance metrics to the Embeddings calculated in the previous section. The Proposed Methodology in chapter 7 will discuss the possible combinations for generating the relatedness matrix used for proposing connections between texts.

7
Proposed Methodology

This chapter presents the proposed solution for creating an interconnected graph version of a text collection. The chapter is organized by section 7.1 presenting the goals and motivation for the methodology, section 7.3 explaining in detail the methodology for selecting the edges and concept nodes to connect the text nodes in the graph representing the text collection, section 7.5 provides an explanation for the necessary adjustments for the graph to be navigated using Obsidian, a knowledge base oriented note-taking software tool.

7.1
Goals and Motivation

The major motivation behind this dissertation may be summed up by the goal of increasing human capacity to **acquire knowledge faster and more profoundly**, i.e. learn faster while thinking wider and deeper. The motivations and contexts behind the proposed methodology were deeply discussed in the Research Context, chapter 3.

The envisioned scenario to enable the goal of acquiring knowledge faster and more profoundly is by providing users with an **External Collection of Knowledge** through which the user could navigate their's and other people's ideas. This follows directly from a convergence between the ideas of the Extended Mind Thesis, (Clark and Chalmers, 1998), and the External Repository of Knowledge, (Bush, 1945), coming together to enhance human cognition and capabilities through this external collection of knowledge.

From this higher level scenario, a more tangible way of reaching the initial goal would be to explore the combination between modern **Note-Taking** and **Natural Language Processing** tools to propose a system that can help users to **connect their ideas** by transforming any collection of texts into an interconnected, human-readable, and human-navigable knowledge graph.

The intended path to reach the main goal for this methodology is represented by the problem statement, proposed in section 1.1, and shown below.

Problem: How to automatically generate connections to transform a siloed text collection into an interconnected and inter-navigable text collection, represented by a graph?

Specific Details: How to propose knowledge connections between texts using shared concepts and semantically related texts? How to leverage modern note-taking software tools to enable navigation using the generated connections?

When looking at this problem statement, it is important to remember a couple of important aspects related to Knowledge Management, mentioned in section 3.2. First, within the **4 processes in Knowledge Management**, (Alavi and Leidner, 2001), this dissertation is most interested in the two processes of 1. Knowledge Creation and 2. Knowledge Storage and Retrieval.

Second, with regard to the **Knowledge Representation models**, (Markman, 2013), the idea is to combine the *spatial representation* of a page with the *network representation* by adding connections between the texts.

Finally, when looking at the **Knowledge Visualization functions** for the proposed system in this chapter, the intended purpose for the final output of this methodology is to enhance the tasks of *Recall*, *Elaboration*, and *New Insight*, (Burkhard, 2005).

These are considered important guidelines for the design of the proposed system and will be used in the decision making process, throughout this chapter.

7.2 Methodology Design

This section briefly outlines the intended design for the methodology, by expanding on the intended format for the chapter, while also mentioning some relevant design decisions.

The first important comment is regarding the separation of the entire methodology into 3 distinct chapters, namely chapter 5, chapter 6 and chapter 7, instead of including everything in this chapter.

The main idea behind this separation is to treat the data collection separately from the design decisions of actually proposing the final system to create the interconnected Text Collection. chapter 5 and chapter 6 were written in support of the methodology, to detail the data collection and capture specific technical implementations that would bring unnecessary information to this chapter, which is much more oriented to qualitative and subjective aspects.

This means that when looking at Research Question 2: "How to propose connections between any two given texts present in a text collection?", the solution for this question is actually spread out in different chapters. The information needed to solve the problem was captured in the previous chapters, and the qualitative aspects of the design rationale behind proposing connections and building the graph are presented in this chapter.

The intended use of this methodology is to provide a system that is capable of applying the procedures described in this section to **any given text collection**. The process of generating nodes and edges to connect the graph can also be adapted and personalized to specific needs. This includes multiple versions for a single text collection and also dealing with different text characteristics and file formats.

7.2.1 Inputs and Outputs

When thinking in terms of Inputs and Outputs of the system, the **Input is a collection of text files**, or a dataset containing text, organized together with any relevant information, such as the author, separation by chapters or any other information.

The Output of the methodology is a combination of **a graph** with **a note taking tool that allows for navigation** of the graph. Initially, the output format is a graph, but it needs a means to be navigated otherwise there is no point in making connections. This is done through transforming the nodes in the graph into pages inside a note-taking tool, in this case, Obsidian. Each node is represented as a markdown file, which contains information about the node itself and about any outgoing edges.

7.2.2 Motivating Example

This section will illustrate the procedures in this chapter with visual examples that may enhance the understanding of these procedures, by representing the inputs for the methodology together with the intended output for navigating the text collection.

The input for this motivating example is a combination of the same example text, together with the supporting information extracted in the previous chapters. While the output is the page for the example text in the Interconnected and Navigable Text Collection, with hyperlinks to related concepts and to related texts.

The inputs are represented by the example text, Figure 5.1, as well

Table 7.1: Final Concepts and Mentions list for adding concepts connections

Entity Name	Position		Mention Text
	Start	End	
Personal computer	21	39	personal computers
Technology	77	87	technology
Management	121	127	manage
Money	132	137	money
Productivity	152	164	productivity
Personal knowledge management	187	216	personal knowledge management
Innovation	276	287	innovations
Technology	291	301	technology
Eternity	444	452	timeless
Principle	468	478	principles

Table 7.2: Most Related Texts based on Semantic Relatedness and Concept-based Relatedness

Semantic Relatedness

	Related Text	Score
Same Author	Building a Second Brain Note 19	0.794
	Building a Second Brain Note 3	0.578
Different Author	Measure What Matters Note 24	0.421
	The Almanack of Naval Ravikant Note 11	0.393

Concepts-based Relatedness

	Related Text	Score
Same Author	Building a Second Brain Note 19	0.539
	Building a Second Brain Note 5	0.52
Different Author	The Almanack of Naval Ravikant Note 16	0.647
	The Almanack of Naval Ravikant Note 20	0.568

as by Table 7.1 with the final concepts linked to the text, the Table 7.2 with an example of the most related texts to the example text, used for the recommendation system, and finally Table 7.3 showing the available information, collected for each concept.

The outputs are represented with two images showing the final environment for page navigation. Red boxes are used to bring attention to the different navigation mechanisms in the pages. Each word in blue represents a hyperlink for a linked page, Figure 7.1 presents the final Text Page for the example text, with concepts linked in the text and with a section of recommended texts. While Figure 7.2 shows the final Concept Page for the concept of “Knowl-

Table 7.3: Collected information for each Concept, available for using in the Methodology

Category	Description
DBpedia URL	The UI for the concept
Entity Name	The name of the concept
Description	DBpedia Abstract describing the concept
Image	Link to image illustrating the concept
DBpedia Types	The associated Types to the concept, following OWL or DBpedia Ontology
Subject	The topics of the resource
Wiki Links OUT	Links from this concept's Wikipedia page to other Wikipedia pages
Wiki Links IN	Link from other Wikipedia pages to this concept's Wikipedia page
ConceptNet Relatedness	ConceptNet Quantitative Relatedness to other concepts
ConceptNet Relations	ConceptNet Qualitative Relationships with other concepts

edge", showing some of the related Concepts, as well as the texts that mention "Knowledge", in the right-hand side panel.

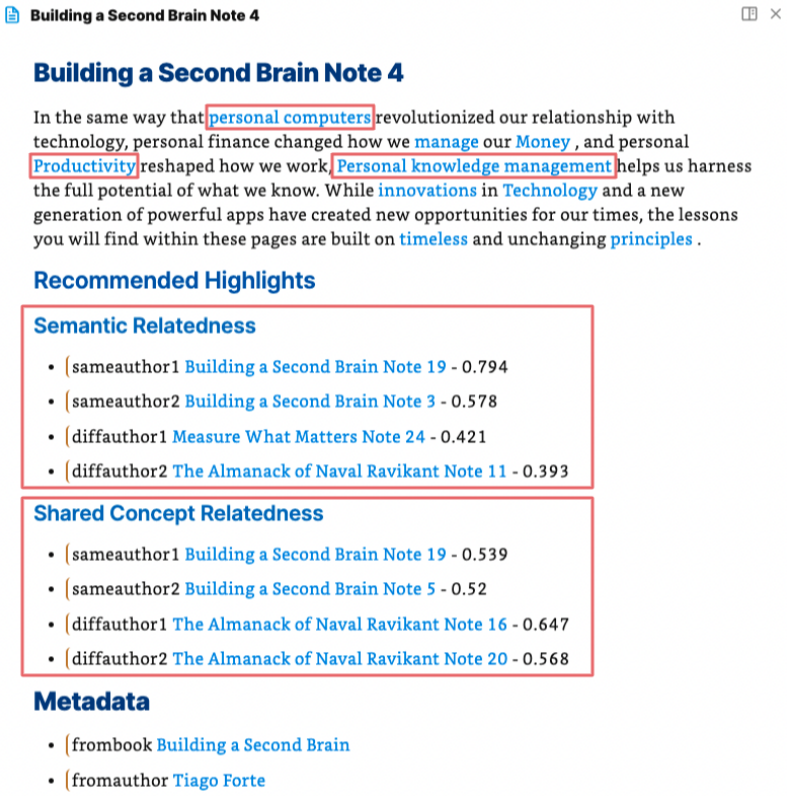

Figure 7.1: The final representation of the example text, within the Obsidian environment with links for the concepts mentioned and for related texts

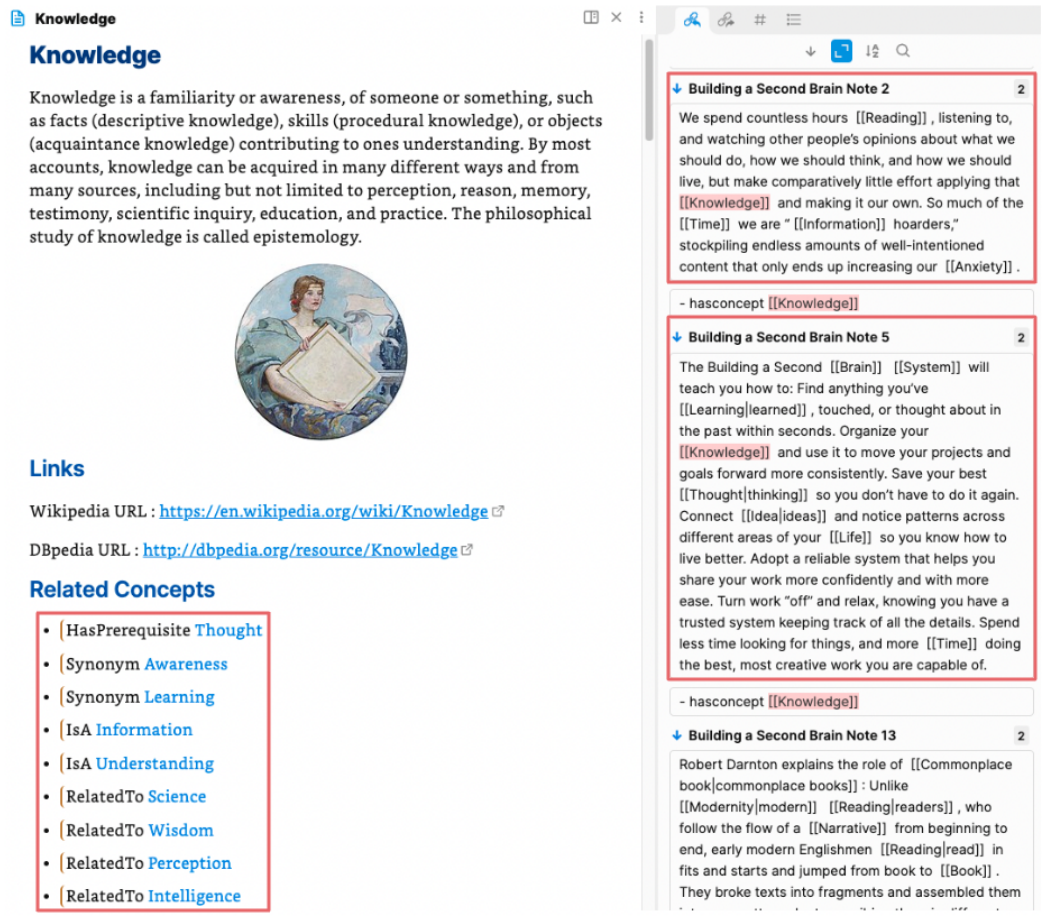

Figure 7.2: The final representation of the Knowledge Concept Page, with connections of texts that mentions "Knowledge" as incoming links (backlinks) to this page.

7.2.3 Procedures

The chosen format for this proposed methodology is split into 3 different sections, Building the Interconnected Graph, section 7.3, Content Selection for Nodes, section 7.4, and Adding Navigation with Obsidian, section 7.5.

This division is done to separate the decision-making process, mainly to separate the task of proposing the knowledge connections in section 7.3, from the task of presenting the connections by choosing the content and format of the Node pages, in section 7.4.

The final section on Adding Navigation with Obsidian, section 7.5, deals with specific features of the Obsidian note-taking software, specifically how the graph is represented using markdown files in a way that Obsidian can process the information and users can navigate the text collection.

7.3 Building the Interconnected Graph

The first step of the methodology is selecting which nodes and edges will form the interconnected version of the text collection, the knowledge graph, G_T. This is mostly a problem of decision making; the data is already available, it is simply a matter of putting data together while making sure that relevant connections are made.

This Section presents the methodology for building the graph that represents the interconnected version of the text collection while explaining the design decisions for adding each type of connection to the text. This includes the rationale behind deciding what concepts to include in the graph, what are the most relevant relationships between concepts, and also how to define the connections between texts.

According to the mathematical problem definition, first presented in subsection 1.1.2, the objective of this section is:

Given a text collection T, create an equivalent, yet connected knowledge graph G_T that can be navigated by users. By adding new node and edge types to the graph.

The graph, G_T, is defined as:

$$G_T = (V, E) \tag{7-1}$$

$$V \subseteq (T, C, A) \tag{7-2}$$

1. T: Text nodes
2. C: Concept nodes
3. A: Author nodes

Each node represents a page, which contains text that may be accessed, consumed, and edited. From these three node types, the goal is to generate edges that represent all combinations of connections between the nodes, which are bi-directional:

$$E \subseteq (\overleftrightarrow{TT}, \overleftrightarrow{TC}, \overleftrightarrow{CC}, \overleftrightarrow{AA}, \overleftrightarrow{AC}, \overleftrightarrow{AT}) \tag{7-3}$$

Consider the reflection made in subsection 1.1.2, which states that to create an interconnected graph, the underlying problem to be solved is simplified to solving for 9 graph components (3 node types, and 6 edge types), while 2 node types and 1 edge type are already known, ($\{T, A\} \in V$; $\overleftrightarrow{AT} \in E$).

The 9 remaining components (3 node types, and 6 edge types) are obtained from 3 distinct tasks. The decision-making for each of these tasks will be discussed in the remaining of this graph-building section.

1. **Text** $\leftrightarrow$ **Concept**: Selecting Concepts for Text Collection
2. **Concept** $\leftrightarrow$ **Concept**: Relation Types between Concepts
3. **Text** $\leftrightarrow$ **Text**: Semantic Recommendations between texts

These three tasks are closely related to the structure of two sub-problems outlined in the introduction section 1.1:

1. Extract Concepts mentioned in the text and their Relationships.
   a. Extract Concepts mentioned in the text. ($C \in V$; $\{\overleftrightarrow{TC}, \overleftrightarrow{AC}\} \in E$)
   b. Identify Semantic Relations between Concepts. ($\overleftrightarrow{CC} \in E$)
2. Compute Semantic Relatedness between Texts. ($\{\overleftrightarrow{TT}, \overleftrightarrow{AA}\} \in E$)

All of the information needed to determine the nodes and edges that will transform the text collection into an interconnected graph are already available, as exemplified by Table 7.1 and Table 7.2.

Next, we detail the design rationale for each segment in building the graph with the information that was collected in chapter 5 and chapter 6. The goal of the following subsections is to justify the design decisions for including or excluding specific nodes and edges while reflecting on the decision-making process.

7.3.1 Text $\leftrightarrow$ Concept: Selecting Concepts for Text Collection

This subsection details the design rationale behind the concept selection (filtering) for the Graph Building phase. This section has as a starting point, the database of mentions extracted from the text, which has detailed information on the recognized Entities. Here, we will discuss the decision-making behind the filters applied to this database of mentions, described in subsection 5.4.2, of Concepts Filtering.

– **Node Type involved:**

  Concept: $C \in V$

– **Edge Types involved:**

  Text $\leftrightarrow$ Concept: $\overleftrightarrow{TC} \in E$

  Author $\leftrightarrow$ Concept: $\overleftrightarrow{AC} \in E$

The final selection of Concepts will determine what Concept nodes will be created, while the edges between Texts and Authors to Concepts will follow from the selected nodes. Edges from texts to concepts are added whenever the concept is mentioned in the text, and the edges from authors to concepts are defined by the most used concepts by each author.

The filters applied will include the Types of Entities, the Confidence Level of the Entity Recognition Task, and Concepts that occur in a single text. The filters are implemented using the available information for the extracted entities. The Table 7.4 shows all of the mentions identified by the Entity Extraction task, as well as relevant information for filtering the mentions database.

Table 7.4: Mentions Database with Concepts extracted from the Example Text, with special detail to Entities that are **filtered out**

Entity Name	Position		Confidence	DBpedia Type
	Start	End		
Personal computer	21	39	0.612	
Interpersonal relationship	59	71	0.503	
Technology	77	87	0.636	
Personal finance	89	105	0.704	
Money	132	137	0.578	
Productivity	152	164	0.681	
Personal knowledge management	187	216	0.865	
Innovation	276	287	0.589	
Technology	291	301	0.636	
Mobile app	335	339	0.47	
Equal opportunity	357	370	0.406	
Timeless (TV series)	444	452	0.667	/TelevisionShow
Principle	468	478	0.458	

Starting with the collected Mentions Database for the Text Collection, several filters are applied, in order to select concepts that are relevant.

7.3.1.1 Entity Types Filter

The Concepts included in the interconnected Text Collection go through a filtering process with the objective to create a graph with concepts that are useful for navigation, while keeping the precision as high as possible.

There is a default setting for which DBpedia Resource Types will be excluded, based on Types that are considered to be noisy or damaging to the precision of the Entity Recognition task. It is also possible to customize the list of Entity Types that are filtered out of the final selection of Concept Nodes.

There are Entity Types that are considered to be damaging to the Entity Recognition task, this happens because DBpedia is a direct mapping of all Wikipedia pages (Lehmann et al., 2015), which means it contains a lot of fine-grained entity types. This results in a lot of mentions being identified as exact string matches of very granular entity types.

Huge amounts of fine-grained entity types usually lead to erroneous tags with high confidence, when the text contains phrases that are exact matches of names of songs, albums, movies, TV series episodes, magazines, etc. These are usually uncommon phrases, but since there are millions of Resources, this happens with a relatively high frequency.

An example of this type of erroneous match is depicted in Table 7.4, where the TV Show Timeless is identified in the text whereas the mention actually refers to the definition of "timeless", the idea of eternal existence. This type of mention is removed from in this filtering stage.

The solution for this challenge, as mentioned above, is to filter out a couple of DBpedia resource types for every text collection, unless explicitly stated otherwise. The types belonging to the DBpedia ontology which are by default filtered out of the mentions database are the following:

- /Actor
- /Actress
- /Band
- /Film
- /Magazine
- /MusicalWork
- /Musician
- /TelevisionShow
- /VideoGame

The common theme among all of these resource types is they are related to the field of Entertainment. While it is certainly possible to represent knowledge about entertainment, in a general way knowledge and entertainment are found on opposite sides of the "Education → Entertainment spectrum" (Cole, 2020), so it makes sense to exclude all entities belonging to these types from the database, and eventually from the connections between texts and concepts, especially so because a considerable amount of them contain erroneous tags due to exact string matching.

The default list may be augmented, according to specific use cases. Categories such as "*/Person*", "*/Location*" and "*/Organisation*" may also be filtered out of the Dataset.

The "*/Person*" and "*/Organisation*" types can be especially important to include or exclude from the Concepts Selection, depending on the characteristics of the Text Collection. For example, if the texts are articles on entrepreneurship and technology companies, then the /Person and /Organization types are very important. On the other hand, if the content of the texts is about Science, a Textbook on Chemistry, then the Person and Organisation should be excluded from the Concepts Selection.

Other potential Resource Types to exclude could be those of "*/FictionalCharacter*", "*/WrittenWork*", "*/SportsTeam*", "*/ChemicalSubstance*", "*/Biomolecule*", "*/AnatomicalStructure*", etc. These would often not represent a huge amount of noise, as any recurrent appearance of these Resource Types would probably be in coherence with the Text Collection actually containing Entities from these Types. Nonetheless, there is always the possibility to further filter the Concepts included in the final Graph.

7.3.1.2 Confidence Level Filter

The Entities (Concepts) included in the graph are also filtered by the confidence level for the mentions that represent each entity. The confidence level filter's main objective is to improve the precision of the entities identified.

Different thresholds for the confidence level were tested, and the default parameter was chosen as 0.60. The removed Concepts from the example text in this stage of the filtering process are also signaled as red in Table 7.4.

The parameter for the Confidence level filter may be adjusted according to needs, by either prioritizing recall or prioritizing precision. Both cases are discussed below.

A lower confidence level can be used for obtaining a higher recall of concepts (more concepts), and populating the dataset with more connections between texts and concepts. It is worth noting that this would decrease the significance of each individual connection, as there would be more alternatives to navigate the dataset, and the probability of utilizing any specific connection would decrease. Another relevant point is that dealing with more concepts means performing more queries for enhancing the information on the concepts, which would demand more computational time and resources.

Decreasing the confidence level would be a useful approach for the task of New Insights, looking to find unusual connections between texts or simply

being able to have more breadth on the navigation possibilities.

A higher confidence level could be used for obtaining a higher precision of concepts (fewer concepts), and this would result in a more sparse graph, with fewer connections between texts. This would increase the quality of connections and would be a constraint on the navigation to more distant texts.

The resulting selection of mentions and concepts after all the undesired concepts are removed for the example text is presented in Table 7.5. It is worth remembering that the idea here is to remove concepts that are not relevant for the **entire text collection**, not for this specific text. This is important for the next step in the methodology.

Table 7.5: Concepts after Entity Type and Confidence Filter

Entity Name	Position		Confidence
	Start	End	
Personal computer	21	39	0.612
Technology	77	87	0.636
Personal finance	89	105	0.704
Productivity	152	164	0.681
Personal knowledge management	187	216	0.865
Technology	291	301	0.636

7.3.1.3 Enhancing Mentions database for the filtered Concepts

The next step following the filtering of the initial concepts is to find additional mentions for the selected concepts. This process will run another layer of Entity Extraction on all the texts, this time using DBpedia Spotlight. This is done to identify any missed or deleted mentions in the texts while adding them to the mentions database.

This is done for three main reasons.

First, by applying a second Entity Recognition algorithm, it is possible to capture any occurrences of the selected concepts that may have been missed by the Dandelion API Entity Extraction. This widens the range of mentions captured while maintaining only concepts that already were “approved” in the filtering process.

Second, by applying the DBpedia Spotlight, we are able to retrieve some of the deleted mentions in the confidence step which actually represent relevant concepts for the text collection. Since the motivation for the confidence filter is to remove concepts that are not relevant for the **entire text collection**, and not for this specific text, this is a measure to read mentions that were deleted, but which are actually related to the text collection.

Third, this step serves as a preparation for the next filtering process, which removes any concepts that appear in **only one text**. This reason builds on top of the first one, by using a second tool for extracting concepts, we raise the chances that any missed occurrence of a concept may be missed, with the intent of it not being filtered out of the interconnected text collection in the next filtering process.

The final result for enhancing the mentions database of the filtered concepts list from the previous sections is presented in Table 7.6, where all additional mentions for the example text are added and signaled as green if they were (re)introduced in this step.

Table 7.6: Concepts after Enhancement with DBpedia Spotlight: Mentions Database for the Example Text

Entity Name	Position		Source
	Start	End	
Personal computer	21	39	Dandelion
Technology	77	87	Dandelion
Personal finance	89	105	Dandelion
Management	121	127	Expansion
Money	132	137	Expansion
Productivity	152	164	Dandelion
Personal knowledge management	187	216	Dandelion
Innovation	276	287	Expansion
Technology	291	301	Dandelion
Eternity	444	452	Expansion
Principle	468	478	Expansion

7.3.1.4 Multiple Occurrences of Concepts Filter

The last significant filtering process applied to the Concepts included in the graph is that of considering only Concepts that have Multiple Occurrences throughout the Text Collection. A concept that has Multiple Occurrences is a concept that appears in more than one text over the entire Text Collection.

This filtering process can be turned on or off using a parameter, according to user preference. The default behavior is to remove the concepts with single mentions to reduce computation time.

The main objective of this filter is to assure that any tagged concept will represent a means for navigation and not a dead end. A dead end is when a concept is only tagged in one text, which means accessing that concept's Node would only allow for navigation back to the original Node where the user was already located in.

This is exactly the case for the concept of "Personal Finance", identified in the example text. This is the reason why the concept is marked in red in Table 7.6, because the example text is the only text in the entire Text Collection in which this concept appears.

This situation is usually undesired because the user would have an energy cost as well as an emotional cost of visiting a Concept Node page in the anticipation of being able to navigate to another text which also portrays such a concept, but in reality, would be forced to navigate back to where the user was.

There are though, situations where the concept page with no connections would be useful, by showing a definition and additional resources for a given concept. Another argument for including isolated concepts because they could still be connected to other concepts that could aid the exploration toward a specific direction.

Table 7.7: Concepts after Multiple Occurences Filter: Mentions Database for the Example Text

Entity Name	Position		Mention Text
	Start	End	
Personal computer	21	39	personal computers
Technology	77	87	technology
Management	121	127	manage
Money	132	137	money
Productivity	152	164	productivity
Personal knowledge management	187	216	personal knowledge management
Innovation	276	287	innovations
Technology	291	301	technology
Eternity	444	452	timeless
Principle	468	478	principles

The default setting of not including isolated concepts is mainly due to reducing computational resources and having a cleaner visual representation whenever possible. Removing the concept altogether allows for reduced visual pollution, and more focus on connections that are productive.

The results for the last filtering stage for the concepts are included in Table 7.7. This table is equivalent to Table 7.1, presented in the Motivating Example in subsection 7.2.2, with the final list of concepts linked to the example text.

7.3.1.5 Manual Filtering

The last resort for filtering the Concept Types included in the graph would be Manual Filtering. This is usually not needed but is an important mechanism to deal with problematic entity types for specific scenarios.

This is usually the case when exact string matches occur between a part of the text and a granular entity type that does not belong to the default list of Entity Types used for filtering.

7.3.2 Concept ↔ Concept: Relation Types between Concepts

This subsection details the design rationale for the selection of relations between concepts that will be added to the graph. Here, we will discuss how these relations are used in the graph building, outline the possible relations that can be added to the graph, as well as propose a hierarchy for these relations.

– **Edge Types involved:**

Concept ↔ Concept: $\overleftrightarrow{CC} \in E$

The first comment is that the level granularity of relation types presented in this subsection is possible due to the ConceptNet Knowledge Base, (Speer et al., 2018), which portrays Commonsense Knowledge in an organized and accessible API. Commonsense Knowledge captures knowledge that humans intuitively understand in the format of relations between concepts, such as Studying → Causes → Knowledge or Reading → Causes → Knowledge.

This is the main reason behind using ConceptNet as a source of information, this type of information is fundamentally different from simply knowing that the Wikipedia page for "Knowledge" has a link to "Reading". The implications of this distinction will be discussed further in this section.

Given that Commonsense knowledge relations are available through ConceptNet, it is interesting to understand what the available relations are. It is also important to identify what are the benefits that different relation types can present to the user.

7.3.2.1 Concepts Relations in the Graph

Before outlining the relation types, it is important to define how the Relations will be added to the graph.

The idea is to portray **relevant** relations between concepts. This means that relations will only be included whenever **both concepts** belonging

to a relation triple, represented by (concept, relation, concept) or (subject, predicate, object), are present in the final Concept list.

7.3.2.2 Possible Relation Types

The possible relations between concepts come from two different sources, DBpedia and ConceptNet. The main distinction between them is that ConceptNet presents a detailed categorization of the relations between concepts, whereas DBpedia simply presents the information that a concept's Wikipedia page has a link to another concept's page.

That said, the possible relations from DBpedia are very simple to outline, either concept A has a Wikipedia link TO another concept B, FROM another concept B, or BOTH TO and FROM the other concept B.

- dbo:WikiPageWikiLink TO
- dbo:WikiPageWikiLink FROM
- dbo:WikiPageWikiLink BOTH

All of the possible relationship types present in ConceptNet are detailed in their online Wiki. From the complete list of relations, the relations considered to be informative or relevant to the purpose of this dissertation were selected. The initial list of relations is described in subsection 5.4.3.

The idea for this large initial list is to capture a wide range of data on relations between concepts, that can either be filtered for specific use cases or simply maintained for exploration purposes. It is worth noting that, since the relation types are very granular, and both concepts must be present in the relation triple for the relation to be added to the graph, usually there are not many relationships triples belonging to each category, with the exception of "*/r/RelatedTo*", which is the most general relation between concepts.

7.3.2.3 Relation Types Hierarchy

In order to be able to prioritize when looking at all of the relationships extracted from ConceptNet and DBpedia, we chose to categorize the relation types into 3 tiers according to the usefulness of the information provided.

This selection is based on the following metrics: the **specificity** of information provided; the **number of relations** identified belonging to each type; the three main functions of visualizing knowledge, Recall, Elaboration, and New Insight.

High value

- Causality
  - /r/Causes
  - /r/CapableOf
  - /r/MotivatedByGoal
  - /r/Desires
- Equivalency
  - /r/Synonym
  - /r/SimilarTo
- Opposition
  - /r/Antonym
  - /r/DistinctFrom
- Dependency
  - /r/HasPrerequisite
  - /r/HasContext
  - /r/HasProperty
  - /r/PartOf
  - /r/UsedFor

The first tier of relations provides information on causality, equivalency, opposition, and dependency. These 4 categories of information were considered by the author to be of high value when exploring a given text collection, This is because they present specific relations between concepts and may be able to guide the exploration of a given subject in a valuable direction.

Average value

- /r/IsA
- /r/RelatedTo

This second tier of relations presents only two relations, which are considered to be less specific than the relations in the first tier, but more specific than the relations in the last tier.

Low value

- dbo:WikiPageWikiLink TO
- dbo:WikiPageWikiLink FROM
- dbo:WikiPageWikiLink BOTH

This last tier presents relations that are considered to be non-specific. The Wikipedia links from one page to another do present a valid source of connection, but it is the most abundant source of connections, and the least specific one, which means these relations are considered to be less valuable as means of meaningful navigation of a text collection.

All the extracted relations for the concept of "Knowledge" are presented following the proposed hierarchy in Table 7.8. The idea here is to visualize how the level of specificity gets lower as we transition form Higher value relations to Lower value relations.

7.3.3
Text ↔ Text: Semantic Recommendations between texts

This subsection details the design rationale behind the creation of connections between texts following the relatedness between texts, explored in chapter 6 and the relatedness between the concepts in the texts, explored in subsection 5.4.5. Here, we will discuss the combination of these two sources of connections, by looking at how both types of connection may complement each other, as well as exploring the semantic relatedness connections individually.

With regard to the Semantic Relatedness connection, the discussion will involve the chosen methods for calculating the relatedness between texts, including the encoding option and the distance metric used.

– **Edge Types involved:**

Text ↔ Text: $\overleftrightarrow{TT} \in E$

Author ↔ Author: $\overleftrightarrow{AA} \in E$

7.3.3.1
Text Semantic Relatedness Connections

With regard to the Text Semantic Relatedness connections, the main comment to be made is on the difference between Simple Word Embeddings, Language Model Encoders, and Sentence Embedding from Sentence BERT.

As previously mentioned, pre-calculated word embeddings do not present the ability to distinguish between different meanings of polysemic words, this is the desired functionality for the Semantic Recommendation system in this methodology. For this reason, pre-calculated word embeddings such as Word2Vec and GloVe were not considered.

The two methods that were considered, as explained in chapter 6 were Language Model Encoders and Sentence Embedding from Sentence BERT.

Table 7.8: Example of the defined Hierarchy for the concepts related to "Knowledge"

Category	Related Concepts to "*Knowledge*"
/r/Causes	
/r/CapableOf	[increase_value, change_people_greatly, advance_mankind, seed_ideas, open_mind, open_human_mind, make_person_happy, make_person_sad]
/r/MotivatedByGoal	
/r/Desires	
/r/Synonym	[cognition, knowingness, ken, awareness, learning, cognizance, knowledge]
/r/SimilarTo	
/r/Antonym	[ignorance]
/r/DistinctFrom	
/r/HasProperty	[powerful, unlimited, power]
/r/HasPrerequisite	[thought]
/r/PartOf	[innovation, understanding]
/r/UsedFor	[cutting]
/r/IsA	[powerful_thing, good_thing, power, applied_information, power_if_used_correctly, information, understanding]
/r/RelatedTo	[erudition, knowing, data, complete, know_how, information, prospective, information, intercourse, science, significant, know, study, known, wisdom, course, carnal_knowledge, perception, place, awareness, know, epistemology, information, intelligence, wisdom, brain, power, gathered, gathered_intelligence, education]
/r/HasContext	[philosophical, archaic, legal]
Wiki Links OUT	['Writing', 'Learning', 'Understanding', 'Technology', 'Belief', 'Mind', 'Personal_knowledge_management', 'Wisdom', 'Post-scarcity_economy', 'Fact', 'Decision-making', 'Peace']
Wiki Links IN	['Writing', 'Learning', 'Understanding', 'Technology', 'Belief', 'Mind', 'Personal_knowledge_management', 'Wisdom', 'Wealth', 'Risk', 'Government', 'Hierarchy', 'Image']

Among these two methods, the chosen one was Sentence Embeddings using Sentence BERT. The main reason for choosing SBERT embeddings is the ability to compare embeddings generated for different "batches" of texts, that

is, embeddings generated in different runtimes of the BERT Model. This is a major limitation of calculating embeddings using the full BERT Model because in order for them to be compatible, the embeddings need to be generated with **ALL** the texts inputted to the Encoder.

This is a very useful feature for the possibility of comparing the embeddings of a newly inputted text with the previously calculated embeddings of the texts that were already present in the Text Collection without needing to calculate them again.

Another reason for choosing the SBERT Embeddings is the intended Knowledge Visualization functions for this methodology, those of Recall, Elaboration, and New Insight. These functions are more broad and exploratory than narrow and precise, there is no need to choose a more computationally expensive method in a tradeoff for more precision.

More precision in comparing texts is not one of the most important aspects of the creation of the interconnected Text Collection. The precision which is actually important is that of extracting concepts from the text and relations between concepts. Other than that, a high level of precision is not a pre-requisite.

With regard to the distance metrics, we chose to keep the default distance metric of the cosine-similarity.

7.3.3.2 Recommendation System

The recommendation system will be built using all the possible connections available, opting for a more divergent approach, presenting multiple possibilities for navigation of the text collection.

This means that initially, both the Concept-based Relatedness and Semantic Relatedness metrics will be used. These will be presented separately from each other, in order to inform the user which hyperlink corresponds to each of the relatedness metrics.

The recommendations are built by accessing the Relatedness Matrices built in the corresponding sections to each of the relatedness metrics, in subsection 5.4.5 for Concept-based relatedness and in section 6.4 for the Semantic relatedness.

In order to exemplify this selection from the relatedness matrix, Table 7.9 presents the three most similar texts to the example text:

> "In the same way that personal computers revolutionized our relationship with technology, personal finance changed how we manage

> our money, and personal productivity reshaped how we work, personal knowledge management helps us harness the full potential of what we know. While innovations in technology and a new generation of powerful apps have created new opportunities for our times, the lessons you will find within these pages are built on timeless and unchanging principles."

Relatedness Score	Text
0.795	I've come to believe that personal knowledge management is one of the most fundamental challenges—as well as one of the most incredible opportunities—in the world today. Everyone is in desperate need of a system to manage the ever-increasing volume of information pouring into their brains. Those who learn how to leverage technology and master the flow of information through their lives will be empowered to accomplish anything they set their minds to.
0.578	To be able to make use of information we value, we need a way to package it up and send it through time to our future self. We need a way to cultivate a body of knowledge that is uniquely our own, so when the opportunity arises—whether changing jobs, giving a big presentation, launching a new product, or starting a business or a family—we will have access to the wisdom we need to make good decisions and take the most effective action. It all begins with the simple act of writing things down.
0.545	This digital commonplace book is what I call a Second Brain. Think of it as the combination of a study notebook, a personal journal, and a sketchbook for new ideas. It is a multipurpose tool that can adapt to your changing needs over time. In school or courses you take, it can be used to take notes for studying. At work, it can help you organize your projects. At home, it can help you manage your household. However you decide to use it, your Second Brain is a private knowledge collection designed to serve a lifetime of learning and growth, not just a single use case. It is a laboratory where you can develop and refine your thinking in solitude before sharing it with others.

Table 7.9: The three most Related texts to the Example text, according to Semantic Relatedness metric

All three of the most related texts to the example text are from the same author and same book. This is considered to be limiting following the objective of proposing connections directed at Elaboration and New Insight because the connections would be limited to texts from the same author, which narrows the possibilities of being exposed to new ideas.

Same Author and Different Author

With the intent of diversifying the proposed connections, two types of recommendations are made, to texts from the Same Author and to texts from Different Authors. The objective here is that of being exposed to Elaboration from the same author as well as to New Insight by different authors, allowing for the two options while navigating from a given text node to other nodes.

Connections from the same author are useful to deepen the understanding of the ideas presented in the text by being prompted by supporting texts from the same author, which probably come from the same worldview, and possibly form a trail of thought.

Connections from different authors, however, are useful to present alternative knowledge related to the present text, which does not necessarily follow the same worldview, and may compose the big picture of the subject by suggesting less predictable paths of related texts.

Finally, it is worth noting that the number of connections displayed is a parameter of the methodology. The parameter refers to the number of each connection type shown in the final output. It is worth noting that the parameter is flexible and may be changed for each distinct connection type, the two different metrics; Concept-based Relatedness, and Semantic Relatedness, as well as for texts; from the Same Author, and from Different Authors.

This means that if the parameter for all connection types is set to 3, the system will actually generate 12 different connections to other recommended texts.

Additional Option

Here we present an additional alternative that is not implemented but could be interesting for specific use cases of the proposed methodology.

Regarding the specific function of "New Insight", there is an interesting possibility of generating connections using a non-deterministic approach, using a normal distribution to suggest n texts, instead of relying solely on the n most related texts. This would widen the scope of the connections even further, and could be a source of even more serendipity between texts (Eichler et al., 2017).

7.4 Content Selection for Node Pages

This section details the last step in creating an interconnected Text Collection, which is selecting the content that will be added to the Markdown files belonging to each node type. For each of the three node types, Text, Concept, and Author, we present a small discussion as to what information will be available for each node, and how this should enhance the experience of

navigating the Text Collection.

The essence of this section is to detail how the different navigation paths are organized and prioritized. With this in mind, each node type will be analyzed by discussing two different components, the body of the node's text, which contains outgoing links, and the incoming links to that node, which may be presented in a sidebar on the right-hand side or at the bottom of the page, as an extension to the page.

As explained in subsection 7.5.2 on Obsidian Functionalities, Obsidian supports YAML metadata to represent node types, and it must be the first information in each file. The metadata appears before the title itself, which makes a clear distinction between what belongs as metadata and what is part of the actual content of a given node. This subsection discusses the content and the order of the contents after the YAML metadata and the title, which are always going to be the first visible information.

7.4.1 Text nodes

Body of Page:

The fundamental information inside the Text Collection is the individual content belonging to each Text node. Naturally, this will be the most important information represented in the Text nodes as well. Being so, the text will unsurprisingly be the first and central piece of information displayed inside the Text nodes.

The next most important information after the text itself could be either the concept nodes mentioned in the text or the semantic recommendations to other text nodes. The choice between which option should be represented first, though, is simple.

The subsection 7.5.3 on integrating the text to Obsidian explains that the hyperlinks to the concepts mentioned in the text are represented using double brackets inside the text itself. This means that both the text and the concept nodes mentioned in it can be represented simultaneously as the centerpiece for the Text nodes.

This leaves us with the last proposed path for navigation, which is the text recommendations to other text nodes. The semantic and concept-based recommendations will come right after the text, as the secondary navigation paths.

Here, the recommendations to other text nodes are divided between the two relatedness metrics, Concept-based and Semantic, and by texts from the same author and belonging to different authors. The idea behind this is to

allow for multiple options while navigating from a current text node.

The last piece of information in the body of the Text nodes is the metadata regarding the edge type. This is used mainly for integration with the Juggl[1] extension, to enable visualizing different edge types in the graph view using different colors that help better understand and visually navigate the Text Collection, this will be explained in section 7.5.
The final layout for the body of Text nodes:

- Node Metadata
- Title
- Text with Hyperlinks to Concepts
- Semantic Recommendations
- Edges Metadata
- Incoming links

Incoming Links:

The only type of information portrayed in the incoming links for Text nodes is the other Text nodes that have semantic relatedness with the current Text node. This information can be quite redundant, as the scores for relatedness are symmetric, so the highest-scoring outgoing links should have a large overlap with the highest-scoring incoming links.

The final design for the Text Node page is presented in Figure 7.3, putting together all the elements explained in this subsection and using the text node corresponding to the example text as the chosen node. The incoming links are presented on the right-hand side for ease of fitting all of the information into a single image.

7.4.2 Concepts

Incoming Links:

The incoming links play a central function for the concept node pages, as it provides a connection between the current concept node and texts that mention that concept. This leads to the incoming links section being the most important component of the concept node's page because this is how the texts which mention the concept may be visualized and accessed.

It is worth noting once again that the Incoming links may be added to the bottom of the page, or viewed in the right-hand side, as it is displayed in Figure 7.4.

[1] https://juggl.io/Juggl

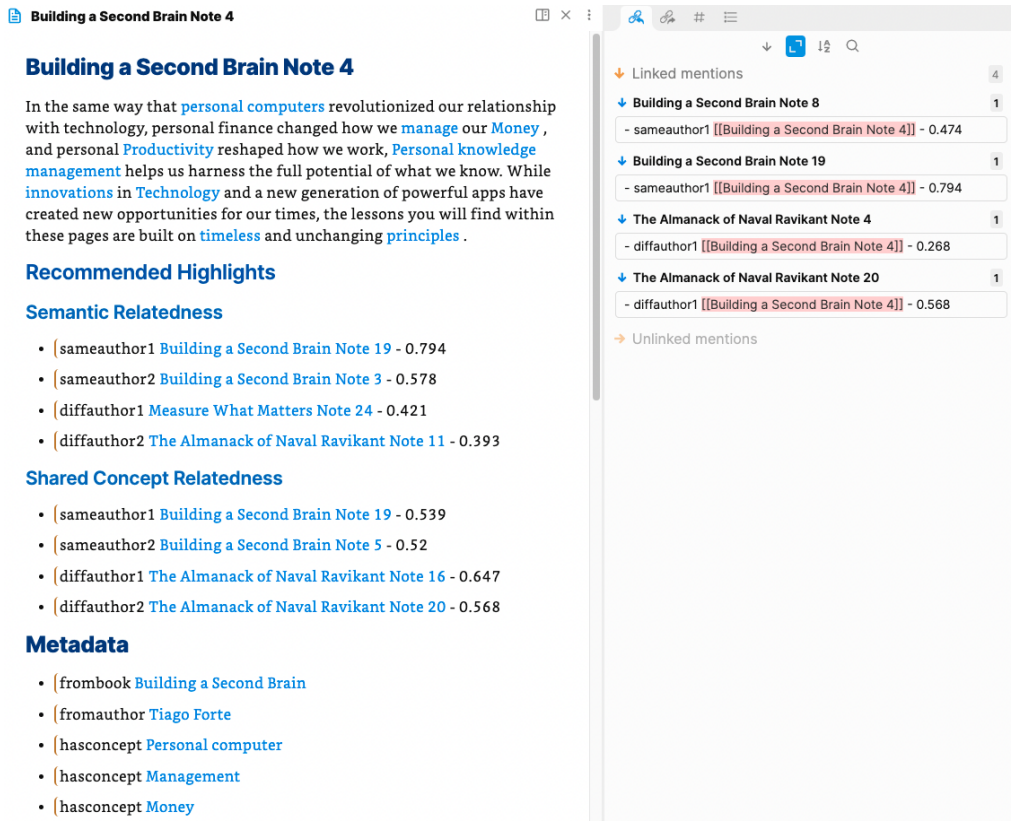

Figure 7.3: Design for the Pages of Text nodes, illustrated by the page of the example text

Body of Page:

The body of Concepts node begins with the description of the concept, the text describing the concept is considered to be the best representation of that specific concept. The first information to be displayed regarding a concept should be its best representation. There is also the feature of Transclusion, whenever hovering over a hyperlink to a page, the first paragraph appears without having to navigate to that page. For concepts, what will appear within the Transclusion is the description of the concept.

The next representation is the collected image for the Concept. The purpose of the image is not really to inform anything, but rather to compose the concept‘s page and to provide a visual representation of the concept.

Following the images, we present two links for additional resources on the Concepts, the Wikipedia link and the DBpedia link. Since all the extracted entities are mapped to a DBpedia resource, all extracted concepts will have a corresponding Wikipedia page, which is linked in order to support any further research and exploration of any given concept and subject. The DBpedia link is also provided as a means for additional information, for any case in which it may be useful to access the DBpedia information for a concept.

The last piece of information presented on the Concepts page is the set

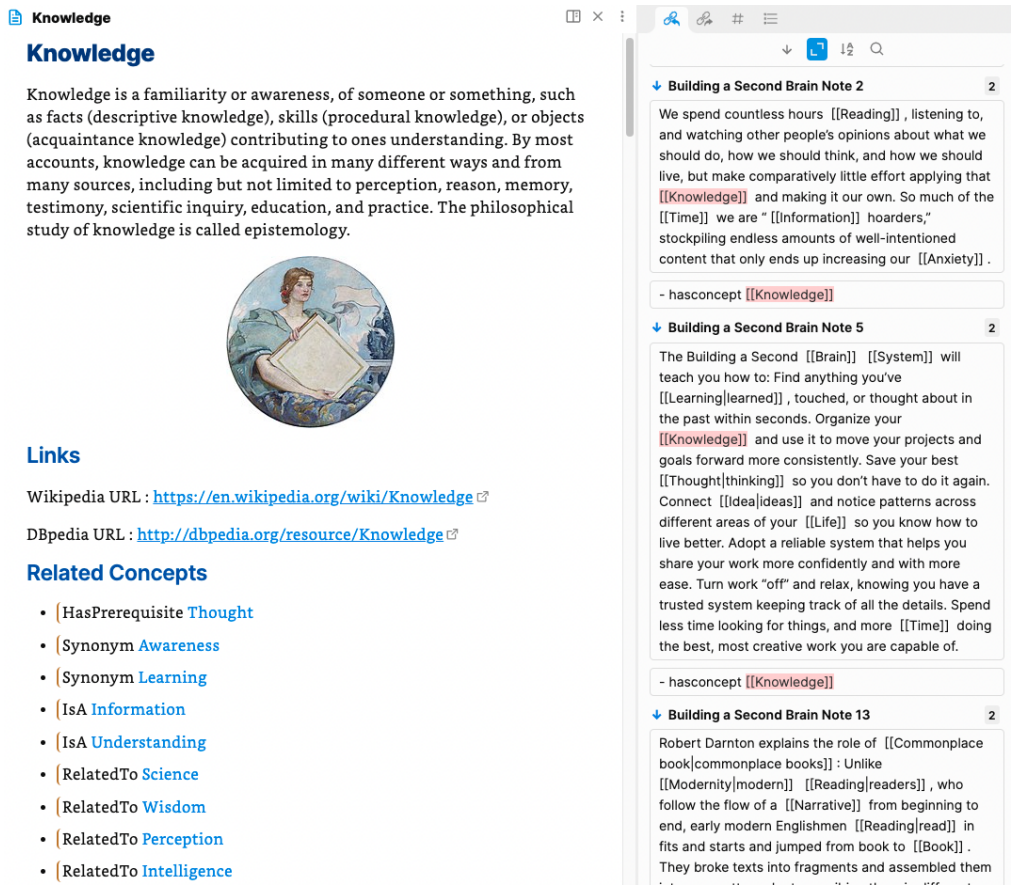

Figure 7.4: Design for the Pages of Concept nodes, illustrated by the page for "Knowledge"

of Related Concepts. These are added to the page as a means of navigating to other concepts, exploring the text collection, and additionally, as a means of better understanding the knowledge portrayed in the Text Collection. A visual representation that two concepts are connected may be valuable information for associative thinking and creating new mental connections.
The final layout for the body of Concepts nodes:

- Node Metadata
- Title
- Description of the Concept
- Links to Wikipedia and DBpedia
- Links to Other Concepts
- Incoming links

The final design for the Concept Node page is presented in Figure 7.4, combining all the elements explained in this subsection and using the familiar "Knowledge" concept as an example. The incoming links are again presented

on the right-hand side for ease of fitting all of the information into a single image.

7.4.3 Author

Body of Page:

The body of the Author nodes begins with the most used concepts by the Author. This is a parameter for the methodology, meaning the number of concepts may be altered at will. The concepts most mentioned by an author are the most relevant information following the proposed connections in this dissertation, as they provide a means of navigating to other pages which may have relevant knowledge from the author, as well as from other authors.

Another reason for starting with the most used concepts is that the body of Author node does not have any text content or description, as do the Text and Concept nodes, as well as not having much information to be displayed, which means there are not many options to chose from to start.

Following the links to concepts, we present connections to other related authors, which is a short piece of information, making it an easy choice to come before the Texts list, which can be long, depending on the number of texts present in the initial Text Collection.

Finally, the last piece of information is a list of all the Texts belonging to the Author. This is presented in order to collect all the texts from the same author in one single place, with the additional benefit of being able to use the Transclusion feature and see the content of every text by hovering over the hyperlink.
The final layout for the body of Author nodes:

- Node Metadata
- Title
- Most used Concept
- Links to other Author
- Links to Texts by Author

The final design for the Author Node page is presented in Figure 7.5, combining all the elements explained in this subsection and using the author “Tiago Forte” as an example.

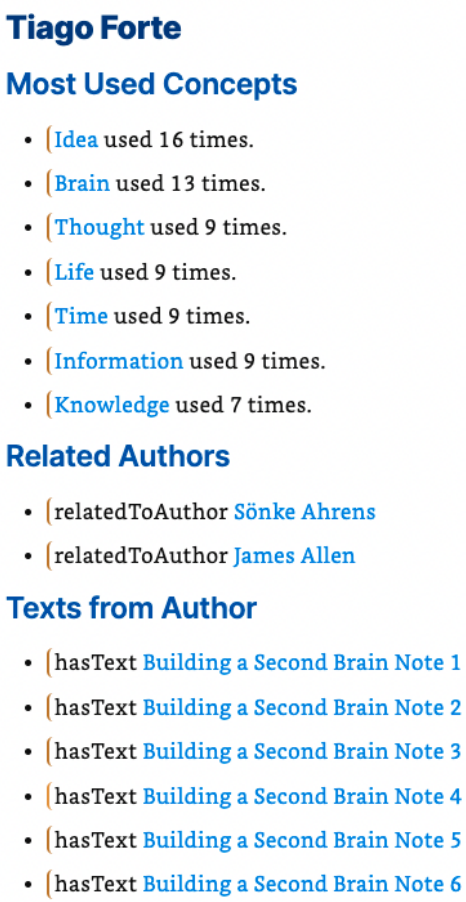

Figure 7.5: Design for the Pages of Author nodes, illustrated by the page for author "Tiago Forte"

7.5 Adding Navigation with Obsidian

This Section presents the methodology for using the graph constructed in section 7.3 together with the Text Collection and the pages designed in section 7.4, integrating these items together into the note-taking software called Obsidian, to create an interconnected Text Collection. This is done by adapting the original texts belonging to existing nodes in the collection and creating the text for new nodes, according to the proposed design.

It is important to note that the main focus of this subsection is Navigation. The connections already exist, each node is interconnected into the graph, and the missing element is Navigation.

Given the set of final nodes and edges following Equation 7-2 and Equation 7-3, the idea is that users are able to navigate through the Text Collection using the note-taking software Obsidian. This depends on representing the nodes as markdown files containing the desired text and linking conventions that enable Obsidian to show the desired information.

This section is divided into 4 subsections, subsection 7.5.1 explains the reasons for selecting Obsidian as the means for Navigating the Text Collection, subsection 7.5.2 explains the basic functionalities of the software, subsection 7.5.3 describes the adaptations performed on the text to suit to how Obsidian works, and finally subsection 7.5.4

7.5.1 Reasons for selecting Obsidian for Navigation

Before jumping into how the integration and the text selection for each node are done, we first outline the reasons for choosing Obsidian as the tool for Navigating the data.

There were three features that were considered essential in a note-taking software for it to be considered as the navigation device for the proposed methodology:

Simple **convention to add connections (hyperlinks)** between pages. **Connections between pages are accessible in both directions** (to & from) Possibility of **automatically generating files** for the software to read.

Obsidian contains all of these features, there is a simple convention for adding hyperlinks, they are shown on the pages for both nodes that share a connection, and Obsidian works using local Markdown files, which can be easily edited and automatically generated.

In addition to the essential features, Obsidian also has several other functionalities that make it an attractive solution for navigating the graph representing the Text Collection. Obsidian is available for free, there is a built-in graph view of nodes and edges (both global and local), and hyperlinks show a preview of other pages (nodes), additionally, there is also a wide range of community plugins available for enhancing the experience of using the tool (End-user development).

7.5.2 Obsidian Functionalities

Obsidian is a popular note-taking software, at the time of writing, having reached 60,000 followers on Twitter and 40,000 users in the official subreddit community. The home page for the website describes the tool as being “a powerful knowledge base on top of a local folder of plain text Markdown files.” This means Obsidian is a note-taking software based on local files, which is also a knowledge base.

One of the most interesting features in Obsidian is that it combines Hierarchical structures with Networked operations. Obsidian presents the Hier-

archical structure of files and folders alongside the Networked functionalities of bi-directional hyperlinks. This opens up several possibilities with regard to making the most out of each of these two types of operation with data. Both of these types of organizing data are useful means of navigation and may be leveraged for an improved user experience.

Files inside Obsidian represent nodes, each node is a page inside the notes knowledge base and is represented by a Markdown file, which has a unique title that serves as a Unique Identifier for nodes.

Edges are represented by hyperlinks and live inside nodes. The information which defines an edge is represented by the text of a node, by referring to another node following a naming convention. The edge is rendered automatically by the Obsidian software, displaying the connection between two nodes.

One of the most useful features in Obsidian is that it displays the backlinks, or incoming links for every node, which means that whenever users are navigating, it is possible to access all of the nodes which have connections to the present node.

The naming convention for representing a link from “node A” to “node B” is to include the title for “node B” inside double brackets [[node B]] as a text for the file representing “node A”, as shown in Figure 7.6.

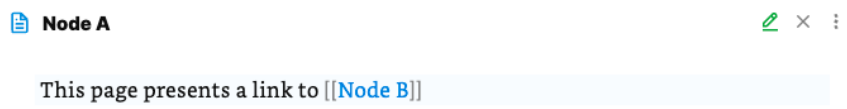

Figure 7.6: Example of “node A” with a link to “node B”.

As with a conventional graph, edges in Obsidian depend on nodes to exist, but in this case, edges need only one node. This happens because, differently from the RDF representation, where the edge is defined by a triple of (node, edge, node), in Obsidian, edges live inside the nodes. This means that if an edge (a hyperlink) points to a node that does not exist yet, i.e. does not have a file representing it, the edge still exists and is displayed in the software, but to an empty node.

Another interesting functionality for Obsidian is the use of metadata for defining node and edge types. YAML can be used as metadata to define node types as a header for the file representing a node, while to define edge types, it is possible to use a community-based plugin to display edge types by following a metadata convention.

The plugin used for this use case is called Juggl[2], and it was built for customizing and enhancing the graph view and navigation by identifying node and edge types and adding colors and shapes to differentiate different nodes and relationships.

7.5.3 Adapting the Text to Obsidian

This subsection explains in very light technical detail the step-by-step procedures for editing the texts for integrating the Text Collection with the way that Obsidian works behind the scenes. The procedures are divided into 3 parts, creating new pages (nodes), adding hyperlinks (edges), and adding metadata (types).

Creating Pages

The first task is creating the pages to represent the nodes in the graph. This is a necessary step for new nodes as well as for the existing text nodes, because the final output format for each node is a Markdown file, and the original Text Collection is represented by a database. This means that texts need to be represented in a format that Obsidian understands and displays, which is as local Markdown files.

Each node has a Markdown file generated for it containing the text selected for that node, following the design decisions explained in section 7.4.

Adding Hyperlinks (Connections)

The next step is to create hyperlinks that represent the edges of the graph. A hyperlink is created to represent each and every edge of the graph. The hyperlinks are created inside the outgoing node, according to the convention detailed in the previous subsection, subsection 7.5.2.

Here, we briefly explain how this convention is used together with the practical differences between using mentions or entities when generating hyperlinks.

Firstly, each and every node will have a unique title which is a Universal Identifier that represents each node. Whenever an edge is created towards a Text or Author node, the title for the desired node is mentioned inside double brackets: [[title]].

When dealing with connections from Text nodes to Concept nodes, the node which contains a mention to a specific concept is edited to include a hyperlink from the text node to the concept node. This is done in either of two ways, depending if the mention has an exact string match with the Concept's title or not.

[2]https://juggl.io/Juggl

If there is an exact match, the original string of the text is edited to include the title inside double brackets: [[concept_title]]. If an exact match is not found, then an alias is created in order to maintain the original text in its exact wording while being able to navigate to a Concept's page, by using the format of: [[title|alias]], in our case [[concept_title|mention_text]].

Adding metadata for node and edge types

Metadata is added by simply using a YAML parameter of "tags" followed by the corresponding value, which is one of the node types in Equation 7-2, Text, Concept, or Author. It is worth noting that nodes' YAML metadata must be included as the first inputs to the file, otherwise they are invalid YAML parameters.

The edge types metadata work a little bit differently, they are added by adding a second hyperlink between the two connected nodes, this time around by using the metadata that describes the type of relationship between them.

The most important relations between nodes are hasConcept, fromAuthor, semanticRec, and also any specific relation between Concept nodes.

7.5.4 Navigation options in Obsidian

This subsection presents a brief overview of the available options to navigate through Obsidian. The objective is to demonstrate some of the different alternatives when exploring a Text Collection.

The basic functionalities to navigate a dataset are by using searching, navigating straight to the desired Text Node or Concept Node using the File Explorer, or by navigating through the present page into other pages. The Figure 7.7 presents these three types of basic navigation with colors to show where each is located, search is in red, file explorer in orange, and links navigation in yellow.

It is worth noting that Obsidian has a distinction between edit mode and reading mode, which makes navigation more effective by having a dedicated functionality for navigation. One of the features that are available in reading mode is that of Transclusion. Transclusion is when content from one page is included in another page, without having to leave the original location. An example is demonstrated in Figure 7.8

In Obsidian, it is possible to open multiple pages at once, by navigating to the desired page by holding Ctrl(Cmd) + Click. This can be used to open different pages related to the same subject and study them simultaneously. An example of this is shown in Figure 7.9, where two different texts which contain the concept of Knowledge are opened simultaneously on the right-hand side.

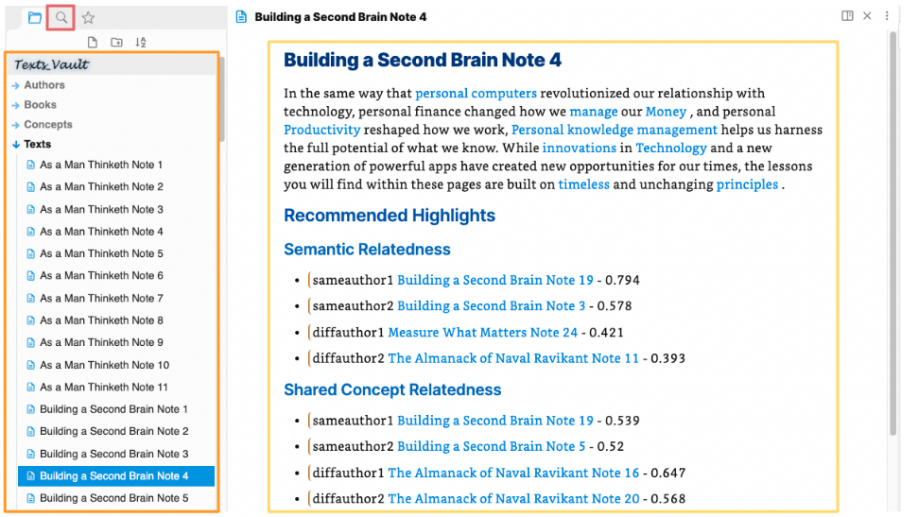

Figure 7.7: The basic navigation functionalities in Obdsidian.

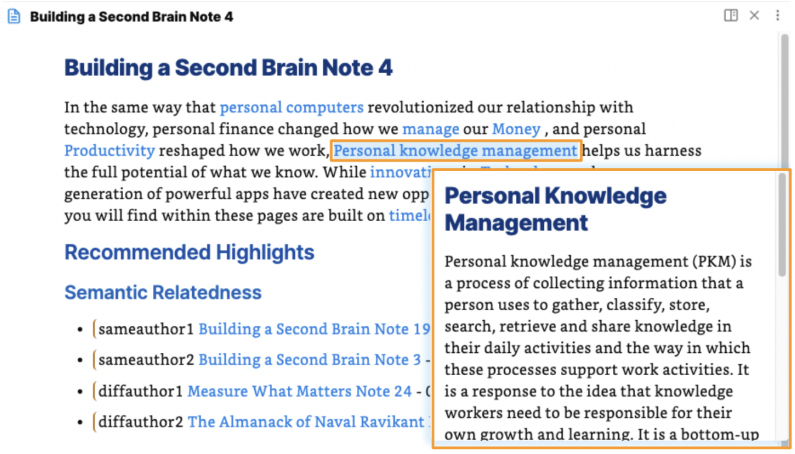

Figure 7.8: The functionality of Transclusion in Obdsidian.

Obsidian also has graph views which allow for additional information. There is the global graph view, which usually does not present much information outside of a visual representation of the number of edges connected to a node, as exemplified in Figure 7.10.

Finally, there is also a local graph view that shows neighboring nodes to the currently selected node. In order to enhance this visualization, we added an extension called Juggle, which is capable of assigning shapes and colors to node types. In Figure 7.11 we present the graph view for the example text used throughout the methodology, with the nodes that share edges with this page, it is worth mentioning that the exact same links are available in the Text Node page and in the local graph view. Concepts nodes are represented by red pentagons, text nodes by blue circles, and the Author node by a green triangle.

The combination of features presented in this section provides several different mechanisms for navigating the Text Collection, which support the

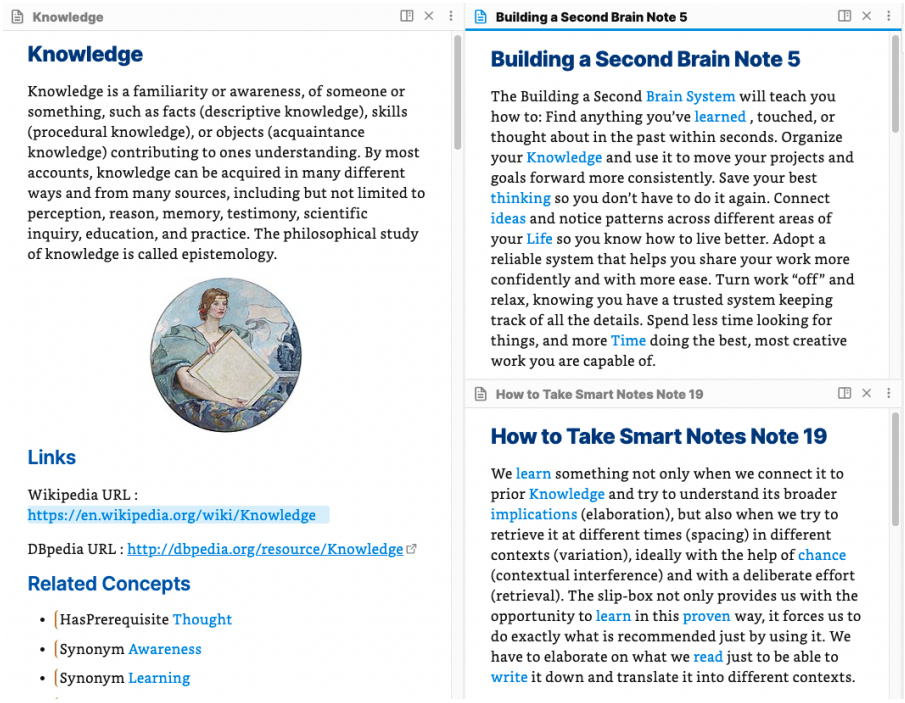

Figure 7.9: Multiple pages open in Obsidian

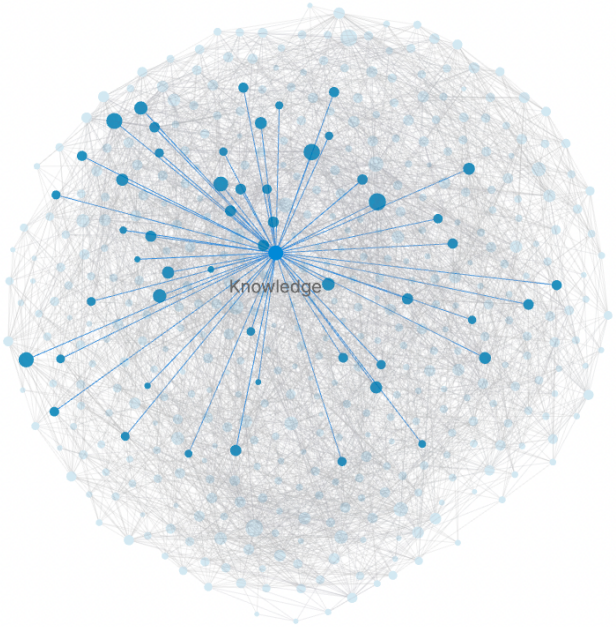

Figure 7.10: Global Graph view in Obsidian

task of exploring the Text Collection and providing the user with multiple options for doing so.

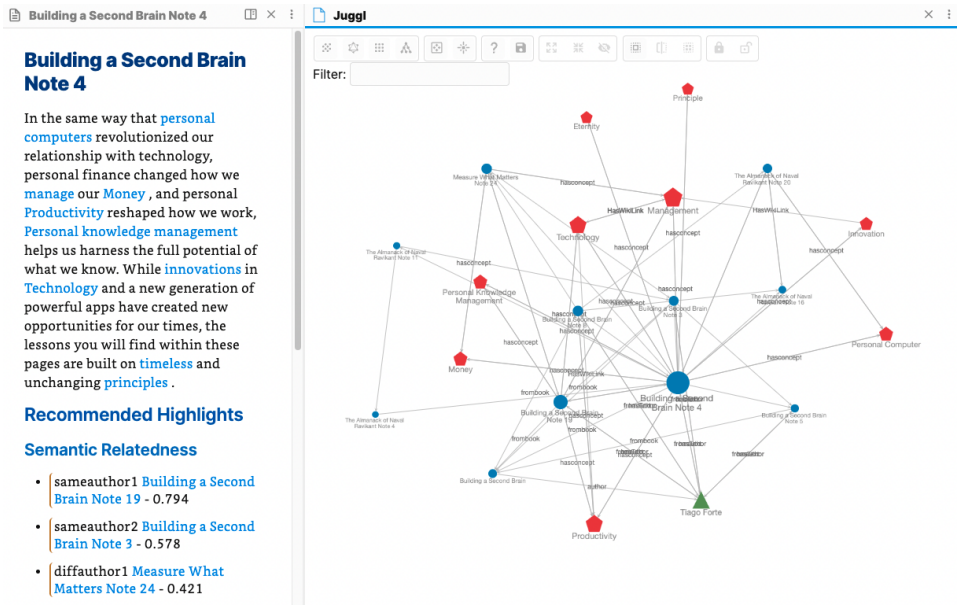

Figure 7.11: Local Graph view in Obsidian with the Juggle extension

8
Evaluation

This chapter explains how the proposal was evaluated, following a quantitative metric of coherence between connections and a qualitative experiment of the generated navigation system. This section also discusses the relevance of this proposal to the activity of exploring knowledge connections from the perspective of Recall, Elaboration, and New Insight.

8.1
Evaluation Planning

The evaluation of the proposed methodology is split into two distinct parts, the first is the main evaluation, which looks at the fundamental functioning of the implemented system, to determine whether the implemented system corresponds to what was proposed in the methodology. This part corresponds to the main evaluation in this chapter.

A second, and less structured, part of the evaluation is regarding the possibilities that the proposed methodology actually allows for, this is a discussion-oriented evaluation, where the chosen Research Questions will be analyzed through the lens of what was actually implemented and what is possible with it.

The **computational experiment** designed to evaluate the proposed methodology is centered around performing an analysis of the fundamental features of the system.

In order to carry out the experiment, two different-sized subsets of the text collection were used. The selected text collection is composed of book passages (highlights) from several different books. The highlights were collected from the Kindle digital reader using the Readwise[1] tool. The two subsets are organized in the following way:

The **small test-set**, or test-set 1, has 52 passages from two books, this set has connections that can be easily interpreted. This small test set is used because it is much easier to analyze the fundamental functioning of the system in a small test set, where the connections can be manually analyzed and easily interpreted.

[1]https://readwise.io/ – A knowledge integration software.

The **medium test-set**, or test-set 2, has 182 passages from eight books, and has connections that are less obvious to interpret, but still under control for human interpretation. The idea behind this test set is to interpret how the fundamental features of the system scale for a slightly larger text collection.

Table 8.1: Books and Authors that are included in the two test sets.

Small Test-Set	**Medium Test-Set**	**Author**
Feel the Fear and Do It Anyway	Feel the Fear and Do It Anyway	Susan Jeffers Ph.D.
Courage is Calling	Courage is Calling	Ryan Holiday
	The Obstacle is the Way	Ryan Holiday
	As a Man Thinketh	James Allen
	Building a Second Brain	Tiago Forte
	How to Take Smart Notes	Sönke Ahrens
	Measure What Matters	John Doerr
	The Almanack of Naval Ravikant	Eric Jorgenson

Table 8.1 presents the books selected for each of test set. The configuration of the small test set is very simple, two books with easy-to-interpret relationships were chosen, in this case, portraying the general topics of fear and courage.

The books for the medium test set were selected while trying to fulfill a number of properties:

- Treat different topics.
- Expand on the small test set.
- Include multiple books from the same author.
- Include both; concepts that are closely related, and others that are not related.
- Depict the topic of Personal Knowledge Management, mainly because it is closely related to this dissertation.

These properties seek to structure the raw input into the system such that the system may be tested and evaluated on important functionalities for the functioning of the methodology.

8.1.1 Objective

The main goal of this evaluation is to propose and answer a set of questions. These questions seek to analyze whether the **fundamental** features for the functioning of the system are implemented according to the proposed methodology.

Based on the problem definition, we selected the features that are necessary for the system to be considered a successful implementation to solve the problem. The problem definition, as stated in section 1.1 is:

Problem: How to automatically generate connections to transform a siloed text collection into an interconnected and inter-navigable text collection, represented by a graph?

Specific Details: How to propose knowledge connections between texts using shared concepts and semantically related texts? How to leverage modern note-taking software tools to enable navigation using the generated connections?

With this in mind, the features considered fundamental for solving the problem are:

1. Inter-connectivity of the Text Collection
2. Navigability of the Text Collection
3. Accurate Graph Representation of Text Collection
4. Coherence of Knowledge Connections
    a. Shared Concepts Connections
    b. Text Semantic Relatedness Connections
    c. Coherence between both types of connections

These features will be explored using a combination of qualitative and quantitative analysis. The focus is to analyze how the features have been implemented, and whether the implementations are considered to be successful in solving the respective segment of the problem definition.

8.2 Procedures

This section outlines in further detail the chosen procedures to evaluate the fundamental features of the system. Whenever possible, a combination of quantitative and qualitative metrics was applied for each feature, in order to produce an all-around evaluation.

It is extremely important to note that the parameters used in the proposed methodology directly influence the quantitative results analyzed in this section. Changing the parameters for the confidence of Concept Recognition and the number of Text Recommendations for each Text would generate different results by modifying the nodes and the edges of the graph.

Even though the quantitative results vary according to different parameters, this is not a problem for the Evaluation, since the objective here is to analyze the extent to which the proposed features are actually present in the implementation. This is an analysis that tends to be binary, either the proposed features are present or they aren't, the objective of this first main part of the evaluation is to determine whether the system is functional at a fundamental level.

The parameters chosen for the evaluation were of **one (1) text recommendation** for each of the 4 combinations between Semantic Relatedness vs. Concept-based Relatedness, and Same Author vs. Different Author, and **confidence of 0.6** for accepting concepts to the text collection.

It is worth noting that for any analysis based on graph features, Author nodes and corresponding edges were not used, as they would provide for very fast navigation between text nodes, and are not the focus of this evaluation.

8.2.1 Interconnectivity of the Text Collection

The interconnectivity of the Text Collection is the most important feature together with the Navigability. The interconnectedness is the essence of this dissertation, it represents the act of moving from a paradigm where ideas are kept separate from each other to a knowledge organization paradigm where ideas are connected, accessible, and malleable, a paradigm that allows for interaction between different disciplines and different perspectives.

The idea behind evaluating the feature of interconnectivity is to determine to what extent all of the text nodes are connected with each other. It is worth noting that, as a matter of design, the text nodes will very probably be connected to all other nodes, given that edges are added to each text node.

The proposed evaluation to analyze the extent to which text nodes are

interconnected is primarily based on the ease of access. One way to evaluate this is the shortest path between nodes. The shortest path is defined as, starting from one node, how many edges need to be navigated in order for the user to reach the other node. Another relevant metric for analyzing the extent to which nodes are interconnected is the degree of a node, which is defined by the number of edges that are arriving or leaving a given node.

A set of questions was proposed in order to validate the feature of the Interconnectivity of the Text Collection. The questions focus on the Text Nodes and look to understand the extent to which the text nodes are connected to each other. The questions seek to understand the average scenario as well as the worst-case scenario, to make sure the entire text collection is interconnected, and not only a selected section.

Question 1.1: What is the **average** and the **longest** shortest path between text nodes?

This question seeks to understand if the texts are actually connected with each other. The essential element of the Text Collection are the texts, so this is the place to start evaluating if the collection is interconnected.

Question 1.2: Are the pairs of nodes that share the longest shortest path actually unrelated to one another?

This question is meant to understand if the text nodes should actually be distant from one another. Do they represent unrelated ideas or are they actually somewhat related and could have been more closely connected by the system?

Question 1.3: What is the **average** and the **smallest** Degree for all of the text nodes?

Analyzing the degree is a way of looking at the number of navigation options for a given node. This question is directed at, given a specific node, determining the average number, and the worst-case scenario of available options for navigation. This question is a great of the level of interconnectivity for the text nodes, which are considered to be the most important.

Question 1.4: Are the nodes with the lowest Degree dissonant with the other texts in the text collection?

This question serves the purpose of analyzing the coherence of the degree metric, to determine if the most isolated nodes are actually not as related to the rest of the text collection or if they should be more connected to the rest of the texts.

8.2.2 Navigability of the Text Collection

The navigability of the Text Collection is the other major feature alongside Interconnectivity. Navigability determines the usefulness of the interconnected graph. Navigation, as proposed in this work, is not possible without interconnectivity, but interconnectivity is useless for users without navigation.

Here, we chose to evaluate the feature of navigability by looking at the extent to which navigation is possible and analyzing whether navigation is appropriate and user-friendly. A positive evaluation of both navigation and interconnectedness would mean that the general implementation of the system is a success.

It is worth noting that the usefulness of the navigation will be addressed in the feature of "Coherence of Knowledge Connections", subsection 8.2.4, which seeks to determine if the possible options for navigation are coherent. This feature seeks to interpret if navigation is possible, not if it presents quality options.

The evaluation to understand the extent to which navigation is possible is based on analyzing the nodes and edges. Here, we measure this by looking at how many options are available for navigation. The metric used to analyze this aspect of navigation is again the degree of a node, the number of edges that are arriving or leaving the given node.

The evaluation to analyze whether navigation is appropriate and user-friendly is more subjective, it is based on the chosen vehicle for navigating the text collection, which is composed of a combination of the two connection types and the note-taking tool that enables navigation, which is Obsidian. The idea is to qualitatively determine if the features that Obsidian presents are user-friendly, intuitive, and appropriate for navigation.

A set of questions was proposed in order to validate the feature of Navigability of the Text Collection. The questions focus on the Text Nodes and look to understand the extent to which the text nodes are connected to each other. The questions seek to understand the average scenario as well as the worst-case scenario, to make sure the entire text collection is interconnected, and not only a selected section.

Question 2.1: What is the **average** and the **smallest** Degree for Text and Concept Nodes?

This question is not specifically directed at text nodes but focuses on the possibilities for navigation for both the text and concept node types. The main interest here is to identify the general possibility of navigation for both these types.

This question seeks to determine a navigability score in some way. The higher the degree, the more navigation options there are, which is not necessarily good or bad, as this depends on the use case, but the general idea is to understand if there are **enough** alternatives for navigation.

Question 2.2: Are the functionalities available for Navigation intuitive and user-friendly?

This is a subjective question, it is meant to promote discussion on the available features and functionalities for navigation. Which seeks to determine if the basic features are simple and intuitive to use and understand.

Question 2.3: Are the functionalities available appropriate for navigation?

This question is also subjective, this one seeks to understand if the available features are able to provide an appropriate navigation experience for the user through the Text collection. This question is more directed at all the possible features as a means to understand what are the capabilities of the system, more than how easy it is to use it.

8.2.3 Accurate Graph Representation of Text Collection

The feature represented by the Accurate Graph Representation of the Text Collection has the objective of validating the final structure of the Interconnected Text Collection by comparing it to the proposed graph structure in the mathematical problem statement in subsection 1.1.2.

The idea here is to analyze the output of the system by understanding if the proposed structure, is present and accurate, represented here by the node and edge types. Obsidian has a graph view functionality, which can be enhanced by using an extension called Juggl, as explained in section 7.5, which expands on the graph view functionalities and adds information about the node

and edge types.

The proposed evaluation will seek to understand if the structure of the graph presented in the graph view functionality corresponds to the graph structure designed in the problem definition. This is represented by a single question that seeks to validate the accuracy of the graph representation.

Question 3.1: Are all of the node and edge types in the mathematical problem statement represented in the graph view of Obsidian?

This is a very simple question, simply stated to check if all the proposed node and edge types proposed in the problem statement are actually present in the actual system.

8.2.4 Coherence of Knowledge Connections

The final feature selected to evaluate the implemented system for this dissertation is the Coherence of the Knowledge Connections. This is not explicitly present in the problem statement, but it was selected for the evaluation for being an important aspect to determine if the knowledge connections generated are appropriate and serve their intended purpose.

We define Coherence of Knowledge Connections as being mainly **the coherence, or similarity, between the two categories of connections proposed**, Semantic Relatedness connections, and Concept-based connections. There is an important prerequisite to this definition; the connections within each category must be coherent individually.

This means that the feature of the Coherence of Knowledge Connections is divided into two parts. Firstly to study the coherence, or relevance, of the two connection types individually. Second, to mathematically calculate the coherence between the two types, in order to compare the two types of connections.

The proposed evaluation to analyze the two connection types individually is based on a subjective assessment of the proposed connections for a given text node, looking to understand if the possible options for navigation are broadly related to the original text node. Here, it was not possible to create an elaborate metric for the coherence of all connections for both categories. Instead, we chose to analyze individual examples subjectively to provide a general overview of the satisfiability of the connections belonging to each category.

The chosen evaluation design for defining the overall coherence of the system is composed of analyzing the coherence between the two types of

connections. This was done using the matrices of relatedness between texts, obtained for each of the two connection types. The Relatedness Matrix for the Text Semantic Relatedness is explained in section 6.4, while the Relatedness Matrix for the Concept-based Connections is explained in subsection 5.4.5.

The two relatedness matrices are compared by applying the Mantel Test (Mantel, 1967) as a metric for the coherence or similarity between them. The Mantel Test is one of the most popular statistical tests, it calculates the correlation between two matrices and returns a measure of the correlation ranging from -1 to 1.

The Mantel Test was chosen because, if the test presents a positive correlation between the two relatedness matrices, then the two relatedness metrics may be defined as being coherent with one another. This occurs because the Mantel Test evaluates the similarity between the matrices. If the connections proposed by the Semantic Relatedness metric are similar to the connections proposed by the Concept-based metric, then, by definition, they are coherent.

We further perform a simple grid search by varying one parameter in the calculation of each of the relatedness matrices; the confidence threshold for concepts extraction in the Concept-based connections matrix.

The set of questions presented to analyze the coherence of Knowledge Connections is presented below:

Question 4.1: Are the available options for semantic text-relatedness navigation for a given text node related to its text?

This question is aimed at promoting a qualitative evaluation of how relevant are the recommended texts following the semantic text-relatedness metric. It is meant to broadly assess how the suggestions for navigation for a given text actually relate to the current text. This question represents the navigation path of direct connections between text nodes.

Question 4.2: Are the text nodes that share concepts actually related to one another?

This question is directed at the navigation path of using concepts as a bridge between texts. The evaluation is a broad subjective analysis of how relevant the possible options for navigation are compared to the context of the text, and the concept used for navigation.

Question 4.3: What is the coherence between the two navigation methods presented? Represented by the coherence between the relatedness matrices proposed for each method.

The last question is the most important one. This question presents an evaluation of the coherence of the entire system, by comparing the two different navigation paths proposed in terms of how coherent they actually are.

This question is the placeholder for the grid search performed to determine what confidence threshold provides the best coherence between the two types of connections.

8.3 Results

This section presents the results of the proposed Evaluation, starting with an analysis for each of the questions proposed in section 8.2, in preparation for further discussions regarding the potential use cases of the proposed methodology, in a more open-ended approach.

Here we present answers and discussions for each of the outlined questions in the previous section. The objective is to evaluate if the fundamental features proposed for the system are present in the actual functioning system, more than to evaluate the use cases, the degree to which features are useful, or more complex analysis.

It is worth repeating that for any analysis based on graph features, Author nodes and corresponding edges were not used, as they would provide for almost immediate navigation between text nodes, and would mask the results.

8.3.1 Interconnectivity of the Text Collection

Question 1.1: What is the **average** shortest path and the **longest** shortest path between text nodes?

These shortest paths include the two paths for navigation proposed in the methodology, navigation through concept pages and direct navigation between texts, with 4 total text recommendations for each text.

The average and maximum shortest paths between text nodes indicate that the text collection is indeed interconnected, since it is possible to navigate between any two text nodes in less than 7 hyperlinks, for both test sets.

Table 8.2: Average and Longest Shortest Path between Text Nodes

	Shortest Path	
	Average	Longest
Small Test-Set	2.628	5
Medium Test-Set	2.810	6

Question 1.2: Are the pairs of nodes that share the longest shortest path actually unrelated to one another?

For the Small test set, the nodes that shared the longest path between, and their titles, are:

Text 1: Courage Is Calling Note 4

> Courage is risk. It is sacrifice ... ... commitment ... perseverance ... truth ... determination. When you do the thing others cannot or will not do. When you do the thing that people think you shouldn't or can't do. Otherwise, it's not courage. You have to be braving something or someone.

Text 2: Feel the Fear and Do It Anyway Note 30

> Lighten up. We live in a world where most people take themselves and their decisions very seriously. I have news for you. Nothing is that important. Honestly! If as a result of a decision you make, you lose some money, no problem—you learn to deal with losing money.

The two passages presented above share a semantic relatedness score of 0.092, which is considered to be very low. The meaning of both passages is not related in any specific way, and may even be considered to be somewhat diverging, so it is safe to say that the "most distant" text nodes for Test set 1 are actually not related.

For the Medium test set, the nodes that shared the longest path between, and their titles, are:

Text 1: How to Take Smart Notes Note 17

> The first type of links are those on notes that are giving you the overview of a topic. These are notes directly referred to from the index and usually used as an entry point into a topic that has already developed to such a degree that an overview is needed or at least becomes helpful. On a note like this, you can collect links to other relevant notes to this topic or question, preferably with a short indication of what to find on these notes

Text 2: The Almanack of Naval Ravikant Note 27

> Then, you have to figure out how to scale it because if you only build one, that's not enough. You've got to build thousands, or hundreds of thousands, or millions, or billions of them so everybody can have one.

The two passages presented above share a semantic relatedness score of 0.052, which is even lower than the previous pair. The meaning of both passages is also not related in any way, with the second passage being difficult to interpret, seemingly missing some context to be fully understood. This makes it harder for it to be related to others.

After studying two pairs of "most distant texts", the impression is that they are actually unrelated to one another, and should be distant from one another.

Question 1.3: What is the **average** and the **smallest** Degree for all of the text nodes?

Table 8.3: Average and Smallest Degree for Text Nodes

	Degree	
	Average	Smallest
Small Test-Set	9.846	2
Medium Test-Set	13.302	2

For text nodes in the Small test set, the average degree was 9.846, with 6.115 of these edges being to and from other text nodes, and 3.731 of the edges going to concept nodes.

For text nodes in the Medium test set, the average degree was 13.302, with 6.505 of these edges being to and from other text nodes, and 6.797 of the edges going to concept nodes.

This means that on average there are almost 10 options for navigation from each text node. This is a high enough number to ensure that the dataset is considered to be interconnected with enough alternatives for navigation.

It is interesting to comment that, regarding the most isolated text nodes, both texts do not have any concepts in them, which contributes to the nodes becoming isolated. This happens for two reasons, first, the obvious fact that there are no concepts to navigate to, and second, the concept-based recommendations are not applied in this specific case, since there are no concepts to compare to.

Question 1.4: Are the (3) nodes with the lowest Degree dissonant with the other texts in the text collection?

The most isolated nodes for both of the test sets are presented below.
Small test set:

> 'They are like the rest of us: all trying to do the best they can and all uncertain about whether they're good enough. It never varies.'

Medium test set:

> 'Then, you have to figure out how to scale it because if you only build one, that's not enough. You've got to build thousands, or hundreds of thousands, or millions, or billions of them so everybody can have one.'

When looking at both of these texts, it is possible to notice that they are missing a context. It is not easy to understand exactly what they are referring to, which makes it hard to relate them to other texts. This is also an obstacle to applying semantic relatedness comparison to other texts, exemplified by the fact that no other text in the collection is highly related to these texts.

In a general way, both of the texts are indeed dissonant with other texts in the collection, both for not having any concept in them and for not being easy to relate to other texts.

8.3.2 Navigability of the Text Collection

Question 2.1: What is the **average** and the **smallest** Degree for Text and Concept Nodes?

Table 8.4: Average and Smallest Degree for Text and Concept Nodes

	Degree			
	Text Nodes		Concept Nodes	
	Average	Smallest	Average	Smallest
Small Test-Set	9.846	2	11.754	2
Medium Test-Set	13.302	2	17.322	3

The results for the average and smallest edge densities for each of the two node types are presented in Table 8.4, with the clear observation that concept nodes have a higher degree than text nodes.

Considering a subjective navigability score for the interconnected text collection, the results for this question suggest that **there are enough alternatives for navigation**. Especially when assuming that the nodes which have a low degree are indeed NOT highly related to the text collection as a whole, as the answer to Question 1.4 suggests.

Question 2.2: Are the functionalities available for Navigation intuitive and user-friendly?

This question builds upon the available features for navigation presented in subsection 7.5.4 about Navigation options in Obsidian.

The first comment is that Obsidian follows Wikipedia-like navigation, with the addition of other features, the most relevant one being the backlinks. With regard to the basic features for navigation, presented in Figure 7.7, it is definitely the case that they are intuitive and user-friendly, because they follow basic computer skills and semiotic signs of search, file explorer, and hyperlinks.

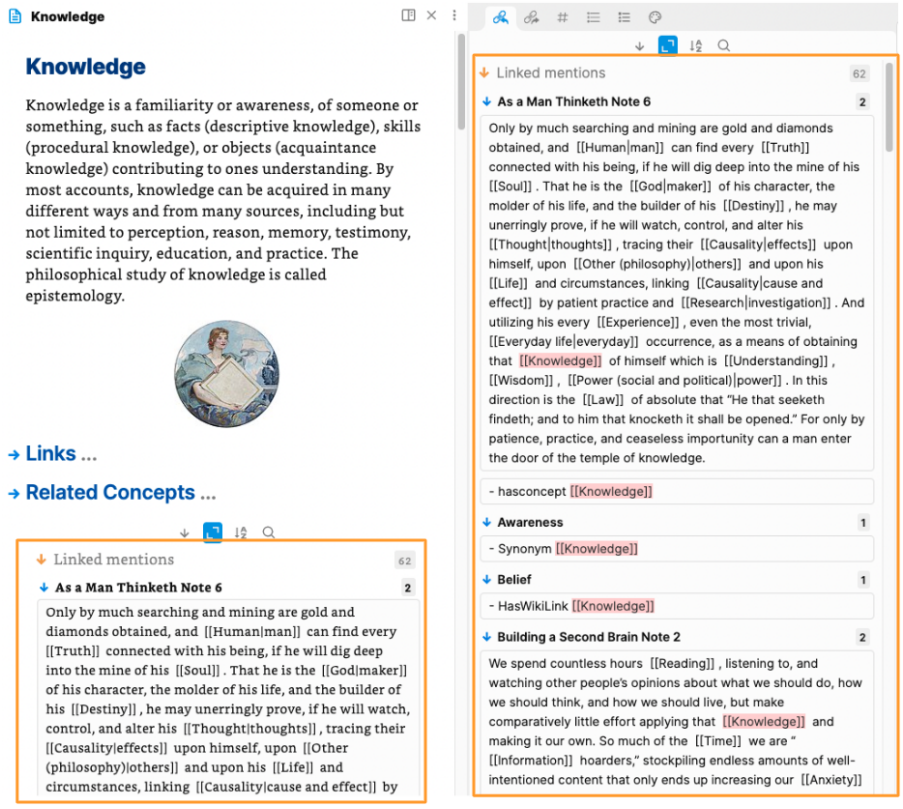

Figure 8.1: The different positions for presenting the Incoming Links in Obsidian

Even though backlinks could be a new experience for most users, the functionalities for navigating using them can be considered to be user-friendly. First, the backlinks are presented in a separate part of the screen, creating a

notion of being part of a different function, and second, backlinks are represented visually in a slightly different format, with the specific incoming link to the current page being highlighted in a different color, again, demonstrating a different functionality.

As mentioned earlier, there are two different positions for the incoming links to a given node. Both these positions for the backlinks are outlined in orange in Figure 8.1.

Question 2.3: Are the functionalities available appropriate for navigation?

This question is more oriented towards the different available features for navigation, which again, is highly related to the available features for navigation, presented in subsection 7.5.4.

The different features presented in the section on Navigation options cover an interesting range of options, where users are able to perform a series of different actions to navigate the text collection.

The most relevant functionalities are the basic navigation using the search and file explorer, using hyperlinks in both directions, the previewing of a neighboring node's content using Transclusion, the possibility of opening multiple pages at once, and the local graph view to spatially represent the connections with other nodes.

8.3.3 Accurate Graph Representation of Text Collection

Question 3.1: Are all of the node and edge types in the mathematical problem statement represented in the graph view of Obsidian?

The best way to answer this question is to look individually at the local graph view for each of the three node types. Since the global graph view is usually very densely populated and is not as adequate to understand its contents. Additionally, the Juggle extension was used to better identify different node types in the local graph view.

The very fact that three different node types are being analyzed individually is enough to answer that the three node types proposed in the problem statement are actually present in the graph representation of the text collection. This question then turns into identifying the 6 edge types within the different graph views for each node type.

The edge types that need to be present in the graph representation of Obsidian are defined by Equation 8-1. The three node types will be analyzed looking to identify these edge types.

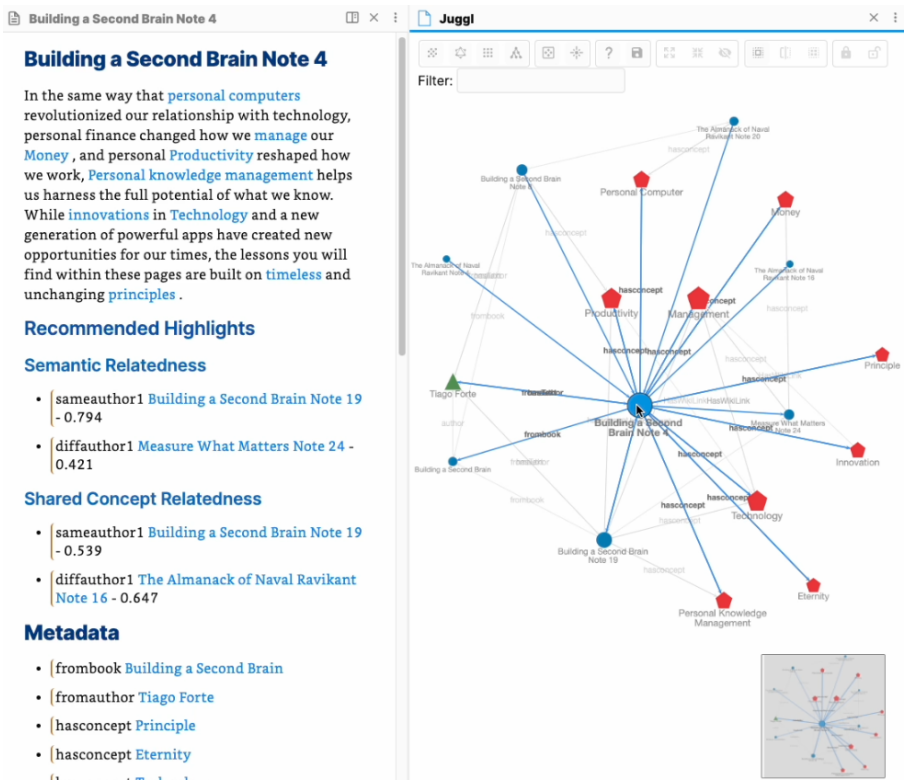

Figure 8.2: The local graph view for an example Text Node

$$E \subseteq (\overleftrightarrow{TT}, \overleftrightarrow{TC}, \overleftrightarrow{CC}, \overleftrightarrow{AA}, \overleftrightarrow{AC}, \overleftrightarrow{AT}) \tag{8-1}$$

1. $\overleftrightarrow{TT}$: Text $\leftrightarrow$ Text edges
2. $\overleftrightarrow{TC}$: Text $\leftrightarrow$ Concept edges
3. $\overleftrightarrow{CC}$: Concept $\leftrightarrow$ Concept edges
4. $\overleftrightarrow{AA}$: Author $\leftrightarrow$ Author edges
5. $\overleftrightarrow{AC}$: Author $\leftrightarrow$ Concept edges
6. $\overleftrightarrow{AT}$: Author $\leftrightarrow$ Text edges

When looking at the example local graph view for the Text node of the example text, in Figure 8.2, it is possible to identify the three edge types that involve text nodes: Text $\leftrightarrow$ Text edges, $\overleftrightarrow{TT}$, between blue circles, Text $\leftrightarrow$ Concept edges, $\overleftrightarrow{TC}$, between blue circles and red pentagons and Author $\leftrightarrow$ Text edges, $\overleftrightarrow{AT}$, between the green triangle and blue circles.

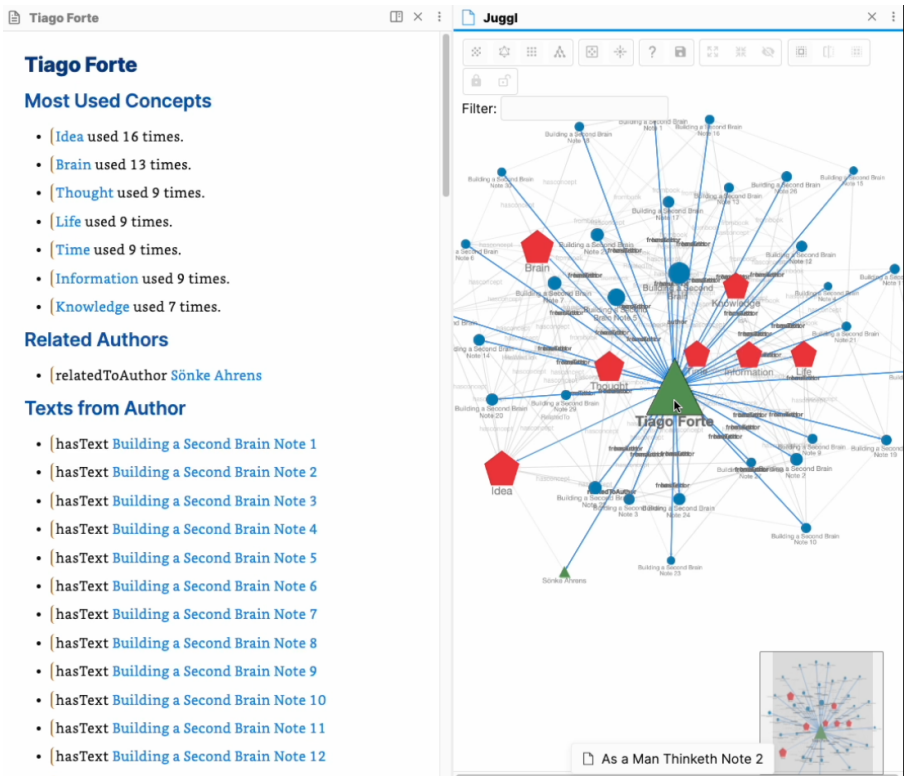

Figure 8.3: The local graph view for an example Author Node

When looking at the example local graph view for the Author node of Tiago Forte, in Figure 8.3, it is also possible to identify the three edge types that involve author nodes: Author ↔ Author edges, $\overleftrightarrow{AA}$, between the green triangles, Author ↔ Concept edges, $\overleftrightarrow{AC}$, between the big green triangle and red pentagons and again Author ↔ Text edges, $\overleftrightarrow{AT}$, between the big green triangle and blue circles.

Finally, when looking at the example local graph view for the Concept node of "Wisdom", in Figure 8.4, it is possible to identify two edge types that involve concept nodes, the first of them being the only remaining edge type to complete the 6 types detailed in Equation 8-1. We can observe the Concept ↔ Concept edges, $\overleftrightarrow{CC}$, between the red pentagons, as well as the Text ↔ Concept edges, $\overleftrightarrow{TC}$, between the blue circles and red pentagons.

After looking individually at the local graph views for each node type, it is possible to affirm that all the elements proposed in the problem statement are included in the actual system.

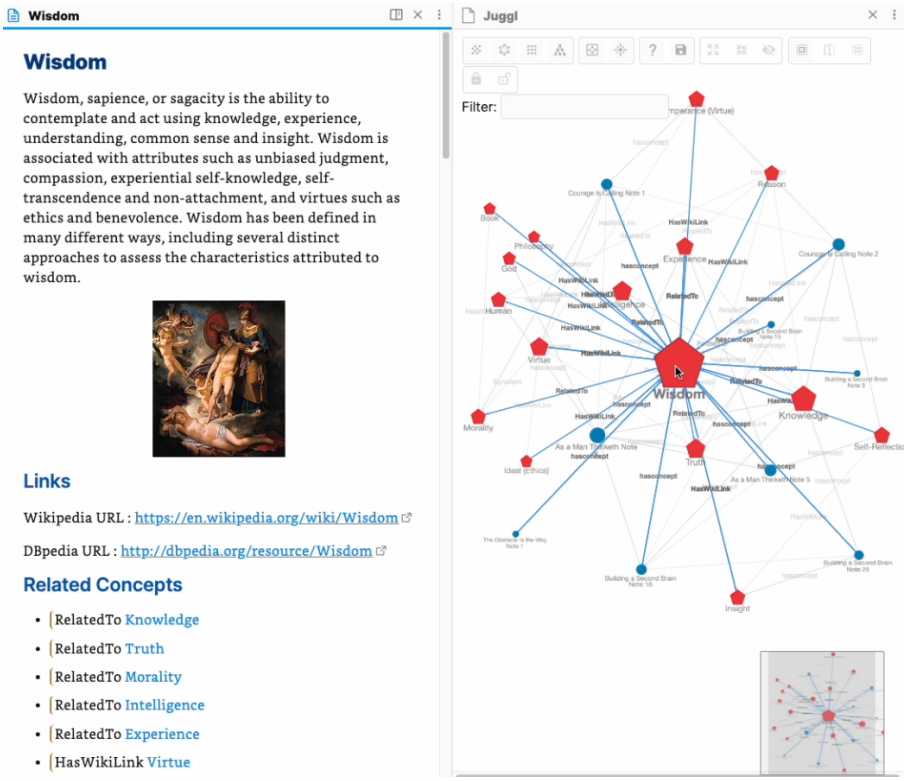

Figure 8.4: The local graph view for an example Concept Node

8.3.4 Coherence of Knowledge Connections

Question 4.1: Are the available options for semantic text-relatedness navigation for a given text node related to its text?

This is a subjective and broad question, its objective is not to perform an endless and tiresome evaluation of all the possible connections between text nodes, but rather to discuss the general functioning of the semantic text relatedness connections.

As a means of providing examples for this discussion, two comparisons are available, the first example is illustrated in Table 7.9, showing the most related texts when compared to the example text, used throughout the dissertation.

The second example is presented below and is a simple comparison between two specific texts that are present in both of the test sets and share a relatedness score of 0.669, which is considered to be very high.

Text 1: Courage Is Calling Note 7

> The Stoics, the Christians — they didn't fault anyone for having an emotional reaction. They only cared what you did after the shine of that feeling wore off. "Be scared. You can't help that," William Faulkner

> put it. “But don’t be afraid.” It’s an essential distinction. A scare is a temporary rush of a Feeling. That can be forgiven. Fear is a state of being, and to allow it to rule is a disgrace.

Text 2: Feel the Fear and Do It Anyway Note 8

> Obviously, the real issue has Nothing to do with the fear itself, but, rather, how we hold the fear. For some, the fear is totally irrelevant. For others, it creates a state of paralysis. The former hold their Fear from a position of power (Choice, Energy, and action), and the latter hold it from a position of Pain (helplessness, depression, and Paralysis).

Both examples present supporting evidence to the theory that connections available through text relatedness are actually related to one another.

The first example, in Table 7.9, captures perfectly the similarity between both texts when talking about the importance of Personal Knowledge Management as a tool for succeeding in the world today, while also presenting a high score of relatedness between the example text and two other texts that mention Knowledge, and the idea of building an external repository of knowledge.

With regard to the second example, presented above, the high level of relatedness captured between texts can be considered to make sense. Both texts talk about Fear, but the high degree of relatedness is due to more than that. Both texts mention two sides of different situations involving fear, both texts use the word “*but*”. They also deal with a “*state*” of being or “*position*”, both texts use different words to convey this idea, but the relatedness is captured nonetheless.

These two examples present a high degree of relatedness, but they are not anomalies within the two test sets analyzed. In a general way, suggestions based on semantic relatedness are useful and provide connections to texts that are indeed related to the current text node.

Question 4.2: Are the text nodes that share concepts actually related to one another?

This second question is not as easy to bring multiple examples, because there are two variables in play, the initial text and the chosen concept to be navigated to.

The most important of these variables, though is the chosen concept. The concept node will show the same information, regardless of which text node the user was previously at. In order to illustrate possible navigation options, we present a Concept and three example text nodes that are available for navigation from that concept.

As exemplified by the three texts below, within all the possible text nodes available for navigation **some options usually share common themes** with the original text. The texts may not always be related, or talk about the exact same subjects, but generally speaking, the connections using common concepts do provide related texts in some way.

The chosen concept to illustrate this analysis is "*Risk*", and the three passages presented are from three different books:

Text 1: Courage Is Calling Note 14

> This moment is a test. They're called "trying times" for a Reason. It's good that it's happening now, instead of later—because later, you'll be better for having gone through it today. Got it? You think it'd be better if things were easy. You wish you didn't have to take this **Risk**. If only the leap didn't look so damn dangerous. That's just the Fear talking. It's good that it's hard. It deters the cowards and it intrigues the courageous.

Text 2: Feel the Fear and Do It Anyway Note 4

> In all my Life I have never heard a mother call out to her child as he or she goes off to school, "Take a lot of **risks** today, darling." She is more likely to convey to her child, "Be careful, darling." This "Be careful" carries with it a double message: "The world is really dangerous out there" ... and ... "you won't be able to handle it." What Mom is really saying, of course, is, "If something happens to you, I won't be able to handle it." You see, she is only passing on her lack of trust in her ability to handle what comes her way.

Text 3: Measure What Matters Note 35

> To succeed, a stretch goal cannot seem like a long march to nowhere. Nor can it be imposed from on high without regard to realities on the ground. Stretch your team too fast and too far, and it may snap. In pursuing high-effort, **high-risk** goals, employee commitment is essential. Leaders must convey two things: the importance of the outcome, and the Belief that it's attainable.

The three passages mention Risk in different contexts, which means that the essence of each of these passages is not intrinsically related, they are not dealing explicitly with the same subject.

Even though they are not portraying the exact same message, these three passages are obviously related in some way, the first two passages are clearly stating that risk is seen by society as being something bad, while each text provides a different approach to dealing with this information. The last passage is probably the most unrelated among the three, but it still presents an idea that is related to the other two, by mentioning the concept of high-risk goals and also how to deal with this situation in a corporate environment.

An interesting exercise is to imagine from which text the user would have arrived from, this is when the other variable of the initial text comes in, which is very influential for the situation, and changes the intention of the user when facing the available navigation options for a certain concept node.

Question 4.3: What is the coherence between the two navigation methods presented? Represented by the coherence between the relatedness matrices proposed for each method.

Table 8.5: Grid Search of Coherence scores, while varying Confidence threshold for Entity Extraction

	Coherence Score	
Confidence	Small Test Set	Medium Test Set
0.55	0.679	0.519
0.60	0.665	0.499
0.65	0.642	0.508
0.70	0.681	0.481

The Coherence of Knowledge Connections is defined as being the coherence, or similarity, between the two categories of connections, namely the Semantic Relatedness connections, and Concept-based connections.

By this metric, all of the results in Table 8.5 present a positive correlation. According to Swinscow and Campbell (2002), correlations from 0.40 to 0.59 are considered to be moderate, and from 0.6 to 0.79, as strong. This means that the relatedness metrics are considered to be **strongly correlated for the small test set, and moderately correlated for the medium test set**.

By applying the significance test with $P < 0.001$, the correlation coefficients are considered to be highly statistically significant for all the obtained results. This signals that the two different approaches for generating connections are indeed coherent with one another.

The last two questions looked at the individual coherence of connections for the different ways of connecting text nodes. We judge the results as supporting the fact that both connection types are considered relevant and present coherent options for navigation.

This final question ties everything together, by looking at the general definition of coherence, and evaluating the coherence of connections within the context of the entire methodology proposed in this dissertation.

The positive correlation between the different means of navigation sug-

gests that the methodology presented throughout this dissertation is valid. The generated connections are individually coherent in terms of texts that are suggested for each navigation path, and also quantitatively coherent with one another.

With regard to the grid search for the optimum confidence, by considering both test sets, it is possible to say that lower confidence thresholds when extracting concepts lead to a higher coherence between the relatedness metrics.

This makes some intuitive sense because this means that there are more concepts to work with, which apparently increases the performance of the concept-based relatedness, as there is more information available.

Regardless of why the coherence between the metrics improves, the results are somewhat similar for both test sets, the confidence threshold of 0.55 presents the best overall coherence.

Even though the best coherence between the two metrics was obtained at 0.55 confidence, the chosen default parameter for confidence was 0.60, mainly because of the computational cost of running the system. Whenever the size of text collections grows, more concepts are extracted, which means that lower confidence could lead to very long runtimes.

8.4 Discussion

This section will present additional discussions regarding the Evaluation Results, as well as reflect on the proposed methodology overall, with special attention to the potential use cases that this technology allows for.

8.4.1 Results Discussion

The obtained results will be discussed by looking back at the fundamental features outlined to evaluate the functionality of the implemented system, looking to take a general look at the features, and judge if the system was considered to be a successful implementation of the proposed methodology.

The fundamental features outlined in the Evaluation Planning, subsection 8.1.1 are those of:

1. Inter-connectivity of the Text Collection
2. Navigability of the Text Collection
3. Accurate Graph Representation of Text Collection
4. Coherence of Knowledge Connections

a. Shared Concepts Connections

b. Text Semantic Relatedness Connections

c. Coherence between both types of connections

These four features were analyzed in detail, through quantitative and qualitative metrics, and the clear takeaway is that the system may be considered to have been successfully implemented. All of the features are present and functional, providing evidence that the implemented system is capable of executing its intended functions.

The proposed methodology was successfully represented in the functional version of the system, which is 1. interconnected, as explored by the metrics of shortest path and degree, 2. there are enough alternatives as well as features for navigation, 3. the final implementation successfully corresponds to the proposed graph representation, and 4. the Connections can be considered to be coherent, individually and collectively.

What follows from these features being present is a discussion as to what extent do they fulfill the intended objective of increasing human capacity to acquire knowledge faster and more profoundly, outlined in the Goals for the Methodology, section 7.1.

This is an open-ended question, and in the scope of this dissertation, we did not set out to perform the necessary studies to validate if the proposed system is able to accomplish such an ambitious objective.

What the scope does allow for is a discussion regarding the proposed methodology as means to reach the intended research objectives.

8.4.2 Proposed Methodology Overview

This subsection looks to present some discussions regarding the Research Questions presented in section 1.2, looking to carry out additional reflections on the final implementation through the lens of the research objectives.

The three proposed Research Questions (RQs) are presented below.

1. Can the combination of NLP with Networked note-taking tools improve the Knowledge Management functions of Recall, Elaboration, and New Insight?

2. How to propose connections between any two given texts present in a text dataset?

3. Are Concept Nodes a useful mechanism for navigating a text collection?

The first comment here is regarding the composition of the three Research Questions, the first RQ is directed at the **inspirations and reasons** for proposing the methodology and building the corresponding system. The second question is a more practical question, that seeks to understand **how to technically and conceptually connect different texts**. Finally, the third and last question is directed at the **utility of introducing the Concept Nodes** as means of connecting and enriching a given text collection.

Each research question plays a specific part in influencing the proposed methodology, the second RQ was most relevant for the theory backing up the proposed methodology and also relevant for the practical evaluation, whilst the first and third RQs are more influential with regard to design decisions and how the system is supposed to work, which is more plausible of being openly discussed.

When analyzing the proposed methodology through the lens of the three intended functions of Knowledge Visualization, Recall, Elaboration, and New Insight, it is interesting to mention some characteristics of the envisioned system.

These three functions share a common duality, which is the presence of **Divergent and Convergent aspects**. Each function has an element of divergence, of spreading out wide and exploring new connections, while also presenting the convergence element, of collapsing to one tangible connection or event.

The function of Recall has the divergence of searching for a specific item, say an idea or a concept, across a wide range of options, while eventually converging to the actual retrieval of the desired item(s). Elaboration follows a similar path, of being presented with different options for elaboration and eventually converging to one possibility to elaborate on a given topic, usually one at a time. New Insight, in turn, is the most divergent of all three features, seeking mainly to be exposed to *new* ideas and possibilities that aren't previously known, but it surely has the convergence which happens in order to form an *Insight*, which is when two or more ideas connect and form a new piece of knowledge.

The proposed system was inspired by these three functions, as it was designed to generate connections according to these two ways of operating, seeking divergence and convergence, in a balanced way, each of the two different options for navigating the text collection has different components of divergence and convergence.

The main utility for the Concepts Connections is directed at *Divergent thinking*, by proposing a set of related ideas that use common concepts, with

the important characteristic of **not discriminating** between ideas according to the specific topic covered in the texts or what exactly is being mentioned regarding the specific concept. The convergent aspect of the Concept Nodes Connections comes from the fact that concepts are able to connect different texts through a common theme, which is a shared concept between them.

On the other hand, the Text Relatedness connections seek to initially promote *convergent thinking*, by generating connections between texts that are indeed related to each other, while also presenting elements of divergent thinking, with the possibility of contrasting different ideas from different authors.

These two modes of operation are essential aspects of the proposed system as they play a part in both connection types and were designed into the system to simultaneously represent the three functions of Recall, Elaboration, and New Insight.

Another analysis of the system along these lines is regarding the tradeoff between divergence and convergence, between broad exploration and precise connections.

Whenever possible, the chosen priority for the system is **divergence and exploration**, instead of valuing convergence and precision. The idea is to provide the structure for semi-organized divergent thinking while providing the tools for the user to use this system for convergence, according to their own terms. Since the system is supposed to be used by human users, it is expected that the user would use the system with a specific outcome in mind, in the format of a tangible project or even a specific reflection or contemplation of ideas.

This also means that the coherence metric between the two types of relatedness metrics would not be considered essential for the correct functioning of the proposed system since it has a strong element of divergence, which doesn't depend on the two connections leading to similar places. The coherence metric is an indication that the connections proposed are appropriate, indicating an efficient and organized divergence between ideas.

To end this subsection, it is important to note that connections generated with the proposed methodology are not mandatory, they are by no means final or represent a foundational truth, instead, they are mainly suggestions and possibilities. The idea of the interconnected text collection is to promote a means for exploration using divergence, and a means to connect texts and promote convergence, not necessarily to connect perfectly related texts.

8.4.3 Use Cases for the proposed Technology

This final subsection will explore a couple of the potential use cases for the proposed methodology, outlining specific tasks that may be carried out using the proposed system as well as more general use cases of what can be done with this technology.

Personal Knowledge Management

The first and most important use case for this technology is within the field of Personal Knowledge Management, following the main inspiration for this dissertation. The idea is to apply the system to generate connections for a text collection composed of a person's external repository of knowledge, with notes belonging to a single person, collected across disciplines and throughout the years.

The main idea behind a Personal Knowledge Management System is to store past knowledge in a safe place so that it may be later recycled and reused. Automatic connections between ideas may be a catalyst for retrieving and "*applying*" the knowledge in some useful way. There are a series of more specific use cases into which this initial general use case may be divided into.

The first and most obvious use case would be **creation**. Creation of a specific deliverable for work, creation as in a research project, and, of course, the epitome of creation: **writing**. Writing a short article, an academic paper, a non-fiction book, or even a novel, any type of writing could benefit from an interconnected external collection of knowledge, in such a way that multiple ideas can be accessed and recycled into a new piece of content, a new convergence of these ideas.

Another use case within the overarching use case of Personal Knowledge Management is helping with the retrieval of related knowledge to any specific text, general topic, or even search queries. The text, topic, or query could be inserted into the system, which would extract the concepts and obtain any range of related tests. This could be used for several different reasons, for example resurfacing pieces of knowledge to build an argument with supporting evidence, or for sharing knowledge with a client, a coworker, or an employee.

Another potential use case for an interconnected text collection would be learning. Learning by comparing and relating a new piece of information to previous pieces of knowledge is one of the centerpieces of the Zettelkasten method for note-taking, explained in How to Take Smart Notes, (Ahrens, 2017). This is due to the fact that when a person holds two different ideas in their mind, they are led to reflect more carefully on what is being portrayed, consolidating that which was learned. This goes together with the idea that

learning a new piece of information is more effective whenever there is a previous structure of knowledge available to attach to the new piece of information. The analogy for this would be a tree trunk, to which new branched and leaves are attached, by having a place to hold onto, the new piece of knowledge has more chances of being truly learned and remembered.

By automatically showing the user what are previous pieces of knowledge that *may* be related to the new information being learned, the system of generating connections is capable of aiding the process of comparing a new piece of information to previous knowledge by making it easier, faster and more powerful.

It is worth noting that contrasting a new piece of information with the available knowledge is one of the central functionalities when dealing with Knowledge Management. This is useful for adding connections between notes and creating a network of ideas that may be explored and used for new insight.

A more specific use case for this functionality would be to add the connections using semantic relatedness and concept nodes to other ways of organizing knowledge, such as the Discourse Graph, (Chan, 2020). An example use case would be automatically generated suggestions of related texts that could help human users to identify and add structured connections between nodes, for example by identifying a text that supports a claim or answers a research question.

An interesting observation is that these three more granular use cases may be loosely mapped to the three functions of Knowledge Visualization, respectively, Elaboration, Recall, and New Insight. Elaboration when creating something, Recall when retrieving knowledge, and New Insight when learning and connecting ideas.

Knowledge from Different Topics

Another specific use case within the personal context of managing knowledge would be **organizing knowledge from different topics**. The main specific example of this would be High School and University students organizing their notes across different disciplines using automatic connections between their personal notes, and even within segments of digital Textbooks.

This could have a positive impact in two ways. First, promoting connections across different parts of the syllabus, which works for ideas about the same discipline and from different disciplines. Second, showing relations **between concepts**, which can help with deepening of tacit, commonsense, knowledge, by explicitly showing what concepts are related to other concepts.

These two aspects combine to provide students with a **big-picture view** of what they are learning, something which is not as easily available, due to

traditional education being very keen on separating the content belonging to different subjects instead of promoting an interdisciplinary exchange of information.

A generalization of organizing knowledge from different topics, as applied to any non-student, would be to use the system to read and study multiple books at once, while easily navigating between the ideas presented in these books. The idea here would be to read any combination of books in whatever desired order, linearly or intercalating books while highlighting the passages that present ideas the user wishes to ponder about or study more deeply. The next step would be retrieving the highlighted passages, which may be done automatically using specific tools, or manually. The text collection of passages may be inputted to the system, which will effortlessly generate connections to transform the text collection into an interconnected version with all the inputted texts and connections between them.

The outputted interconnected text collection may then be navigated, and **most importantly, edited and incremented**, because the system is hosted inside a note-taking app, meaning the user may use the software to create new notes and elaborate on the initially collected ideas.

Collective Knowledge Management

Another important use case for this technology is **connecting ideas from different people** inside a group, using the system to connect and compare texts from more than one person. This is very aligned with what Doug Engelbart defines as the two most important aspects of the Collective IQ level. First, the process, i.e. How well a group develops, integrates and applies its knowledge, and second, the assets produced by that process, i.e. How effective the group's shared repository of knowledge is, and how easily information can be synthesized, stored, retrieved, and updated.

The proposed system is an interesting approach to facilitate the process of integrating different people's knowledge and also enhancing a potential knowledge asset by automatically generating connections for it.

9
Conclusion

This chapter concludes the dissertation, by providing closing remarks on the proposed methodology and results, outlining this work's main contributions, and suggesting future paths for exploration.

Recent advancements in the fields of Natural Language Processing and Personal Knowledge Management present a powerful opportunity to combine them. Specifically by enhancing modern note-taking tools with Artificial-Intelligence-based features, taking advantage of existing functionalities as a starting point.

In this dissertation, we have proposed and successfully implemented a methodology to automatically generate connections between texts. The proposed methodology employs a combination of NLP tools and note-taking apps to transform a given text collection into an interconnected and navigable version of the same texts.

Interconnectedness is obtained by transforming a text collection into a graph. Thiw way, texts become Text nodes, Concept nodes are introduced, and connections can be easily added using edges.

Navigation, in turn, is added to the text collection using a modern note-taking tool called Obsidian. The tool combines the hierarchical organization of files and folders with the networked organization of bidirectional hyperlinks.

The results and evaluation suggest that the proposed system is indeed interconnected and navigable, as well as an adequate representation of the system proposed in the introduction, chapter 1. Lastly, the two different paths for generating connections are coherent within themselves, which suggests that the connections generated by the system are reliable.

9.1
Main Contributions

The main contributions and possible impacts on the field of Personal Knowledge Management are described as follows.

Contribution 1: The creation of a system that automatically creates connections between texts. Connections between texts are generated following two different options, Semantic Relatedness between texts and

through Concept Nodes. The connections are considered to be coherent and reliable for the intended functions of Recall, Elaboration, and New Insight. Whenever this tool is available to the public, it shall provide users with powerful capabilities to generate connections and explore relations between ideas.

Contribution 2: A theoretical and practical workflow for introducing NLP capabilities to modern note-taking tools. To the best of our knowledge, this work presents the first academic production that explicitly unites the fields of Note-taking apps together with Natural Language Processing. This dissertation opens up paths for more AI tools to be created in the field of Personal Knowledge Management, by detailing the underlying NLP tasks, and practical implementations to generate this system. The information and thought-process hereby presented may be extremely useful when designing and building note-taking applications that use NLP.

Contribution 3: A novel technique for generating connections between ideas using Concept Nodes. The proposed navigation using concept nodes structures relevant information about concepts, and creates bridges to texts that mention the same concepts. With the added possibility of calculating the text relatedness using the concepts mentioned in the texts. This idea and implementation of creating Concept Nodes as a means of navigating a text collection open up interesting opportunities that can be easily extended to other applications and use cases. Education and knowledge acquisition are considered to be the most promising opportunities for this technique.

These contributions come at a good time for the field of Personal Knowledge Management and supporting note-taking tools, with many researchers being interested in this specific intersection. The possible implications of this work are to lower the entrance barrier for applying Artificial Intelligence and Natural Language Processing to the process of Knowledge Management, which may have potentially powerful implications with regard to enhancing collective intelligence and collective capabilities to overcome the challenges that humanity shall face in the near future.

9.2 Future Work

The immediate Future Work following this dissertation is **evaluating to what extent is the proposed system capable of having a positive impact on learning and operating with knowledge**. This is an immediate follow-up from this work, that seeks to validate the overarching goal of providing a system to enhance the experience of learning faster and with broader connections, which is the main inspiration for building the system.

The initial idea for further evaluation would be to test the system with users and students. Potentially, track performance metrics for groups of users when executing a set of tasks. The same task could be tested in different versions of the text collection, both *with* and *without* the knowledge connections, while still providing a questionnaire to obtain feedback on the user's opinions on the system.

Given a positive evaluation of the system being applied to learning, a natural posterior work for the proposed methodology would be to **apply the methodology for educational and epistemological purposes**.

The application of technology that automatically generates connections for learning purposes would aim at providing better recall of information, as well as a greater ability to generate new insights. This could be a very useful tool for acquiring knowledge in the new age of digital education, in which online courses and even online diplomas are getting more and more popular.

Enhancements to the System

There are several possibilities for future work directed at enhancements of the proposed methodology for creating connections.

A possible starting point would be to create a Knowledge Management System where the notes have two different versions, one of the versions contains the original notes, without the automatic connections, and the other version is the interconnected one, with all the generated connections. This way, the additional nodes and edges are not always "turned ON", it is possible to **activate concept nodes and connecting edges on demand**.

Another similar addition would be to enable users to edit the data, both by adding and removing, edges and nodes from the knowledge collection. These alterations would be available with a User Interface and could be saved for future uses. This would give users the ability to edit and organize the data in whatever way they desire.

A strong addition to the user experience using the system would be to present a ***Big Picture*** **view** of the collection, using clusters of texts and concepts. The clusters could represent topics and groups of concepts, providing a high-level view of what is inside the text collection. Similar to a map of what is contained inside the text collection

Clusters could be created using the concepts mentioned together with the results of a topic modeling on the collection's texts. This would allow for a multidimensional distribution where clusters could be calculated and displayed in a 2D or 3D representation of the clusters of nodes.

Similar to an overview, another option for enhancing the user's participation is to identify and **provide *entry points*** to the knowledge collection,

based on concepts and topics the user is interested in. Entry points would provide a set of starting points for exploration and could be both text and concept nodes. They could be calculated by comparing a given set of concepts with the central concepts in the graph, according to node centrality data.

Another useful addition to the system would be to create a functionality to filter a large text collection using a UI or SPARQL Queries to select specific knowledge automatically for the building of an interconnected collection, where the user would be able to easily generate multiple different versions of interconnected sub-collections of the original large collection.

A final interesting functionality would be to create a universal input encoder, where the system is able to automatically **process different input formats**, including note databases belonging to note-taking apps. This way, the inputs wouldn't need to be in raw text format and the inputs could be in different formats, a very interesting functionality when thinking about creating a bridge between originally isolated ideas in separate folders.

Expansion to Large Online Databases

Another continuation would be scaling this technology to large online databases. This could even include subsets of massive **online collections**, such as the Ar5iv[1]. The Ar5iv is a collection of HTLM web pages for a big significant part of papers published to ArXiv.

Exploring big amounts of data with automatic connections could be a massive step for working with knowledge. Enabling the selection of any set of topics with automatic generation of hyperlink connections between papers and concepts, promoting navigation and exploration between different academic works and areas.

When scaling to a very high volume of texts, it would be valuable to present connections from concepts to papers following filtering and ranking of the proposed connections, instead of showing *all* the texts that use a given concept. This ranking system would demand further reasoning to find a suitable way to create personalized recommendations, while taking into consideration large amounts of data about each paper, including; the concepts mentioned, metadata from DBLP (Ley, 2002), the paper from which the user arrived, and possibly historical data of the user's navigation.

Knowledge Base Completion

Lastly, another Future work for this dissertation is to adapt and apply the hereby proposed methodology for the task of **Knowledge Base Completion**, (Li et al., 2016). Knowledge bases, such as DBpedia, (Lehmann et al., 2015) usually have missing fields and relationships for its resources, it is possible to

[1] ar5iv.labs.arxiv.org/

adapt the methodology in this dissertation to identify missing links based on data extracted from documents and complete missing links in existing knowledge bases.

9.3 Final Remarks

The development of any Artificial Intelligence system that enhances or replaces human effort comes with a responsibility of thinking about its implications. The usage of the technology described in this dissertation could change the way users interact with knowledge, and this brings opportunities and risks. Opportunities such as faster navigation, easy connection between topics, and a graph structure. Risks such as replacing human activities and discouraging independent reasoning.

According to the Extended Mind Thesis, (Clark and Chalmers, 1998), human cognition is made up of the interactions between internal entities and external entities. It is extremely important to understand how the addition of new functionalities and possibilities will shape the *external entities* that humans interact with. As this will play an important part in the way humans behave.

For example, a system that automatically generates connections and allows for easy navigation of a text collection could potentially lead humans into a less active, and more passive role when dealing with knowledge. Users could get used to simply following easily laid-out suggestions and paths, instead of actively engaging with different sources and thinking critically.

When using the system, it is important to promote active roles for humans, in which users actively cocreate connections and are an inherent part of the puzzle. This could avoid users from being too passive, and is a matter that should be studied. Some of the initial contexts for these active roles could be: choosing what texts to include in the collection, active ways of generating and editing connections, and finally, adding personal notes to texts and concepts nodes, then utilizing them as output.

Bibliography

Ahrens, S. (2017). *How to take smart notes: One simple technique to boost writing, learning and thinking*. Sönke Ahrens.

Alavi, M. and Leidner, D. E. (2001). Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS quarterly*, pages 107–136. Publisher: JSTOR.

Anand, R. and Kotov, A. (2015). An empirical comparison of statistical term association graphs with dbpedia and conceptnet for query expansion. In *Proceedings of the 7th forum for information retrieval evaluation*, pages 27–30.

Asghar, N. (2016). Automatic Extraction of Causal Relations from Natural Language Texts: A Comprehensive Survey. *arXiv:1605.07895 [cs]*. arXiv: 1605.07895.

Bach, N. and Badaskar, S. (2007). A review of relation extraction. *Literature review for Language and Statistics II*, 2:1–15.

Becker, M., Korfhage, K., and Frank, A. (2021a). COCO-EX: A tool for linking concepts from texts to ConceptNet. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 119–126.

Becker, M., Korfhage, K., Paul, D., and Frank, A. (2021b). CO-NNECT: A Framework for Revealing Commonsense Knowledge Paths as Explicitations of Implicit Knowledge in Texts. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 21–32, Groningen, The Netherlands (online). Association for Computational Linguistics.

Becker, M., Liang, S., and Frank, A. (2021c). Reconstructing implicit knowledge with language models. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 11–24.

Blanco-Fernández, Y., Gil-Solla, A., Pazos-Arias, J. J., Ramos-Cabrer, M., Daif, A., and López-Nores, M. (2020). Distracting users as per their knowledge: Combining linked open data and word embeddings to enhance history learning. *Expert Systems with Applications*, 143:113051.

Bollacker, K., Cook, R., and Tufts, P. (2007). Freebase: A shared database of structured general human knowledge. In *AAAI*, volume 7, pages 1962–1963.

Bordes, A., Chopra, S., and Weston, J. (2014). Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*.

Brew, A. (2008). Disciplinary and interdisciplinary affiliations of experienced researchers. *Higher Education*, 56(4):423–438.

Budanitsky, A. and Hirst, G. (2006). Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47.

Burkhard, R. A. (2005). Towards a framework and a model for knowledge visualization: Synergies between information and knowledge visualization. In *Knowledge and information visualization*, pages 238–255. Springer.

Bush, V. (1945). As we may think. *The atlantic monthly*, 176(1):101–108.

Buzan, T. and Buzan, B. (2006). *The mind map book*. Pearson Education.

Canales, R. F. and Murillo, E. C. (2017). Evaluation of Entity Recognition Algorithms in Short Texts. *CLEI ELECTRONIC JOURNAL*, 20(1):13.

Chabchoub, M., Gagnon, M., and Zouaq, A. (2018). FICLONE: Improving DBpedia Spotlight Using Named Entity Recognition and Collective Disambiguation. 5(1):17.

Chan, J. (2020). Knowledge synthesis: A conceptual model and practical guide. *Open and Sustainable Innovation Systems (OASIS) Lab*.

Chandrasekaran, D. and Mago, V. (2021). Evolution of Semantic Similarity – A Survey. *ACM Computing Surveys*, 54(2):1–37. arXiv: 2004.13820.

Choi, E., Levy, O., Choi, Y., and Zettlemoyer, L. (2018). Ultra-Fine Entity Typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia. Association for Computational Linguistics.

Choi, Y., Breck, E., and Cardie, C. (2006). Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 431–439.

Clark, A. and Chalmers, D. (1998). The extended mind. *analysis*, 58(1):7–19. Publisher: JSTOR.

Cole, N. (2020). *The art and business of online writing: how to beat the game of capturing and keeping attention*. OCLC: 1302730850.

Costello, E. K., Lauber, C. L., Hamady, M., Fierer, N., Gordon, J. I., and Knight, R. (2009). Bacterial community variation in human body habitats across space and time. *science*, 326(5960):1694–1697. Publisher: American Association for the Advancement of Science.

Dai, D., Xiao, X., Lyu, Y., Dou, S., She, Q., and Wang, H. (2019). Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6300–6308. Issue: 01.

Dalvi, N., Kumar, R., Pang, B., Ramakrishnan, R., Tomkins, A., Bohannon, P., Keerthi, S., and Merugu, S. (2009). A web of concepts. In *Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–12.

de Souza, C. S. (2021). Inteligência Artificial e IA: Filósofos Analisam Dois Projetos Distintos à Luz da Interdisciplinaridade. *C. S.*, page 15.

Derczynski, L., Nichols, E., van Erp, M., and Limsopatham, N. (2017). Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

Dessì, D., Osborne, F., Recupero, D. R., Buscaldi, D., and Motta, E. (2021). Generating Knowledge Graphs by Employing Natural Language Processing and Machine Learning Techniques within the Scholarly Domain. *Future Generation Computer Systems*, 116:253–264. arXiv:2011.01103 [cs].

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, N., Xu, G., Chen, Y., Wang, X., Han, X., Xie, P., Zheng, H.-T., and Liu, Z. (2021). Few-NERD: A Few-Shot Named Entity Recognition Dataset. Technical Report arXiv:2105.07464, arXiv. arXiv:2105.07464 [cs] version: 6 type: article.

Ebbinghaus, H. (1964). *Memory*. .

Eichler, J. S. A., Casanova, M. A., Furtado, A. L., Ruback, L., Leme, L. A. P. P., Lopes, G. R., Nunes, B. P., Raffaetà, A., and Renso, C. (2017). Searching Linked Data with a Twist of Serendipity. In Dubois, E. and Pohl, K., editors, *Advanced Information Systems Engineering*, volume 10253, pages 495–510. Springer International Publishing, Cham. Series Title: Lecture Notes in Computer Science.

Engelbart, D. (2004). Augmenting society's collective IQs. In *Proceedings of the fifteenth ACM conference on Hypertext and hypermedia*, pages 1–1.

Engelbart, D. and Lehtman, H. (1988). Working together. *Byte*, 13(13):245–252.

Engelbart, D. C. (1990). Knowledge-domain interoperability and an open hyperdocument system. In *Proceedings of the 1990 ACM conference on Computer-supported cooperative work*, CSCW '90, pages 143–156, New York, NY, USA. Association for Computing Machinery.

Engelbart, D. C. (1992). Toward high-performance organizations: A strategic role for groupware. In *Proceedings of the GroupWare*, volume 92, pages 3–5. Citeseer.

Engelbart, D. C. (1995). Toward augmenting the human intellect and boosting our collective IQ. *Communications of the ACM*, 38(8):30–32. Publisher: ACM New York, NY, USA.

Ernst, P., Siu, A., and Weikum, G. (2015). Knowlife: a versatile approach for constructing a large knowledge graph for biomedical sciences. *BMC bioinformatics*, 16(1):1–13. Publisher: Springer.

Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying Relations for Open Information Extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Falke, T., Meyer, C. M., and Gurevych, I. (2017). Concept-Map-Based Multi-Document Summarization using Concept Coreference Resolution and Global Importance Optimization. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 801–811, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Fang, S., Huang, Z., He, M., Tong, S., Huang, X., Liu, Y., Huang, J., and Liu, Q. (2021). Guided Attention Network for Concept Extraction. volume 2, pages 1449–1455. ISSN: 1045-0823.

Finkel, J. R., Grenager, T., and Manning, C. D. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05)*, pages 363–370.

Forte, T. (2022). *Building a second brain: A proven method to organize your digital life and unlock your creative potential*. Atria Books. tex.lccn: 2021057379.

Frodeman, R., editor (2017). *The Oxford Handbook of Interdisciplinarity*, volume 1. Oxford University Press.

Gillick, D., Lazic, N., Ganchev, K., Kirchner, J., and Huynh, D. (2016). Context-Dependent Fine-Grained Entity Type Tagging. Technical Report arXiv:1412.1820, arXiv. arXiv:1412.1820 [cs] type: article.

Girard, J. and Girard, J. (2015). Defining knowledge management: Toward an applied compendium. *Online Journal of Applied Knowledge Management*, 3(1):1–20.

Gomaa, W. H. and Fahmy, A. A. (2013). A Survey of Text Similarity Approaches.

Haller, H. (2011). *User Interfaces for Personal Knowledge Management with Semantic Technologies*. PhD thesis.

Huang, L., Milne, D., Frank, E., and Witten, I. H. (2012). Learning a concept-based document similarity measure. *Journal of the American Society for Information Science and Technology*, 63(8):1593–1608. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.22689.

Ilkou, E. (2022). Personal Knowledge Graphs: Use Cases in e-learning Platforms. *ArXiv*.

Joshi, M., Levy, O., Zettlemoyer, L., and Weld, D. (2019). BERT for Coreference Resolution: Baselines and Analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China. Association for Computational Linguistics.

Kambhatla, N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the ACL interactive poster and demonstration sessions*, pages 178–181.

Kenter, T. and Rijke, M. d. (2015). Short Text Similarity with Word Embeddings. *CIKM*.

Kimbro, L. (2003). How to Make a Complete Map of Every Thought You Think.

Klein, J. (2015). *Interdisciplining Digital Humanities: Boundary Work in an Emerging Field*. digitalculturebooks, Ann Arbor, MI.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From Word Embeddings To Document Distances. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 957–966. PMLR. ISSN: 1938-7228.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Leacock, C. and Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283.

Leal, J. P., Rodrigues, V., and Queirós, R. (2012). Computing semantic relatedness using dbpedia. In *1st Symposium on Languages, Applications and Technologies*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., and Auer, S. (2015). Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195. Publisher: IOS Press.

Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., and Shepherd, M. (1990). Cyc: toward programs with common sense. *Communications of the ACM*, 33(8):30–49. Publisher: ACM New York, NY, USA.

Ley, M. (2002). The DBLP computer science bibliography: Evolution, research issues, perspectives. In *International symposium on string processing and information retrieval*, pages 1–10. Springer.

Li, J., Sun, A., Han, J., and Li, C. (2020). A Survey on Deep Learning for Named Entity Recognition. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

Li, X., Taheri, A., Tu, L., and Gimpel, K. (2016). Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1445–1455.

Lin, B. Y., Chen, X., Chen, J., and Ren, X. (2019). KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning. arXiv:1909.02151 [cs].

Ling, X. and Weld, D. (2021). Fine-Grained Entity Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 26(1):94–100.

Liu, H. and Singh, P. (2004). ConceptNet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226. Publisher: Springer.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Luo, G., Huang, X., Lin, C.-Y., and Nie, Z. (2015). Joint Entity Recognition and Disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888, Lisbon, Portugal. Association for Computational Linguistics.

Mantel, N. (1967). The detection of disease clustering and a generalized regression approach. *Cancer research*, 27(2_Part_1):209–220. Publisher: AACR.

Markman, A. B. (2013). *Knowledge representation*. Psychology Press.

Martinez-Rodriguez, J. L., Lopez-Arevalo, I., and Rios-Alvarado, A. B. (2018). OpenIE-based approach for Knowledge Graph construction from text. *Expert Systems with Applications*, 113:339–355.

Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011). DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the*

7th International Conference on Semantic Systems, I-Semantics '11, pages 1–8, New York, NY, USA. Association for Computing Machinery.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41. Publisher: ACM New York, NY, USA.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244. Publisher: Oxford University Press.

Milo, N. (2022). In what ways can we form useful relationships between notes?

Murre, J. M. and Dros, J. (2015). Replication and analysis of Ebbinghaus' forgetting curve. *PloS one*, 10(7):e0120644. Publisher: Public Library of Science San Francisco, CA USA.

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26. Publisher: John Benjamins.

Navigli, R. and Ponzetto, S. P. (2010). BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225.

Nelson, T. H. (1965). Complex information processing: a file structure for the complex, the changing and the indeterminate. In *Proceedings of the 1965 20th national conference*, pages 84–100.

Nguyen, H. T., Duong, P. H., and Cambria, E. (2019). Learning short-text semantic similarity with word embeddings and external knowledge sources. *Knowledge-Based Systems*, 182:104842.

Ni, Y., Xu, Q. K., Cao, F., Mass, Y., Sheinwald, D., Zhu, H. J., and Cao, S. S. (2016). Semantic Documents Relatedness using Concept Graph Representation. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 635–644, San Francisco California USA. ACM.

Nicolescu, B. (2002). *Manifesto of transdisciplinarity*. Suny Press.

Parameswaran, A., Garcia-Molina, H., and Rajaraman, A. (2010). Towards the web of concepts: Extracting concepts from large datasets. *Proceedings of the VLDB Endowment*, 3(1-2):566–577. Publisher: VLDB Endowment.

Pawar, S., Palshikar, G. K., and Bhattacharyya, P. (2017). Relation Extraction : A Survey. *arXiv:1712.05191 [cs]*. arXiv: 1712.05191.

Pedersen, T., Pakhomov, S. V. S., Patwardhan, S., and Chute, C. G. (2007). Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics*, 40(3):288–299.

Piao, G. and Breslin, J. G. (2015). Computing the semantic similarity of resources in dbpedia for recommendation purposes. In *Joint International Semantic Technology Conference*, pages 185–200. Springer.

Rebele, T., Suchanek, F., Hoffart, J., Biega, J., Kuzey, E., and Weikum, G. (2016). YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *International semantic web conference*, pages 177–185. Springer.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Technical Report arXiv:1908.10084, arXiv. arXiv:1908.10084 [cs] type: article.

Resnik, P. (1995). Using Information Content to Evaluate Semantic Similarity in a Taxonomy. arXiv:cmp-lg/9511007.

Ringland, N., Dai, X., Hachey, B., Karimi, S., Paris, C., and Curran, J. R. (2019). NNE: A Dataset for Nested Named Entity Recognition in English Newswire. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5176–5181, Florence, Italy. Association for Computational Linguistics.

Roth, D. and Yih, W.-t. (2004). A linear programming formulation for global inference in natural language tasks. Technical report, Illinois Univ at Urbana-Champaign Dept of Computer Science.

Sang, E. F. T. K. and De Meulder, F. (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. *arXiv:cs/0306050*. arXiv: cs/0306050.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sarawagi, S. (2008). Information extraction. *Foundations and Trends® in Databases*, 1(3):261–377. Publisher: Now Publishers, Inc.

Schmidt, J. F. (2014). Der Nachlass Niklas Luhmanns–eine erste Sichtung: Zettelkasten und Manuskripte. *Soziale Systeme*, 19(1):167–183. Publisher: De Gruyter Oldenbourg.

Serrat, O. (2017). Social Network Analysis. In Serrat, O., editor, *Knowledge Solutions: Tools, Methods, and Approaches to Drive Organizational Performance*, pages 39–43. Springer, Singapore.

Singh, P., Lin, T., Mueller, E. T., Lim, G., Perkins, T., and Zhu, W. L. (2002). Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 1223–1237. Springer.

SpazioDati, . (2012). Dandelion API | Semantic Text Analytics as a service.

Speer, R., Chin, J., and Havasi, C. (2018). ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. arXiv:1612.03975 [cs] version: 2.

Strauss, B., Toma, B., Ritter, A., de Marneffe, M.-C., and Xu, W. (2016). Results of the WNUT16 Named Entity Recognition Shared Task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144, Osaka, Japan. The COLING 2016 Organizing Committee.

Swinscow, T. D. V. and Campbell, M. J. (2002). *Statistics at square one*. Bmj London.

Tomaszczyk, J. and Matysek, A. (2020). Digital Wisdom in Research Work.

Tran, V.-H., Phi, V.-T., Kato, A., Shindo, H., Watanabe, T., and Matsumoto, Y. (2021). Improved decomposition strategy for joint entity and relation extraction. *Journal of Natural Language Processing*, 28(4):965–994. Publisher: The Association for Natural Language Processing.

Tutek, M., Glavas, G., Šnajder, J., Milic-Frayling, N., and Basic, B. D. (2016). Detecting and Ranking Conceptual Links between Texts Using a Knowledge Base. *CIKM*.

Van Harmelen, F., Lifschitz, V., and Porter, B. (2008). *Handbook of knowledge representation*. Elsevier.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention Is All You Need. arXiv:1706.03762 [cs].

Waldis, A., Mazzola, L., and Kaufmann, M. (2018). Concept Extraction with Convolutional Neural Networks:. In *Proceedings of the 7th International Conference on Data Science, Technology and Applications*, pages 118–129, Porto, Portugal. SCITEPRESS - Science and Technology Publications.

Wang, Q., Mao, Z., Wang, B., and Guo, L. (2017). Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

Weikum, G., Dong, X. L., Razniewski, S., and Suchanek, F. (2021). Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases. *Foundations and Trends® in Databases*, 10(2-4):108–490.

Weischedel, R., Palmer, M., Marcus, M., Eduard, H., Pradhan, S., Ramshaw, L., Xue, N., Taylor, A., Kaufman, J., Franchini, M., El-Bachouti, M., Belvin, R., and Houston, A. (2013). OntoNotes Release 5.0. Type: dataset.

Weston, J., Bordes, A., Yakhnenko, O., and Usunier, N. (2013). Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973*.

Witten, I. H. and Milne, D. N. (2008). An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. Publisher: AAAI press.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wu, F. and Weld, D. S. (2010). Open information extraction using wikipedia. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 118–127.

Yan, Y., Okazaki, N., Matsuo, Y., Yang, Z., and Ishizuka, M. (2009). Unsupervised relation extraction by mining wikipedia texts using information

from the web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1021–1029.

Yang, C., Zhang, J., Wang, H., Li, B., and Han, J. (2020). Neural Concept Map Generation for Effective Document Classification with Interpretable Structured Summarization. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1629–1632. Association for Computing Machinery, New York, NY, USA.

Yazdani, M. and Popescu-Belis, A. (2013). Computing text semantic relatedness using the contents and links of a hypertext encyclopedia. *Artif. Intell.*

Yu, B., Zhang, Z., Shu, X., Wang, Y., Liu, T., Wang, B., and Li, S. (2019). Joint extraction of entities and relations based on a novel decomposition strategy. *arXiv preprint arXiv:1909.04273*.

Zhang, F., Yuan, N. J., Lian, D., Xie, X., and Ma, W.-Y. (2016). Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362.

Zhang, N., Xu, X., Tao, L., Yu, H., Ye, H., Xie, X., Chen, X., Li, Z., Li, L., Liang, X., Yao, Y., Deng, S., Zhang, W., Zhang, Z., Tan, C., Huang, F., Zheng, G., and Chen, H. (2022). DeepKE: A Deep Learning Based Knowledge Extraction Toolkit for Knowledge Base Population. Technical Report arXiv:2201.03335, arXiv. arXiv:2201.03335 [cs] type: article.