

On The Automatic Generation of Knowledge Connections

Felipe Poggi A. Fraga¹^a, Marcus Poggi¹^b, Marco A. Casanova¹^c, Luiz André P. Paes Leme²^d

¹*Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro RJ, Brazil*
{ffraga,poggi,casanova}@inf.puc-rio.br

⁴*Universidade Federal Fluminense, Niterói RJ, Brazil*
lapaesleme@ic.uff.br

Keywords: Personal Knowledge Management, Natural Language Processing, Semantic Similarity, Concept Extraction, Text Relatedness, Connections Generation, Note-taking Apps, Bi-directional Hyperlinks

Abstract: 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*, as well as 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 Natural Language Processing (NLP). This opens up an interesting opportunity to apply NLP techniques to operating with knowledge. This paper proposes a methodology that uses NLP and networked note-taking apps to transform a siloed text collection into an interconnected and inter-navigable text collection. The navigation mechanisms are based on shared concepts and semantic relatedness between texts. The paper proposes a methodology, presents demonstrations using examples, and describes an evaluation to determine if the system functions correctly and whether the proposed connections are coherent.

1 Introduction

The recent surge in popularity of the Personal Knowledge Management (PKM) field has led to a myriad of new note-taking tools being released since 2016. These innovative tools provide the general public with brand-new functionalities to operate with knowledge that were not previously available, an example being *bidirectional hyperlinks*.

Even though advancements in note-taking tools are happening simultaneously with the accelerated development of Natural Language Processing (NLP), as of today, these two fields still interact in a very superficial way. This presents an opportunity to combine the fields of PKM and NLP, by enhancing the features of note-taking tools using Artificial Intelligence.

In a structured way, this work is inspired by the intersection of three elements: (1) Personal Knowledge Management; (2) New generation of Note-Taking Tools; (3) Natural Language Processing.


Personal Knowledge Management deals with creating an external and persistent collection of a person’s


knowledge. Recently, PKM has become increasingly popular, as evidenced by the popularity of the books: “How to Take Smart Notes”, (Ahrens, 2017), which explains the note-taking process by the prolific Sociologist Niklas Luhmann, called the Zettelkasten; and “Building a Second Brain”, (Forte, 2022), which explains how to build an external collection of knowledge to work faster and with better quality.


Another piece of evidence for this increase in popularity is the appearance of a new category of note-taking tools. Tools such as Roam Research, Obsidian, and Tana present new features for knowledge organization based on **networked note-taking**. Networked note-taking contrasts with the hierarchical, folder-document systems used by traditional note-taking. The feature that stands out as essential for this work is the use of **bidirectional hyperlinks** to connect notes. Representing both directions of the link, *front-links* (*outgoing*) and *back-links* (*incoming*), provides a fundamentally different type of knowledge to work with.


The main contributions and possible impacts on the field of Personal Knowledge Management are:

Contribution 1 – The creation of a methodology that automatically generates connections between texts: Connections between texts are generated following two different options: *Semantic Relatedness* be-

^a <https://orcid.org/0000-0002-1385-9098>

^b <https://orcid.org/0000-0003-0827-7111>

^c <https://orcid.org/0000-0003-0765-9636>

^d <https://orcid.org/0000-0001-6014-7256>

tween texts and *Shared Concept*. The connections are considered to be coherent and reliable for the intended functions of *Recall*, *Elaboration*, and *New Insight*.

Contribution 2 – A theoretical and practical workflow for introducing NLP capabilities to modern note-taking tools: To the best of our knowledge, this paper is the first work to explicitly unite the fields of note-taking apps and Natural Language Processing. The methodology presented in the paper contributes to the design and implementation of note-taking applications that use NLP.

Contribution 3 – A technique for generating connections between texts using Shared Concept: The proposed navigation using concept nodes provides relevant information about concepts and creates bridges between texts that mention the same concept. With the added possibility of calculating the text relatedness using the concepts mentioned in the texts.

The possible implications are, therefore, to lower the entrance barrier for applying Natural Language Processing to Knowledge Management, which may enhance collective intelligence and capabilities to overcome the challenges that humanity soon must face.

The rest of this paper is organized as follows. Section 2 defines the problem to be solved. Section 3 covers related work. Section 4 outlines the proposed methodology. Section 5 explains the experiments that evaluate the performance of the proposed methodology. Section 6 discusses the implications of this work. Finally, Section 7 contains the conclusions.

2 Problem Statement

The objective of this work is to apply NLP techniques to leverage and enhance the current functionalities in note-taking apps. To facilitate, empower, or replace human effort in operating with knowledge, specifically inside networked note-taking apps.

The approach is intended to help users explore a text collection by automatically proposing connections between texts. An essential element of the approach is to use networked note-taking apps to visualize and navigate through the connections.

Furthermore, the problem statement and the methodology are designed to ensure that users have an active role. Automatically generated connections could nudge users into a passive role, with Artificial Intelligence doing all the “thinking”.

Given an initial text collection, a user is responsible for selecting the highlights s/he wants to insert into the system for the automatic generation of connections. Hence, the user has a direct influence on the result of the methodology. A *highlight* is a selected passage

from a text in the initial text collection that is used to create a set of highlights, H , where the methodology will be applied.

Two types of connections are proposed to navigate through the highlights:

1. *Concepts Connections:* Navigation between highlights through shared concepts, by first navigating from a highlight h to a concept c mentioned in h , and then from c to other highlights that mention c .
2. *Text Relatedness Connections:* Navigation based on a recommendation system, with texts suggested according to Semantic Relatedness.

The problem addressed is informally defined as:

What: Automatically generate connections to create a highlights collection that is interconnected and inter-navigable, represented by a graph.

How: Create connections between highlights using shared concepts and semantic relatedness.

Where: Use networked note-taking tools to navigate.

In what follows, recall that, in an undirected graph, an edge is an unordered pair $\{x,y\}$ of graph nodes (contrasting with a directed graph, where a directed arc is an ordered pair (x,y) of nodes).

Given two sets X and Y of nodes, let \overleftrightarrow{XY} denote the set of all edges $\{x,y\}$ such that $x \in X$ and $y \in Y$.

The problem is then more precisely defined as follows. Given a set of highlights H , create an undirected graph $G_H = (V,E)$, called an *interconnection graph for H* , such that:

$$V \subseteq H \cup C \cup A \quad (1)$$

where H is a set of *Highlight nodes*, C is a set of *Concept nodes*, and A is a set of *Author nodes*, and

$$E \subseteq \overleftrightarrow{HH} \cup \overleftrightarrow{HC} \cup \overleftrightarrow{CC} \cup \overleftrightarrow{AA} \cup \overleftrightarrow{AC} \cup \overleftrightarrow{AH} \quad (2)$$

where the edges represent connections between the nodes and are always bidirectional.

We assume that all edges in \overleftrightarrow{AH} are given, while the others must be computed, as well as the Concept nodes, C .

3 Related Work

3.1 Frameworks and Systems

Becker et al. (2021b) presents CO-NNECT, a framework that proposes connection paths between sentences according to the concepts mentioned and their relations. This work uses concepts in ConceptNet and

language models trained on knowledge relations from ConceptNet. Maria Becker suggests this framework could be used to enrich texts and knowledge bases.

Ilkou (2022) uses Entity Extraction and the DBpedia Knowledge graph to generate Personal Knowledge Graphs with specific e-learning user's 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 answers to distract the user in multiple-choice questions. The system uses knowledge bases and semantic relatedness between texts.

3.2 Concept Recognition

This section briefly outlines works that focus on extracting concepts.

(Mendes et al., 2011) introduces DBpedia Spotlight, which approaches concept extraction as a text annotation task, and can annotate mentions with DBpedia resources (which include abstract concepts).

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

Fang et al. (2021) proposes GACEN (Guided Attention Concept Extraction Network), a technique of attention networks feeding a CRF to extract concepts using the title, topic, and clue words.

Recent Surveys on Named Entity Recognition (Li et al., 2020) and (Canales and Murillo, 2017) point to a set of industry-based tools. One tool mentioned in both surveys is Dandelion API, (SpazioDati, 2012), which can identify conceptual entities. Dandelion API performs a high-quality identification of entities, as well as entity linking to the DBpedia knowledge base.

3.3 Using Knowledge Graphs

This section briefly outlines a few works that use knowledge bases instead of building them.

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 concepts by how much information they have in common.

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.

Anand and Kotov (2015) utilizes DBpedia and ConceptNet to perform query expansions and retrieve additional results to a given query.

3.4 Text Semantic Relatedness

This section presents works that perform the tasks of Text Semantic Relatedness and Text Semantic Similarity. This task is usually divided into two main categories of algorithms: Knowledge-based methods and Corpus-based methods (Gomaa and Fahmy, 2013).

3.4.1 Knowledge-based methods

Speer et al. (2017) describes the ConceptNet Numberbatch, a relatedness measure between concepts using the ConceptNet knowledge base.

Yazdani and Popescu-Belis (2013) presents a relatedness metric based on Visiting Probability using Random Walks between two texts. Visiting probability is calculated using relations matrices between concepts. The metric uses two different relation types, Wikipedia links and a relatedness score between concepts.

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.

3.4.2 Corpus-based methods

With the introduction of word embeddings, (Mikolov et al., 2013), similarity metrics shifted to using the semantic meaning of the words. Nonetheless, pre-trained word embeddings still have limitations when dealing with polysemic words, which have multiple meanings, such as "bank" which can mean a river bank or a financial bank.

One solution to this problem is BERT, presented in Devlin et al. (2019). BERT (Bidirectional Encoder Representations from Transformers) uses a Transformers architecture (encoder-decoder), (Vaswani et al., 2017), to create word embeddings that capture the meaning of surrounding words. BERT pre-trains language models considering both directions of the text so that words further along in the text still influence the vectorial representation of earlier words.

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

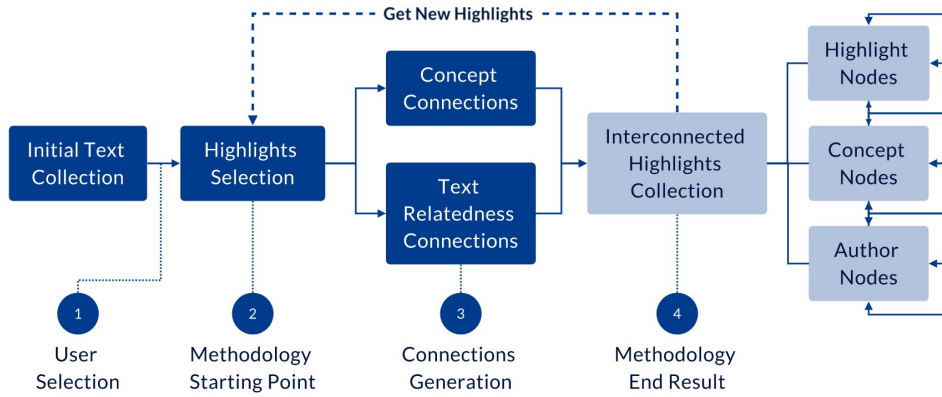


Figure 1: Overview of the proposed Methodology

4 Methodology for Construction of Interconnection Graphs

4.1 Overview of the Methodology

The proposed methodology is to create an interconnected, human-readable, and human-navigable collection of highlights, G_H . Instead of being a one-time execution, the methodology is recurrently applied to continuously updated sets of Highlights H_i selected from the initial text collection.

The methodology creates a first interconnection graph G_{H_0} and then updates the initial graph with newly added highlights, $G_{H_i} \rightarrow G_{H_{i+1}}$. This is done to allow for the highlights set, H_i to be constantly changing as users add new texts and passages.

Each interconnection graph, G_{H_i} , can be navigated by transforming the nodes in the graph into pages inside a networked note-taking tool that enables navigation and editing. The chosen tool for this task is Obsidian, (Li and Xu, 2020).

Figure 1 details the methodology overview, including this background check for new highlights. The remainder of this section details the processes for a single pass of the methodology for a fixed set of highlights, H . This process occurs between stages 2-4.

To create an interconnection graph, G_H , the underlying problem is simplified to solving for 9 graph components (3 node types and 6 edge types). We assume that H , A , and \overleftrightarrow{AH} are given and known in advance. The six remaining components (1 node type and 5 edge types) are obtained from the two paths described in section 2:

1. Concepts Connections

- *Concept Recognition*: C , \overleftrightarrow{HC} , and \overleftrightarrow{AC}
- *Concept Relationships*: \overleftrightarrow{CC}

2. Text Relatedness Connections: \overleftrightarrow{HH} , and \overleftrightarrow{AA}

Concepts Connections represent paths between any given highlight and other highlights that mention the same concept. Text Relatedness Connections create a direct bridge between highlights, providing a clear picture of other highlights and ideas that are related to any given highlight.

4.2 Concept Connections

Concept Connections refer to all the connections that involve concepts, the most important type being between Highlights and Concepts, which is identified through the task of Concept Recognition. Concept Relationship, subsection 4.2.2, plays an exploratory role and collects important information for Section 4.3 on Text Relatedness.

4.2.1 Concept Recognition

This section details the tasks of entity recognition and the filtering of these entities to obtain concepts. The pipeline starts with entities, and as the filtering process occurs, the term concepts will be employed. Concepts are defined as conceptual entities, such that every concept is also an entity, but not all entities are concepts, Equation 3. In special, for the scope of this paper, Named Entities are not considered to be concepts.

$$C \subseteq E \quad (3)$$

where C is the set of concepts and E is the set of entities.

The first task in the pipeline for this section is identifying the entities mentioned in a highlight's text. Since all types of entities are initially extracted, the task is described as Entity Recognition.

The task is the joint task of Entity Recognition together with Entity Linking, which is formalized as follows:

1. Spot mentions in the text.

2. Collect candidate entities for each mention.
3. Select the most likely entity represented by each mention.

Several tools were considered to perform the joint task, and two different tools were chosen.

The Dandelion API (SpazioDati, 2012) was used for initial Entity Recognition for a couple of reasons. First, it has the functionality of identifying concepts. Second, it provides an exact confidence level for each mention-entity occurrence, and third, it links each mention to a DBpedia resource (entity).

The other tool is DBpedia Spotlight (Mendes et al., 2011), which also links mentions to a DBpedia resource and is used to identify any mention-entity occurrence that may have been missed by Dandelion API. DBpedia Spotlight is used for this task because it has a broader reach and captures more mentions.

It is worth mentioning that both tools do not require the traditional data cleaning of removing stop words and performing lemmatization on the input text for the task of Entity Recognition.

Upon extraction of the information regarding the mentions and corresponding entities, a **mentions database** is built. The database contains important information that is used in the filtering process, including mention surface form, entity name, entity DBpedia URL, location in the text, and confidence score.

The following step is filtering the mentions database according to unwanted entities. This step marks the transition from dealing with entities to dealing with concepts. The term “concepts” refers to the set of entities after the filtering process, which prioritizes conceptual entities.

Let H be a collection of selected highlights and M be a set of mention-entity occurrences, denoted (m, e) , identified in the text corpus of highlights, where m denotes the text fragment used to represent the entity e . Let also $\alpha(e)$ be a set of taxonomic categories that classify e , $\beta(m, e)$ be the confidence of m being an entity e , and $\gamma(e)$ be the frequency of the entity e in the texts, i.e. $\gamma(e) = 1$ means that the entity occurs in just one text $t \in T$.

The filtering expression for the set M is as follows.

$$\alpha(e) \cap L = \emptyset \wedge \beta(m, e) > 0.6 \wedge \gamma(e) > 1 \quad (4)$$

where L is a set of unwanted categories (such as the DBpedia categories /Film, /Band, /Magazine, and /TelevisionShow), α is a DBpedia-type filter, β is a confidence level filter, and γ is a single occurrence filter. There are user-defined parameters for each of these filters.

After the filtering process, a final *concepts list*, C , is defined with the remaining entities’ DBpedia URLs.

4.2.2 Concept Relationships

This section details the procedures to identify relationships between the concepts in the graph. All entities that passed the filtering process are regarded as concepts from here onward.

This task is composed of finding all relationships between concepts contained in the graph G_H , given by the concepts list, C . To find all relationships between each possible pair of concepts, information is retrieved from two Knowledge Graphs, DBpedia (Lehmann et al., 2015) and ConceptNet (Speer et al., 2017).

Queries are made to each knowledge graph to determine all relationships $r \in R$ between each concept pair in the concepts list, C . The data is gathered by posting queries to both knowledge graphs and collecting the results from these queries. Queries and their results follow the triples format (subject, predicate, object).

$$(\forall c_1, c_2 \in C)(\text{find all } r \in R) \text{ such that:} \quad (5)$$

$$r \text{ IN } (c_1, p, c_2) \quad (6)$$

where R is the chosen set of relationships and (c_1, p, c_2) is the result triple with predicate p . The set of relationships considered, R is composed of relations from DBpedia, R_D , and ConceptNet, R_C .

DBpedia was accessed to capture relations between concepts based on Wikipedia page links. DBpedia relations are Wikipedia links from one concept’s page to another. These relations are the least specific and provide the lowest value information.

DBpedia relations (R_D)

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

ConceptNet was used to obtain commonsense knowledge between concepts. Commonsense knowledge presents more granular relationship types between concepts, providing richer information on how concepts connect with one another.

ConceptNet relations (R_C)

- Causality
 - /r/Causes
 - /r/CapableOf
 - /r/MotivatedByGoal
- Equivalency
 - /r/Synonym
 - /r/SimilarTo

- Opposition
 - /r/Antonym
 - /r/DistinctFrom
- Dependency
 - /r/HasPrerequisite
 - /r/HasContext
 - /r/HasProperty
 - /r/PartOf
- General
 - /r/IsA
 - /r/RelatedTo

It is worth noting that all relationships are used as bidirectional links between concepts. Technically, the inverse of each relation is also captured.

4.3 Text Relatedness Connections

Text Relatedness Connections are direct connections between highlights. They represent an important bridge between ideas, creating direct pathways between highlights. This section details the creation of connections between highlights using two different relatedness metrics: *Knowledge-based Shared Concepts Relatedness* (*kbr*) using concepts; and *Corpus-based Semantic Relatedness* (*cbr*). A final Relatedness Score is achieved by combining the two metrics:

$$S(h_1, h_2) = \frac{kbr(h_1, h_2) + cbr(h_1, h_2)}{2} \quad (7)$$

where S is the final relatedness score, kbr is the knowledge-based relatedness, and cbr is the corpus-based relatedness.

It is worth noting that recommendations to other highlight nodes are also divided between highlights that are from the *same author* and belonging to *different authors*. This is done to provide multiple options when navigating between highlight nodes.

4.3.1 Knowledge-based Relatedness

Knowledge-based shared concepts relatedness between highlights is calculated according to the concepts that are mentioned in each highlight’s text together with the relatedness between each concept.

$$kbr(h_1, h_2) = \frac{\sum_{c_1 \in h_1, c_2 \in h_2} rel(c_1, c_2)}{|h_1| \cdot |h_2|} \quad (8)$$

$$rel(c_1, c_2) = \frac{nr(c_1, c_2) + sr(c_1, c_2)}{2} \quad (9)$$

where $|h|$ is the number of concepts in highlight h , $nr(c_1, c_2)$ is the numerical relatedness and $sr(c_1, c_2)$ is the shared relationship between concepts.

Numerical Relatedness is a quantitative score that represents the direct semantic relatedness between concepts, and *Shared Relationship* is a binary value that signals the presence of any descriptive relation between the concepts.

The *numerical relatedness score*, $nr(c_1, c_2)$, is a value between 0 and 1 that indicates the relatedness between two concepts as defined by the ConceptNet Numberbatch (Speer et al., 2017), which calculates word embeddings based on shared neighbors in a ConceptNet graph with additional retrofitting using GloVe and word2vec embeddings. This score was further normalized to represent relatedness scores between 0 and 1.

An example of the two concept-relatedness metrics is presented in Table 1, showing the relatedness between the concept of “Knowledge” as compared to selected concepts.

Table 1: Relatedness scores for the concept of “Knowledge”

Concept	Relatedness to: “ Knowledge ”	
	Numerical Relatedness	Shared Relationship
Information	0.460	1
Wisdom	0.442	1
Understanding	0.434	1
Intelligence	0.332	1
Learning	0.328	1
Memory	0.181	1
Thought	0.120	1
Biology	0.063	0
Innovation	0.017	1
Nutrient	-0.007	0
Pricing	-0.080	0

Shared Relationship is when two concepts share one or more relationships with each other. A relationship happens when two concepts are part of a relation triple, (subject, predicate, object), extracted in subsection 4.2.2. i.e. One concept appears in the subject position, and another concept in the object position, while sharing a predicate (relation) between them.

The presence of a relationship between concepts is used to create a *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; the number 1 is used to represent that a relationship between two given concepts exists, and 0 is used when there is no relationship.

$$sr(c_1, c_2) = \max_{r \in R} (r \in R) \quad (10)$$

where R is the set of all relationships considered.

The relationship types R considered were detailed in Section 4.2.2 and were selected from DBpedia and

ConceptNet. These relationships were applied in both directions, meaning that if concept *A* is related to concept *B*, then concept *B* is automatically related to concept *A*. This is done to satisfy the four properties of a distance metric: non-negativity, symmetry, the identity of indiscernibles, and triangle inequality.

A *Knowledge-based Shared Concept Relatedness Matrix* is created by calculating the overall relatedness between each pair of highlights in the collection. The method to calculate the relatedness between two given highlights is based on the average connection strength between the concepts of each highlight’s text. This is inspired by one of the features for document similarity using concepts, presented in (Huang et al., 2012).

For each of the metrics, the following procedures are performed. First, the concepts mentioned in each of the highlights are identified. Next, an average relatedness is calculated considering all the relationships between the concepts in highlight A with the concepts in highlight B. When a concept appears in both highlights, the relatedness score is 1. The final relatedness between the two highlights 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.

4.3.2 Corpus-based Semantic Relatedness

Corpus-based Semantic Relatedness is defined as the similarity between the semantic meaning of two texts. This relatedness metric is calculated using SBERT Sentence Transformers (Reimers and Gurevych, 2019), an adaptation that generates Sentence Embeddings using the BERT architecture:

$$cbr(h_1, h_2) = \cos(SB(h_1), SB(h_2)) \quad (11)$$

where $SB(h)$ is the Text Embedding for highlight h .

The process consists of two stages: (1) Encode the highlights’ texts; (2) Create a Relatedness Matrix.

Encode the Text

To calculate the relatedness (or distance) between two given highlights, it is necessary to encode each highlight’s text into a vectorial representation. This numerical format, a tensor, represents the semantic meaning of text across multiple dimensions so that it is possible to apply similarity metrics to compare two highlights.

Sentence-BERT, SBERT, is a model trained only to generate embeddings, which means it only contains the encoding architecture and does not contain a decoder component. 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 computationally very efficient and

run in a fraction of the time needed to run the entire BERT architecture. SBERT also does not require the removal of stop words and lemmatization.

Sentence-BERT works by automatically adding a pooling operation to the output of BERT. It performs fine-tuning of a neural network architecture composed of siamese and triplet networks to produce sentence embeddings that are meaningful and can be compared using similarity metrics.

Sentence-BERT outputs fixed-sized Sentence Embeddings, $SB(h)$, that can be easily explored to calculate the relatedness between them. All the embeddings generated with the same model checkpoint will be compatible with one another. This does not depend on the texts being encoded together in the same batch.

Creating a Relatedness Matrix

Once a vectorial representation, or a Text Embedding, is generated for all highlights in the text collection, the relatedness between these vectors is calculated to generate a relatedness matrix.

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

To build a relatedness matrix, the cosine similarity was calculated between the vectors representing all pairs of highlights in the collection.

4.4 Methodology End Result

The result of the methodology is an interconnection graph, G_H , with the proposed connections. For users to actively use this graph, the nodes and edges are respectively “translated” to pages and hyperlinks in the networked note-taking tool Obsidian, (Li and Xu, 2020).

Creating pages and hyperlinks turns the graph into an accessible collection of highlights. Two figures briefly illustrate what this collection looks like. Figure 2 shows the Highlight Node’s page for an example

Building a Second Brain Note 4

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. New generation of powerful apps have created lessons you will find within these pages are [principles](#).

Recommended Highlights

Semantic Relatedness

- [sameauthor1 Building a Second Brain](#)
- [sameauthor2 Building a Second Brain](#)
- [diffauthor1 Limitless Note 2 - 0.445](#)

Personal Knowledge Management

Personal knowledge management (PKM) is a process of collecting information that a person uses to gather, classify, store, search, retrieve and share knowledge in their daily activities and the way in which these processes support work activities. It is a response to the idea that knowledge workers need to be responsible for their

Figure 2: Example Highlight Node page and hyperlinks

highlight. The figure shows the different alternatives for navigation, where red rectangles take users to Concept Nodes, and blue ones take users to Highlight Nodes. Figure 3 presents the Concept Node’s page for an example concept, again demonstrating the options for navigation following the same key.

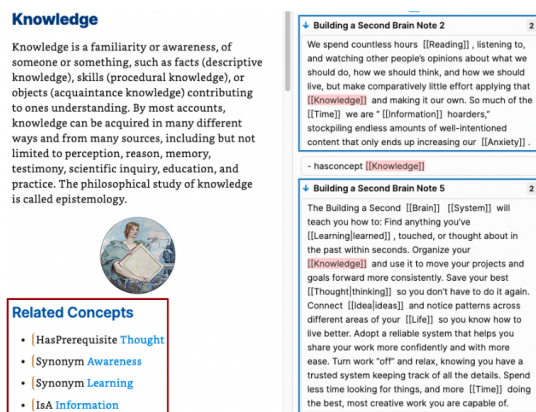


Figure 3: Example Concept Node page and hyperlinks

5 Evaluation

The objective of the evaluation is to analyze whether the methodology implementation successfully solves the defined problem. Two metrics were selected as being necessary to consider the system a successful implementation:

1. Accurate Graph Representation.
2. Coherence of Knowledge Connections.

To carry out the experiment, two different-sized subsets of the highlight collection were used. The original Text collection T was composed of several different books. The selected highlights H were book passages collected using the Kindle digital reader for each book. The subsets of the text collection were two highlights selections:

The *small test-set* had 52 highlights from two books, with connections that can be easily interpreted. This small test set makes it easier to analyze the fundamental functioning of the system in a limited scope, where the connections can be manually analyzed and interpreted.

The *medium test-set* had 182 highlights from eight books, with connections that were less obvious to interpret, but still under control for human interpretation. This test set was selected to interpret how the fundamental features of the system scale for a slightly larger highlights collection.

5.1 Accurate Graph Representation

Question 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. It simply checks if all the node and edge types proposed in the problem statement are actually present in the final structure of the interconnection graph G_H .

The note-taking tool Obsidian, which is used to navigate the generated connections, has a functionality called the graph view, where it is possible to visualize the generated graph. This view mode is further enhanced by an extension called Juggl, which makes it possible to include additional information to identify the node and edge types in the graph.

The chosen approach to answering this question is to look individually at the local graph views representing the three node types. The Highlight nodes are represented by blue circles, the Concept nodes are shown as red pentagons, and the Author nodes as green triangles.

The fact that it is possible to look at the three different node types individually is enough to understand that the three node types, Highlight, Concept, and Author, are present in the graph representation. The question then translates into determining if the 6 edge types defined in Equation 2 are present within the graph views for each node type.

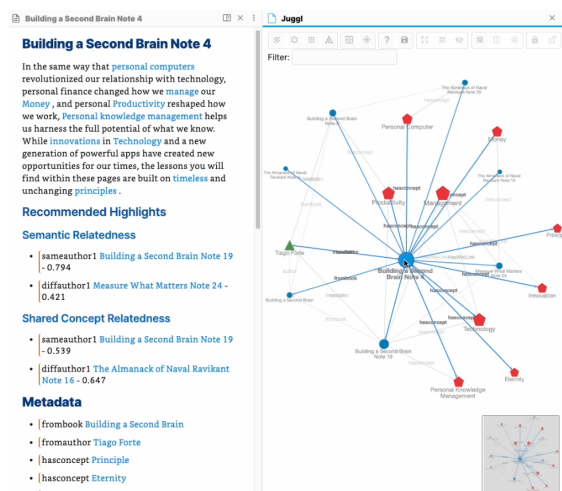


Figure 4: Local graph view for an example Highlight Node

When looking at the local graph view for the Highlight node of an example text, in Figure 4, it is possible to identify all the three edge types involving Highlight nodes. \overrightarrow{HH} , between blue circles, \overrightarrow{HC} , between blue circles and red pentagons and \overrightarrow{AH} , between the green triangle and blue circles.

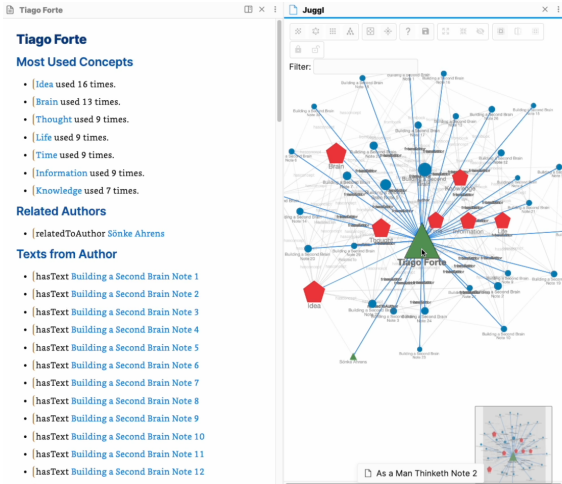


Figure 5: Local graph view for an example Author Node

When looking at the local graph view for the Author node of Tiago Forte, in Figure 5, it is possible to identify the three edge types that involve author nodes. \overrightarrow{AA} , between the green triangles, \overrightarrow{AC} , between the big green triangle and red pentagons, and once more \overrightarrow{AH} , between the big green triangle and blue circles.

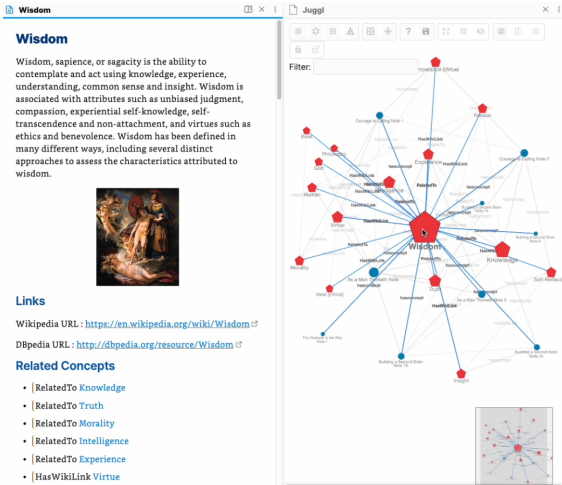


Figure 6: Local graph view for an example Concept Node

Finally, when looking at the local graph view for the Concept node of “Wisdom”, in Figure 6, it is possible to identify the final edge type to conclude that the implementation has all the proposed elements, the edges \overrightarrow{CC} , between the red pentagons. It is also possible to observe the edges \overrightarrow{HC} , between blue circles and red pentagons.

After analyzing the local graph views for each node type, it is possible to assert that all elements proposed in the problem statement are present in the implementation.

5.2 Coherence of Knowledge Connections

Question 2: What is the coherence between the two connection types? i.e. The coherence between the relatedness matrices generated for each method.

This evaluation determines if the generated knowledge connections are appropriate and serve their intended purpose.

The *Coherence between Connections* is defined as being the *coherence*, or *similarity*, between the two types of proposed connections, Shared Concepts (knowledge-based) and Semantic Relatedness (corpus-based).

The metric for coherence between the two types of connections is calculated by comparing their respective relatedness matrices using the Mantel Test (Mantel, 1967). The Mantel Test is a popular statistical test that returns a measure of the correlation between two matrices, ranging from -1 to 1.

The Mantel Test evaluates the similarity between the matrices. If the connections proposed by the Semantic Relatedness metric are similar to the ones proposed by the Shared Concepts metric, then they are coherent. A positive correlation between the two relatedness matrices implies that the two relatedness metrics are similar and, thus, coherent with one another.

A simple grid search is also performed when calculating the relatedness matrices, to find the best coherence while varying the parameter $\beta(m, e)$, the confidence threshold for concepts extraction.

Table 2: Grid Search of Coherence scores, while varying Confidence threshold for Entity Extraction

Confidence	Coherence Score	
	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

Following the proposed metric, all results in Table 2 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 positive correlation between the different connection types suggests that the methodology presented throughout this paper is valid. The generated connections are 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 provides more concepts to work with, which would increase the performance of the concept-based relatedness, as there is more information available.

Regardless of the reason for the coherence between the metrics improving, 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, because of the computational cost of running the system. More concepts mean a longer runtime to run the methodology. This implies that running the methodology with a lower confidence threshold could lead to very long runtimes for larger highlights collection.

6 Discussion

This section presents a discussion regarding the evaluation of the results and a reflection on the proposed methodology, with special attention to the potential use cases for this technology.

The proposed methodology was successfully represented in the functional version of the system, the final implementation successfully corresponds to the proposed graph representation, and the Connections can be considered to be coherent with one another.

6.1 Proposed Methodology Overview

Two questions are proposed to carry out additional reflections on the implementation of the methodology.

1. Can the combination of NLP with *Networked note-taking tools* improve the Knowledge Management functions of *Recall, Elaboration, and New Insight*?
2. Are Concept Nodes a useful mechanism for navigating a highlights collection?

The first question is directed at the inspirations and reasons for proposing the methodology and implementing it. The second question is directed at the

utility of introducing the Concept Nodes as means of connecting and enriching a highlights collection.

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.

The 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 across a wide range of options and eventually converging to the retrieval of the desired item(s). Elaboration presents different options for elaboration and eventually converges to one of the possibilities to elaborate on a given topic. New Insight is the most divergent of all three features, seeking mainly to be exposed to *new* ideas and possibilities that aren't previously known. The convergent aspect of insights is when two or more ideas connect and form a new piece of knowledge.

The proposed methodology was inspired by these three functions, which means it was designed to seek divergence and convergence in a balanced way.

The two Connection types used to navigate the highlights collection have different levels of divergence and convergence. The **Concepts Connections** are directed at *divergent thinking* by uniting related ideas that use common concepts, with the important feature of *not* discriminating between ideas according to the general topic of the highlights. The convergent aspect of the Shared Concepts Connections comes from connecting different highlights through a common concept between them.

On the other hand, the **Text Relatedness Connections** seek to initially promote *convergent thinking* by generating connections between highlights that are indeed related to each other. In turn, it also presents elements of divergent thinking, with the possibility of contrasting ideas from different authors.

The two modes of operation – divergence and convergence – are essential to the proposed system, as they play a part in both connection types and are designed into the system to simultaneously represent the three functions of Recall, Elaboration, and New Insight.

Whenever possible, the chosen priority for the system is to provide solid options for divergence and exploration instead of convergence and precision. The navigation mechanism provided by Concept Nodes is considered to be a key factor for this divergence and exploration.

The focus of the proposed methodology is to *provide the structure for semi-organized **divergent thinking*** while also providing the tools and context for the human user to bring forth convergence, according to their personal terms. This means humans are still responsible for interpreting connections and selecting the most relevant ones, something humans do inherently well.

The methodology promotes an *active role* for humans, with users actively cocreating connections and being an inherent part of the puzzle. This prevents users from being too passive and is an important aspect of the development of AI technologies.

6.2 Use Cases for the Proposed Technology

The methodology is supposed to be used by human users and depends directly on user input. It is expected that the user would deploy the methodology with a specific outcome in mind, in the format of a tangible project or even a specific reflection or contemplation of ideas.

The first category of use cases is applying the methodology for Personal Knowledge Management (PKM). The idea behind a PKM System is to store a person's past knowledge in a safe place to be later recycled and reused.

A potential combination would be integrating the interconnected highlights collection G_H into a given user's PKM System. This way, the user could access their highlights through the generated connections and use them as catalysts for retrieving and "*applying*" the knowledge contained in the highlights for several more granular use cases.

A clear use case for this combination would be *creation*. This could be creating a specific deliverable for work, a research project, and of course, the epitome of creation, *writing*.

Another use case within the overarching use case of Personal Knowledge Management is to help users *retrieve* any piece of knowledge related to specific texts, concepts, or topics in a fast and efficient way.

Another potential use case within PKM is *learning*. Generating connections automatically can help users compare a new piece of information with previously acquired knowledge in an easier, faster, and more powerful way.

A very interesting use case would be *organizing knowledge from different topics*. A specific example of this would be students organizing their notes across different disciplines using automatic connections between passages from digital Textbooks, online articles, and personal notes.

A generalization applied to any non-student would be to use the system to read and study multiple sources at once, while easily navigating between the ideas presented in them. By highlighting the passages that present important ideas that the user wishes to ponder about or study more deeply, the user would be able to receive an interconnected version with connections between the selected highlights.

The interconnected text collection may then be navigated, and most importantly, **edited and incremented**. Since the system is hosted inside a *note-taking app*, the user may easily use the software to create new notes and elaborate on the initially collected ideas.

A final, important use case for this technology is *connecting ideas from different people*. Using the system to connect and compare texts from a group of people is very aligned with what Doug Engelbart defines as the two most important aspects of the Collective IQ level, (Engelbart, 1992). First, the process – how well a group develops, integrates, and applies its knowledge. Second, the assets produced by that process – how effective the group's shared repository of knowledge is and how easily information can be synthesized, stored, retrieved and updated.

7 Conclusions

Recent advances in *Natural Language Processing* and *Personal Knowledge Management* present an opportunity to combine these two fields. Specifically, by providing Artificial-Intelligence-based features to note-taking tools, using the recently available functionalities as a starting point.

This paper proposes and successfully implements a methodology to automatically generate connections between highlights. The proposed methodology employs a combination of NLP tools to transform a given highlights collection, H , into an **interconnected and navigable** graph representing the same highlights, G_H .

Interconnectedness is present in the graph where highlights become Highlight nodes, and connections are added with multiple edge types and Concept Nodes. In turn, *navigation* is added to the text collection using Obsidian, a note-taking tool that combines the hierarchical organization of files and folders with the networked organization of bidirectional hyperlinks.

The results and evaluation suggest that the end result of the proposed methodology is an adequate representation of the proposed graph, G_H , in the Problem Statement, section 2. Finally, the two different paths for generating connections are coherent within themselves, which suggests that the connections generated by the system are reliable.

REFERENCES

- Ahrens, S. (2017). *How to take smart notes: One simple technique to boost writing, learning and thinking*. Sönke Ahrens.
- Anand, R. and Kotov, A. (2015). An empirical comparison of statistical term association graphs with dbpedia and conceptnet for query expansion. In *Proc. 7th forum for information retrieval evaluation*, pages 27–30.
- Becker, M., Korfhage, K., and Frank, A. (2021a). COCO-EX: A tool for linking concepts from texts to Concept-Net. In *Proc. 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 *Proc. 14th International Conference on Computational Semantics (IWCS)*, pages 21–32.
- 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.
- Canales, R. F. and Murillo, E. C. (2017). Evaluation of Entity Recognition Algorithms in Short Texts. *CLEI Electronic Journal*, 20(1):13.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 4171–4186.
- Engelbart, D. C. (1992). Toward high-performance organizations: A strategic role for groupware. In *Proc. GroupWare*, volume 92, pages 3–5. Citeseer.
- 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.
- 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.
- Gomaa, W. H. and Fahmy, A. A. (2013). A Survey of Text Similarity Approaches. *International Journal of Computer Applications*, 68(13):13–18.
- 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).
- Ilkou, E. (2022). Personal knowledge graphs: Use cases in e-learning platforms. In *Proc. WWW '22: Companion Proceedings of the Web Conference 2022*, page 344–348.
- Leal, J. P., Rodrigues, V., and Queirós, R. (2012). Computing semantic relatedness using dbpedia. In *Proc. 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.
- 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*, 34(1):50–70.
- Li, S. and Xu, E. (2020). Obsidian [Computer software].
- Mantel, N. (1967). The detection of disease clustering and a generalized regression approach. *Cancer research*, 27(2_Part.1):209–220. Publisher: AACR.
- Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011). DBpedia spotlight: shedding light on the web of documents. In *Proc. 7th International Conference on Semantic Systems*, pages 1–8.
- 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.
- 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 *Proc. 9th ACM International Conference on Web Search and Data Mining*, pages 635–644.
- 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.
- Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proc. 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992.
- Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proc. 14th International Joint Conference on Artificial Intelligence - Volume 1 - IJCAI'95*, page 448–453.
- SpazioDati, . (2012). Dandelion API | Semantic Text Analytics as a service.
- Speer, R., Chin, J., and Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proc. 31st AAAI Conference on Artificial Intelligence – AAAI'17*, page 4444–4451.
- Swinscow, T. D. V. and Campbell, M. J. (2002). *Statistics at square one*. Bmj London.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proc. 31st International Conference on Neural Information Processing Systems – NIPS'17*, page 6000–6010.
- Yazdani, M. and Popescu-Belis, A. (2013). Computing text semantic relatedness using the contents and links of a hypertext encyclopedia. *Artif. Intell.*, 194:176–202.