Semantic trajectory analysis for the prediction of the physical state of the collections at the BnF

Alaa Zreik

université PARIS-SACLAY

Semantic trajectory analysis for the prediction of the physical state of the collections at the BnF

Analyse de trajectoires sémantiques pour la prédiction de l'état physique des collections à la BnF

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, Sciences et Technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique
Graduate School : Informatique et sciences du numérique
Référent : Université de Versailles–Saint–Quentin–en–Yvelines

Thèse préparée dans l'unité de recherche **DAVID** (Université Paris–Saclay, UVSQ), sous la direction de **Mme Zoubida KEDAD**, Maîtresse de conférences

Thèse soutenue à Versailles, le 16 Janvier 2023, par

Alaa ZREIK

Composition du jury

Membres du jury avec voix délibérative

Nacera SEGHOUANI Professeure, Centrale Supelec, Université Paris–Saclay	Présidente
Amel BOUZEGHOUB Professeure, Telecom Sud Paris	Rapporteur & Examinatrice
Dimitris KOTZINOS Professeur, CY Cergy Paris Université	Rapporteur & Examinateur
Fayçal HAMDI Maître de conférences (HDR), Conservatoire National des Arts et Métiers	Examinateur

THESE DE DOCTORAT

NNT : 2023UPASG003

université PARIS-SACLAY | ÉCOLE DOCTORALE Sciences et technologies de l'information et de la communication (STIC)

Titre: Analyse de trajectoires sémantiques pour la prédiction de l'état physique des collections à la BnF

Mots clés: Représentation de trajectoires, Matching de trajectoires, Analyse de trajectoires, Graphe de connaissance, Pondération de concepts

Résumé: La Bibliothèque nationale de France (BnF) a pour mission de collecter, conserver, enrichir et communiquer le patrimoine documentaire national. Elle conserve près de quarante millions de documents. L'une des missions de la BnF est de maintenir les documents qui composent ses collections en bon état afin d'assurer leur disponibilité auprès des lecteurs. La définition d'une politique de conservation/restauration par les experts suppose l'identification des documents qui sont en mauvais état ; pour cela, l'état physique des documents doit être vérifié régulièrement afin d'identifier ceux qui nécessitent des interventions urgentes. Mais cette tâche très chronophage est impossible en pratique en raison du volume très important de documents. L'objectif de notre travail est de fournir un support aux experts dans la définition de leurs politiques de conservation/restauration, et de fournir un système d'aide à la décision permettant de caractériser l'état physique des documents par l'intégration et l'analyse des données disponibles dans les bases de données des différents départements de la BnF. En considérant que chaque document est décrit par un historique de conservation/restauration qui inclut toutes les informations susceptibles d'avoir un impact sur son état physique, les principales questions auxquelles nous sommes confrontés sont d'un part celle de la représentation de ces historiques et leur comparaison en tenant compte de leur hétérogénéité terminologique, d'autre part la définition d'un processus d'analyse de ces historiques permettant de caractériser l'état des documents et de le prédire. Notre travail vise à proposer des contributions pour un système d'aide à la décision pour des experts en conservation/restauration à la BnF. Nous avons proposé une représentation des historiques de conservation–restauration sous la forme de trajectoires sémantiques et nous avons introduit des mesures de similarité adaptées permettant de résoudre l'hétérogénéité terminologique des données en utilisant une base de connaissance externe, élaborée en collaboration avec les experts. Nous avons également défini un processus d'analyse fondé sur un algorithme de clustering afin de caractériser l'état physique des documents. Enfin, nous avons proposé une méthode originale de pondération des concepts qui permet de définir l'importance de ces derniers en considérant une tâche d'analyse spécifique.

université PARIS-SACLAY | ÉCOLE DOCTORALE Sciences et technologies de l'information et de la communication (STIC)

Title: Semantic trajectory analysis for the prediction of the physical state of the collections at the BnF
Keywords: Trajectory representation, Trajectory matching, Trajectory analysis, Knowledge graph, Concepts weighting

Abstract: The mission of the National Library of France (BnF) is to collect, preserve, enrich and make available the national documentary heritage. Its collections comprise nearly forty million documents.

One of the BnF's missions is to maintain the documents of its collections in good condition in order to ensure their availability to readers. The definition of a conservation/restoration policy by the experts requires the identification of the documents in poor condition; to this end, the physical state of the documents must be checked regularly to identify those requiring urgent interventions. But this time-consuming task is impossible in practice due to the large volume of documents.

The objective of our work is to provide a support to the experts in the definition of their conservation/restoration policies and to provide a decision support system allowing the characterization of the physical state of documents by the integration and analysis of the data available in the databases of the various departments of the BnF.

Considering that each document is described by a conservation–restoration history, which includes all the information likely to have an impact on its physical state, the main questions we are faced with are, on the one hand, the representation of these histories and their comparison taking into account their terminological heterogeneity, and on the other hand, the definition of an analysis process of these histories enabling to characterize the state of the documents and to predict it.

Our work aims to propose some contributions towards a decision support system for conservation/restoration experts at the BnF. We have proposed a representation of conservation–restoration histories as semantic trajectories, and we have proposed appropriate similarity measures to resolve the terminological heterogeneity of the data using an external knowledge base developed in collaboration with experts. We also have defined an analysis process based on a clustering algorithm to predict the documents' physical state. Finally, we have proposed a novel concept weighting approach that allows to define the importance of the concepts considering a specific analysis task.

Résumé substantiel en français

La Bibliothèque nationale de France (BnF) a pour mission de collecter, conserver, enrichir et communiquer le patrimoine documentaire national. Elle conserve près de quarante millions de documents.

L'une des missions de la BnF est de maintenir les documents qui composent ses collections en bon état afin d'assurer leur disponibilité auprès des lecteurs. La définition d'une politique de conservation/restauration par les experts suppose l'identification des documents qui sont en mauvais état ; pour cela, l'état physique des documents doit être vérifié régulièrement afin d'identifier ceux qui nécessitent des interventions urgentes. Mais cette tâche très chronophage est impossible en pratique en raison du volume très important de documents.

L'objectif de notre travail est de fournir un support aux experts dans la définition de leurs politiques de conservation/restauration, et de fournir un système d'aide à la décision permettant de caractériser l'état physique des documents par l'intégration et l'analyse des données disponibles dans les bases de données des différents départements de la BnF.

En considérant que chaque document est décrit par un historique de conservation–restauration qui inclut toutes les informations susceptibles d'avoir un impact sur son état physique, les principales questions auxquelles nous sommes confrontés sont d'un part celle de la représentation de ces historiques et leur comparaison en tenant compte de leur hétérogénéité terminologique, d'autre part la définition d'un processus d'analyse de ces historiques permettant de caractériser l'état des documents et de le prédire.

Notre travail vise à proposer des contributions pour un système d'aide à la décision pour des experts en conservation/restauration à la BnF. Nous avons proposé une représentation des historiques de conservation–restauration sous la forme de trajectoires sémantiques et nous avons introduit des mesures de similarité adaptées permettant de résoudre l'hétérogénéité terminologique des données en utilisant une base de connaissance externe, élaborée en collaboration avec les experts.

Nous avons également défini un processus d'analyse fondé sur un algorithme de clustering afin de prédire l'état physique des documents. Le processus d'analyse proposé est formé de différents modules afin de réaliser une telle prédiction. Il s'appuie sur le clustering de trajectoires visant à caractériser chaque classe de document, hors d'usage et communicable, par un ensemble de trajectoires représentatives. L'état physique d'un document est prédit en évaluant la similarité entre sa trajectoire et les trajectoires représentatives générées pour déterminer la classe du document.

Enfin, nous avons proposé une méthode originale de pondération des concepts qui permet de définir l'importance de ces derniers en considérant une tâche d'analyse spécifique. Dans une classification de trajectoire, certains événements peuvent avoir plus d'importance que d'autres dans la distinction entre les classes. Par conséquent, lors de l'analyse des trajectoires, il est important d'accorder un poids plus important aux événements à fort pouvoir discriminant lors de le processus d'analyse. Par exemple, de tels poids pourraient être pris en compte lors du calcul des scores de similarité entre trajectoires. Certains travaux se sont concentrés sur le contenu informationnel des concepts, et d'autres ont abordé le problème de la pondération des concepts représentés sous forme de structure hiérarchique. Dans ce travail, nous introduisons une nouvelle approche de pondération de concept qui prend en compte les classes prédéfini des trajectoires dans l'ensemble de données, correspondant à une tache d'analyse spécifique. L'approche transforme une ontologie en un réseau de neurones personnalisé où chaque nœud représente un concept de l'ontologie, et les relations entre les nœuds représentent les relations entre les concepts. Basée sur la régression, l'approche apprend les poids qui donnent la meilleure séparation des classes de trajectoires, et attribue des poids pour les concepts.

Acknowledgement

First of all, I would like to express my sincere gratitude to my thesis supervisor, Prof. Zoubida Kedad for the continuous support throughout this thesis, the insightful comments, and immense knowledge.
Besides my supervisor, I would like to thank the rest of my thesis committee. Specifically, I value the effort of Prof. Amel Bouzeghoub and Prof. Dimitrios Kotzinos in reviewing my dissertation. I truly appreciate their constructive feedback and for the time they have kindly devoted to it. I also deliver my sincere gratitude to Prof. Nacera Seghouani and Prof. Fayçal Hamdi, it is an honor having you as my jury member and having my work validated by you.
In addition, I would like to thank my friends and colleagues from the DAVID laboratory, and the BnF. We were not only able to support each other by deliberating over our problems and findings but also happily by talking about things other than just our papers.
Also, I would like to thank my family: my parents Ali and Louna, my sister Nour, and my brother Rida for supporting me spiritually throughout writing this thesis and my life in general. Everything I have reached is because of your continuous support and love.

Contents

List of Figures

List of Tables

1 - Introduction

Contents

1.1 Context and Motivation

The National Library of France (BnF) manages and maintains almost forty millions of documents in its collections. Some documents are exposed in the readers' halls, and the others are stored in the collection department. The documents are available as either hard copies or digital versions accessed through the BnF online services.
One of the key missions of the BnF is to define and implement the conservation–restoration policies to ensure that these documents remain in a good physical state. The physical state of the documents should be monitored continuously to ensure that they can be made available to the readers who request a hard copy; if not, the readers will be suggested to access a digital version. The documents that are in a bad physical state should undergo some conservation or restoration processes before any other communication to the readers.

In order to identify the documents that are in bad physical state at an early stage, it is essential to check their physical state continuously. This task is time-consuming and impossible in practice due to the large volume of documents. To facilitate the identification of the documents which are the most likely to become unavailable to the readers and which require some conservation–restoration operations, we aim to propose some contributions towards a decision support system that can predict a document's physical state by integrating and analysing the data stored in the BnF databases.

There are several departments at the BnF to manage the documents, the communication and the interventions that keep these documents in a good physical state. The collection department manages the storage, physical state checking and the communication of the documents to the readers. The conservation–restoration department manages the interventions on the documents to conserve them in a good physical state and consequently to keep them available to the readers who request them. These departments generate and store data in their local databases designed for their specific activities. For example, the collection department keeps track of all document communication requests and records the date, the reader, and other details. Additionally, in a separate database, the conservation and restoration department keeps

informations about how documents are treated and the degradations that have been observed.

The early identification of the documents which are likely to deteriorate would be beneficial to the experts to prioritize the ones which have to undergo some conservation processes in order to prevent further major degradations. Unfortunately, continuous physical state checking is impossible due to the volume of the documents.

Our goal is to propose a set of contributions towards a decision support system supporting the conservation/restoration experts, capable of integrating the relevant data, analysing it, and learning to predict the documents' physical state. The system will help the experts identify the documents that urgently need some conservation or restoration process. Different problems face the creation of such a system, such as the integration of the data, the analysis of the relevant data related to the documents' physical state, and the implementation of an analysis pipeline that uses machine learning algorithms to predict the physical state.

1.2 Challenges

Providing a support to the identification of documents requiring conservation–restoration processes at the BnF raises several problems. One issue is identifying and integrating the data related to the documents' physical state, which should be part of the analysis. For a given document, these data should include all the events that have occurred in the conservation–restoration history of this document, as well as any other events that might have an impact on the physical state of this document. One problem that has to be tackled is to find an appropriate representation for these data that can be used for for further analysis and for knowledge extraction. A critical task in any analysis is the ability to compare the elements on which the analysis will be performed. In our context, the elements are the conservation–restoration histories describing the documents.

Data on conservation–restoration histories may have been recorded for a few days or months but may also have been recorded decades ago. This leads to very heterogeneous terminologies in the description of the data. One of the problems to be addressed is to propose conservation history comparison tools that take into account this terminological heterogeneity when comparing these events and evaluating their similarity. Such similarity is necessary to identify the ones sharing similar histories to find some correlations or reasons for their physical state.

The effective prediction of the physical state of a document requires the design of the analysis pipeline suitable for deriving such prediction from the available date describing conservation–restoration histories. Besides selecting the most appropriate learning processes, this pipeline should enable the integration of domain knowledge, which integrates whenever possible to enhance the analysis. The design of such a pipeline is also a key challenge in our context.

Finally, the analysis pipeline could be enriched by weighting the events that constitute the conservation–restoration histories. The events that constitute the conservation–restoration

history might have different importance to the analysis objective. Therefore, giving importance to the events depending on their impact on the analysis results is another issue we aim to tackle in our work.

1.3 Contribution

In this work, our aim is to provide some contributions towards a decision support system for conservation–restoration experts at the BnF. The first part of our contributions is related to the generation, the representation, and the matching of conservation–restoration histories. We propose to represent these histories as semantic trajectories consisting of sequences of conservation–restoration events. In addition, we propose a trajectory-matching process that takes into account terminological heterogeneity and the semantic relationships between the elements composing the trajectories. The proposed similarity measure uses an external knowledge base representing the experts' knowledge to solve this heterogeneity. In order to take this knowledge into account during the matching, we have defined a set of semantic relationships between elements based on the structure of the knowledge graph. We have also proposed a specific knowledge graph describing the conservation–restoration concepts that are used in the BnF databases.

The second part of the contributions is related to the analysis of the conservation-restoration histories, represented as semantic trajectories. Numerous analysis pipelines have been proposed in existing works, with different goals, such as predicting the next element in a trajectory or classifying trajectories. Our aim is to design an analysis pipeline that enables the prediction of a document's physical state. One of the proposed analysis pipeline modules is a trajectory clustering module aiming to characterize both the class of the deteriorated documents and the class of available ones by a set of representative patterns. The physical state of a document is determined by assessing the similarity between its conservation-restoration trajectory and the generated patterns. The analysis process is based on a knowledge graph representing the domain experts' knowledge. We will show that adding more semantics into the analysis process improves the prediction results.

Finally, our last contribution is determining the importance of the elements composing the conservation–restoration trajectories for a specific analysis task. We propose a novel weighting approach that helps assign weights to the trajectory elements, where the weight is proportional to the element's importance for the considered analysis task.

1.4 Organization of the Manuscript

This manuscript consists of five chapters apart from this introduction.
Chapter 2 presents state-of-the-art related to the problems we have addressed. We survey the recent works on trajectory analysis, including their representation, similarity computation and trajectory data mining. We present the spatial and semantic trajectories and different

approaches to compute their similarity. In addition, we survey recent works on trajectory data mining, especially trajectory filtering, clustering, classification and prediction. We also present recent works integrating knowledge graphs in the similarity computation, as well as some approaches aiming to determine the weights of a set of concepts, representing their importance in a given context. We also present some of the existing knowledge graphs in the cultural heritage field represented as ontologies.

Chapter 3 deals with our semantic trajectory matching process. The process consists of integrating the experts' knowledge to fill the terminological gap between the elements composing the trajectories. We present the analysis of the BnF databases and the creation of the semantic trajectories. In addition, we present our proposed similarity measure to compute the similarity between the semantic trajectories. Finally, an experimental evaluation is provided to compare the matching results with and without the external source of knowledge.

Chapter 4 presents an analysis pipeline to predict the documents' physical state. We present the trajectory clustering process based on the proposed similarity measure. In addition, a filtering and prediction rules creation process is discussed. Furthermore, we present the creation of the prediction model that predict the documents' physical state. Finally, an experimental evaluation is provided to compare the prediction results with and without using the experts' knowledge represented by the knowledge graph.

In chapter 5, we present a novel weighting method that gives weights for the trajectory elements based on the considered analysis task. It is based on a neural network representation of the concepts hierarchy. We present the automatic transformation of the hierarchy into a customized neural network. In addition, we present the normalization of the neural network. Finally, we discuss the weights learning using forward and backward propagation.

Finally, we provide a conclusion in chapter 6, where we summarise our contributions and show how our proposal can help analyse semantic trajectories. We discuss the open problems and present some future works.

2 - State of the art

Contents

2.1 Introduction

In many applications domain, the data are represented as trajectories. For instance, the trajectories used in human movement mining could be represented as sequences of time-stamped locations that show how people move. In the healthcare field, a trajectory could be a sequence of events that indicates a patient's medical history. These trajectories may represent a huge volume of information that can not be analysed manually. In order to learn and extract some knowledge from a massive trajectory dataset, many works have proposed analysis pipelines relying on machine learning algorithms.

In our context, we are interested in analysing the conservation–restoration histories of the documents stored at the BnF, which could be represented as semantic trajectories. For this reason, we are interested in developing a pipeline that analyses these trajectories. Therefore, we present different works on trajectory analysis related to the tasks that will be part of our analysis pipeline, such as the similarity measures and the mining approaches. One important problem is the formalization of trajectory data, which consists in finding the best representation for the trajectories. Another one is comparing the trajectories because the ability to compute the similarity between them is crucial for further analysis. In addition, the application of data mining algorithms to analyse them is used in different contexts.

In addition, we are interested in resolving the terminological heterogeneity of the conservation–restoration histories during the analysis, which may decrease the accuracy of the analysis pipeline. Therefore, to resolve the terminological heterogeneity and improve the analysis, we will present approaches that use external knowledge during the analysis and specifically to compute the similarity between semantic concepts.
Moreover, we are interested in the weighting approaches that refine the analysis by identifying the importance of concepts. Such weighting could increase the performance by focusing on the important concepts. The weights can be integrated into the similarity computation process, where more priority can be given to concepts with higher weights.

Furthermore, one possible way to represent knowledge is to use knowledge-base graphs that contain nodes representing the concepts, i.e., events, to describe where the edges between them represent the relationships. One way to express knowledge in a domain is ontologies, and many ontologies in the cultural heritage field are available and describe concepts and their semantic relationships. We will present these ontologies and discuss their suitability for our context and objectives.

In this chapter, we will study some of the approaches that have been proposed in the semantic-based data mining field, especially for trajectory data. In the first part, presented in section 2.2, we study different aspects related to the trajectory analysis. Mainly, we will focus on the representation of the trajectories where we distinguish between the spatial and semantic trajectories, the similarity measures used on trajectories, and the trajectory data mining, which

includes trajectory clustering, predicting and classification. The second part is devoted to knowledge-based similarity computation and presented in section 2.3. We will present the related works and the challenges raised by this problem. In the third part, presented in section 2.4, we will present the ontologies in the cultural heritage and the conservation-restoration fields and their limitations to be used in our context. Finally, we will provide a conclusion in section 2.5 highlighting the open problems.

2.2 Trajectory Analysis

Trajectory analysis aims to extract some meaningful knowledge from trajectory data. For example, this knowledge could be a label for each trajectory, the identification of unusual ones, extracting patterns characterizing similar ones, or the prediction of their future elements to cite a few. Many machine learning algorithms could be used in the analysis of the trajectories such as clustering and classification.

Trajectories can be classified into two categories, the spatial trajectories and the semantic ones. Spatial trajectories are sequences of geographical coordinates. Semantic trajectories are sequences of events or semantic elements. Spatial trajectories are sequences of time-stamped places that are sorted according to their time [129]. For example, a trajectory $t = [< (long\_1, lat\_1), 8am >, < (long\_2, lat\_2),\ 10am >, < (long\_3, lat\_3), 5pm >]$ can be used to describe the movement of a person who was in $(long\_1, lat\_1)$ at 8am, $(long\_2, lat\_2)$ at 10am, and $(long\_3, lat\_3)$ at 5pm. A semantic trajectory is one in which the elements are not geographical coordinates. For example, the trajectory t could be represented as a sequence of semantic elements $t = [< home, 8am >, < work,\ 10am >, < restaurant, 5pm >]$.

The approaches proposed to analyse trajectories rely on different tasks such as their preprocessing, the computation of the similarity between them, the goal of the analysis and the appropriate machine learning method. In order analyse trajectories, one problem is to find the best representation of the elements that constitute the trajectories, which could take the form of sequences of locations if the trajectories represent spatial data or events if the trajectories are semantic. Another problem is to find a way to calculate the similarity between the trajectories and their elements. Finally, the best machine learning algorithm should be selected to perform the analysis task at hand.

In the sequel, we will provide a broad overview of trajectory representation in section 2.2.1 where we distinguish between the representation of spatial and semantic trajectories. Section 2.2.2 deals with the similarity evaluation between the trajectories where we discuss the measures used in the fields where the data are represented as sequences or strings. Some of the existing knowledge extraction and learning algorithms are discussed in section 2.2.3, such as the trajectory filtering, clustering, prediction and classification. Finally we provide a discussion in section 2.2.4.

2.2.1 Trajectory Representation

In this part we give an overview of fields where data are represented as trajectories and highlight their different types, which are the spatial trajectories and the semantic ones. In section 2.2.1.1 we present how some works represent the data as spatial trajectories. In section 2.2.1.2 we present some works that represent the data as semantic trajectories, which sometimes are considered as sequences of events.

2.2.1.1 Spatial Trajectories

The representation of spatial data generated in various fields such as maritime traffic [88] and people movement analysis [11] is the first fundamental step in analyzing this data. Most of the analysis works represent this data as spatial trajectories, i.e. sequences of locations ordered by their timestamp.
Spatial trajectories could also be enriched with more information that helps with the analysis task. A spatial trajectory is represented as $Tr = e_1,\ e_2,\ ...,\ e_{len_{TR}}$ and what differs between the analysis works is the representation of the elements e_i in Tr and the information they contain.

The authors in [58] represent the trajectories as sequences of multi-dimensional points. For example, a trajectory TR_i is represented by $TR_i = p_1 p_2 ... p_{len_i}$, where p_j is a d-dimensional point with $1 \leq j \leq len_i$, and len_i is the length of the trajectory. In addition, the segments are clustered in order to discover the common sub-trajectories among the initial set of trajectories.

Similarly in [51], the trajectories are also represented as a sequence of spatio-temporal points. A trajectory is defined as a list $T = \{tp_0, tp_1, ..., tp_n\}$, with $tp_i = (x_i, y_i, t_i, w_i)$; each point is characterized by (x_i, y_i) coordinates, a timestamp t_i and a set of numerical features w_i. In this work, trajectories are partitioned into segments, i.e. sub-trajectories, and each segment is enriched with a semantic label and set of segment features.

The works presented in [97] and [79] enrich the spatial trajectories with the speed information. [97] represents a trajectory as a list of spatio-temporal points $< p_0, p_1, ..., p_n >$, where $p_i = (x_i, y_i, t_i)$ and $x_i,\ y_i,\ t_i \in \mathbb{R}$, and partitions it in smaller pieces called stops and moves using the speed feature. The stops are explicitly defined by the user and the moves are the sequences between two stops, or between t_0 and the first stop, or the last stop and t_n, or $[t_0, t_n]$. The authors in [79] propose to further partition the stops and moves into sub-stops and sub-moves in order to enrich more the trajectories.

2.2.1.2 Semantic Trajectories

Semantic trajectories are used in several works to refer to temporally ordered sequences of time-stamped semantic elements. The elements that constitute these sequences could be, for example, semantic locations or events. Other works call these sequences simply semantic sequences, and when the elements forming them are events, some works refer to them as event sequences [15][67][45].

In the healthcare field, the approach presented in [86] represents each patient by a temporal trajectory where the events are healthcare actions on the patient. Consider that ε is the set of all the events, an event $e \in \varepsilon$ is a tuple $e =< p, action, time >$ that associates the patient p to the action at a specific time. For example, $e_1 =< p_1, BMI\ obese, 05/2017 >$ means that patient p_1 had a BMI in May 2017 categorizing him as obese. The trajectory of a patient p is defined as a sequence of temporally sorted events $T(p) = \{< p, action, time > \in \varepsilon\}$.
[52] proposes the representation of semantic knowledge on paths by adding additional semantic layers. [10] transforms the geotagged data into semantic trajectories and represents them by sequences of location types such as restaurant or park $SemT =< type_1, type_2, ..., type_n >$. [56] analyses the maritime traffic and uses the raw data such as the speed and the position to create a semantic layer of two types of nodes called way-points and traversals.

Another field where the representation is very similar to the semantic trajectories is where the data is sequences of events. This data can be found in various applications and fields such as healthcare, marketing analytics and advertising. For example, in Electronic Health Records, the events could be the patients' diagnosis, procedure, and medication codes. Different ways exist to represent the event sequences. The survey presented in [38] identifies five categories of visual representation: chart-based visualization [69], timeline-based visualization [83], hierarchy-based visualization [119], sankey-based visualization [96] and matrix-based visualization [80].

2.2.2 Trajectory Similarity Measures

Different similarity measures have been proposed to calculate the similarity or the distance between sequences. These measures can also be extended and used to calculate the similarity between spatio-temporal trajectories or semantic ones. In the survey presented in [108], the authors classify the trajectory distance measures depending on different criteria.

The similarity of spatio-temporal trajectories depends on the geographical proximity between the points composing them. This does not apply to the semantic trajectories presented in the section 2.2.1.2. This is because the distance between the elements which constitute the semantic trajectories can not be calculated using the distance metrics applied for spatio-temporal ones, such as the euclidean distance for example.

In this section, we survey some of the existing similarity measures and their applicability for spatio-temporal trajectories or semantic ones. The measures are classified into two categories: sequences similarity measures (presented in section 2.2.2.1) and string similarity measures (presented in section 2.2.2.2). The string similarity measures are compared to the sequences similarity measures as the strings can be considered as sequences of characters.

2.2.2.1 Sequence Similarity Measures

Sequence similarity measures have been proposed and used for both spatial trajectories and semantic ones. Recall that in a spatial trajectory, each item represents a geographical location. In a semantic trajectory, each item can be either an event or a semantic element.
The authors in [86] [85] define a similarity measure to calculate the similarity between semantic trajectories representing the patients' healthcare histories. For example, figure 2.1 shows two

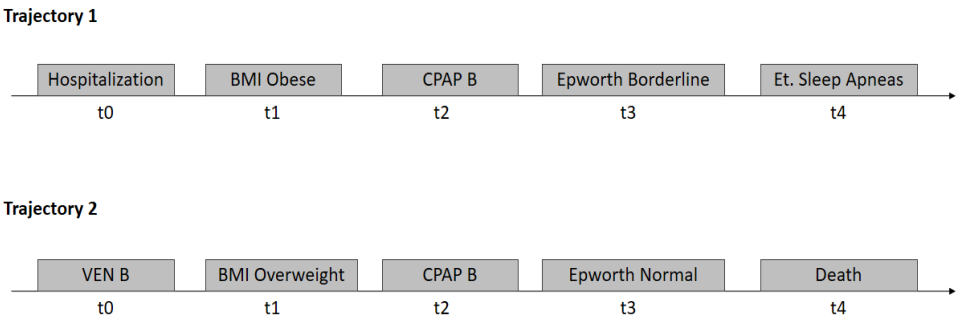


Figure 2.1: Healthcare Trajectories

healthcare semantic trajectories, where the elements of the trajectories are healthcare events ordered by their time. The similarity measure searches for the matches between the healthcare events and compute the similarity between two trajectories depending on the number of matches. Once the similarity is computed between the trajectories the authors propose to create groups containing similar trajectories called "cohorts" using clustering. Two events are considered as matching events if they have the same name, and their similarity score depends on their associated time and the length of the trajectories to which they belong. The authors in [115] compare the similarity measures in event-based data, i.e. data containing sequences of events, by defining nine features to characterize the event sequences similarity measures and assess the suitability of these measures to a given context. The authors highlight the necessity of carefully defining what an event-sequence is because the applicability of the measures may vary accordingly.

In the survey presented in [108], the authors classify the trajectory similarity measures depending on the considered data type according to two criteria: (i) the discrete or continuous nature of the data and (ii) the existence of a temporal dimension for the data.

A trajectory is considered discrete if there is no movement between two consecutive items. For example, a semantic trajectory composed of events fits this case as there is no link or path between two events. When a trajectory is discrete, it contains a finite number of items. The second criterion is whether the measure compares the ordered items in the trajectories using only the spatial attribute or both spatial and temporal information. In other words, the second criterion is whether the measure uses the temporal information in the calculation or not. Distance measures that only depend on the spatial distance between trajectories are called sequence-only distance measures. Distance measures that apply both the spatial and temporal distance between trajectories are called spatial-temporal measures.

2.2.2.1.1 Discrete and Sequence-only The first category corresponds to discrete sequences. The measures in this category depends only on the spatial dimension. In addition, the compared sequences contain a finite number of elements. Examples of measures in this category are the Euclidean Distance (**ED**) [63], the Longest Common SubSequence (**LCSS**) [87] and the Edit Distance on Real sequence (**EDR**) [75]. The EDR and the LCSS measures are related, as the LCSS calculates the length of the longest common sub-sequence, and the

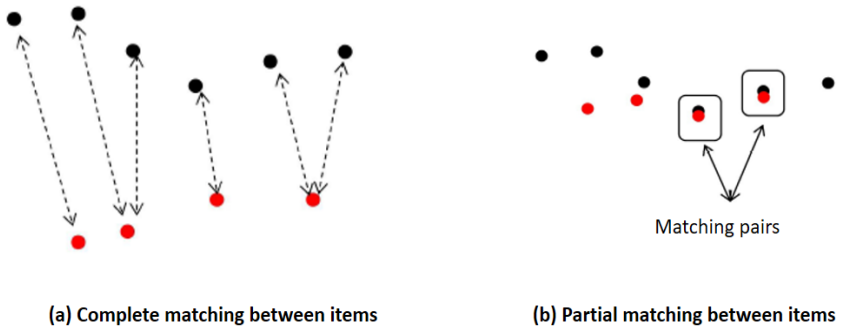


Figure 2.2: Comparison between complete matching and partial matching

edit distance calculates the number of changes needed between the compared sequences. Measures in this category can be divided into two types, the ones that provide complete match and the ones that provide a partial match.

For two compared sequences s_i and s_j, complete match strategy requires every sample elements of s_i and s_j should be in a match pair. On the contrary, partial match strategy does not require every sample elements of s_i and s_j should be in a match pair.

Figure 2.2 shows two examples of complete (a) and partial (b) matching. We can see that all of the elements belong to at least one match pair when computing the similarity between the two sequences using complete matching. On the contrary, using partial matching, we can see that some items are not matched.

Euclidean distance (ED). The Euclidean distance is the most commonly used distance measure, also known as $L_2 - norm$. To calculate the distance between two sequences s_i and s_j having the same length n, the distance between every pair of elements at the same position in the two sequences is computed. In this case, the complexity of this process is $O(n)$.
In the case of two sequences with two different lengths, the ED measure is used with a sliding window having a size equal to the shortest length, and the complexity is $O(mn)$ where m is the difference between the lengths of the two sequences and n is the length of the shortest one.

$$d_{Euclidean}(T_i, T_j) = \frac{\sum_{i=1}^{n} d(p_{1,i}, p_{2,i})}{n} \tag{2.1}$$

$$d_{Euclidean}(T_i, T_j) = \min_{j=1}^{m-n+1} \frac{\sum_{i=1}^{n} d(p_{1,i}, p_{2,j+i})}{n} \tag{2.2}$$

Equations 2.1 and 2.2 show the euclidean distance definition when the compared trajectories or sequences have the same size and different sizes, respectively.

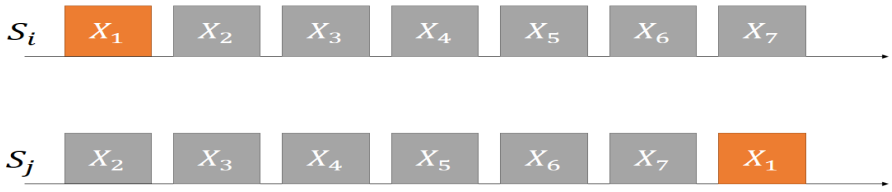


Figure 2.3: Shifting effect on the ED measure

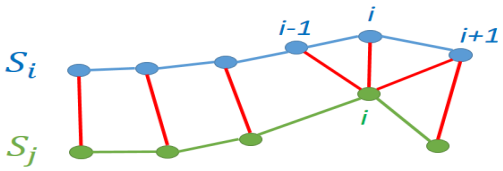


Figure 2.4: DTW between two sequences

The euclidean distance is a complete match measure. If the two compared sequences have the same length, every matching pair will contain the elements that have the same index (i.e. position). If the lengths are different, the complete match requires that all the elements from the smallest sequence should be in at least one match pair. A limitation of the algorithm is that one shifting can change the similarity value completely. Figure 2.3 shows the effect of one shifting on the similarity value between two sequences s_i and s_j where the elements are the same but in different positions and the ED between the two sequences will be minimal, e.g. the similarity equal to 0 although the two sequences have a common sub-sequence [x_2, ..., x_7] of length equal to six.

Dynamic time warping (DTW). The DTW [81] is a complete match measure. The difference with ED is that an element in the first sequence can be matched to more than one element from the second sequence. Figure 2.4 shows an example of DTW between two sequences s_i and s_j where the elements at the positions i-1, i and i+1 from s_i respectively are matched with the nearest element from s_j which is at position i. Equation 2.3 shows the definition of the DTW measure. The DTW can be extended to be a partial match where the number of events matters, and for every skipped event, i.e. for each event without a match, a maximal distance α can be added.

$$\mathrm{d}_{DTW}(T_i, T_j) = \begin{cases} 0, & \text{if } n = 0 \; and \; m = 0 \\ \infty, & \text{if } n = 0 \; or \; m = 0 \\ d(Head(T_i), Head(T_j)) + min\{d_{DTW}(T_i, Rest(T_j)), & \\ d_{DTW}(Rest(T_i), T_j), d_{DTW}(Rest(T_i), Rest(T_j))\} & \text{otherwise} \end{cases} \tag{2.3}$$

Piecewise dynamic time warping (PDTW). The PDTW [54] is an extension of the DTW, which is also a complete match. The goal is to reduce the algorithm's complexity by merging each set of sequential elements and representing them by their mean if it can be calculated.

This measure could be used for spatial trajectories but not for semantic ones.
Consider two spatial trajectories $t_i = [l_1, l_2, l_3, ...l_{len_i}]$ and $t_j = [l'_1, l'_2, l'_3, ...l'_{len_j}]$ represented as sequences of geospatial points. For example, if the merge is performed each three consecutive points, the two trajectories will start by mean(l_1, l_2, l_3) and mean(l'_1, l'_2, l'_3) respectively. During trajectory analysis, the evaluation of the similarity will be performed by comparing the means of the elements instead of the elements themselves.
Event merging cannot be performed in semantic trajectories or any other context in which the mean of a set of elements cannot be calculated or is meaningless. For example, given a trajectory of healthcare events $T = [< BMIObese, t_1 >, < HospitalizationE, t_2 >]$, it is impossible to calculate the mean of these events, and therefore, the PDTW measure cannot be applied.

Longest common subsequence (LCSS). The LCSS [87] is a partial match measure. It is used to calculate the longest common sub-sequence. When using the algorithm on spatial trajectories, two elements, i.e. locations on two different trajectories, are considered as matching when their distance is less than a threshold ϵ. LCSS can be used with event sequences by considering that each matching pair should contain events with the same name from the two different sequences. The order of the elements is essential when using the LCSS measure, while the index, i.e. the position of the elements in their respective trajectories, is not. In other words, two elements can be matched regardless of their position, but they can not be matched if the algorithm has already matched elements with a higher index in the sequence.
Equation 2.4 shows the LCSS distance, and equation 2.5 shows the definition of LCSS measure.

$$d_{LCSS}(T_i, T_j) = 1 - \frac{S_{LCSS}(T_i, T_j)}{size(T_i) + size(T_j) - S_{LCSS}(T_i, T_j)} \tag{2.4}$$

$$\mathrm{S}_{LCSS}(T_i, T_j) = \begin{cases} \emptyset, & \text{if } l_i = 0 \text{ or } l_j = 0 \\ S_{LCSS}(Rest(T_i), Rest(T_j)) + 1, & \text{if } d(H(T_i), H(T_j)) \leq \epsilon \\ max\{S_{LCSS}(T_i, Rest(T_j)), S_{LCSS}(Rest(T_i), T_j)\}, & \text{otherwise} \end{cases} \tag{2.5}$$

With l_i and l_j representing the length of T_i and T_j respectively. The H function returns the first element in the trajectory, and the Rest function returns the trajectory without its first element. For example, given two sequences s_i=[X_1, X_2, X_3, Y_1, X_4, X_5, X_6, X_7] and s_j=[X_1, X_2, X_3, Y_2, X_4, X_5, X_6, X_7], the longest sub-sequence will be s=[X_1, X_2, X_3, X_4, X_5, X_6, X_7], the longest common sub-sequence length will be $S_{LCSS}(s_i, s_j) = 7$, and the distance equal to $1 - (7/8)$.

Edit distance on real sequence (EDR). The edit distance [75] is a string metric for calculating the extent to which two sequences differ. It is a partial match measure defined as the minimal number of edits needed to transform one string into another and it can be used for events sequences. The considered edit operations are the insertion, the deletion and the

substitution. For every unmatching pair of elements, i.e. elements having a distance higher than a predefined threshold, the required edit operations are determined to transform one element to the other. The cost of each operation is equal to one, and the distance is computed as the sum of the costs of the transformation operations. For example, consider the three healthcare trajectories T_1, T_2 and T_3, with their respective events sequences T_1=[<BMI Obese, t_1>, <chronic bronchitis, t_2>], T_2=[<chronic bronchitis, t_1>] and T_3=[<BMI Overweight, t_1>, <chronic bronchitis, t_2>]. The transformation of T_2 to match T_3 needs one *insert* operation, but the transformation of T_3 to match T_1 requires one *substitution*. The cost of each of the two transformations is equal to one. But we can see that T_3 is more similar to T_1 than T_2, because T_3 and T_1 share a BMI element, and should therefore have a smaller distance to T_1.

Edit distance with real penalty (EPR). EPR [14] is an extension of EDR, but it does count not only the number of required operations but also combines Lp-norm and Edit Distance. After each operation, the distance between each unmatched pair will be taken into consideration, and unlike EDR, the cost may vary between one operation and another. This measure is best suited for semantic trajectories if the similarity between the elements could be calculated. For example, consider the three trajectories T_1=[<BMI Obese, t_1>, <chronic bronchitis, t_2>], T_2=[<chronic bronchitis, t_1>] and T_3=[<BMI Overweight, t_1>, <chronic bronchitis, t_2>]. The distance between T_1 and T_3 should be less than the distance between T_1 and T_2. The difference between T_1 and T_3 is the first element, which is somehow similar but not identical. If the cost of the *substitution* operation is less than the cost of the *insert* operation, this will lead to a smaller distance between T_1 and T_3 than T_1 and T_2.

2.2.2.1.2 Discrete and spatial-temporal

After presenting the discrete measures which depend only on the spatial dimension, we will now present the second category which corresponds to the discrete and spatial-temporal measures.

Some similarity measures in this category use both the temporal and the spatial data to calculate the similarity; other measures calculate two distinct similarity values, one of which depends on the spatial information, and another depends on the time dimension. Some of the measures in this category is the Spatio-Temporal LCSS (**STLCSS**) introduced by [114] and used in [109], and the Spatio-Temporal Linear Combine Distance (STLC).

Spatio-Temporal LCSS (STLCSS). STLCSS [114] is an extension of the LCSS measure which has been created for string matching and not for timestamped data. This extension takes into consideration the location and the time of each item to calculate the similarities. Two items are considered the same if the distance between them and the time difference between them are less than two thresholds ϵ and δ respectively. Equation 2.6 shows the definition of STLCSS.

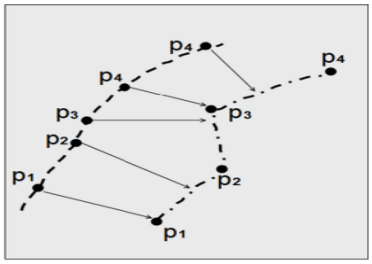


Figure 2.5: Example of OWD projection between two sequences

$$\mathsf{S}_{STLCSS}(T_i, T_j) = \begin{cases} 0, & \text{if } Tr_i \text{ or } Tr_j \text{ is empty} \\ 1 + S_{STLCSS}(Rest(T_i), Rest(T_j)), & \text{if } \mid \text{Head}(T_i).\text{x-Head}(T_j).\text{x} \mid < \epsilon \text{ and} \\ & \mid \text{Head}(T_i).\text{y-Head}(T_j).\text{y} \mid < \epsilon \text{ and} \\ & \mid \text{Head}(T_i).\text{t-Head}(T_j).\text{t} \mid \leq \delta \\ max\{S_{STLCSS}(T_i, Rest(T_j)), & \\ S_{STLCSS}(Rest(T_i), T_j)\}, & \text{otherwise} \end{cases} \tag{2.6}$$

Spatio-Temporal Linear Combine Distance (STLC). This measure combines the spatial distance and the temporal distance between the compared trajectories. The algorithm calculates separately the spatial and temporal dimensions. It is possible in some contexts one dimension is more important than another. Thus, they can not be treated equally. Similarly to the DTW, an element from a trajectory can be matched to more than one element from another trajectory.

2.2.2.1.3 Continuous Trajectory Distance Measures In this category, the shape between elements matters when computing the similarity between trajectories. The shape between the elements, if they represent locations, can be the road that connects them. The measures in this category deal with continuous trajectories, with or without considering the time dimension. These measures can be used on spatial trajectories. Some of the measures in this category are the Edit Distance with projections (EDwP) [93], the One Way Distance (OWD) [64] and the Spatio-Temporal Locality In-between Polyline distance (STLIP). However, in the context of events or semantic trajectories, it is challenging to define a shape between two events.

The OWD is a continuous sequence-only distance measure. Consider two trajectories T_1 and T_2. When searching the minimum distance between an element in T_1 and T_2 in order to find a match for the element, it is possible that the minimum distance is not with an actual element from T_2. It can be a projection on the shape between two elements. For example, the figure 2.5 shows the OWD projection of a sequence S_1 to another sequence S_2 to calculate their distance. We can see that the projection of p_4 is S_1 is not to an actual element in S_2, but it is on the linear representation between p_3 and p_4 of S_2. The linear representation of a sequence is not possible when the trajectories are semantic.

2.2.2.2 String Similarity Measures

Several string matching algorithms have been proposed to calculate the similarity between two strings [113]. Some of them can be extended to be used for trajectories or semantic sequences such as the edit-based, token-based, and sequence-based similarity measures. Other algorithms can not be used for trajectories. For example when the algorithm rely on the string sound when spoken, or only on one subset of the string, such as its prefix and postfix. In this section, we focus on the categories that can be extended and used for trajectories.

2.2.2.2.1 Edit-based Measures

When computing the similarity between two strings, the edit-based measure counts the number of modifications required to transform one string to another. The simplest distance in this category is the Hamming distance, which allows only substitution operations when two strings have the same length, and it only enables the insertion at the end of a string when the lengths are not equal.

The Hamming distance [99] compares the characters at the same index. If the two compared characters are not the same, the edit cost increases by one, and one of the characters will be substituted to be identical.

The second distance in this category is the Levenshtein distance [74], where more operations are considered. The Levenshtein distance allows the deletion, insertion and substitution operations, and the cost of each operation is equal to one. The Damerau-Levenshtein distance [73] is an extension of the Levenshtein distance which allows one additional operation, the transposition of two adjacent characters. The new operation can reduce the number of operation required. For example, given two strings s_i='xy' and s_j='yx', the Levenshtein distance is equal to two because two operations are required, which are the deletion of 'x' from the sequence s_i and the insertion of 'x' at the end of this sequence, but the Damerau-Levenshtein distance will be equal to one as only one transposition operation is required.

Jaro distance [47] is another distance in this category that matches character at different indexes in two strings within a limited range defined as max($|s_i|$, $|s_j|$)$/2+1$, where $|s_i|$ and $|s_j|$ are the lengths of the two strings s_i and s_j respectively. Equation 2.7 shows the definition of the Jaro distance between two string s_x and s_y, where m is the number of matching characters and t is half of the number of transpositions.

The Jaro-Winkler distance [117] is an extension of the Jaro distance. It gives more importance to the first characters of the string, i.e. its prefix of the string. Equation 2.8 shows the definition of Jaro-Winkler distance, where l is the length of the common prefix (max to 4), and p have a standard value equal to 0.1.

$$d_j = \frac{1}{3}\left(\frac{m}{|s_x|} + \frac{m}{|s_y|} + \frac{m-t}{m}\right) \tag{2.7}$$

$$d_w = d_j + (lp(1-d_j)) \tag{2.8}$$

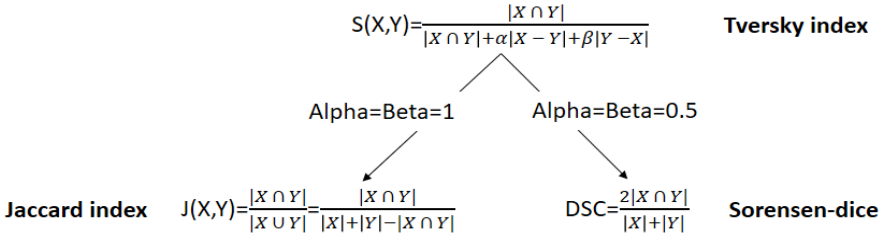


Figure 2.6: Relations between some Token based distances

2.2.2.2.2 Token-based Similarity

The distance measures belonging to this category depend on the intersections and the unions between two strings, i.e. the common characters and the total number of distinct characters in both strings. The intersection of two strings is the common characters between them regardless of their index. The Tversky index [112] is token-based, and generalises both the Jaccard index [41] and the Sorensen-dice index [20]. Figure 2.6 shows the definition of the Jaccard, Sorensen-dice and Tversky index distances and the relations between them, where X and Y two sets of elements. The Jaccard index is a specialisation of the Tversky index with α and β equal to one, and the Sorensen-dice is a specialisation with α and β equal to 0.5.

Another token based distance is the overlap coefficient, that measure the overlap between two finite sets. Equation 2.9 shows its definition, where X and Y represent two strings.

$$overlap(X, Y) = \frac{|X \cap Y|}{min(|X|, |Y|)} \tag{2.9}$$

2.2.2.2.3 Sequence-based Similarity

The distance measures in this category consider the order of the shared items in the strings. One of these measures is Ratcliff/Obershelp [94], which computes the number of matching characters multiplied by two and divided by the total number of characters of both strings. It starts by searching the longest common substring (LCS) and then searches the next longest sub-string on both sides of LCS.

The longest common substring and the longest common sub-sequence are very similar. The only difference between the two measures is that the gap between the common characters in the longest common substring is not allowed. In other words, the common sub-string should contain a sequence of consecutive characters.

2.2.2.3 Comparison of the Similarity Measures

As we have seen in the previous sections, many measures have been proposed to compute the similarity between either sequences or strings. These two categories of measures have some similarities because a string can also be considered as a sequence of elements. In order to analyse these measures and find the most suitable one to be used in our context we compare

the measures using the following five characteristics:

Position dependence. This characteristic indicates whether the matching between the items from different sequences depends on their positions, i.e. index. For example, using the euclidean distance, matching items always have the same index in the two compared sequences. On the contrary, using the LCSS measure, matching items can have different indexes.

Time dependence. This characteristic indicates if the matching between the items from different sequences depends on the temporal dimension. A measure that depends on the temporal dimension may consider two items are matching only if the difference between their timestamps is less than a predefined threshold. For example, using the STLC measure, two items with a high difference in the temporal dimension can not be matched, and therefore this measure depends on the time.

Order dependence. This characteristic indicates if the items' matching can intersect. For example, consider two sequences S_1=[p_1, p_2] and S_2=[p_2, p_1]. If the similarity measure does not depend on the order, the similarity between the two sequences will be equal to two as the sequences contain the same items. On the contrary if the measure depends on the order, it will select only one match, i.e. if p_1 is selected from the two sequences to be the first match, p_2 can not be selected as its position in S_2 is before p_1.

Partial and Total match. This characteristic represents, as mentioned in section 2.2.2.1 if the items of a sequence should be at least in one matching pair. Using a partial match, it is possible that an item does not belong to any matching pair. On contrary, a complete match requires that each item belongs at least to one matching pair.

Multiple and Single match. This characteristic indicates if an item of a sequence can have more than one match. A multiple match is similar to 1 to n relationship where one item can be matched to n items from another sequence. A single match is similar to 1 to 1 relationship where one item can be matched only to one item from another sequence.

Table 2.1 shows the characteristics of the presented measures. The measures are characterized by five attributes, and each one represents one of the defined characteristics. We can see that most of the measures are based on partial matching, which allows focusing on similar parts of the sequences. Most of the measures take into account the order of the items. In addition, only two measures of the studied ones depend on the time, but we think that the integration of the time dimension is possible for most other measures.
Depending on the context, some measures are more appropriate than others. We think that for the analysis of the semantic trajectories in the our context, choosing a single matching measure is best because the number of occurrences of the semantic elements is essential. For example, two documents with the same degradation in their conservation history but one of them had it two times are distinct as they do not share the same history, therefore, they should not considered as identical histories. In our context, the measure to compute the similarity

between the conservation histories should be partial, single, and depend only on the order.

	Matching		Dependency		
Measures	**Partial/Complete**	**Single/Multiple**	**Position**	**Time**	**Order**
ED	Complete	Single	✓	✗	✓
DTW	Complete	Multiple	✗	✗	✓
PDTW	Complete	Multiple	✗	✗	✓
LCSS	Partial	Single	✗	✗	✓
EDR	Partial	Single	✗	✗	✓
EPR	Partial	Single	✗	✗	✓
STLCSS	Partial	Single	✗	✓	✓
STLC	Partial	Single	✗	✓	✓
Levenshtein	Partial	Single	✗	✗	✓
Damerau-Levenshtein	Partial	Single	✗	✗	✓
Hamming	Partial	Single	✓	✗	✓
Jaro/Jaro Winkler	Partial	Single	✓	✗	✓
Tversky index	Partial	Multiple	✗	✗	✗
Sorensen-dice	Partial	Multiple	✗	✗	✗
Jaccard	Partial	Multiple	✗	✗	✗
Overlap coefficient	Partial	Multiple	✗	✗	✗
Ractliff/obershelp	Partial	Single	✗	✗	✓

Table 2.1: Measures characteristics

2.2.3 Mining Trajectory Data

Trajectory data mining approaches [68, 118, 92] aim to provide solutions to extract knowledge from possibly large sets of trajectories. These approaches include clustering [111], which groups similar trajectories, classification, which classifies the trajectories into different categories, anomaly and interesting location detection, which identifies the outliers and interesting locations in trajectories, and the prediction of future locations or elements to cite only a few. Due to the possibility that the quality of the trajectory data could be low, data preprocessing [120] is necessary before any further analysis.

In this section, we present some of the most commonly used trajectory data mining approaches. We present the trajectory filtering techniques. In addition, we give an overview of recent work on trajectory clustering, prediction of trajectory data, and trajectory classification.

2.2.3.1 Trajectory Filtering

Trajectory filtering is the process of preparing the trajectories to be analysed. Trajectory filtering involves techniques such as noise filtering, data cleaning, and data compression or other preprocessing techniques such as trajectory segmentation.
Noise filtering is the process of eliminating unexpected mistakes in the data. For example, when analyzing vehicle trajectories on a given map, their location should always be within a route, and any noise value should removed or replaced. Most of methods used for noise filtering could

be categorized into three types, Mean filters [128], Kalman filters [60, 6] and particle filters [43, 39]. Consider the trajectory in figure 2.7 represented as a sequence of points, where the point p_5 is a noise point. Using the mean filter with a sliding window equal to 5 on the z coordinate, the coordinate of p_5 will be modified to be equal to $\sum_{i=1}^{5} p_i.z/5$. However, the size of the sliding window is important because small sliding windows can not filter correctly consecutive noise points, for example, p_{10}, p_{11} and p_{12}.

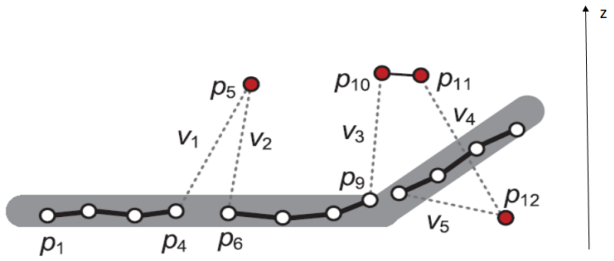


Figure 2.7: Noise points in a trajectory [128]

For spatial-temporal trajectories, stay point detection is the detection of the points where the geographical position does not change over a relatively long period. This approach could be used as a preprocessing technique because such points have a special meaning to the moving objects. For example, such point can help identify the nature of a stop. Many stay point detection approaches are based on the concepts of density and nearest neighbours. Density-based clustering [90], nearest neighbour [127, 126], or content-based [116] are among the algorithms used for stay point detection.

2.2.3.2 Trajectory Clustering

A cluster is a group of similar elements; elements from different clusters are not alike. Clustering is an important tool in exploratory data analysis and is used in several disciplines, such as artificial intelligence, pattern recognition, and information retrieval. A clustering algorithm generates clusters from the definitions of elements, and cluster analysis is the formal analysis of these algorithms. [46].
Clustering has been used in many applications such as healthcare, security, web search and outlier detection. In the context of trajectory analysis, clustering can be used, for example, for trajectory labelling, the extraction of representative trajectories known as “trajectories patterns", the detection of outliers, and the detection of common sub-trajectories. Furthermore, clustering could be used as a preliminary step in the development of a trajectory classifier. Most of the clustering algorithms can be classified into the following categories: partitioning algorithms, hierarchical algorithms, density-based algorithms and grid-based algorithms.

In this section, we present an overview of clustering algorithms. First, we give an overview of the basic clustering algorithms. Then, we provide an overview of some approaches that specifically target trajectory clustering.

2.2.3.2.1 Clustering Algorithms

Many trajectory analysis approaches rely on clustering algorithms in order to extract meaningful knowledge from the data, such as representative patterns for example. In this section, we present some of the most widely used clustering algorithms. In [40] the authors identify the following four categories of clustering algorithms: partitioning, hierarchical, density-based, and grid-based algorithms. The authors in [7] categorize learning algorithms into three types: unsupervised, supervised, and semi-supervised. In supervised algorithms, a label is associated to each element in the training set, representing its category. In unsupervised algorithms, the elements are unlabeled. The semi-supervised algorithms combines a small amount of labeled elements with a large amount of unlabeled data during training. The unsupervised category contains the density, hierarchical and spectral models and the supervised category contains the nearest neighbor algorithm, statistical models and Neural network. The semi-supervised algorithms fall between the two categories and are based on their algorithms. Unsupervised algorithms aim to describe the hidden relationships between similar items that belong to the same cluster by a label. In the unsupervised category, we will present some of these algorithms which are the density-based spatial clustering of applications with noise (DBSCAN) [26], k-means [42] and hierarchical clustering [82] (agglomerative and divisive) algorithms. The supervised algorithms aim to learn a function using a labelled data set, called a training set, to predict items' labels from a testing set. Some of the algorithms in this category are the nearest neighbor, statistical and neural networks algorithms.
We present in the following a brief description of the most widely used supervised and unsupervised clustering algorithms.

DBSCAN [26] Density-based clustering aims to identify dense regions of arbitrary shape, where the clusters are dense regions and separated by sparse regions. The algorithm relies on the definition of core objects, which are the ones having a number of neighbors exceeding a predefined threshold. The algorithm starts by computing, for each item, the number of items close to it, i.e. those having a distance less than a user-defined threshold α. Another parameter to be defined by the user is the density threshold, i.e. required number n of close items to a core item in order to initiates a cluster. In other words, a core item is an item that has n or more close items. All the core items that are directly reachable to each other, i.e. the distance between them is less than α belong to the same cluster. For example having three core items c_1, c_2 and c_3, if c_1 is close to c_2 then they will belong to the same cluster, and if c_3 is close to c_2, regardless its distance to c_1, it will be added to the cluster containing c_1 and c_2. Unlike core items, non-core items can not extend a cluster. A non-core item can join a cluster if it is close to a core item in it. The DBSCAN algorithm could be used for any data by providing a distance matrix that contains the distances between all pairs of items.

K-means [42] The k-means clustering algorithm is a partitioning method that aims to organize the objects into k clusters. The shape of the clusters is spherical, and each cluster have a representative which is the mean of the items in the cluster. The algorithm starts by randomly selecting k items where each will form a cluster. The items will then be added to the most similar cluster, i.e. the cluster with the closest mean. The clusters' means will be updated at each

iteration, and the distance between the items and the means will be re-computed. The algorithm ends when the clustering becomes stable; in other words, a high percentage of the items stay in the same cluster after the means update. Other versions of k-means are k-means++ [3], k-medoids [53] and k-modes [13]. K-modes is an extended version of k-means used for categorical variables where it is not possible to calculate the mean. The most commonly used method to select the number of clusters k for this algorithm is the elbow method. The K-means algorithm can be used for trajectory data by providing a suitable similarity measure between the trajectories.

Hierarchical models [82] There are two types of hierarchical models: agglomerative and divisive. Agglomerative clustering is a bottom-up algorithm, which starts by comparing all the items and merging the two most similar into a new cluster. Depending on the items type, a mean, a centroid or a medoid is used to represent the cluster. It is also possible to compute the distance between items and clusters according to single-linkage or complete-linkage. Using the single-linkage approach, the distance between an item and a cluster is the distance between the item and the closest point in that cluster. Using the complete-linkage approach, the distance between an item and a cluster is the distance between the item and the farthest point in that cluster. At each iteration, the algorithm merges clusters or items until generating one cluster that contains all the items in the data set. Finally, a process of cluster detection, referred to as tree cutting, is performed to define the resulting clusters. Unlike the agglomerative approach, the divisive one, begins with a cluster containing all the items and splits it recursively to meet some predefined requirements such as the number of clusters, the maximum number of items in the clusters or the maximum distance to the means.

Nearest Neighbor [24] The nearest neighbor is a widely used classification approach that could be extended to be used for clustering [12]. The nearest neighbor algorithm clusters an item depending on its neighbors, i.e. the most similar items to it. The most commonly used algorithm is the K-Nearest Neighbor (KNN) which uses a voting system to determine the cluster or category of a new coming item. Given a set of labelled items and a new item, the distance between the new item and all the labelled ones is calculated, and its k nearest neighbors vote on its label. As a result, labelling each new item necessitates computing its distances to all labelled items in the data set, which is time-consuming if the data set is vast.

2.2.3.2.2 Trajectory Clustering Approaches

Trajectory clustering approaches for both spatial and semantic trajectories generally rely on the basic clustering algorithms. Most trajectory clustering algorithms deal with a trajectory as a single item when grouping trajectories. Some approaches deal with a trajectory as a set of segments to detect possible common sub-trajectories in dissimilar trajectories. We present in this section two trajectory clustering algorithms that are based on partition-and-group and community detection, respectively. In addition, we present an approach to cluster semantic trajectories taking into account the semantic relationships between the elements constituting

the trajectories.
When dealing with a trajectory as a single item, it is possible to miss the common characteristics between trajectories that share some sub-trajectories. [58] introduces the new partition-and-group approach on spatial trajectories that aims to detect similar trajectories, in addition to, similar sub-trajectories. The approach is based on the DBSCAN algorithm. Figure 2.8 illustrates the clustering process that starts by partitioning each trajectory into a set of segments. Similar segments are grouped into a cluster represented by a trajectory. The partitioning de-

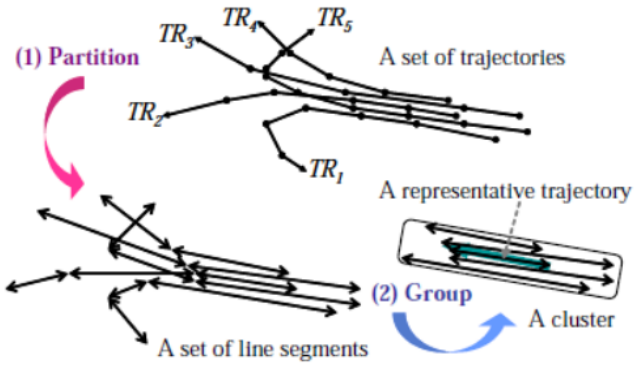


Figure 2.8: An example of trajectory clustering in the partition-and-group framework [58]

pends on the minimum description length (MDL) and uses predefined distances, depending on the segments' angle and length. Therefore, to use this approach for semantic trajectories, it is necessary to define a new partitioning approach suitable to the context and type of data. After defining the neighborhood of a segment as the set of segments that are close to it, the core segment as a segment with a neighborhood higher than a predefined threshold, and conditions to call a segment reachable or connected, the approach uses an adapted DBSCAN algorithm that considers the number of segments of the base trajectories. This extension of DBSCAN prevents the creation of a cluster containing segments extracted from the same or a small number of trajectories where they should be considered non-reachable or outliers.

The approach presented in [66] uses the k-nearest neighbor network construction approach, where a node represents each trajectory. For each node n, the algorithm searches the K most similar nodes and adds an edge between n and these nodes. Finally, a community search algorithm is applied to generate semantic trajectory clusters on the created network. Figure 2.9 shows a classic example of a graph containing three communities represented in different colours. We can notice that the nodes that belong to the same community are fully connected while the number of connections between nodes in other communities is low.
To compute the similarity between the trajectories, the authors define a semantic trajectory similarity measure based on the work presented in [89, 34] and on the ontology theory. The ontology is used to match the categorical attributes by searching a common hypernym. The authors in [33] provide a detailed review and user guide to the methods of community detection in networks.

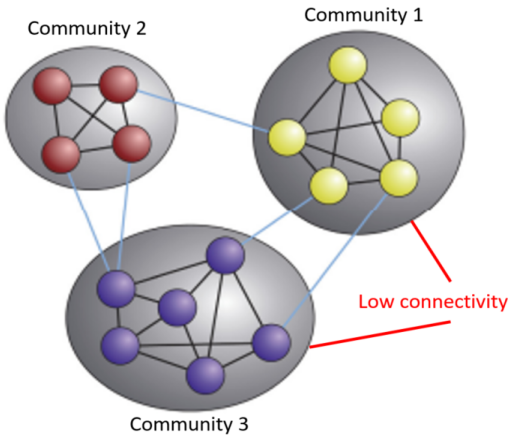


Figure 2.9: Schematic picture of a network with three communities [33]

In the approach presented in [78], the authors test several clustering algorithms on multi-dimensional semantic trajectories with the use of UMAP [71] for dimension reduction. The similarity measure used during the clustering has been introduced in [77], where the authors propose a Contextual Edit Distance CED, a similarity measure for semantic sequences. It is an extension of the Edit Distance that considers the context similarity between the elements (events). The value of the semantic similarity between each pair of elements is between 0 and 1. The calculation is based on a proposed directed acyclic graph that represents the relationships between the elements. Figure 2.10 shows ten semantic trajectories and a concept hierarchy

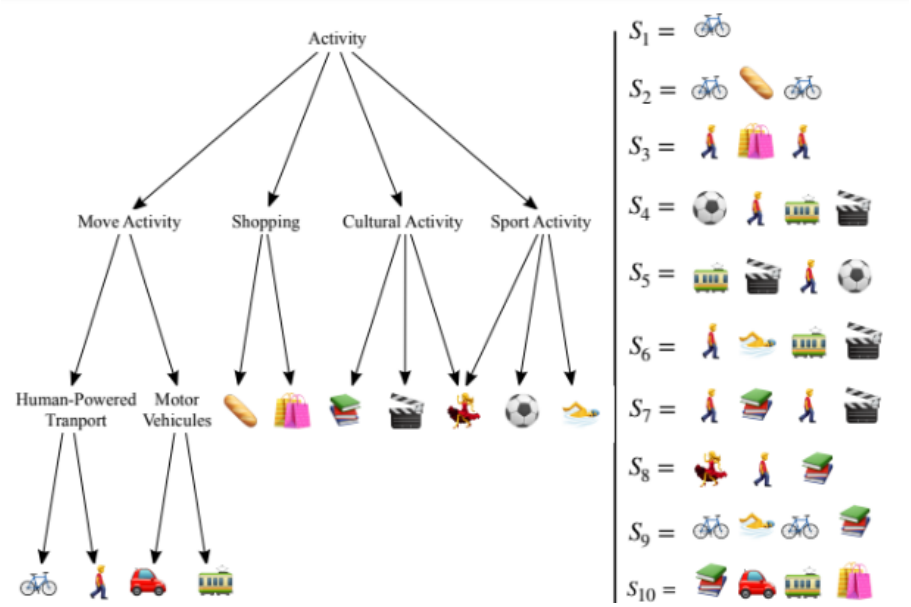


Figure 2.10: Ten semantic trajectories and the associated concept hierarchy based on an simple ontology [77]

representing the semantic elements that constitute the trajectories. The edition cost depends on the similarity between the element to edit and the most similar element in the another sequence. The repetition and permutation editions of similar elements have a lower cost.

2.2.3.3 Prediction Approaches for Trajectory Data

Prediction approaches for trajectory data aim to determine the following elements in the trajectory based on the previous ones. In many location-based services, next location prediction is an essential technique, including route navigation, applications like location-based advertising, dining location recommendations, and traffic planning and control, to mention a few.
In the survey presented in [120], the authors review the existing location-prediction methods for temporal and spatio-temporal trajectory data. The prediction methods are divided into different types such as content-based and pattern-based. The content-based methods use the Markov model, and the pattern-based methods aim to find common behaviours.
The authors in [123] propose an approach called semantics-enriched recurrent model (SERM) that enriches the prediction of the next location by semantic information such as a text message that describes the activity of the user. The trajectories are considered as sequences of records, for example, a trajectory of a user u_i is represented as a sequence of records $t(u_i) = \{r_1(u_i),\ r_2,\ ...,\ r_K(u_i)\}$. Each record is such that: $r_k(u_i) = (t_k, l_k, c_k)$, where t_k is the time, l_k is the location of the user at time t_k, and c_k is the text message describing the activity. To predict the next location, the authors propose a three-layer model. The first layer embeds the elements in each record, the second layer is a recurrent layer where each node h_k encodes the information observed until step k, and the third layer is the output layer. Finally, the authors define the objective function and use the Stochastic Gradient Descent and the backward propagation to learn the parameters to optimize the objective function.

Content-based methods

These methods learn the probability of the location transition. Given a sequence of location history, the methods aim to predict the next location based on this history using the Markov model. In this model, a user's next location depends on his current location [124]. In a k-order Markov model, the next location depends on the last k locations in the history. Formally, if the user trajectory is $tr = \{l_1,\ l_2,\ ...,\ l_{n-k+1},\ ...,\ l_n\}$, the k-order context is the sequence $(l_{n-k+1},\ ...,\ l_n)$ and the Markov model is defined as follows:

$$P(l_{n+1} = l^* | L(1,n)) = P(l_{n+1} = l^* | c = (l_{n-k+1}, l_n)) \tag{2.10}$$

Another approach based on the content to predict the next location is the compression-based location predictors [106]. A trajectory is first partitioned into distinct sub-trajectories. Then a tree is built to store the sub-trajectories, where each node represents an element and its number of occurrences at a specific index. For example, consider a trajectory $tr = gbdcbgcefbdbde$,

the partitioning results is a set of sub-trajectories g,b,d,c,bg,ce,f,bd,bde. The tree representing the locations occurrences is represented in figure 2.11. The node b:4 indicates that location b appears four times at the beginning of the sub-trajectories as its position in the tree in the first layer after the root node. The node d:2 indicates that the location d appears two times in the second position in the sub-trajectories as its position in the tree in the second layer.
The following equation 2.11 is used to compute the occurrence probability of a location l given a prefix p, where N(p,l) denotes the number of p occurring as a prefix for p,l in the sub-trajectories, and L is the location sets.

$$P(l_{n+1} = l) = \frac{N(p, l|L)}{N(p|L)} \tag{2.11}$$

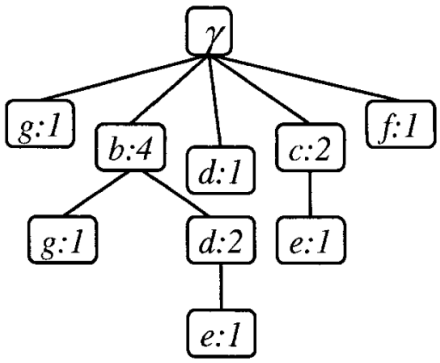


Figure 2.11: Example of a tree containing sub-trajectories based on compression [106]

Pattern-based methods

Trajectory pattern mining includes methods based on sequential patterns [35, 55], frequency patterns [116], integrated data patterns [62], and periodic patters [125].

The basic frequent sequential pattern (FSP) is the problem of finding all the frequent sequences in a database of sequences TrSet, where each element of each sequence is a timestamped set of items. The FSP problem was first introduced in [1]. Elements in the sequence are arranged according to their time-stamps. A frequent sequence is defined as a subsequence of a large percentage of sequences of TrSet.

The authors in [35] introduce trajectory patterns as concise descriptions of frequent behaviours in terms of space and time. The authors define a trajectory pattern as a pair (S,A), where S=[(x_0,y_0), ..., (x_k,y_k)] is a sequence of points in $\mathbb{R}^2$, and A=[α_1, ..., α_k] is the temporal annotation of the sequence. The trajectory pattern is represented as (S,A)=$(x_0,y_0) \xrightarrow{\alpha_1}$ $(x_1,y_1) \xrightarrow{\alpha_2}$ ... $\xrightarrow{\alpha_k}$ (x_k,y_k).
A trajectory pattern is considered frequent if the number of input trajectories that contain this

pattern is higher than a predefined threshold s_{min}. In addition, the authors define a spatio-temporal containment relationship $\preceq$ between a trajectory pattern and an input trajectory, which determines when a trajectory pattern is considered contained in an input trajectory. The support(S,A) is defined as the number of input trajectories T such that (S,A) $\preceq T$. Finally, a frequent trajectory pattern is each trajectory pattern (S,A) for which support(S,A) $\geq s_{min}$.

2.2.3.4 Trajectory Classification

The goal of trajectory classification tasks is to find trajectories, sub-trajectories or elements that best discriminate the different classes.
In [103], the authors divide the classification methods into three categories depending on whether the methods extract local features [8, 18, 105], global features [130, 102, 50], or both [21, 122, 27].
The approach presented in [21] extracts local and global features from trajectories such as speed, acceleration and direction change between points. Global features are fix for the entire trajectory, such as the maximum speed. Once the features are extracted, the authors propose a process to classify the trajectories. The process starts by selecting the features, followed by a principal component analysis (PCA) to reduce the number of original features. Finally, the support vector machine [17] is used to classify the selected trajectories.
The authors in [103] survey the state of the art in trajectory classification and compare existing methods. A raw trajectory is defined as a sequence of n points Tr=[p_1, p_2, ..., p_n] in which p={x,y,t}, where x,y is the position of the moving object is space and t is the timestamp at which the point was collected. A multiple aspect trajectory is defined as a sequence of multiple aspect points where a point p={x,y,t,A}, for such that x, y is the position in space at time t, and A is a set with r aspects A = {a_1, a_2, ..., a_r}. The authors separate between the classification of raw and multiple aspect trajectory classification.
In [59], the authors introduce TraClass for raw trajectories classification, a technique to classify spatial-raw trajectories. TraClass first divides the space in a grid and iterates to reduce the size of the grid cells until most trajectories inside a cell belong to the same class. A region space is considered homogeneous if only one class c_{major} has a number of trajectories which is higher than a predefined threshold, and all the other classes do not. A cell is picked as a feature and no longer divided into smaller sizes if most of the trajectories it contains belong to the same class. If not, the splitting process for this cell continues until it reaches the smaller possible size determined by a predefined threshold.
When considering both the space and time dimensions, different features could be extracted from the trajectories, such as speed, acceleration etc. When a feature depends on the entire trajectory, it is called a global feature; when it depends on sub-trajectories or on trajectory points, it is called a local feature.

For multiple aspect trajectories, the authors in [72] provide a novel approach to represent and enrich a multiple aspect trajectory, defined as a sequence of points that can be enriched with three different types of aspects: volatile, long-term, and permanent. Whereas the volatile aspect of an object, such as the time attribute, frequently changes during the object movement, the long-term aspect does not change in the same trajectory. It may still change in a future

trajectory of the same object. The permanent aspect is related to the object and lasts for the duration of its lifecycle. The authors present a conceptual model for defining an aspect as a set of attributes.
The work presented in [70] introduces a Recurrent Neural Network RNN-based approach to classify the multiple-aspect trajectories using space, time and semantic embeddings. Different methods are used to encode the attributes of the points, such as one-hot and geohash encoding, depending on the type of each one. Then each encoded attribute is multiplied by its respective embedding matrix to extract its embedded representation, where all the attributes are embedded in the $\mathbb{R}^n$ space. Finally, the attributes are aggregated, and a vector represents each point in $\mathbb{R}^n$. The trajectories are then fed to the recurrent module.
The authors in [29] propose a method called movelets that extracts relevant sub-trajectories that can be used to classify trajectories. A relevant sub-trajectory is defined as a trajectory capable of discriminating the existing classes. This method extends the shapelets technique used to analyze time series. The authors consider the sub-trajectory as a contiguous sub-sequences.

2.2.4 Discussion

In section 2.2, we have presented different works related to the trajectories and sequences analysis, which are similar to the type of analysis we aim to propose to analyze the conservation–restoration histories at the BnF. In particular, we have presented the representation of the trajectories, and different approaches related to analysis tasks that are interesting in our work, such as the methods to compute the similarities between trajectories and existing methods used for trajectory data mining.

Several representations have been used for trajectories and sequences. Some of these are used for spatial trajectories where the provided data contains locations, and others for semantic trajectories. In the existing approaches, the representations used for semantic trajectories could be used for conservation–restoration trajectories, while the representations of spatial trajectories can not be used because the conservation–restoration documents' history are composed of events representing semantic elements.
Similarly to sequences of events, the conservation–restoration histories are discrete. In addition, the number of occurrences of their elements should be taken into consideration. Hence, a history of one treatment of type x is partially similar but not identical to a history containing two treatments of type x. Therefore, the measures that allow an event to be matched with multiple events can not be used in our context. Furthermore, conservation–restoration histories vary in length. As a result, if an event cannot appear in more than one match, a complete match cannot be used. Regarding the temporal dimension, the order and type of elements are more important than the time of their occurrence according to the domain experts. The measure to calculate the similarities between the conservation sequences should therefore be a discrete and partial match. It should take into consideration the order of the events and give more importance to their types than their time. As a result, the measure used to calculate the similarities between conservation event sequences should be a partial and single match, and it should depend only on the order of the events. As a consequence, six of the measures presented in section 2.2.2 can be used in our context: the LCSS, EDR and EPR from the trajectories

similarities measures context, and the Levenshtein with its extension Damerau-Levenshtein and the Ratcliff/obershelp from the string similarity context. Note that these measures are related, as the EDR, EPR, Levenshtein and Damerau-Levenshtein are based on the edit distance. The LCSS and the Ratcliff/obershelp are similar; they calculate the longest common sub-sequence.

Several approaches have been proposed for trajectory and sequence analysis, with different goals. Depending on the context and objectives, the best suitable approach should be selected for trajectory data mining to cluster the trajectories, predict the following elements or classify the trajectories. The analysis method selection also depends on the available data and their types. One of the limitations of the nearest neighbor method is the required computation of the distance with all the elements in training set for each new item to classify or to cluster, and in our context, more than forty million histories exist. Partition-and-group algorithm is a new idea used on spatial trajectories, and one of its limitations is that it can not be used on semantic trajectories. The method is based on pre-defined distances depending on the length and trajectory angle. The adaptation of the method is required to use it on semantic trajectories.
In our context, we aim to provide a model that predicts the documents' physical state based on their conservation–restoration histories. Therefore, we are interested in providing an analysis pipeline to achieve such a prediction. One module of this pipeline will aim to extract knowledge using the appropriate clustering and classification approaches. In addition, some adaptions are required to integrate more semantics into these approaches.

2.3 Knowledge-based Similarity Computation

As trajectories or sequences are composed of elements, their matching relies on the comparison and evaluation of the similarity between these elements. The matching between spatial trajectories is based on the selected distance metric and a predefined threshold to find similar elements.
Matching the elements in semantic trajectories is a complex process, where the elements could have hidden relationships known only by the domain experts. In addition, it is possible to have matching problems due to the heterogeneity in the terminology used of the semantic elements. For example, in our context, it is possible to find two conservation–restoration events with different names but refer to the same event.
For this reasons, the experts knowledge could be represented to help in matching the semantic elements.

The knowledge of an expert is heuristic in nature. Because experts often face challenges articulating the rules of thumb they use to solve problems efficiently, acquiring their expertise is generally a complex and challenging task. This phenomenon is known as the knowledge acquisition bottleneck [44]. According to [49], "the more competent domain experts become, the less able they are to describe the knowledge they use to solve problems!". Several knowledge acquisition techniques are available for facilitating knowledge transfer from experts, including analysis and interviews protocols. Expert knowledge can take several forms, such as ontologies, thesauri, and taxonomies of concepts. It can be represented by a graph where the nodes are the concepts, and the links represent the semantic relationships between them. These representations store the experts' knowledge and can be processed automatically to enhance complex systems. Such knowledge graph could enrich the analysis tasks by integrating semantic information along the analysis process. The authors in [104] study the impact of integrating domain knowledge in the analysis by comparing the results of different data mining classification methods with and without incorporating domain knowledge and shows that incorporating domain knowledge improves the classification results.

Some works have been proposed to calculate the similarity between concepts using external knowledge represented by knowledge graphs. For example, some measures have been proposed to calculate the similarities between concepts using graphs representing their relationships. In addition, approaches have been proposed to refine the graph-based similarity calculation by associating for each concept in the graph a weight based on many criteria such as its position in the graph. In this section, we will present an overview the concepts similarity computation methods in section 2.3.1 and graph weighting methods in section 2.3.2.

2.3.1 Concepts Similarity

The matching between concepts could be enriched with domain experts' knowledge when computing the similarity between semantic trajectories represented as sequences of concepts. This knowledge could be used to match concepts with different names but share hidden common characteristics known only by domain experts. The experts' knowledge could be expressed in

different ways, such as a knowledge graph representing the relationships between the concepts or by a set of features enriching the concepts with their characteristics. Therefore, feature-based measures have been proposed to calculate the similarity between concepts, each one represented by a set of features. In addition, different methods have been proposed to compute the similarity between concepts in a graph, and we refer to these as knowledge-based concept similarity measures. In this section, we present the feature-based concepts similarity measures in section 2.3.1.1, and the knowledge-based concepts similarity measures in section 2.3.1.2.

2.3.1.1 Feature-based Concepts Similarity Computation

Different approaches have been proposed to compute the similarity between concepts by enriching their description with a set of characteristics related to the context. Each concept can be described by a set of attributes, which we will call features. Given two concepts c_1 and c_2 with their set of features f_1 and f_2 respectively, different characteristics could be extracted from such representation, such as the number of shared features between the concepts and the number of features that exist for a concept and do not for the another. We present a set of operations between the two concepts used in [112] as follows:

- $c_1 \cap c_2$: the common features between f_1 and f_2.
- $c_1 - c_2$: the features that belong to f_1 but not to f_2.
- $c_2 - c_1$: the features that belong to f_2 but not to f_1.

Using set theory and the ratio model, the authors in [112] define a similarity measure between concepts based on the ratio of the shared features between the concepts. The similarity measure is defined as follows:

$$S(c1,c_2) = \frac{f(c_1 \cap c_2)}{f(c_1 \cap c_2) + \alpha f(c_1 - c_2) + \beta f(c_2 - c_1)} \tag{2.12}$$

Where α, $\beta \geq 0$ and f measures the contribution of any particular feature to the similarity between objects. By changing α and β the model generalizes several set-theoretical similarity measures proposed in the literature [37, 25, 9]:

- If α=β=1, $S(c_1, c_2) = \frac{f(c_1 \cap c_2)}{f(c_1 \cup c_2)}$ [37].
- If α=β=$\frac{1}{2}$, $S(c_1, c_2)$ equals $\frac{2f(c_1 \cap c_2)}{f(c_1)+f(c_2)}$ [25].
- If α=1 and β=0, $S(c_1, c_2)$ reduces to $\frac{f(c_1 \cap c_2)}{f(c_1)}$ [9].

The authors in [98] propose a method to calculate the similarity between concepts in different ontologies. The method is based on the normalization of Tversky's similarity model [112], and defined as follows:

$$S(c1,c_2) = \frac{|c_1 \cap c_2|}{|c_1 \cap c_2| + \alpha(c_1,c_2)|c_1 - c_2| + (1-\alpha(c_1,c_2))|c_2 - c_1|} \tag{2.13}$$

Where $|S|$ is the cardinality of a set S, and α is a function that defines the importance of the non-common characteristics depending on the depth of the compared entities in the hierarchy and it is defined as follows:

$$\alpha(c_1,c_2) = \begin{cases} \frac{depth(c_1)}{depth(c_1)+depth(c_2)} & depth(c_1) \leq depth(c_2) \\ 1 - \frac{depth(c_1)}{depth(c_1)+depth(c_2)} & depth(c_1) > depth(c_2) \end{cases} \tag{2.14}$$

To compute the similarity between two entities from different ontologies, the algorithm starts by connecting the two ontologies by adding a new imaginary root T, and by making each of the ontologies' root a direct descendant of T. The similarity depends on the depth of the concepts as show in equation 2.14.
For example, figure 2.12 shows the connection between two ontologies through the insertion of a new root, denoted by "anything". In order to compute the similarity between the two entities "building" in the two ontologies (a) and (b) respectively, the depth of a.building is equal to five, and the depth of b.building is equal to two. Finally, α(a.building, b.building) will be equal to 0.28 based on the equation 2.14.

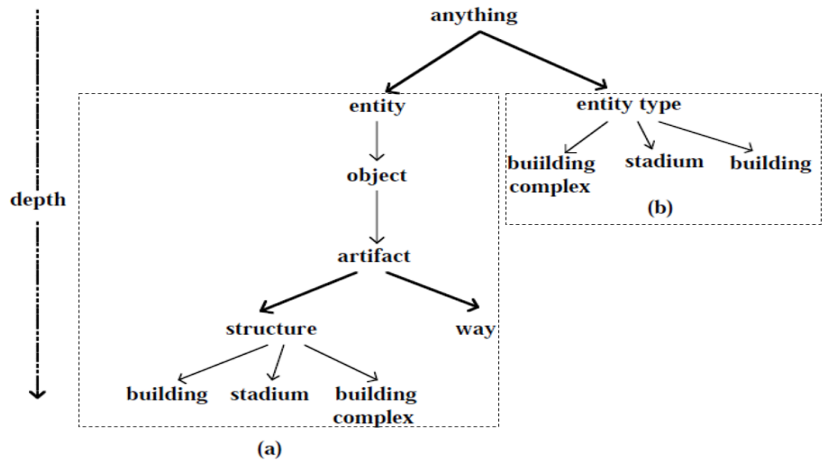


Figure 2.12: Connecting independent ontologies: (a) partial WordNet ontology and (b) partial SDTS ontology. [98]

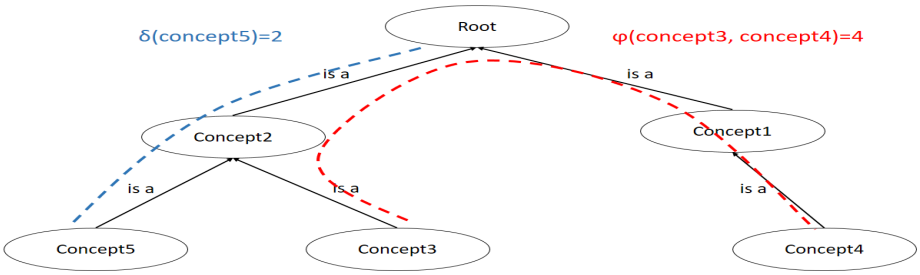


Figure 2.13: Depth and distance between concepts

2.3.1.2 Knowledge-based Concepts Similarity Measures

The knowledge-based concepts similarity measures differ from feature-based ones. The knowledge-based measures depend on the relationships between the concepts and not on a set of features representing the concepts. These relationships are represented by a graph which will be used by the considered measure. This section presents an overview of graph-based semantic similarity measures. The methods are represented by their year of publication in figure 2.14.

Most of the approaches for computing the similarity between two concepts based on a graph search for their first common hypernym, also called the Least Common Subsumer (LCS) [16]. These methods rely on some characteristics such as the length of the paths between the concepts and their LCS, the information content or the weight of the concepts or the depth of the concepts, i.e. the respective lengths of the paths between each concept and the root of the hierarchy. The authors in [2] categorise the semantic similarity measures into two types, the first one is based on the semantic distance between concepts, i.e. number of edges between concepts, and the second one is based on the statistical distribution of concepts in a given corpora. Figure 2.13 illustrates the depth δ and distance between concepts φ, where the depth of concept5 is equal to two, and the distance between concept3 and concept4 is equal to four.

We present in the following the most commonly used semantic similarity measures methods to calculate the similarity between concepts. Some of the methods are based on the information content. An information content indicates the amount of information provided by a concept and its degree of generality. We provide an overview of the methods used to calculate the information content in section 2.3.2. Three of the methods in this category are based only on the information content [95] [65] [48]. The others integrate the depth or the length in the similarity computation.

Sim_{Res} [95] was the first to use the information content theory in the calculation of the similarity between two concepts c_1 and c_2, which was defined as the information content of their LCS, as shown in the following definition:

$$sim_{res}(c_1, c_2) = IC(LCS(c_1, c_2)) \tag{2.15}$$

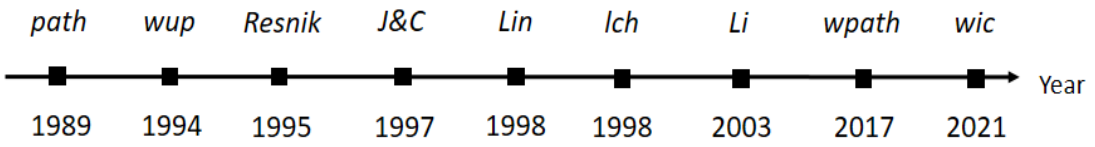


Figure 2.14: Semantic similarity measures by year of publication

$Sim_{J\&C}$ [48] extends the resnik method by calculating the distance between two concepts as the difference between their IC and the IC of their subsumer. The distance and the similarity between two concepts is defined as follows:

$$dis_{j\&c}(c_1, c_2) = IC(c_1) + IC(c_2) - 2 \times sim_{res}(c_1, c_2) \tag{2.16}$$

$$Sim_{j\&c}(c_1, c_2) = \frac{1}{1 + dis_{j\&c}(c_1, c_2)} \tag{2.17}$$

Sim_{Lin} [65] is another extension of the resnik method where the similarity between two concepts is measured as the ratio of the shared information content between them. The similarity, denoted sim_{lin}, is defined as follows:

$$Sim_{lin}(c_1, c_2) = \frac{2 \times sim_{res}(c_1, c_2)}{IC(c_1) + IC(c_2)} \tag{2.18}$$

Sim_{Path} [91] uses the distance feature in the calculation of the similarity between two concepts, and it is defined as follows:

$$Sim_{path}(c_1, c_2) = \frac{1}{1 + \varphi(c_1, c_2)} \tag{2.19}$$

Sim_{wup} [121] is based on the depth of the concepts and their LCS to calculate the similarity. The similarity is inversely proportional to the distance between the compared concepts and their LCS, i.e. the difference between their depth and the LCS's depth in the graph. For example, when the LCS of two concepts c_1 and c_2 is low in the hierarchy of concepts, it means these concepts have many common characteristics. On the contrary, if the LCS of two concepts is at a very high level in the hierarchy, the two concepts do not have common characteristics. The definition of the Wup similarity measure is the following:

$$Sim_{wup}(c_1, c_2) = \frac{2\delta(LCS(c_1, c_2))}{\delta(c_1) + \delta(c_2)} \tag{2.20}$$

Sim_{lch} [57] takes into account the depth of the graph in the calculation of the similarity, i.e. the maximum distance between a leaf concept and the root node, referred to as $\mathbb{L}$, and the definition of the similarity is the following:

$$Sim_{lch}(c_1, c_2) = -log\left(\frac{\varphi(c_1, c_2)}{2\mathbb{L}}\right) \tag{2.21}$$

Sim_{Li} was presented in [61]. The authors investigate the effectiveness of their method by defining different strategies to combine the depth, the distance and a local density, represented by the information content, to calculate the similarity between concepts. For example, one of the strategies combines the depth of two concepts with their shortest length to calculate their similarity, and its definition is as follow:

$$Sim_{li}(c_1, c_2) = e^{-\alpha\varphi(c_1,c_2)}\left(\frac{e^q - e^{-q}}{e^q + e^{-q}}\right) \tag{2.22}$$

Where $q = \beta\delta(LCS(c_1, c_2))$, with α and β are two predefined parameters.

Sim_{wpath} was presented in [132]. It combines the path information with the information content of the subsumer, and its definition is as follows:

$$Sim_{wpath}(c_1, c_2) = \frac{1}{1 + \varphi(c_1, c_2)k^{IC(LCS(c_1,c_2))}} \tag{2.23}$$

Sim_{wic} [2] tackles the uniform distance problem of the wpath method. The problem is that the method provides the same similarity for each pair of concepts having the same distance, i.e. shortest path, regardless of their level of abstraction. For example, in Figure 2.13, the concepts pairs $(concept5, concept3)$ and $(concept1, concept2)$ have the same shortest path, and hence the same similarity regardless that the concepts are in different levels of abstraction and the concepts $concept5$ and $concept3$ are more specific and should be more similar than $concept1$ and $concept2$. The authors combine the depth information and the information content of the subsumer. Their approach extends the Wup measure by calculating the difference between the depth of the concepts to be compared and their subsumer as follows:

$$\Delta = \delta(c_1) + \delta(c_2) - 2\delta(LCS(c_1, c_2)) \tag{2.24}$$

Then a weight of contribution is defined as follows:

$$\omega = \frac{\lambda}{\lambda + \Delta + 1} \tag{2.25}$$

Where $\lambda = N$ or $\lambda = 10^{-N}$. Finally the similarity between two concepts is defined as follows:

$$Sim_{wic}(c_1, c_2) = \omega * IC(LCS(c_1, c_2)) \tag{2.26}$$

We can see that the existing knowledge-based semantic similarity measures could be grouped into three groups. (i) the measures based only on the information content, (ii) the ones that depend only on structural information, such as the depth and distance between the compared concepts, and (iii) the ones based on both the information content and the structural information.

2.3.2 Considering the Importance of Concepts During Similarity Calculation

The elements of a trajectory do not always have the same importance for a given analysis task. The elements in a sequence or a semantic trajectory may sometimes require to be given different weights during the analysis process For example, given an analysis task in the healthcare field that aims to predict the patient's condition based on his healthcare history, a heart operation is an event that significantly impacts the patient's condition. Therefore, when it is shared between two patients' healthcare histories, their similarity score should be strongly affected to be higher. On the contrary, when an event that does not have a critical impact on the patient's condition appears in healthcare history, it does not bring much information or give specificity to the patient. For example, flu treatment is an event that happens to most patients and is not very dangerous. Therefore, when shared between two patients' healthcare histories, the similarity score should not be affected much.

For this reason, it could be useful to identify the relative importance of the different concepts that compose semantic trajectories or sequences in the context before analysing the semantic trajectories. Many works used to compute the information content [100] of the concepts and use it to express their importance. Other works assign weights to the concepts to represent their importance. These methods depend either on the frequency of the concepts in a given corpus or on the concept's level of abstraction in the graph. More specialized concepts are considered more important as they contain more information. This section presents the most commonly used approaches for information content calculation [65] [101] [131] and concept weighting [30] [31] [19].

2.3.2.1 Information Content

The information Content (IC) of a concept provides an estimate of its degree of generality/concreteness, a dimension which enables a better understanding of concept's semantics [100]. The information content of concepts in a graph could be calculated in different ways. For example, it could depend on the occurrences of each concept in a given corpus or on the structure of the graph, i.e. the number of hyponyms of each concept. This section provides an overview of the most widely used graph-based information content calculation methods. Let us consider the very simple taxonomy shown in figure 2.15 which we will use to illustrate the different calculation methods.

Inverse Probability-based

Many approaches [65, 48, 95] calculate the information content of a concept in a graph based

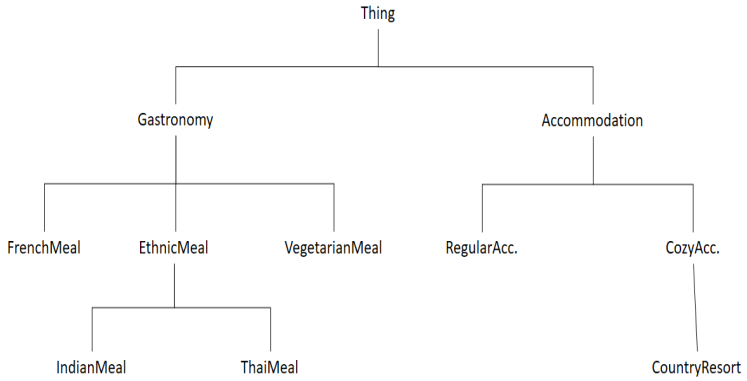


Figure 2.15: An example of concept hierarchy

on its occurrence probability in a given corpus. A corpus is a set of elements represented each by a set of concepts. The idea behind the inverse probability is that the more frequent a concept, the less information it contains. For example, the concept $Thing$ appears in every element in a corpus, regardless of the type of the elements, because based on the hierarchy where $Thing$ is the root concept, we can say that every concept in the hierarchy is a thing. As a result, we can not distinguish between the elements or extract more information to characterize them based on this concept, and the information content of the root of the hierarchy "Thing" is very low. The information content of a concept x based on the inverse probability is computed as follows:

$$IC(x) = -log\ p(x) \tag{2.27}$$

The calculation of the IC of a concept x is achieved by computing the probability p(x) of encountering x or any of its hyponyms in the given corpus. For example, to calculate the information content of the concept $EthnicMeal$ represented in figure 2.15, we calculate the appearance probability of it or any of its hyponyms $IndianMeal$ and $ThaiMeal$. And the p(x) is defined as follows:

$$p(x) = \frac{\sum_{w \in W(x)} count(w)}{N} \tag{2.28}$$

Where $W(x)$ is the set of hyponyms of x and N is the total number of corpus terms, i.e. the total number of concepts representing the corpus elements.

Hyponym-based

This method aims to compute the IC in an intrinsic manner without depending on the corpora. The idea is that a concept with many hyponyms means that more information and specifications could be added to the concept to be transformed into another concept at a lower abstraction level. Concepts with a high number of hyponyms provide less information than the leaf concepts. The first approach to compute the IC of a concept based on its number of hyponyms was

introduced in [101]. The corresponding formula is as follows:

$$IC_{seco\ et\ al}(x) = 1 - \frac{log(hypo(x)+1)}{log(max\_nodes)} \tag{2.29}$$

Where $hypo(x)$ is the number of concepts below x in the graph, i.e. the number of hyponyms, and $max\ nodes$ is the total number of concepts in the graph. In this way, the leaves will have the highest IC, which is equal to one, and the root concept will have the lowest IC, which is equal to zero. For example, the concept $Accommodation$ have three hyponyms below it in the graph represented in Figure 2.15, therefore $IC(Accommodation) = 1 - \frac{log(4)}{log(11)}$.

Hyponym and depth-based

One limitation of the hyponym-based method is that two concepts in different abstraction levels will have the same information content if they have the same number of hyponyms. For example, in Figure 2.15 $RegularAcc.$ and $CountryResort$ will have the same information content value. In order to overcome this limitation, the depth is taken into account in the calculation of the IC, as proposed in [131], where the IC is calculated as follows:

$$IC_{zhou\ et\ al}(x) = k(1 - \frac{log(hypo(x)+1)}{log(max\_nodes)}) + (1-k)\frac{log(depth(x))}{log(max\_depth)} \tag{2.30}$$

Where $depth(x)$ is the distance between x and the root, max_depth is the maximum distance between two concepts in the graph, and k is a tuning factor. For example, with k equal to 0.5, the IC of $RegularAcc$ in figure 2.15. will be equal to $0.5(1 - \frac{log(1)}{log(11)}) + 0.5\frac{log(2)}{log(3)} = 0.81$

Leaf-based

The size of a hyponym tree below a concept depends on many criteria such as the degree of detail and the branching factor, as mentioned in [100]. The authors propose to use only the leaves to compute the information content of a given concept. The leaves represent the semantic of the most specific concepts of a domain and they would accurately define its scope. The leaves are defined as follows:

Definition 1 *Consider a knowledge graph G and its set of concepts C, the leaves of a concept x are defined as follows:*
$leaves(x) = \{l \in C \mid l \in hyponyms(x) \wedge l\ is\ a\ leaf\}$

The method assigns a lower information content to the concepts with many leaves because they are considered as more general and they subsume the meaning of many other concepts. The leaves have the highest information content equal to one, and the root has the lowest value, equal to zero. The leaf-based IC of a concept x is defined as follows:

$$IC_{sanchez\ et\ al}(x) = -log\left(\frac{|leaves(x)|+1}{max\_leaves+1}\right) \tag{2.31}$$

Where $|leaves(x)|$ is the number of hyponym leaves of the concept x, and max_leaves is the number of leaves in the graph. For example, the IC of the concept $Gastronomy$ will be equal to $-log(\frac{5}{7})$.

Leaves and subsumers-based As mentioned in [131], the position of the concept, i.e. its level of abstraction in the graph should be considered in the calculation of the IC. The leaf-based IC can be extended to consider this information. [100] extends the leaves-based IC method by defining and using the subsumers of concepts in the calculation. The subsumers of a concept x are all the concepts in higher abstraction levels than x, and x is their hierarchical specialization. The subsumers are defined as follows:

Definition 2 *Consider a set of concepts C in a graph, and consider that ($\leq$) indicates the hierarchical specialization relation, i.e $x \leq c$ means that x is a specialization of c, The set of $x's$ subsumers is defined as:*

$$subsumers(x) = \{c \in C | x \leq c\} \cup \{x\}$$

The IC of a concept x is defined as:

$$IC_{sanchez\ et\ al}(x) = -log\left(\frac{\frac{|leaves(x)|}{|subsumers(x)|} + 1}{max\_leaves + 1}\right) \tag{2.32}$$

As we can see, different methods have been proposed to calculate the information content of concepts using information related to the structure of the concepts' hierarchy. The difference between the existing methods is which information was used.

2.3.2.2 Weighting Methods

Concepts weighting approaches proceed in a similar way to information content evaluation. In fact, some weighting methods also use the IC in the weight evaluation. Graph concepts weighting consist in assigning a weight for each concept in the graph, representing its importance. Several methods compute the weight of a set of concepts. Some are based on a corpus, and others are based only on the graph's structure. The corpus contains instances, where a set of concepts describes each. The graph is an IS-A hierarchy representing the concepts and their relationships. The methods based on a corpus are called extensional methods, and the methods based on the structure of a graph are called intensional methods. In this section we provide an overview of some well-known weighting approaches. We will present two extensional approaches, namely Concept Frequency (CF) and Annotation Frequency (AF), and two intensional ones, namely the Top-Down Topology-based (TD) and the Bayesian approaches. To illustrate the extensional methods, consider an example of ten elements representing the corpus where each one is represented by a set of concepts, defined according to Figure 2.15 as shown in Table 2.2.

Concept Frequency Method (CF)

As mentioned before, the CF method is extensional and based on the frequency of the concepts.

Element	Annotation concepts
e_1	ac_1=(Gastronomy, RegularAcc.)
e_2	ac_2=(FrenchMeal, Accommodation)
e_3	ac_3=(VegeterianMeal, CozyAcc.)
e_4	ac_4=(ThaiMeal, CozyAcc.)
e_5	ac_5=(FrenchMeal, CountryResort)
e_6	ac_6=(VegeterianMeal, CountryResort)
e_7	ac_7=(IndianMeal, CountryResort)
e_8	ac_8=(FrenchMeal, CozyAcc.)
e_9	ac_9=(EthnicMeal, CozyAcc.)
e_{10}	ac_{10}=(ThaiMeal, CountryResort)

Table 2.2: An example of a dataset of elements represented by concepts

Given a concept x, its frequency is the number of occurrences of x or one of its descendants in the elements' annotation concepts divided by the number of occurrences of all concepts in the graph. Even if an abstract concept does not appear explicitly in an element's annotation concepts, the presence of one of its hyponyms indicates its existence. The CF weight of a concept x is defined as follows:

$$CF(x) = \frac{n(x^+)}{N} \tag{2.33}$$

Where $n(x^+)$ is the number of occurrences of a concept x or its hyponyms concepts in the graph, and N is the total number of the concepts occurrences. Considering the graph in Figure 2.15 and the data set in Table 2.2, the concepts weights according to the concept frequency method are represented in the Table 2.3.

Concepts	CF weight
Thing	1
Gastronomy	0.5
FrenchMeal	0.15
tEthnicMeal	0.2
VegetarianMeal	0.1
IndianMeal	0.05
ThaiMeal	0.1
Accommodation	0.5
RegularAcc.	0.05
CozyAcc.	0.4
CountryResort.	0.2

Table 2.3: CF Concept Weights for the Example in Figure 2.15

Annotation Frequency Method (AF)

The AF method is very similar to the CF method in that both are extensional and based on

the concept occurrences. The difference between the two is that the AF method counts the number of annotation concepts that contain x or any hyponym while computing the weight of a concept x. If two hyponyms of x exist in an element's annotation concepts, they will be counted as a single occurrence. On the other hand, the CF method counts the number of occurrences regardless of whether they represent the same element or not. The CF method divides by the number of occurrences of all the concepts, while the AF divides by the number of elements. The formula corresponding to the AF method was introduced in [30, 31], and is defined as follows:

$$AF(x) = \frac{|E_{x^+}|}{|E|} \tag{2.34}$$

Where $|E_{x^+}|$ is the number of elements containing x or one of it hyponyms, and $|E|$ is the total number of elements. The AF weights of the concepts in Figure 2.15 are represented in Table 2.4.

Concepts	AF weight
Thing	1
Gastronomy	1
FrenchMeal	0.3
EthnicMeal	0.4
VegetarianMeal	0.2
IndianMeal	0.1
ThaiMeal	0.2
Accommodation	1
RegularAcc.	0.1
CozyAcc.	0.8
CountryResort.	0.4

Table 2.4: AF Concept Weights for the Example in Figure 2.15

Top-Down Topology-based Method (TD)

The TD method is intensional and based on the structure of the graph. It is a probabilistic approach that stars by assigning a weight equal to one to the root concept and adopts a uniform probabilistic distribution along the ISA hierarchy [19, 30, 32]. The probability of a concept x is computed as follows:

$$TD(x) = \frac{TD(parent(x))}{|children(parent(x))|} \tag{2.35}$$

Where $|children(parent(x))|$ is the number of the direct hyponyms of the parent of concept x. Considering our example in Figure 2.15, the TD weights of the concepts are represented in Table 2.5.

Concepts	TD weight
Thing	1
Gastronomy	0.5
FrenchMeal	0.16
EthnicMeal	0.16
VegetarianMeal	0.16
IndianMeal	0.08
ThaiMeal	0.08
Accommodation	0.5
RegularAcc.	0.25
CozyAcc.	0.25
CountryResort.	0.25

Table 2.5: TD Concept Weights for the example in Figure 2.15

Bayesian Method The bayesian method is intensional and based on the graph structure in addition to conditional probabilities. Starting from the specialization relationship between a node x and its hypernym, the relationship indicates that the weight of x is influenced by the one of its parent, in other words, parent(x) influences x. The authors in [19] define the bayesian weight of a concept x by the probability that x is True (T):

$$w_b(x) = P(x = T) \tag{2.36}$$

This probability is related to the weight of the parent concept of x and the conditional probability $P(x = T|parent(x) = T)$ which is equal to the TD weight of the concept x. The weight $w_b(x)$ of a concept x is computed based on the probability of its parent as follows:

$$\begin{aligned} w_b(x) = P(x = T|parent(x) = T)P(parent(x) = T)+ \\ P(x = T|parent(x) = F)P(parent(x) = F) \end{aligned} \tag{2.37}$$

Where $P(x = T|parent(x) = F)$ is always equal to zero. Therefore, the simplified equation of $w_b(x)$ is the following:

$$w_b(x) = TD(x)W_b(parent(x)) \tag{2.38}$$

The computation of the bayesian weight is also computed in a top-down manner, starting from the root with $w_b(root) = TD(root)$. The bayesian weights of the concepts in Figure 2.15 are represented in Table 2.6

2.3.3 Discussion

We have presented two categories of concepts similarity computation approaches. The first category considers that each concept is represented by a set of features and calculates the similarity based on these features. The second category considers that the concepts are represented

Concepts	Bayesian weight
Thing	1
Gastronomy	0.5
FrenchMeal	0.08
EthnicMeal	0.08
VegetarianMeal	0.08
IndianMeal	0.0064
ThaiMeal	0.0064
Accommodation	0.5
RegularAcc.	0.125
CozyAcc.	0.125
CountryResort.	0.031

Table 2.6: Bayesian Concept Weights for the Example in Figure 2.15

in a hierarchy of concepts. Then, we provided an overview of the techniques that calculate the importance of the concepts, where we presented the ones that calculate the information content of the concepts and those that calculate their weight.

After presenting the usefulness of knowledge graphs to represent the relationships between domain concepts and the calculation of their similarity, we will discuss the limitations of the existing methods and their applicability in our context. Furthermore, we will discuss the limitations of the information content computation and weighting methods, which are included in the majority of graph-based similarity measure equations.

Graph-based similarity measures can be grouped into information content-based and graph structure-based methods.
The position of the compared concepts in the graph is important, and it should be considered when calculating their similarity. Given two concepts, where one is a subsumer of the second one, we think that inclusion is a strong relationship, and they should have a higher similarity than the concepts that are not in this case. The graph-based methods use only the depth and the path length information in the calculation, which does not help to distinguish such case. In other words, if the concepts have the same path length, we will get the same semantic similarity regardless of whether one of them is a subsumer of the another.

The same problem occur for the information content-based methods, where the same similarity value will be computed between two concepts if they have and their subsumer the same information content regardless of their position in the graph.

Information content calculation and the weighting methods are related and aim somehow to capture the importance of the concepts; they can be grouped into extensional and intensional methods. The extensional methods rely on the data, while the intensional methods rely on the structure of the graph. A significant limitation of the weighting methods or the information content calculation is its assignments of static weights, which are inappropriate for long-lived systems with different analytics tasks. Given the same data but various analysis tasks and goals, the extensional methods will give the same weights that rely only on the data. The same holds for the intensional methods that rely on the graph structure, they will provide the same weights

regardless of the analysis task.

Therefore, concerning the similarity calculation, we aim to tackle the limitation where the position of the concepts is not taken into account, where in some cases, the concepts should have more similarity regarding their position in the graph. Furthermore, the assignment of weights to concepts could be performed considering a specific analysis task, and it would be useful to extend the existing approaches so as to provide different weights for different analysis tasks.

2.4 Cultural Heritage Ontologies

Ontologies aim to represent rich and complex knowledge about things, groups of things, and relations between things. In the cultural heritage domain, various ontologies have been developed in various fields, such as museums, libraries, and academies, among others, because of the richness and variability of cultural data.

In order to integrate domain experts knowledge at the BnF in the analysis of the conservation–restoration histories, we have studied the existing ontologies in the field of cultural heritage which conceptualize the expert knowledge in this field. Many ontologies are designed for cultural heritage data [84][4][22][76]. Some works have dealt with the unification of the terminologies used in the field of conservation–restoration. One of these works is the CIDOC–CRM ontology[84] and its extensions CRM_{CR}[4] and CRM_{SCI}[22].

One of the main ontologies in the cultural heritage field is the CIDOC Conceptual Reference Model (CIDOC–CRM) [84]. Its primary role is to serve as a base for the mediation of cultural heritage information, providing the semantic glue required to transform today's disparate, localized information sources into a coherent and valuable global resource. The CIDOC CRM transforms cultural heritage data from internal institutional databases or catalogues into a highly valuable public resource. Mapping cultural institutions' data to CRM will increase the data's relevance and significance by enriching the things represented in the data with new semantics.
This section will provide an overview of some of the well-known ontologies in the cultural heritage and conservation–restoration field by presenting their most essential dimensions and concepts. In addition, we discuss their fitness for our specific context.

2.4.1 CIDOC–CRM

The CIDOC–CRM is an ontology that provides definitions and a formal structure for describing the implicit and explicit concepts and relationships used in cultural heritage documentation and of general interest for querying and exploring such data. These formal descriptions allow the integration of data from multiple sources[1].
The cultural-heritage concepts are represented by entities in CIDOC–CRM. At the top-level of the ontology, we can find two concepts : temporal entity and persistent entity. Persistent entities, such as people, objects, or ideas, are things that last for an indefinite amount of time. The temporal ones, such as events and actions, are constrained by time. Furthermore, the ontology provides a concept hierarchy using the generalization relationship.
The "thing" entity in CIDOC–CRM, for example, refers to things with a fixed shape, which can be natural or man-made, physical or intellectual, and so on. The thing hierarchy is given in Figure 2.17, where the man-made thing is a direct sub-type of thing with the natural attribute removed. Everything that is not natural or intellectual is considered a physical man-made item. This latter concept can be divided into two categories: man-made object and man-made feature, the former of which is physically separate and the later of which is physically associated.

[1] https://www.cidoc-crm.org/

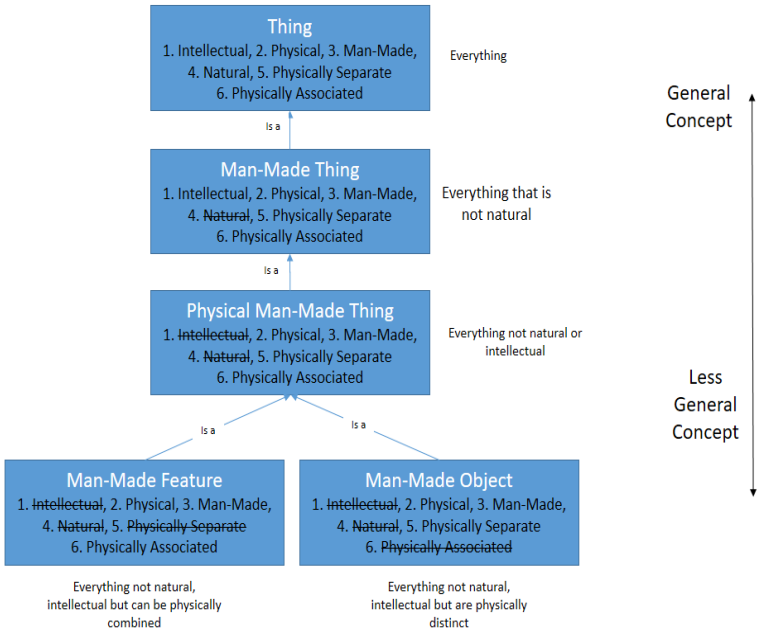


Figure 2.16: Thing Generalisation

The top level of the CIDOC–CRM, which can be used in an integration process, has the temporal entities concept with its events in a central place. The properties of a temporal entity can be within a period, at a specific location, where actors can participate in and affect conceptual objects.

Finally, the CIDOC–CRM facilitates the integration, mediation and interchange of heterogeneous cultural heritage information by providing an uniform representation of concepts in this domain.

2.4.2 CRM–SCI

The CRM SCI (CRM Scientific Observation Model) [22] is a formal ontology that builds on the CIDOC–CRM ontology. It's designed to be utilized in research environments and research data libraries to integrate metadata about scientific observations, measurements, and others. Its main goal is to make managing, integrating, and accessing research data easier by describing by providing a formalization of the concepts related to research environments and the semantic relationships between them.

One of the main concepts in this ontology is the "observable entity" is a subclass of the CRM entity and superclass of the Temporal entity and the persistent item. This class contains instances of E2 Temporal Entity or E77 Persistent Item, i.e. physical entities, their behaviour, states, interactions, or events that may be observed, either directly by human sensory impression or improved using tools and measuring instruments.

Another concept is the "Alteration", it is a subclass of the event concept, and a superclass of

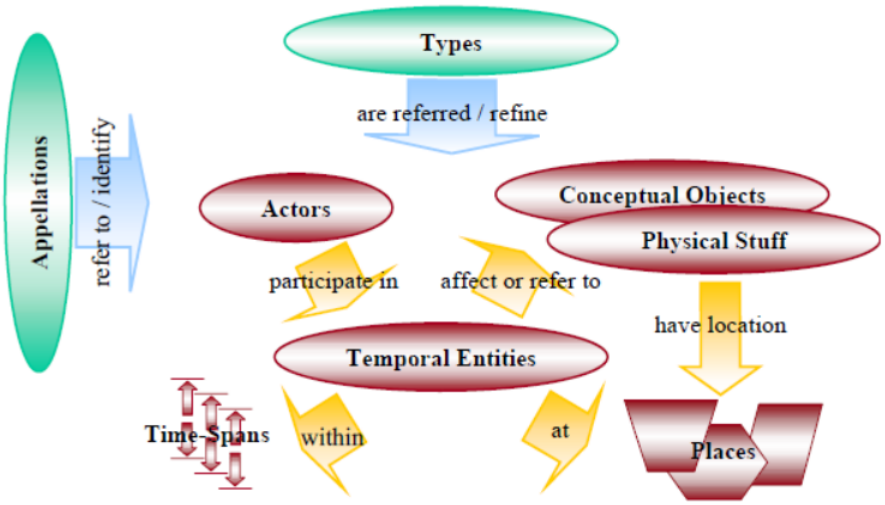


Figure 2.17: CIDOC–CRM top level [23]

modification from CIDOC–CRM. This class includes natural or man-made events that generate, alter, or change physical objects by permanently altering their form or consistency without altering their identity, for example, alterations on depositional features-layers by natural factors or disturbance by roots or insects.

"Physical Feature" is another concept comprises features that are physically attached in an integral way to particular physical objects. This class is a subclass of the physical thing and the place, superclass of the man-made feature, site and segment of matter, and it is equivalent to the physical feature from the CIDOC–CRM.

"Observation" is another class that is a subclass of attribute assignment, and superclass of measurement and encounter event from the CIDOC–CRM. The action of gaining scientific knowledge about specific states of physical reality through empirical data, experiments, and measurements falls under this class. Observation in the natural sciences is described as a type of human activity in which certain Physical Things and their behaviour and interactions are observed, either directly by human sensory impression or improved with instruments and measurement devices, at some Place and within some Time-Span.

2.4.3 CRM–CR

CRM–CR is another extension of CIDOC–CRM. It aims is to support interoperability in the conservation–restoration domain. Cultural heritage institutions manage different types of objects to conserve such as historical monuments in the Research Laboratory for Historical Monuments (LRMH[2]), and artworks held in museums in the French Museum's Research and Restoration Center (C2RMF[3]). The conservation–restoration process generates a large quantity of data. Professionals want to collect and share expertise on the methods employed, the results obtained,

[2]https://www.lrmh.fr
[3]https://www.c2rmf.fr

preventive measures taken, and other relevant information. CRM–CR is an ontological model that represents the concepts related to the conservation–restoration of cultural items and the relationships between them in a unified way.

CRM–CR enables the description of a cultural object based on its fundamental characteristics: identity, physical features and locations, events intervening in the cultural object's life cycle, which can be either degrading or non-degrading, and the consequences of these events. It also enables the description of the instruments used during the events/analysis.

A unified ontology in the conservation–restoration sector can address several issues, including database heterogeneity, which arises from the fact that institutions manage their conservation and restoration data by creating their own databases; as a result, data is represented differently, for example the name of the conservation events could be specific in each cultural heritage institution. Another problem is knowledge incompleteness, which occurs when a database at one institution is lacking information that exists in the databases of another institution and may be transferred between them. CRM–CR is related to the CIDOC–CRM and CRM–SCI ontologies. It contains new concepts that specialize those of CIDOC–CRM. For example, the "Scientific studies and results" concept is a specialization of the concept "E7 Activity" in CIDOC–CRM, and a generalization of the concept "S4 Observation" in CRM–SCI. The top-level concepts of the CRM—CR ontology and the relationships between the two ontologies are shown in Figure 2.18. Furthermore, new notions such as the cultural object, specific events (alterations, scientific studies, documentations, interventions), instruments utilized during events, and scientific studies results are described.

2.4.4 Discussion

After presenting the existing ontologies in the cultural-heritage and conservation–restoration fields, we can observe that the extensions of the CIDOC-—CRM are linked to it in different levels and dimensions. Various ontologies from different domains can be linked in this way. The general goal of CIDOC-—CRM is describing temporal entities, events, actors, etc. CRM—SCI adds new ideas in scientific observation and describe the observation itself, measurement, change, and the observation entity, among other things. CRM—CR might be considered a supplement to the CRM—SCI aiming to cover the conservation–restoration domain. This ontology includes terms to describe the conservation–restoration processes as well as the strategies employed. These ontologies describe general concepts related to the cultural heritage and conservation–restoration fields. In some institutions, it is possible that the terminology used to describe the concepts is different and it is impossible to link directly their data to the existing ontologies. A gap to fill always exists if an institution finds that its data can not be linked directly with the existing ontologies. Therefore, new ontologies representing the institution concepts can be added with links to the existing ones in other ontologies to fill this gap and integrate the institution data.

In our context at the BnF, the terminology used to describe the events in the conservation–restoration histories is very specific. None of the existing ontologies includes the used terminology to describe the conservation–restoration histories at the BnF. Therefore, it is impossible to integrate the domain experts' knowledge describing the events and their relationships in the

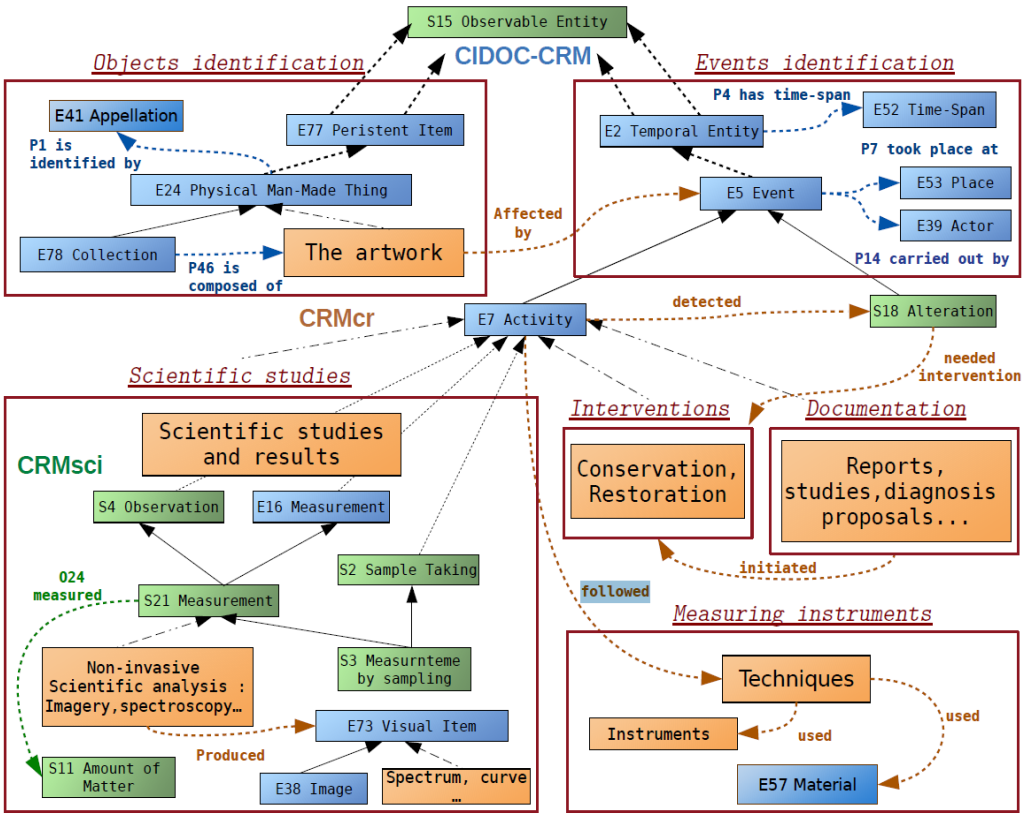


Figure 2.18: CRM–CR and its relation to CIDOC–CRM and CRM–SCI [5]

analysis process using the existing ontologies.

2.5 Conclusion

This chapter includes three parts : A part on the trajectory analysis, a part on the similarity computation using external knowledge, and a part on the cultural heritage ontologies. We have studied the trajectory data preprocessing such as their representation and filtering. In addition, we have studied different trajectory analysis tasks such as the calculation of their similarity and the extraction of knowledge from trajectories. We discussed the different possible representation of the trajectories that can be spatial or semantic depending on the context and the available data. Concerning the similarity measures, we gave an overview on different measures that compute the similarity between sequences, strings or trajectories. The selection of the appropriate similarity measure also depends on the context. In addition, we presented the widely known mining algorithms that can be used on the trajectories. In addition, we discussed how external knowledge has been represented by knowledge graphs and integrated into the computation of the similarity between concepts. Also, we gave an overview of the weighting methods that give more importance to the relevant concepts. Finally, we gave an overview of the graphs in the cultural heritage context in which we develop our work on the analysis of

conservation–restoration histories.

After presenting the different measures used to calculate the similarity between the trajectories, we will mention their limitations and possible extensions. The first category, sequences similarity measures, contains measures requiring a predefined distance metric to check if two elements can be matched. The semantic trajectories are sequences of concepts, and it is unclear how to compute their distance. The presented similarity measures match only the identical ones, i.e. concepts with the same names, and they will miss much semantic information. For example, some concepts can be similar regardless of their names being different. The same idea for the second category, string similarity measures that match only the identical characters. Only identical concepts will be matched if we want to extend these methods to be used on semantic trajectories. Therefore, we can observe that more semantics should be integrated when extending these methods on semantic trajectories. Considering the semantic relationships between the concepts is essential to match the concepts with different names but having some similarities. We gave an overview in section 2.3 on knowledge-based similarity computation methods. These methods can help add more semantics to the matching process. The first aspect that this work target is integrating the experts' knowledge in the computation of the similarity between the semantic trajectories. In other words, we aim to adapt and integrate a knowledge-based similarity computation method in matching semantic trajectories. The knowledge experts will help define the similarities between concepts and refine the similarity measures with this information. Therefore, we introduce in chapter 3 the integration of domain experts' knowledge in matching the semantic trajectories.

Different mining approaches were presented in section 2.2.3, such as clustering, classification and prediction approaches. The goal of these approaches is to extract knowledge from the trajectory data. One possible improvement to existing approaches would be to introduce expert or domain knowledge into the analysis process. External knowledge could be beneficial in different tasks in an analysis pipeline. Therefore, we introduce in chapter 4 an analysis pipeline that analyses the conservation–restoration histories and that takes into account external knowledge representing the domain experts' knowledge.

In section 2.3.2, we gave an overview on the methods that calculate the importance of the concepts, where we presented methods that calculate the information content, and other methods that evaluate the weight of the concepts. All the existing concept weighting approaches assign static weights to the concepts, regardless of the analysis task. In other words, when changing the analysis task, the methods give the same weights as they depend only on the data or on the concepts' hierarchy structure. Some methods used to compute the information content of the concepts are divided into five groups: the probability inverse, hyponym-based, hyponym and depth, leaves, and leaves and subsumers. Other methods calculate the weight of the concepts and are grouped into groups which are the Concepts Frequency (CF), Annotation Frequency (AF), Top-Down Topology-based (TD), and Bayesian methods. In addition, the weighting methods could be divided into intensional and extensional ones, where the former

depends on the structure of the knowledge graph, and the latter depends on the data. The main limitation of the methods that compute the information content or the weight is that they do not rely on the analysis task. When changing the goal of the analysis, the methods will give the same results because the structure and the data are the same. For these reasons, we introduce in chapter 5 a novel approach for graph weighting based on the analysis task.

Finally, in the cultural heritage field, the work to link all the institutions' data is in the beginning, and a lot of contribution is needed to comprise all the contexts in this field. We provided in section 2.4, an overview of some ontologies that exist in the cultural heritage field in which we test our analysis propositions. The ontologies do not cover the terminology used in our context, especially the concepts' names in the semantic trajectories. Therefore a new graph is needed to represent the knowledge and the relationships between the concepts.

In the upcoming chapters, we will present our contributions to integrating the domain experts' knowledge in matching and mining semantic trajectories and the knowledge-based weighting regarding the analysis task.

3 - Semantic Trajectories Matching

Contents

3.1 Introduction

Computing the similarity between trajectories is a key task for trajectory analysis. For example, to analyse the peoples' movements represented as trajectories in a city aiming to extract common behaviours called patterns, a comparison between the trajectories is required to detect those interested in the same places and who follow a similar path. Trajectory matching is the process of calculating the similarity between trajectories. When comparing two trajectories composed of a sequence of elements each, the matching is performed by comparing each pair of elements constituting them.

As mentioned in section 2.2.1, there are two types of trajectories. Spatial trajectories, which are sequences of locations, and semantic trajectories, which are sequences of semantic

elements. There are several ways to measure the similarity between spatial trajectories, but all of these measures rely on the spatial distance between one location and another. In the case of semantic trajectories, the elements that constitute them are not characterized by geographical coordinates, and they should be compared using suitable measures. In addition, hidden relationships between the elements of the trajectories may exist, and these can only be identified by domain experts. For example, suppose that the elements of the trajectories are events, and suppose that despite having different names, they are still very close, according to the domain experts. An automatic matching process would not succeed in identifying the closeness between these events, and they will not be considered as matching ones.

In this chapter we are interested in computing the similarity between the conservation–restoration histories of the documents at the BnF in order to predict their physical state. The data describing the different services at the BnF are distributed. Our first objective is to identify, in the different databases, the relevant information related to the documents' physical state and integrate these data into a conservation–restoration history associated to each document, and containing all the data related to its physical state.

We will first present the process of identification, extraction and integration of this relevant information to create the conservation–restoration histories, and we present in this chapter a representation of these histories which will be used in the different analysis tasks presented in chapter 4.

A critical task in these analyses is comparing and calculating the similarity between the conservation–restoration trajectories, which we represent as sequences of semantic elements. As the information about the elements constituting these trajectories have been extracted from different databases, they are usually expressed using different terminologies. This heterogeneity is due to the evolution of the terms and codifications over time. For example, data stored for a long period of time, sometimes years, uses different terminology than data inserted more recently. The first challenge facing similarity computation is to resolve the heterogeneity of the terminologies used in the different databases.

Another challenge raised by matching the elements constituting semantic trajectories is that some elements may have hidden common characteristics and hidden relationships. For example, assume that two documents D_i and D_j have undergone a conservation process labelled by the BnF as "A1 Réemboîtage, dans couverture d'origin" and "A2 Réemboîtage, dans nouvelle couverture" respectively. Although the two elements seem different, they actually have the same purpose and share some common characteristics. It is possible that the relationships between the elements and their common characteristics are not described anywhere in the databases; they are known only to the domain experts. In this work, we assume that the domain knowledge specific to the conservation and restoration field is available and formalized as an ontology. The problem we are interested in is how to inject the knowledge provided by this ontology in the similarity computation, and consequently in the analysis process.

In this chapter, we will present an approach to match conservation-restoration trajectories taking into account external knowledge representing the semantic relationships between the elements that constitute these trajectories. The rest of this chapter is organized as follows. The problem statement is provided in section 3.2. We present the extraction of the relevant

information from the BnF data sources, and the creation of the conservation–restoration histories in section 3.3. We present the representation of the conservation–restoration trajectories is section 3.4. The events matching using external knowledge is presented in section 3.5. Section 3.6 presents the evaluation of semantic trajectories similarity. The domain knowledge used in our approach is described in 3.7 and our experiments are presented in section 3.8. Finally, we conclude the chapter in section 3.9.

3.2 Problem Statement

The BnF stores more than twenty million of documents, which are described by information registered in different databases. These documents are associated with different events such as the degradation observed on them or the treatments they have undergone to keep them in a good state. Various information related to the documents' physical state should be considered when performing the analysis tasks aiming to predict their physical state. These information are part of the of the documents' conservation–restoration histories.

Consider that each document is described by a conservation–restoration history extracted from the BnF databases and stored in an integrated database. The first problem we faced is how to represent these histories? What is the suitable representation of these histories that is adequate for future analyses to predict the physical state of a document? Consider a set of documents DC=$\{d_1, d_2..., d_n\}$. For each document, we aim to associate a conservation-restoration history that contains all the relevant data related to its physical state and all the relevant events that have happened over time. We represent these histories as conservation–restoration trajectories. Each document d_i will be represented by a trajectory Tr_i representing its conservation restoration history. Each trajectory is a sequence of conservation-restoration elements. One problem when building such trajectories is the identification of the different types of elements relevant to characterize a document's conservation–restoration history.

Given a set of conservation restoration trajectories representing the documents, several analysis tasks can be performed, all of them relying on the comparison of distinct trajectories. One key issue in our context is to perform such comparison considering the heterogeneity of the terminology used to describe conservation–restoration elements in distinct data sources and the integration of the domain's experts knowledge in the comparison process. This leads to the second problem tackled in this chapter, which can be stated as follows: how to evaluate the similarity between two conservation-restoration trajectories corresponding to distinct documents taking into account the terminology heterogeneity? To this end, two similarity functions should be defined:

- $Sim_e(e_i, e_j)$, which evaluates the similarity between two elements e_i and e_j,
- $Sim_s(Tr_i, Tr_j)$, which calculates the similarity between two trajectories Tr_i and Tr_j representing the conservation histories of the documents d_i and d_j respectively.

The elements constituting the conservation–restoration histories may have been recorded several decades ago. Obviously, the names and types of the elements involved in a history can be highly different according to the time they have been recorded. Moreover, the lack of a uniform and standardized vocabulary makes the terminology used in naming the elements very heterogeneous. Evaluating the similarity between elements and trajectories has to take into account the possible semantic relationships between elements. For example, a conservation process recorded for a document twenty years ago may have a different name to an another process recorded this month, but these two processes might still be very similar, which could only be identified by a human expert. In our approach, we assume that the knowledge of conservation-restoration experts exists in a knowledge base. Therefore we aim to integrate this knowledge in the similarity functions Sim_e and Sim_s. In the following sections, we will introduce the identifying of the relevant information in the BnF databases and the creation of the conservation–restoration histories, their representation as conservation–restoration trajectories, then we will present our proposal to evaluate both the similarity between elements and between trajectories.

3.3 Identifying the Relevant Information and Creating Conservation–Restoration Histories

The available data sources at the BnF, which provide data about documents, are created and maintained by different departments, and these sources provide data covering different, possibly overlapping, aspects of document restoration and conservation. For example, the collections department database provides data describing the communication of the documents to the readers requesting them, and the observed degradations on these documents, while the conservation department's database stores data about the conservation and restoration processes performed on the documents to keep them in a good physical state. All the databases have a different structure, different terminologies and attributes. We aim to extract the relevant information from these databases. In the following, we will present, in section 3.3.1, the BnF databases analysis aiming to extract the information that have an impact on the documents' physical state. Such information should be part of further analysis, thus should constitute the conservation histories. In section 3.3.2 we present the integration of the relevant data.

3.3.1 Identification of Relevant Information

To find the information that is related to the documents' physical state and could be used for analytical tasks, we have started by analysing the databases in the different departments.

The databases contain information about different aspects such as the characteristics of the documents, the history of conservation–restoration treatments and the communications with readers. The goal of the analysis is to extract all the available information that can help us to characterize a document and its physical state at a given point in time.

We have studied the existing databases in order to assess their content. Figure 3.1 shows the percentage of the null values in a table that contains information about the documents' physical characteristics. As we can see in the figure, the completeness of the attributes in the table is not

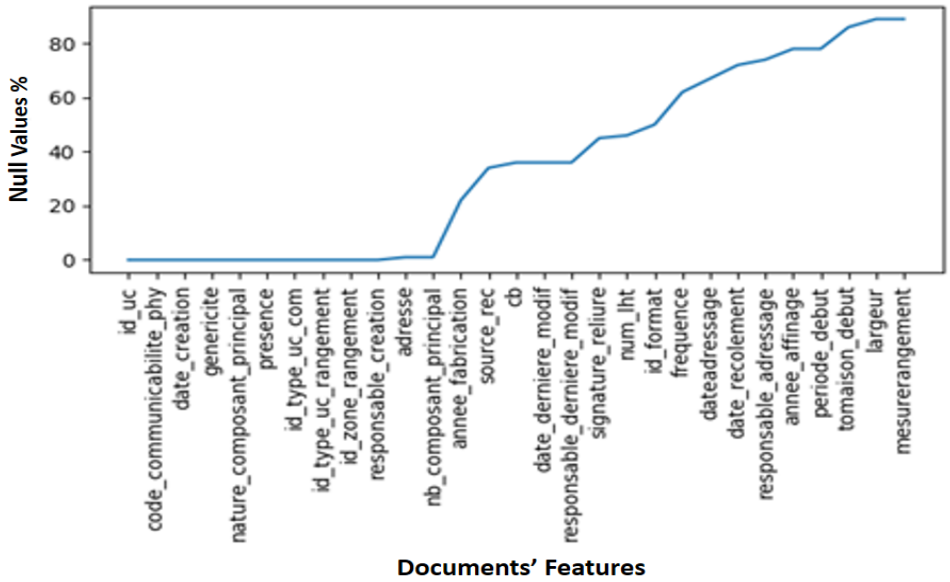


Figure 3.1: Completeness of the data describing the documents physical characteristics

very high. In addition, some of the characteristics are important as they represent information related directly to the documents physical state and could be useful for further analysis such as the paper type, while other features are not relevant.

In this analysis, we have found three types of information with high completeness, and which are relevant to be in a conservation–restoration history. The first one is related to the communication of the documents to the readers, the second one is related to the past degradations recorded for the documents, and the last one describes the past interventions on the documents.

The first dimension of the analysis is the communication of documents. We have analysed the documents average number of communications over different periods of time, and the result is shown in figure 3.2. The results show the documents' communication average according to time periods, which highlights the interest of the readers in the 21st century and a slight increase in interest between 1841 and 1860. At the same time, and by analyzing the physical state of the documents depending on their publication date, we have identified possible correlations or impacts of the communication on the physical state. Figure 3.3 shows the percentage of the out-of-order documents by year of publication. Starting from 1880, we can see a correlation between the two graphs, where the communication requests increase and the percentage of out-of-order also increases. Therefore, we can consider that the communication of the documents should be a part of the conservation–restoration histories.

The other important information we have identified are the past interventions and the degradations on the documents. Where the interventions are the conservation–restoration processes done on the documents to conserve their physical state. This information is distributed on different databases. The old history of interventions, i.e., before 2013, is stored in a separate

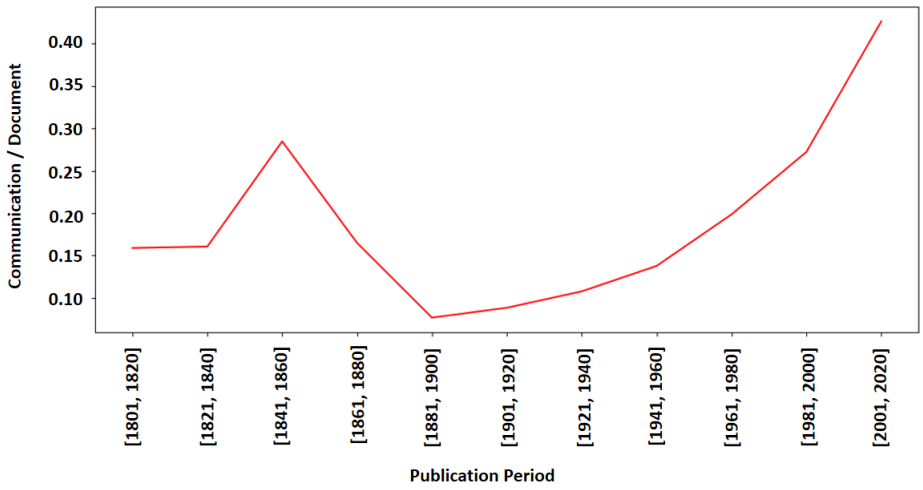


Figure 3.2: Documents' communication average by publication time period

database from the history of interventions after 2003. In the following sections, we present the creation of a database that integrates all the identified important information.

3.3.2 Creation of the Conservation–Restoration Histories Database

We have analysed 30 different databases in the different departments at the BnF containing more than 300 tables in total. We have built an integrated database containing the relevant data that has to be taken into account when building the trajectories describing the conservation-restoration histories. Figure 3.4 shows the tables of the integrated database. The database contains information about almost twenty million documents. The core table of this database (Document) includes information on the documents and their physical characteristics. Each tuple in this table represents a document having an ID as an identifier and characterised by different attributes such as the title, publication date and the author. In addition, three other tables, communication, degradation, and treatment are integrated into this database and linked to the documents. Each row in the communication table represents one request of a reader to access a document. The attributes in this table represent information related to this request, such as the reader, the requested document, the date and the place. Each row in the degradation table represents a degradation event on a specific document. The attributes represent information such as the type of degradation represented by the "description" attribute and the document on which the degradation was detected. Each row in the conservation process table represents a conservation–restoration event and its characteristics, such as its type and the document on which it was performed.

The information in the communication table is extracted from the collections department database. The collections department tracks the documents' communication by storing information about the readers' requests to access the hard copy of the documents. In addition to

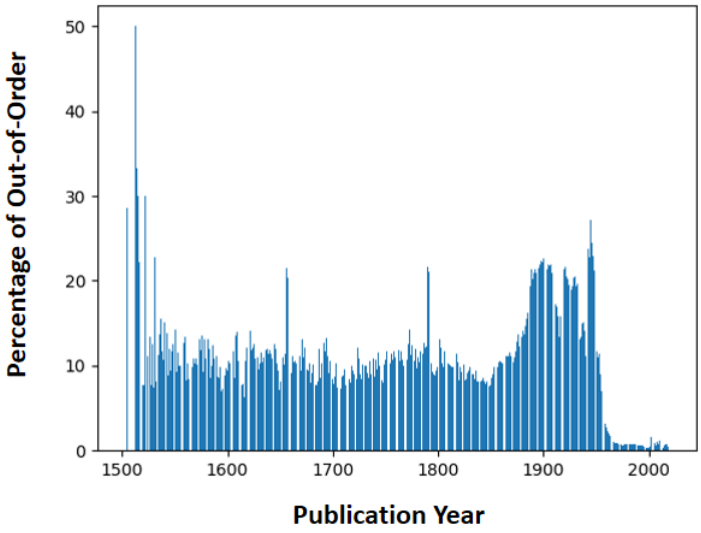


Figure 3.3: Out-of-Order by publication year

that, more data related to the request are stored, such as the status of the request, and the date and the reader identification. The information about the degradation of the documents is also managed by the collections department, but its information is stored in a separate database. The database contains information about the old physical state degradations of the documents, where the domain experts define more than two hundred types of degradation. The stored data describes these degradations, such as the type of degradation, the damaged part of the document, and the date of the detection. The information about the treatment of the documents is managed by the conservation department. The data related to the treatment describes the type of the conservation–restoration process that is accomplished on the document, its date, and the service provider.

3.4 Representation of the Conservation–Restoration Histories As Semantic Trajectories

The conservation–restoration histories contain three types of information: communication, degradation, and the conservation–restoration process. All this information will be represented as timestamped events. We represent the conservation–restoration histories as semantic trajectories consisting of sequences of events, which can be related either to the communications of the document to the readers, to the degradations, or to the conservation–restoration processes. We define a conservation-restoration trajectory as follows:

Definition 3 *Conservation-Restoration Trajectory.*
A conservation-restoration trajectory is a sequence of events of this document ordered by their time. The sequence of events corresponding to a document doc_i *is denoted* Tr_i*. It is such that* $Tr_i = [e_1, e_2, ..., e_k]$*, where each* e_i *is an event described by a triple* $< type_i, name_i, time_i >$*.* $Type_i$ *represents the type of the event,* $name_i$ *represents the designation of the event and*

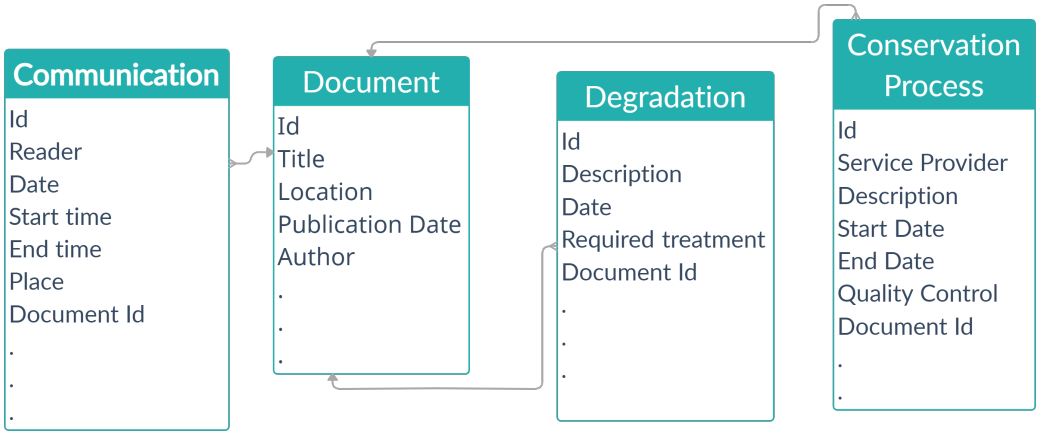


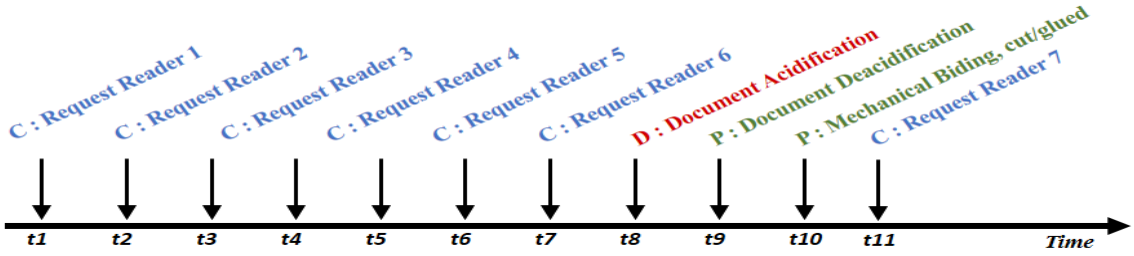
Figure 3.4: Integrated database core tables

$time_i$ represents the time at which the event occurred. The possible values for $type_i$ are P, D or C, corresponding respectively to a conservation process, a degradation or a communication event.

The value of $name_i$ in a trajectory depends on the type of the corresponding event. If $type_i$ = P, then the value of $name_i$ is the designation of the specific conservation process performed on the document. If $type_i$ = D, then the value of $name_i$ is the name of the degradation observed on the document, and if $type_i$ = C, the value of $name_i$ is the level of communication of the document, defined as low, average or high. Obviously, the events are ordered in the sequence according to the ascending order of $time_i$.

An exploration of the integrated database has showed that there were about 150 different conservation-restoration processes which could be performed on a document, and 250 different types of degradations observed by experts on the documents composing their collections.
In order to analyse conservation data, each document will be represented as a conservation-restoration trajectory consisting of a sequence of events that have happened during the document's life. These events can be either a conservation process performed on a document or a degradation observed on a document, characterized by a specific designation. It can also be related to the number of readers who have accessed the document, which can obviously affect its physical state. Therefore, we also consider a specific kind of event capturing the extent to which the readers have requested this document. We are not interested in the exact number of times a document has been requested, we would like instead to characterize the extent to which a document has been requested. We can do so by assigning three distinct values to the communication event: low (≤ 4), average (between 4 and 8) and high (≥ 8), these values have been provided by the domain experts.

Figure 3.5 shows an example of history of events for a given document consisting of several readers requests, followed by an "$Acidification$" degradation observed on the document, then two conservation processes "$Deacidification$" and "$Mechanical\ Binding$" and finally a request from a reader. We can see at the bottom of this figure how this history of events



Tr = [< C, "average", $t_1 - t_7$ >,
< D, "acidification", t_8 >,
< P, "deacidification", t_9 >,
< P, "mechanical binding, cut/glued ", t_{10} >,
< C, "low", t_{11} >]

Figure 3.5: Example of History of Events for a Document and the Corresponding Trajectory

is represented as a trajectory Tr according to our definition. Tr is composed of a communication event for which the name value is "average", representing the six requests from the readers in the beginning of the history, then a degradation event, "$Acidification$", followed by two conservation processes, "$Deacidification$" and "$Mechanical\ Binding$", and finally a communication event for which the name value is "low", corresponding in the example to the request of Reader 7.

3.5 Evaluating Events Similarity

The analysis of the documents' conservation–restoration histories represented as conservation-restoration trajectories composed of sequences of events requires the evaluation of the similarity between each pair of trajectories, which in turn requires the matching of pairs of events, taking into account the terminological heterogeneity of their description.
When matching trajectories, if two compared events have the same name, they will be considered as the same event. Because of the different terminology used in the databases, some events may have different names, but they still could be identical, according to the domain experts. Therefore, only the domain experts could detect a match between such events. Integrating the domain experts in the matching process could be very useful when matching events with hidden similarities. For this reason, we propose to integrate the experts' knowledge as an external source to tackle this challenge when comparing the conservation–restoration trajectories.

If the two compared events have different types, i.e. degradation, conservation process or communication, then they can not be considered as matching events. If the two compared events have the same type, and if this type is either P or D, corresponding respectively to a conservation process or a degradation, then their names are compared to determine if they match or not. If these names are identical then the events are matching ones. But if the names are different, this does not mean that the events do not match. Indeed, the terminologies

used in the description of either the degradations or the conservation processes may differ, or expressed at different levels of detail according to the source they have been extracted from.
In order to overcome this heterogeneity, we suppose that the semantic relationships between the elements constituting the trajectories are represented in a knowledge graph in the form of an ontology, where each event is represented by a concept and their relationships are represented by links between the concepts. We propose to use this ontology in the process of similarity computation. Assuming that each event in a trajectory corresponds to a concpet in the ontology, the idea is to compute the similarity between the two events based on the relationship between the two concepts.

In our work, we have initiated an ontology, called CRM_{BnF} that contains concepts representing all the events that constitute the conservation–restoration trajectories. Figure 3.6 shows an excerpt of this ontology, initiated in close collaboration with domain experts at the BnF. The figure shows the relations between CRM_{BnF} and three existing ontologies, CIDOC-CRM[84], CRM_{CR}[4] and CRM_{SCI}[22], which nodes are represented in green color and dotted lines. We present the process of creating this ontology in section 3.7. Each concept represents either one of the 150 existing conservation-restoration processes or to one of the 250 observed types of degradations. This ontology is expressed using the languages proposed by the W3C [1] Consortium, RDF/S [2] and OWL [3]. Figure 3.6 shows the representation of a subset of both the conservation processes and the degradations related to documents at the BnF. We have identified twenty subclasses for the "BnF:Conservation Process" class; two of them, "BnF:Short Maintenance" and "BnF:Consolidation", are represented in the figure. We have identified twenty-two subclasses for the "BnF:Degradation" class, one of them, "BnF:Headband", is represented in the figure.
The greater the distance between the node and the root, the more precise the name of the conservation process or the degradation corresponding to this node. The ontology was created to provide a unified vocabulary for the events in a conservation history and to help identify the similar ones beyond their terminological heterogeneity. The subclasses of the "BnF:conservation Process" class represent the most generic conservation processes; we refer to these as semantic categories. In the same way, we consider that the twenty-two direct subclasses identified for the "BnF:Degradation" class are also semantic categories. The comparison of two events e_i and e_j is performed by computing a similarity score using the domain ontology. Therefore, to calculate the similarity between the events using the ontology, we distinguish between different cases, and for each we characterize the type of the relationship between the events and the corresponding similarity score.
In the following, we define the relationships between the concepts in section 3.5.1, and we present event similarity in section 3.5.2.

[1] https://www.w3.org/2001/sw/wiki/Main_Page
[2] https://www.w3.org/RDF/
[3] https://www.w3.org/OWL/

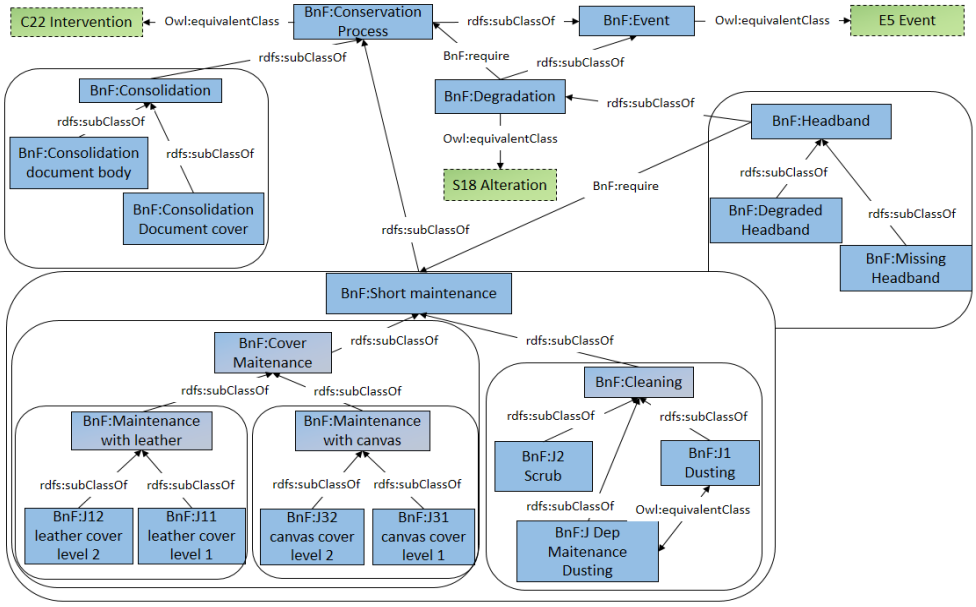


Figure 3.6: Excerpt of the CRM–BnF Ontology

3.5.1 Defining Concepts Relationships

Considering an ontology that contains concepts representing all the existing events in the trajectories, we propose to evaluate the similarity between the events according to the relationship between their corresponding concepts in this ontology. In addition, we propose to consider the position of the concepts in the ontology to define their relationship.
If we consider the case of two events corresponding to two concepts linked by an equivalence relationship in the ontology, these events are equivalents and should be matched. Now consider another example in which on of the two concepts is a subsumer of the other, the events should not be considered as unmatching events as one of them contains all the characteristics of the other.

We have distinguished between five distinct cases depending on the concepts positions in the ontology when comparing two events. Let us consider the concepts corresponding to two events $e_i =< type, name_i, time_i >$, and $< type, name_j, time_j >$ respectively in the ontology. Let us denote Co_i and Co_j these two concepts. The similarity score is computed considering the relative position of Co_i and Co_jin the ontology, and we can identify five different cases describe below:

- **Case 1:** The two concepts Co_i and Co_j are identical, then the two events e_i and e_j correspond both to the same concept in the ontology.

- **Case 2:** The two concepts Co_i and Co_j are such that there is a path P_{ij} between them where all the edges in the path correspond to $owl : equivalent\ Class$ properties. For example, the two concepts "BnF:J Dep Maintenance Dusting" and "BnF:J1 Dusting" in

figure 3.6 fit this case.

- **Case 3:** The two concepts Co_i and Co_j are such that there is either a path P_{ij}=[(Co_i, P_1, Co_1), (Co_1, P_2, Co_2), ...(Co_k, P_{k+1}, Co_j)] or a path P_{ji}=[(Co_j, P_1, Co_1), (Co_1, P_2, Co_2), ...(Co_k, P_{k+1}, Co_i)] where for all the edges in the path, the property P_x is the $rdfs : subclassOf$ property for $1 < x < k + 1$. In this case, the two events e_i and e_j are of the same nature, but one of them is more specific than the other. For example, the two concepts "BnF:J32 canvas cover level 2" and "BnF:Cover Maintenance" in figure 3.6 fit this case.

- **Case 4:** The two concepts Co_i and Co_j are such that there is a concept Co_k in the ontology for which the following two conditions hold: (i) both Co_i and Co_j are included in Co_k, and (ii) Co_k is either a semantic category or is included in a semantic category. For example, the two concepts "BnF:J31 canvas cover level 1" and "BnF:J32 canvas cover level 2" in figure 3.6 fit this case.

- **Case 5:** The two concepts Co_i and Co_j are such that the nearest common ancestor is either the concept "BnF:Conservation Process" when the two concepts have a $type$=P or the concept "BnF:Degradation" when the two concepts have a $type$=D. In other words, the two concepts do not belong to the same semantic category. For example, the two concepts "BnF:J31 canvas cover level 1" and "BnF:Consolidation" in figure 3.6 fit this case.

Based on the previous cases, we will define the relationships between the concepts. We distinguish between four types of relationships between the events: equivalence (case 1 and case 2), inclusion (case 3), closeness (case 4) and dissimilarity relationships (case 5). In the following, we provide the possible relationships, and for each, we show how to discover them in a given ontology.

Equivalence Relationship.

Let us start with the relationship expressing the highest similarity level, the equivalence. When it exists between two events, this relationship means that they are the same and that they have the same characteristics, according to the experts' opinion.

As mentioned before, this relationship exists between two events if their corresponding concepts fit in the case 1 or 2 described above. In other words, if the two events are correspond to the same concept in the ontology (case 1), or if there is a path of equivalence relationships between the two concepts representing the events (case 2).

Figure 3.7 (b) shows an example of the equivalence relation between the two concepts "BnF:J1 Dusting" and "BnF:J Deep Maintenance Dusting", expressed by the OWL:equivalentClass connection between the two concepts. It is also possible that the path length between the two concepts is higher than one, i.e. more than one OWL:equivalentClass connection exists to reach a concept starting from the another.

Inclusion Relationship

An event e_i is considered included in another event e_j if the former is a particular case of the

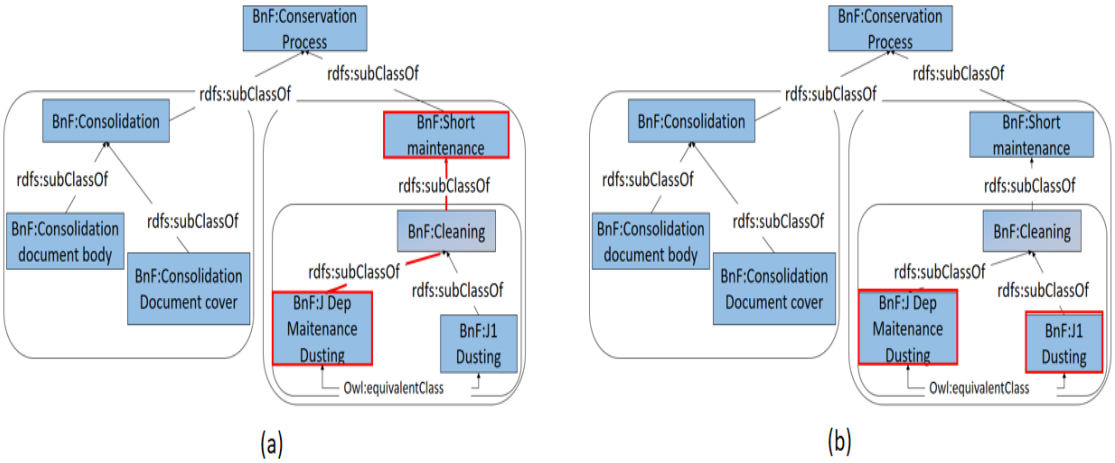


Figure 3.7: Inclusion (a) and Equivalence (b) relationships between concepts

latter. In other words, e_i is considered included in e_j if the corresponding concept to e_i in the ontology is a sub-concept of the concept corresponding to e_j. In addition, the connection between the two concepts can be indirect, i.e. the length of the path between the concepts could be higher than one. Figure 3.7 (a) shows an example of the inclusion relation between two concepts where it is represented by the rdfs:subClassOf connection between the two concepts.

Closeness Relationship

An event e_i is considered close to another event e_j if these events are represented by two concepts that have a common super concept in the ontology. Consider the events represented by the concepts c_1 and c_2, respectively. In addition, let Consider the set of concepts that include the two concepts c_1 and c_2 are $SuperC1$ and $SuperC2$, respectively. When $SuperC1$ and $SuperC2$ have common elements, this means that c_1 and c_2 have a same hypernym. We mention that to consider that closeness is the relationship between two events, the conditions of the equivalence and the inclusion relationships should not be valid, and that because in these two cases the concepts have also common super concept.

In the case where the concepts hierarchy is a tree, all the concepts are included in the same concept, which is the root, and consequently, all the concepts will be considered to have a closeness relationship. Therefore, more constraints should be added, such as the paths' maximum length between the concepts in the ontology. In this work, we consider two concepts are close if they are included in the same semantic category, and the semantic category is defined as follows:

Definition 4 *Semantic Category. A semantic category is a sub-graph of the ontology that starts from a concept and contains all of its sub-concepts. The semantic categories can not*

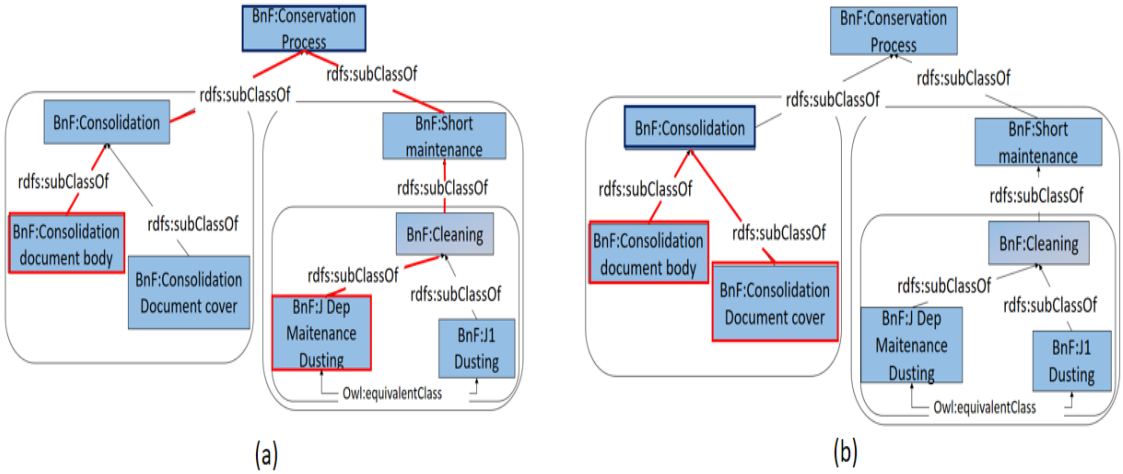


Figure 3.8: Dissimilarity (a) and Closeness (b) relationships between concepts

intersect. The semantic categories in this work starts from the concepts that are directly sub-concepts of the conservation process or the degradation concepts.

Figure 3.8 (b) shows an example of the closeness relationship between the two concepts "BnF:Consolidation document body" and "BnF:Consolidation Document cover".

Dissimilarity Relationship

The last relationship is when two events are dissimilar, and they do not share any common characteristics. This relationship is valid when none of the previous relationships is valid. In other words, this relationship is valid when the concepts representing the two events are in two different semantic categories. Figure 3.8 (a) shows an example of the dissimilarity relation between the two concepts "BnF:Consolidation document body" and "BnF:J Dep Maintenance Dusting".

3.5.2 Event Similarity Score

The similarity between two events is related to the position of their corresponding concepts in the ontology. Different information could be derived from the concepts' positions such as their relationship type and the distance between them. The distance between two concepts in a graph is determined by the shortest path of a precise relationship.

In our work, the similarity score ranges between 0 and 1, and we propose to distinguish between the importance of each relationship by defining a ranking between the different types of relationships defined in the previous section. The highest score is reached if the concepts

are identical or equivalent. The lowest score is reached when the concepts are dissimilar. We consider that the similarity score for the inclusion relationship should always be higher than the similarity score of the closeness relationship.

Furthermore, when there is an inclusion or a closeness relationship between two concepts, we consider that the path length between the two concepts should be reflected in the similarity score: the shorter the path, the higher the score. The similarity score corresponding to each of them is defined as follows:

Definition 5 *Similarity Score. Consider two events e_i and e_j corresponding to two concepts C_i and C_j respectively.*

- ***Equivalence.*** *The similarity score for two equivalent events is equal to 1.*
- ***Inclusion.*** *The similarity score for two events e_i and e_j, such that C_i is included in C_j is comprised in the range $[\alpha, 1[$ and defined as follows:*
$$1 - (1 - \alpha) \times \frac{|P_{ij}|}{depth(CRM_{BnF})}$$
where $|P_{ij}|$ is the length of the path P_{ij} between C_i and C_j, and the depth function return the length of the longest inclusion path in CRM_{BnF}, i.e. the longest path with $rdfs : subclassOf$ properties.
- ***Closenesss.*** *The similarity score for two events e_i and e_j, where they corresponding concepts are both included in a third concept C_k is comprised in the range $]0, \alpha[$ and defined as follows:*
$$\alpha - \alpha \times \frac{(|P_{ik}| + |P_{jk}|)/2}{depth(CRM_{BnF})}$$
- ***Dissimilarity.*** *If this relationship holds between two events, the similarity score is equal to 0.*

The equations of the inclusion and the closeness depend on two parameters. The first one is α, which is an arbitrary value which purpose is only to represent the total order relation defined between the scores corresponding to equivalence, inclusion, closeness and dissimilarity relationships.
The second factor is the relative size of the path between the two compared concepts to the depth of the ontology. When the two concepts are the farthest in the ontology, i.e. longest possible path, this means that they are the less similar among all the concepts along this path, and they should have the minimum possible similarity score.

To illustrate why we are interested in the relative size of the path between the two compared concepts to the depth of the hierarchy, consider the hierarchy in figure 3.9. The depth of the hierarchy is equal to four, and the relationship between all the concepts is inclusion. It is obvious how the "Conditioning with custom light box with filler" concept share more common characteristics with the "conditioning with box" than the concept "treatment". Using the

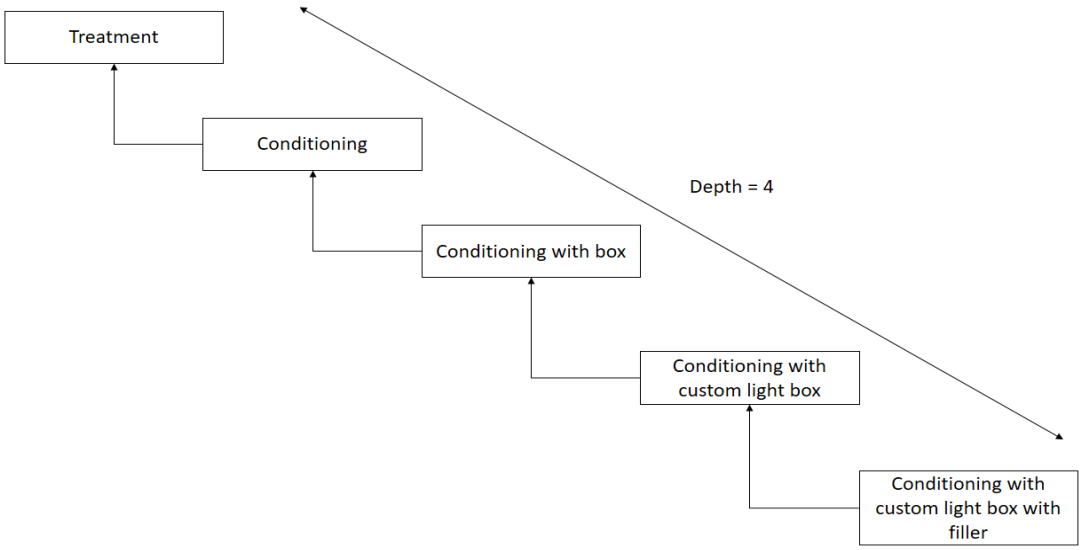


Figure 3.9: Hierarchy of conservation concepts

predefined equation to compute the inclusion similarity, its similarity to the two concepts will be equal to 1-(1-α)$\times\frac{2}{4}$= $\frac{\alpha+1}{2}$, and α respectively. With $\frac{\alpha+1}{2} > \alpha$ when $0 < \alpha < 1$.

The similarity score between two events will be used to define the similarity between conservation-restoration trajectories which is presented in the following section.

3.6 Evaluating of Trajectories Similarity

Evaluating the similarity between trajectories has been addressed by different streams of works, such as semantic trajectories analysis, which involves the evaluation of the similarity between the elements of the trajectory, generally associated with a location, or in the context of string comparisons, where the sequence is a string composed of characters and where distance functions have been proposed to evaluate string similarity. In all of these works, the proposed measures to calculate trajectory similarity have distinct characteristics, as discussed in the survey presented in [108].
The requirements for a similarity measure suitable for our context where trajectories represent histories of conservation events are the followings.

- First, the measure should not depend on the event's position, i.e. its index. Two similar events from two trajectories can match regardless of their positions.
- Another requirement is that the measure should depend on the event's order. For example, if there is a match between two events e_i and e_j from two distinct trajectories, then any following match between the events $e_{i'}$ and $e_{j'}$ is possible only if $i' > i$ and $j' > j$.
- The measure should be independent from the event's time. Two similar events from two trajectories can be matched regardless of their time feature.

- Finally, the measure should rely on a single match for each event, which means that there is at most one match for a given event.

According to conservation experts, the order and the number of occurrences of the events are essential. Therefore the matching should be single and depends on the order of the events. In addition, as the length of the trajectories is different and the matching should be single, it also should be partial. The time and the event's position in the trajectory, i.e., index, are unimportant. After analysing different similarity measures, as shown in chapter 2, we have chosen the Longest Common SubSequence (LCSS) [87] measure among the ones that respect the requirements to be the basis of the conservation–restoration trajectory similarity evaluation. The LCSS measure is a partial match measure. It calculates the longest common subsequence of two compared trajectories. When applied on spatial trajectories, LCSS considers two points of two different trajectories to be a match if the distance between the two locations is less than a given threshold ϵ and for every match it increase the longest common subsequence by one. We present hereafter the definition of LCSS as stated in [108].

Definition 6 *Longest Common SubSequence (LCSS)*

$$S_{LCSS}(Tr_i, Tr_j) = \begin{cases} \emptyset, & \text{if } l_i = 0 \text{ or } l_j = 0 \\ S_{LCSS}(Rest(Tr_i), Rest(Tr_j)) + 1, & \text{if } d(H(Tr_i), H(Tr_j)) \leq \epsilon \\ max\{S_{LCSS}(Tr_i, Rest(Tr_j)), S_{LCSS}(Rest(Tr_i), Tr_j)\}, & \text{otherwise} \end{cases} \quad (3.1)$$

With l_i and l_j representing the length of Tr_i and Tr_j respectively. The H function returns the first event in the trajectory, and the Rest function returns the trajectory without its first event.

The LCSS measure can be also used with semantic trajectories by matching the identical events, and by increasing the longest common subsequence by one for each matching pair.

Figure 3.10 shows an example of the similarity computation between two semantic trajectories using LCSS. The measure starts by comparing the first events in the trajectories, and as they are identical, they will be matched, and the similarity will increase by one. The measure continues with the rest of the two trajectories and considers again the first events, which are events B and event C, which are not the same. The measure continues to find the second match between events C in the two trajectories, and the similarity will be increased by one. Once the end of one of the two trajectories is reached, the final score is the similarity score between the trajectories, and in this example, the score is equal to two.

In order to take into account the terminology heterogeneity, we propose the Longest Common Events SubSequence $LCESS$ measure [134], an extended definition of $LCSS$. $LCESS$ is an ontology-based measure that calculates the maximum possible matches between two semantic trajectories. When comparing two events, $LCESS$ compute their similarity according to the event similarity introduced in section 3.5. It is defined as follows:

Definition 7 *Longest Common Events SubSequence (LCESS)*

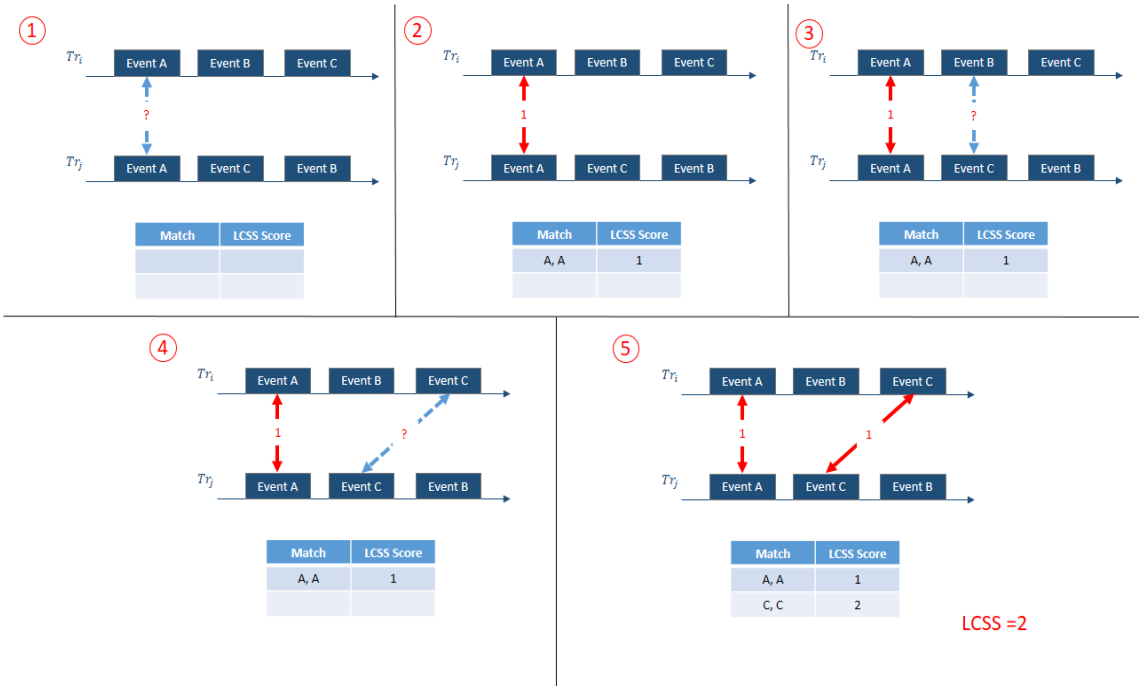


Figure 3.10: Computing the LCSS measure between two trajectories

$$LCESS(Tr_i, Tr_j) = \begin{cases} \emptyset, & \text{if } l_i = 0 \text{ or } l_j = 0 \\ LCESS(Rest(Tr_i), Rest(Tr_j)) + 1, & \text{if } Sim_e(H(Tr_i), H(Tr_j)) = 1 \\ max\{LCESS(Tr_i, Rest(Tr_j)), LCESS(Rest(Tr_i), Tr_j), \\ LCESS(Rest(Tr_i), Rest(Tr_j)) + Sim_e(H(Tr_i), H(Tr_j))\} & \text{if } 0 < Sim_e(H(Tr_i), H(Tr_j)) < 1 \\ max\{LCESS(Tr_i, Rest(Tr_j)), LCESS(Rest(Tr_i), Tr_j)\} & \text{otherwise} \end{cases} \quad (3.2)$$

With l_i and l_j representing the length of Tr_i and Tr_j respectively. The H function returns the first event in the trajectory, and the Rest function returns the trajectory without its first event.

When matching two events e_i and e_j from two trajectories, there are three possible cases:

- $Sim_e(e_i, e_j)$=1. The two events are equivalent, they are considered as matching events.
- $Sim_e(e_i, e_j) \in]0, 1[$. The relationship between the two events is either inclusion or closeness. In this case, they are considered as matching events if there are no other matching event with a higher similarity score.
- $Sim_e(e_i, e_j)$=0. The two events are dissimilar and they are not matching events.

The similarity score Sim_s and the distance dis between two trajectories Tr_i and Tr_j are defined as follow:

$$Sim_s(Tr_i, Tr_j) = \frac{LCESS(Tr_i, Tr_j)}{l_i + l_j - LCESS(Tr_i, Tr_j)} \quad (3.3)$$

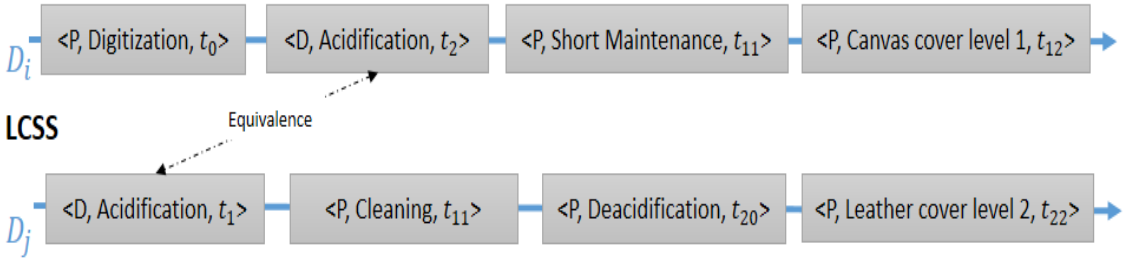


Figure 3.11: LCSS for Matching Conservation–Restoration Trajectories

$$dis(Tr_i, Tr_j) = 1 - sim_s(Tr_i, Tr_j) \tag{3.4}$$

By using $LCSS$ on sequences of events, only identical elements can be matched. Figure 3.11 shows an example of the similarity computation between two semantic trajectories using LCSS, where only identical events are considered as matching ones. The algorithm starts by comparing the first events in the trajectories, i.e., "$Digitization$" and "$Acidification$". As they are different, it will continue by computing the similarity between Tr_i and the rest of Tr_j, and the similarity between the rest of Tr_i and Tr_j, where the rest of a trajectory is the trajectory without its first event. The first match is between the "$Acidification$" events as they are identical. The algorithm continues until it reaches the end of one of the trajectories. In this example, no new matches are found, and the longest common sub-sequence is equal to 1. Regardless of the similarity between the events in these two trajectories, the $LCSS$ measure matches only one pair of events.

Therefore, the main difference between $LCSS$ and $LCESS$ is that the latter takes into consideration the relationships between the concepts corresponding to the events names in the ontology. For each match, $LCESS$ increases the total score by the similarity score between the two matching events. Note that in this work, our goal is to refine the matching between trajectories and propose a measure which enables to identify more matching elements beyond the identical ones, it is possible to further refine this measure, for example by taking into account the length of the trajectories.
Note that $LCESS$ can be used to calculate the similarity between semantic trajectories in other contexts, provided that the suitable domain ontology is provided, representing the semantic relationships between elements in the trajectories of the considered domain.

Figure 3.12 shows an example of the similarity computation between two semantic trajectories using LCESS. Considering the graph containing concepts representing the events A, A', and B and their relationships in Figure 3.12 (a). AS A and A' have a common super concept B', the relationship between them is closeness, and the relationship between these two concepts

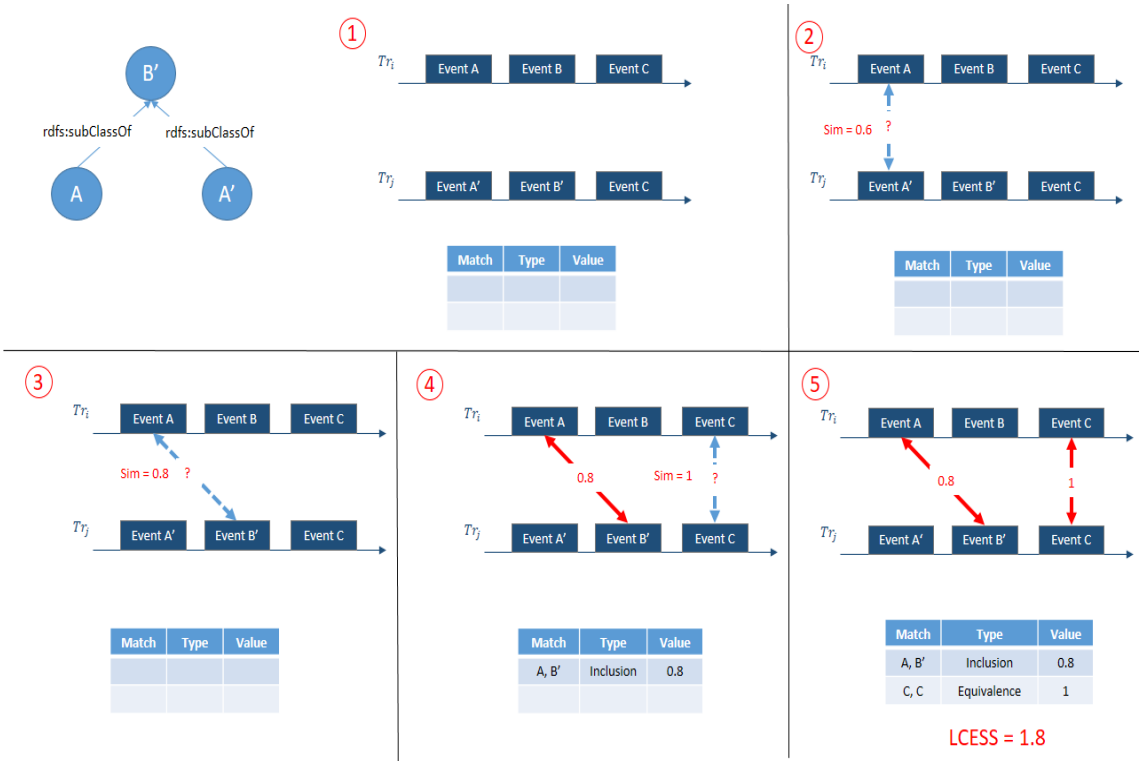


Figure 3.12: Example on the LCESS measure

with B' is inclusion. Therefore the similarities $Sim_e(B', A)$ and $Sim_e(B', A')$ are higher than $Sim_e(A, A')$.
The process starts by comparing the first two events from the two trajectories, where the similarity is supposed to be equal to 0.6. These two events are a possible match if no better match exists. Therefore, the search continues to find a better match where the best first match is between A and B' with a similarity score equal to 0.8. The measure continues and finds another match between identical events (Event C) with a similarity equal to one as they have an equivalence relationship. The process finishes the computation when it achieves the end of one of the two trajectories, and the final similarity score is equal to 1.8.
By using LCSS on these two trajectories, the similarity score will be equal to one with only one match between identical events.

Figure 3.13 shows a comparison between the matching using both $LCSS$ and $LCESS$ on two conservation-restoration trajectories Tr_i and Tr_j of two documents D_i and D_j respectively.
Using $LCESS$, the similarity score is increased as new matching pairs of events are identified. The algorithm starts by comparing the first events. As they are dissimilar, it will then compute the similarity between Tr_i and the rest of Tr_j, and the similarity between the rest of Tr_i and Tr_j, similarly to $LCSS$. The first identified match is between the two "$acidification$" events. The algorithm continues with the rest of the trajectories, and other

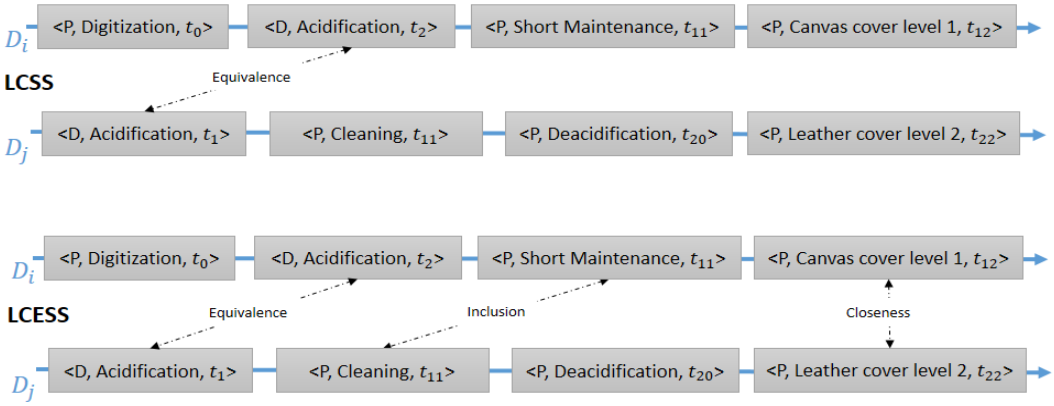


Figure 3.13: LCSS vs LCESS for Matching Conservation-Restoration Trajectories

matches are found, the first between the "$Short\ maintenance$" and the "$Cleaning$" events as they have an inclusion relationship, and the second between the "$Canvas\ cover\ level\ 1$" and "$Leather\ cover\ level\ 2$" events as they have a closeness relationship. This will result in a longest common events subsequence of length equal to $1+SIM_e($ $Short\ Maintenance$, $Cleaning) + SIM_e($ $Canvas\ cover\ level\ 1$, $Leather\ cover\ level\ 2)$. The relationship between the "$Short\ maintenance$" and the "$Cleaning$" events is an inclusion relationship, and the path length between them is equal to one. Assume that the depth of CRM_{BnF} is equal to 6 and α is equal to 0.7. The similarity score between the two events is therefore equal to 0.94. The relationship between the "$Canvas\ cover\ level\ 1$" and the "$Leather\ cover\ level\ 2$" events is a closeness relationship, as the nearest common ancestor between the concepts corresponding to the event names is the "$BnF : Cover\ Maintenance$" as shown in figure 3.6. The distance between the two events and their ancestor is equal to 2, therefore, the similarity score between the two events is equal to 0.47, and the $LCESS$ similarity between the two trajectories is equal to 2.41.

3.7 Towards An ontology for Conservation-Restoration At the BnF

Several ontologies have been proposed in the cultural heritage field, as we have seen in section 2.4. Some of these ontologies are $CIDOC-CRM$[84], CRM_{CR}[4] and CRM_{SCI}[22]. The problem is that they are not suitable to our context as they do not cover the terminology used by the conservation–restoration experts at the BnF. As the existing ontologies are not usable in our work, we have initiated, in collaboration with domain experts, a core of an ontology composed of a hierarchy of concepts representing the relationships between them.
We present in this section a terminology extracted from the existing databases and validated by domain experts which can be used to bridge the terminological gap between different databases at the BnF. This terminology is expressed as an OWL[4] ontology; it represents the concepts and the relationships between them, and could be used for data integration as well as data

[4]https://www.w3.org/2001/sw/wiki/OWL

publishing and exchange between organizations in the conservation domain.
In this section, we introduce the CRM_{BnF} ontology and its creation process. The ontology describes the field of conservation-restoration at the BnF. The creation process consisted of three stages. The first one is the identification of the concepts that should be represented in the ontology, which has been performed through the analysis of the databases in the different departments of the BnF. During the second stage, the relationships between these concepts are identified in cooperation with the domain experts. The first two stages are presented in section 3.7.1. The last stage presented in section 3.7.2 transforms the concepts into resources of the ontology, and the relationships into properties and RDF/OWL[5] triples.

3.7.1 Concepts and Relationships Identification

In order to create our ontology, we have first identified the set of relevant concepts. We have defined these concepts as the ones representing the events in the conservation–restoration trajectories, which will be inserted in the ontology. These concepts have been extracted from the available databases. We have then identified the relationships between these concepts through a methodology enabling the expert to assess the nature of the link between each pair of concept, and derive the appropriate properties to be inserted in the ontology. These two processes are described hereafter.

Concepts Identification.
According to domain experts, two types of events can affect the documents' physical state and that should be represented in the ontology: the conservation-restoration processes and the degradations. By analysing the integrated conservation–restoration database represented in section 3.3.2, we have identified all the types of conservation–restoration processes as well as the types of the possible degradations. The database stores documents' events history generated by different departments. We have identified around 250 different types of degradations and 150 different types of conservation–restoration processes.

In the data instances, there is a codification of the events to indicate their purpose. The domain experts at the BnF have added these terms to indicate the events with a similar purpose. Based on these terms, we have grouped the events having the same purpose, and we have identified 19 high-level conservation–restoration groups from the 150 existing events. We have applied the same process on the degradation events, and we have identified 8 high-level degradation events groups from the 250 recorded degradations. Figure 3.14 shows the eight most frequent conservation-restoration and degradation events groups and the number the different events in each group. These groups and events will serve as a basis to identify the concepts of the CRM–BnF ontology.

[5]https://www.w3.org/RDF/

Conservation-Restoration Groups		
Id	Label	Number of events
1	Reproduction	6
2	Consolidation	13
3	Conditioning	14
4	Deacidification	2
5	Paper treatment	6
6	Binding	16
7	Restoration	5
8	Maintenance less than 2h	24

Degradation Groups		
Id	Label	Number of events
1	Accompaniment	3
2	Body	23
3	Conditioning	16
4	Cover	60
5	Label	16
6	Guards	40
7	Signal	4
8	Support	23

Figure 3.14: Groups of Events in the Existing Databases

Relationships Identification.
In order to identify the semantic links between the concepts, we have considered separately each of the groups of identified conservation restoration processes as well as the sets of identified degradation types. For each group, we have analysed with the experts the events it contains and the relationships between them. We have defined four types of relationships between the events, which are presented and validated by the domain experts, and for each pair of events e_i and e_j, the experts were asked to choose the most appropriate. The events are considered dissimilar in case where none of the relationships are applicable. The relationship types are the followings.

Relationships Types

- Equivalence: This relationship expresses the fact that the events e_i and e_j are equivalent. It is denoted $e_i \equiv e_j$.
- Specialization: This relationship expresses the fact that e_j is a higher-level event than e_i. e_j has several specialized lower-level events, one of them being e_i. The event e_i shares the same characteristics as e_j, but can also have specific ones. This relationship is denoted $e_i \subseteq e_j$.
- Generalization: This relationship expresses the fact that e_i is a higher-level event than e_j. In other words, e_i has several specialized lower-level events, one of them being e_j. This relationship is denoted $e_i \supseteq e_j$.
- Similarity: This relationship expresses the fact that the events e_i and e_j share some common characteristics, but none of them is a generalization or a specialization of the other. This relationship is denoted $e_i \approx e_j$.

Constraints over the relationships
In order to avoid introducing some inconsistencies when inserting new triples and properties in the ontology, we have defined a set of rules. The two first ones ensures that there is no

Database id	Matrix id	Procede
2	1	T1 Reliure mécanisée. Coupé/collé
3	2	T2 Reliure mécanisée. Couture sur cahiers
4	3	T3 Reliure mécanisée. Couture sur surjets
5	4	T4 Reliure mécanisée. Traitement encarté
6	5	T5 Reliure mécanisée. Hors norme
84	6	T6 Reliure mécanisée. Mobile
85	7	T7 Reliure mécanisée. ACLE
262	8	T2 Reliure mécanisée. Couture sur encarté
242	9	T2/2 Reliure mécanisée. Couture sur encarté

Values:
- G : Generalization
- S : Specialization
- SIM : Similarity
- E : Equivalence

id	1	2	3	4	5	6	7	8	9
1		X	X	X	X	X	X	X	X
2			SIM	SIM	X	X	X	SIM	SIM
3				SIM	X	X	X	SIM	SIM
4					X	X	X	SIM	SIM
5						X	X	X	X
6							G	X	X
7								X	X
8									E
9									

Figure 3.15: Mechanical binding group and the relationships between the concepts

inconsistency in both the specified generalisation or the specialisation relationships. The third one ensures the consistency of the equivalence relationships between the concepts. Automatic validation is performed to ensure that these rules are not violated. Given three events e_i, e_j and e_k, we define the three rules as follows:

- Rule 1 : $e_i \supseteq e_j$ AND $e_j \supseteq e_k \rightarrow e_i \not\equiv e_k$ AND $e_i \not\subseteq e_k$

  This first rule aims to avoid the loops in the generalization relationships. If $e_i \supseteq e_j$ and $e_j \supseteq e_k$, then both the relationships $e_i \equiv e_k$ and $e_i \subseteq e_k$ are inconsistent.

- Rule 2 : $e_i \subseteq e_j$ AND $e_j \subseteq e_k \rightarrow e_i \not\equiv e_k$ AND $e_i \not\supseteq e_k$

  This second rule aims to avoid the loops of the specialization relationships. If $e_i \subseteq e_j$ and $e_j \subseteq e_k$, then both the relationships $e_i \equiv e_k$ and $e_i \supseteq e_k$ are inconsistent.

- Rule 3 : $e_i \equiv e_j$ AND $e_j \equiv e_k \rightarrow e_i \equiv e_k$

  This third rule aims to avoid inconsistent equivalence relationships. If $e_i \equiv e_j$ and $e_j \equiv e_k$, then the only possible relationship between e_i and e_k is the equivalence, i.e., $e_i \equiv e_k$

Figure 3.15 shows one of the events groups called "the mechanical binding", which contains nine concepts. The matrix show the relationship between each pair of concepts identified in collaboration with the domain experts.

3.7.2 Initiating the CRM_{BnF} Ontology

CRM–BnF is the first step towards a unified terminology in the field of conservation-restoration in the libraries and towards a well defined vocabulary in this field. This unified terminology will encourage data sharing and the data exchange between the conservation-restoration departments at the BnF and beyond. To understand the transformation process, we define the ontology as follows:

Definition 8 (Ontology) *An ontology is a formal, explicit specification of a shared conceptualisation [107].*
We represent the ontology as a graph $G = (N, E)$ where N is the set of concepts, and E is the set of edges or properties.

The ontology represents all the types of events that can affect a document physical state. We have identified and represented two categories of types of events in the ontology. The first type is the conservation process which represent the events which aim is to improve the physical state of the documents and to fix the detected degradations. The second type is the degradation, which represents all the possible alterations in the documents physical state. The creation process consists firstly in the transformation of each group of events into a specific concept representing this category in the ontology, then the events are transformed into concepts, each one represented by a node in the graph. Finally, the properties between concepts are identified, and added as edges between the corresponding nodes in the graph.

Initializing the Ontology

In the graph G(N, E) representing the ontology, the root node $owl : Thing$ is added to the set of nodes N. The node $Event$ is then added to N, and the edge $< Event, rdfs : subClassOf, owl : Thing >$ is inserted in the set E. Two nodes are then introduced, namely $Conservation\ Process$ and $Degradation$, and added to the set N. Then the two edges $< Conservation\ Process, rdfs : subClassOf, Event >$ and $< Degradation, rdfs : subClassOf, Event >$ are added to the set E.

Transforming Groups of Events

In collaboration with the conservation–restoration experts at the BnF, and after identifying the groups of similar events according to the codification in their name, we have grouped the events into more specific groups according to their similarity. In the ontology, we propose to represent the similar events that share some characteristics by concepts that are subclasses of the same concept that represent these common characteristics. For each group G_i, associated with the label L_i extracted from the corresponding database and representing the purpose of this group, the following actions are executed:

- A new node n_i is added to the set N.
- A new edge $< n_i, rdfs : label, L_i >$ is added to the set E.
- If G_i is a group representing conservation events, then a new edge $< n_i, rdfs : subClassOf, Conservation\ Process >$ is added to the set E.

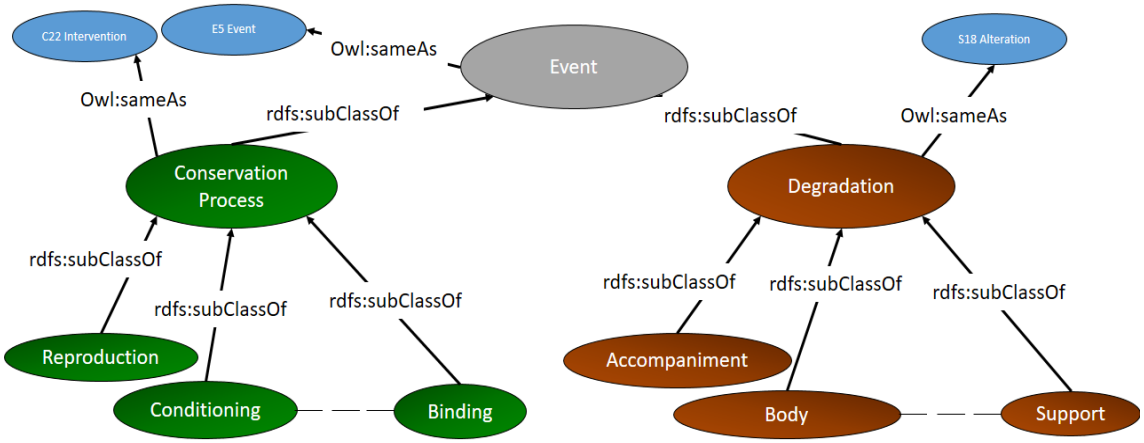


Figure 3.16: High-Level Concepts in the CRM_BnF Ontology

- If G_i is a group representing degradation events, then a new edge $< n_i, rdfs : subClassOf, Degradation >$ is added to the set E.

In the ontology, the nodes will form two rdfs:subClassOf hierarchies: one corresponding to degradations, and the other to the conservation processes. Figure 3.16 shows the two hierarchies, with their respective root "Conservation Process" and "Degradation", represented as sub-classes of $Event$. Each child node of both "Conservation Process" and "Degradation" concepts represents a group of events having the same purpose. The upper node in the hierarchical structure is the Event node created during the initialization step. The figure also shows three $owl : equivalentClass$ links to external nodes in other existing ontologies in the field of cultural heritage and conservation-restoration, represented in blue.

Transforming the events and defining the properties

In the CRM–BnF ontology, represented by the graph G(N, E), each sub-class c_i of the Conservation Process and Degradation nodes represents a group of events having the same purpose, extracted from the explored databases. For each event e_i in the sub-class c_i, a node denoted n_i is added in the set N, and the edge $< n_i, rdfs : subClassOf, c_i >$ is added to the set E. Other properties are identified considering the relationships stated by the experts. The generalization, the specialization and the similarity relationships are transformed into new edges in the set E, corresponding to triples with the $rdfs : subClassOf$ property. The equivalence relationship is transformed into a new edge in E corresponding to a triple with the $owl : sameAs$ property. Consider the two events e_i and e_j with their corresponding nodes n_i and n_j respectively in the ontology represented by the graph G(N, E). We define four transformation rules to generate new properties in the ontology:

Rule 1 *If $e_i \supseteq e_j$, then a new edge $< n_j$, rdfs:subClassOf, $n_i >$ is added in the set E.*

Rule 2 *If $e_i \subseteq e_j$, then a new edge $< n_i$, rdfs:subClassOf, $n_j >$ is added in the set E.*

Rule 3 *If $e_i \equiv e_j$, then a new edge $< n_i$, owl:sameAs, $n_j >$ is added in the set E.*

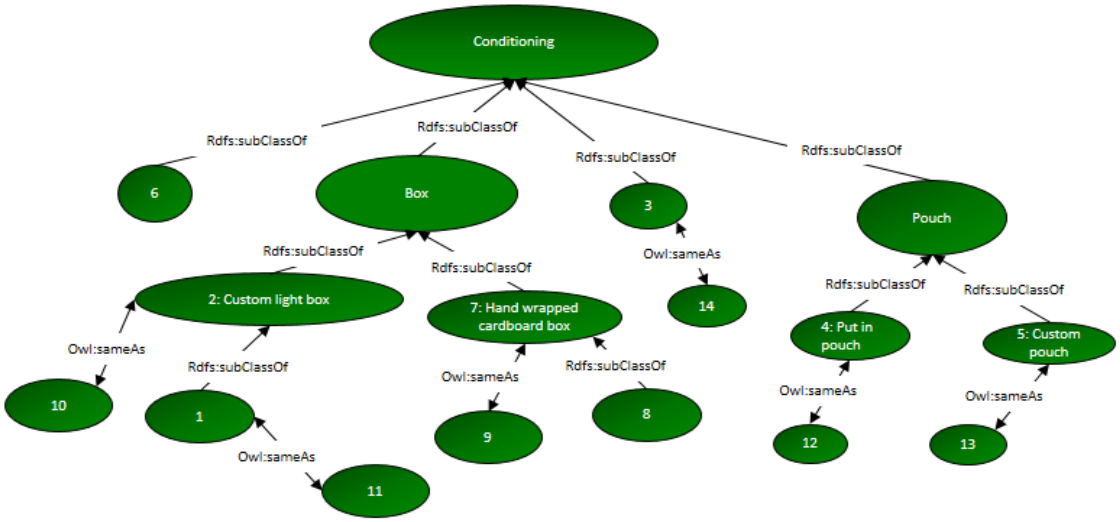


Figure 3.17: Adding Concepts and Properties in CRM_{BnF}

Rule 4 *If $e_i \approx e_j$, then a new node n_k that represents the common characteristics between the two events is added to the set N; the experts assign a label L_k to n_k, then a new triple $< n_k, rdfs : label, L_k >$ is added to the set E. Finally, the two following triples are added in the set E:*
$< n_i, rdfs : subClassOf, n_k >$ *and* $< n_j, rdfs : subClassOf, n_k >$.

Figure 3.17 shows an expert of CRM_{BnF} that represents the nodes and the relationships of the $Conditioning$ group after the application of the transformation rules. Two new nodes were added by the experts, the "Box" node and the "Pouch" node, to represent the common characteristics between the pairs of similar events (2, 7) and (4, 5) respectively. For conciseness, we have denoted some nodes by the ID of the corresponding. Figure 3.18 shows the size of the ontology after the transformation process, and we provide in the appendix A some dimensions of the ontology

3.8 Experimental Evaluation

In this section, we evaluate the process proposed in this chapter for the matching between trajectories. We are interested in showing how much the LCESS measure can increase the number of matches between events when comparing the trajectories. In addition, we present how this increase in the number of matches affects the similarity between the trajectories. Finally, we are interested in showing the quality of the new matches using LCESS by showing how LCESS increases the similarity between the trajectories of the documents having similar conservation–restoration histories based on the opinion of the domain experts.
The experiments presented in this section consist of three parts. The first one is related to event matching, where we show the effectiveness of the use of the ontology for searching for matching events. The second part of the experiments is related to the trajectory similarity.

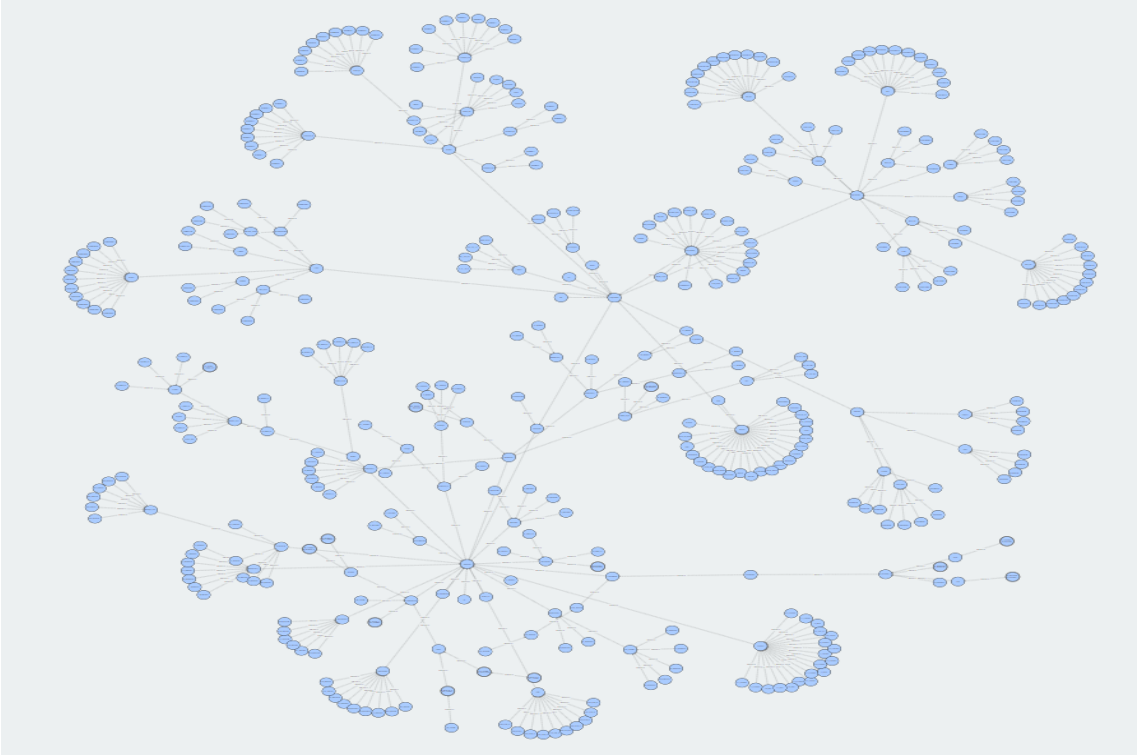


Figure 3.18: CRM_{BnF} Ontology

The third part shows the effectiveness of the ontology and the similarity measure using a set of clusters generated manually by domain experts as a gold standard.

In the experiments, we have used real datasets from the BnF, with 7,317,558 conservation events between 2003 and 2020 for 1,946,760 documents. In our datasets, the maximum trajectory length is 1397 events, the minimum length is 1 event, and the average length is 3.75.

3.8.1 Event Matching

This part of the experiments aims to show how the use of the ontology during event matching improves the results. We run our approach three times on 100 randomly selected trajectories each time. Figure 3.19 (a) shows a comparison between the number of pairs of matching events found after running the three tests with and without the use of the ontology. The first test was done on 100 trajectories containing 120 events in total. The results are shown in figure 3.19 (a), showing that 1242 matching events are retrieved without the use of the ontology, while 1663 are retrieved using the ontology. This represents an increase of 33.89%. The second test was done on 100 trajectories containing 173 events in total. Without using the ontology, 1481 pairs of matching events are found. Using the ontology, this number rises to 2034, showing an increase of 37.33%. The third test was done on 100 trajectories containing 295 events in total. Without using the ontology 2473 matching events are detected, and using the ontology, the

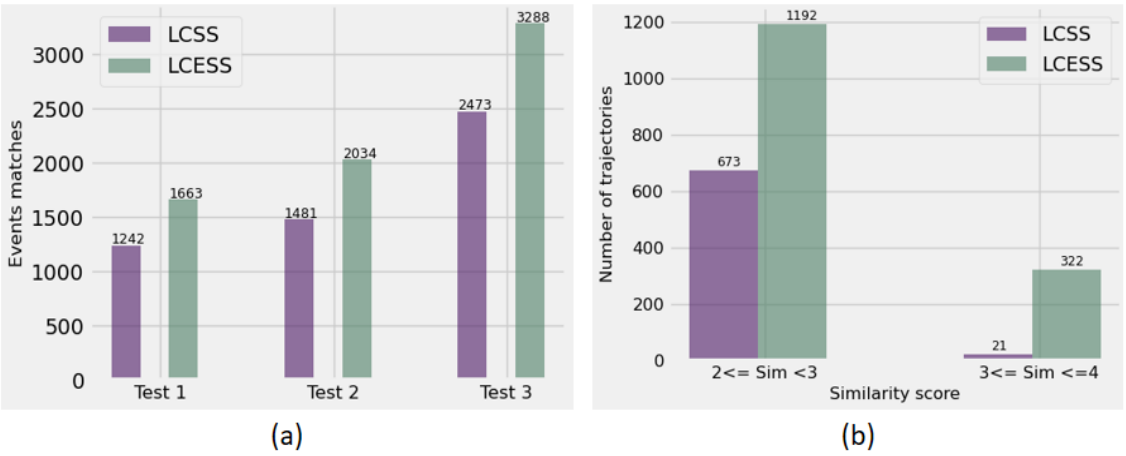


Figure 3.19: Ontology-Based Event Matching (a) and Trajectory Similarity (b)

number of retrieved matching events is 3288, which represents an increase of 32.95%. These tests show a good increase of the number of matching events using the ontology, with an average of 34.72%.

3.8.2 Trajectories Similarity

To show the impact of using the ontology in the evaluation of the similarity between trajectories, we have randomly selected a trajectory Tr and compared it with a set $TrSet$ of 10000 randomly selected trajectories. We have calculated the similarities between Tr and all the trajectories in the set $TrSet$ without using the ontology i.e., using the $LCSS$ measure. We have then selected those having a similarity greater than half of the length of Tr. We have performed the same experiment using the ontology i.e., $LCESS$ measure. Finally, we compare the number of trajectories having a similarity greater than half of the length of Tr using the two measures. Figure 3.19 (b) shows the number of trajectories which are similar to the input trajectory before and after the use of the CRM_{BnF}. The input of the experiment was a random trajectory $Tr = [< C,\ \text{“}High\text{”},\ t_0 - t_2 >,\ < P,\ \text{“}Manual\ Binding\text{”},\ t_4 >,\ < P,\ \text{“}1/2\ Cover\text{”},\ t_6 >,\ < C,\ \text{“}Low\text{”},\ t_9 - t_{10} >]$ with a length equal to 4. The output shows the number of trajectories which have a similarity between 2 and 4 with Tr. The number of trajectories having a similarity greater than 2 without using the ontology was equal to 694. The number increased to 1504 using the ontology, which is an increase of 116.71%. The total similarity average without using the ontology is 0.8 and increases to 0.95.

3.8.3 Quality of the Matching Algorithm

This experiment aims to show the effectiveness of using CRM_{BnF} and the LCESS measure by comparing the computed similarity to a gold standard. In our case, this gold standard is a set of similarities between trajectories set by the conservation experts at the BnF. We started the experiment by selecting a random set of trajectories to be clustered manually by the conservation

experts. We selected a set of 20 trajectories and provided them to the experts. They clustered 14 trajectories into 3 clusters; the 6 remaining trajectories were not assigned to any cluster as they were not similar to any other trajectory according to the experts. Let us denote the resulting clusters by Cl_1, Cl_2 and Cl_3. They contain respectively 5, 4 and 5 trajectories. For the set of considered trajectories, we will compute events matches with and without the use of CRM_{BnF}, and compare the results. The use of the ontology should significantly increase the number of matching events inside the clusters defined by the experts. The number of matches between events from different clusters should not vary significantly.

We distinguish between two situations during event matching, the compared events could either belong to trajectories from the same cluster or to trajectories from different clusters. For each cluster, we compute the number of intra-cluster matches between events. For the cluster Cl_i, this number is equal to the number of event matches between all the pairs of trajectories that belong to cluster Cl_i. We also compute the number of inter-cluster matches between two clusters Cl_i and Cl_j. This number is equal to the number of pairs of matching events (e_i, e_j) such that e_i is an event of a trajectory in Cl_i and e_j is an event of a trajectory in Cl_j.
We define these two numbers as follows. In the definitions, we denote EM the function which returns, for two trajectories, the number of matching events between them.

- Number of intra-cluster matches of the cluster Cl_x :
  $Intra(Cl_x) = \sum EM(tr_i, tr_j)$ where tr_i and $tr_j \in Cl_x$ for $1 < i < n$, $1 < j < n$ and $i \neq j$, and where n is the total number of trajectories in the cluster Cl_x.

- Number of inter-cluster matches between two clusters Cl_x and Cl_y:
  $Inter(Cl_x, Cl_y) = \sum EM(tr_i, tr_j)$ where $tr_i \in Cl_x$ and $tr_j \in C_y$ for $1 < i < n$, $1 < j < m$ and $i \neq j$, where n and m are the number of trajectories in the clusters Cl_x and Cl_y respectively.

Figure 3.20 (a) shows the number of intra-cluster and inter-cluster event matches for the clusters defined by the experts using LCSS. The number of intra-cluster matches for the clusters Cl_1, Cl_2 and Cl_3 is equal respectively to 11, 10 and 13. The number of inter-cluster event matches are as follows: $Inter(Cl_1, Cl_2)$ is equal to 17, $Inter(Cl_1, Cl_3)$ is equal to 8 and $Inter(Cl_2, Cl_3)$ is equal to 21.

The goal of LCESS is to increase the similarity between the trajectories assigned to the same cluster by the experts, i.e. the similar ones, which requires the increase of the number of event matches between them. Therefore, the use of LCESS should notably increase the number of intra-cluster event matches regardless of the total number of inter and intra matches, which depends on both the number of the trajectories in the clusters and the length of the trajectories. We show the effectiveness of using the LCESS measure and the ontology by representing the increasing percentage of the two types of event matches. Figure 3.20 (b) shows the new numbers using LCESS. The number of intra-cluster event matches of Cl_1, Cl_2 and Cl_3 is equal to 21, 17 and 25 respectively. The number of inter-cluster event matches are as follows: $Inter(Cl_1, Cl_2)$ is equal to 18, $Inter(Cl_1, Cl_3)$ is equal to 8 and $Inter(Cl_2, Cl_3)$ is equal to 27. The total number of inter-cluster event matches is computed as follows: $Inter(Cl_1, Cl_2) + Inter(Cl_1, Cl_3) + Inter(Cl_2, Cl_3)$. It was equal to 46 and was increased by 7 new matches,

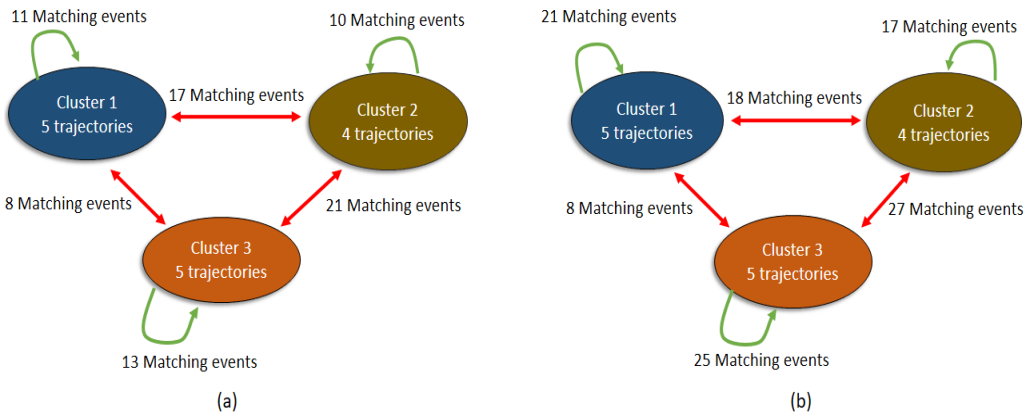


Figure 3.20: Numbers of Inter-Cluster and Intra-Cluster Event Matches Using LCSS (a) and LCESS (b)

which represents an increase of 15.2%. The total number of inter-cluster event matches is computed as follows: $Intra(Cl_1) + Intra(Cl_2) + Intra(Cl_3)$. It was equal to 34 and was increased by 29 new matches, which represents and increase of 85.2%. As we can see, using the LCSS measure, the total number of the inter-cluster event matches was higher than the total number of the intra-cluster event matches, which is contradictory considering that trajectories in same cluster are more similar according to the experts. Using the LCESS measure, the total number of the intra-cluster event matches is equal to 63 and for the inter-cluster is equal to 53.

The use of the ontology results in a high increase of the number of intra-cluster event matches, which in turn increases the similarity between the trajectories in the same cluster. The inter-cluster event matches has increased by 15%, which is not significant, especially compared to the increase of the number of intra-cluster matches. This result also shows the validity of the ontology and the identified relations between concepts, as they represent the actual existing relationships between conservation events and thus enable accurate similarity evaluation between conservation-restoration trajectories. The results show the usefulness of the similarity measure based on the ontology where the similarity values between the trajectories that are similar in the expert's opinion increase significantly compared to those that are not similar.
In addition, the results shows that the use of LCESS in the clustering process of the trajectories provides better results as it matches more events between the similar trajectories

3.9 Conclusion

In this chapter, we have proposed a representation of documents conservation data as conservation-restoration trajectories. We have proposed a process to evaluate the similarity between two given events using an external source represented as an ontology. In addition, we have also proposed an adaptation of an existing trajectory similarity measure to take into account our ontology-based event similarity measure. Finally, we have introduced an ontological model dedicated to encompassing the main conservation and restoration concepts at the BnF.

The proposed process to evaluate the similarity between two given events using an ontological model is based on our proposed relationships between the concepts.We have proposed the LCESS similarity measure, which is an extension of the LCSS measure that consider the semantic relationships between the elements constituting the trajectories. It also searches the maximum possible similarity score. Finally, the introduced ontological model is a first step towards an ontology for conservation-restoration data in libraries. This ontology, called CRM_{BnF}, has been initiated to solve the terminological conflicts arising when heterogeneous databases are compared and integrated. It contains the concepts related to conservation processes as well as document degradations, and the semantic relationships between them. Both concepts and relationships have been identified from the data stored in the existing databases, and validated in collaboration with the domain experts.

The experiments have shown that our approach improves the precision of the matching process. This work shows the ontological model's usefulness for expert knowledge representation and trajectory similarity computation.

Future works will include further enrichment of the proposed ontological model by introducing more relevant properties, such as the physical characteristics of the documents, which could be helpful to be integrated into the analysis process. In addition to extending the proposed similarity measures exploiting these properties.
Another line of work would be to introduce reasoning capabilities to our similarity computation, which would enable us to take into account not only the knowledge explicitly provided in the model but also the one which could be derived through inference.
Finally, we mention that our proposed matching process can be used in different contexts with different semantic trajectories by providing a suitable external source representing the semantic relationship between the elements.

4 - Analysis of Conservation–Restoration Data for Documents' Physical State Prediction

Contents

4.1 Introduction

The BnF stores approximately 20 million documents. In order to detect their degradation at an early stage, it is essential to check their physical state continuously to classify the documents as available or deteriorated; this task is time-consuming and impossible in practice due to the large volume of documents. Instead of a manual assessment of the documents, we propose an approach which analyse the conservation–restoration trajectories introduced in chapter 3 in order to predict the documents physical state. An early detection of the deteriorated documents will help to minimize the number of documents which physical state might deteriorate to the point that they become no longer available to the readers. The early identification of the documents which are likely to deteriorate would be beneficial to the experts to prioritize the ones which have to undergo some conservation processes in order to prevent further major degradations and to define their conservation policies.

Experts define various risk scenarios that should not happen to stop the documents from degrading. For example, some types of treatment followed by high communication can indicate that the documents are at risk of being degraded again. But it is impossible for experts to define all the scenarios or find the all the correlations between the conservation-restoration trajectories

and the physical state.

In this chapter, we propose an analysis pipeline which aims to detect discriminative events or sequences of events that can serve as a basis to define rules to predict the physical state of the documents. The prediction relies on the identification of the trajectory patterns which are most representative of either the deteriorated documents, or the ones in a good physical state.

The prediction of the documents' physical state based on the analysis of its conservation–restoration history is a complex problem as no event could help directly identify the physical state. In addition, documents with similar conservation–restoration history may have different physical states.
The proposed analysis pipeline is formed of different modules in order to achieve such prediction. It relies on trajectory clustering aiming to characterize each class of document, deteriorated and available, by a set of trajectory patterns [133]. The clustering algorithm uses the LCESS similarity measure introduced in chapter 3. The physical state of a document is predicted by assessing the similarity between its trajectory and the generated patterns to determine the document's class. In this chapter, we present the trajectory analysis pipeline to achieve such prediction on the documents' physical state. The rest of this chapter is organized as follows. The pipeline overview to build the prediction system is presented in section 4.2. Section 4.3 presents the conservation–restoration trajectory clustering, and section 4.4 presents the patterns extraction and the generation of the prediction rules. A prototype of clustering and filtering is presented in section 4.5. Our prediction experiments are presented in section 4.6. Finally, we conclude the chapter in section 4.7.

4.2 Conservation Data Analysis Pipeline

The ability to predict the physical state of documents enables the identification of the deteriorated documents without having to manually check their state. Such prediction could done by analysing the data that is related to the documents physical state, which we represent as conservation–restoration trajectories composed of relevant events. Based on such prediction, the conservation experts will be able to target the documents which are at higher risk of deterioration.

Consider the sets of conservation trajectories A and D, corresponding respectively to the trajectories of the available documents and the trajectories of the deteriorated documents. The physical state of a document can be predicted based on the similarity of its trajectory to the two sets A and D. In other words, predicting a document's physical state requires calculating its trajectory similarity to the trajectories of the two sets and classifying its trajectory into one of them. Each set has a wide variety of trajectories, possibly sharing many or them. Comparing a trajectory to all the ones in A and D might not lead to a meaningful result. The two sets may share a high number of similar trajectories, and finding the class of a document might not be straightforward.

For example, consider the distribution of the trajectories in a two-dimensional space accord-

ing to their similarity presented in figure 4.1. Each element in the space represents a trajectory, and the classes of the trajectories are represented by the elements' shape, where two classes exist, class 1 represented by the red circles and class 2 represented by the blue triangles.

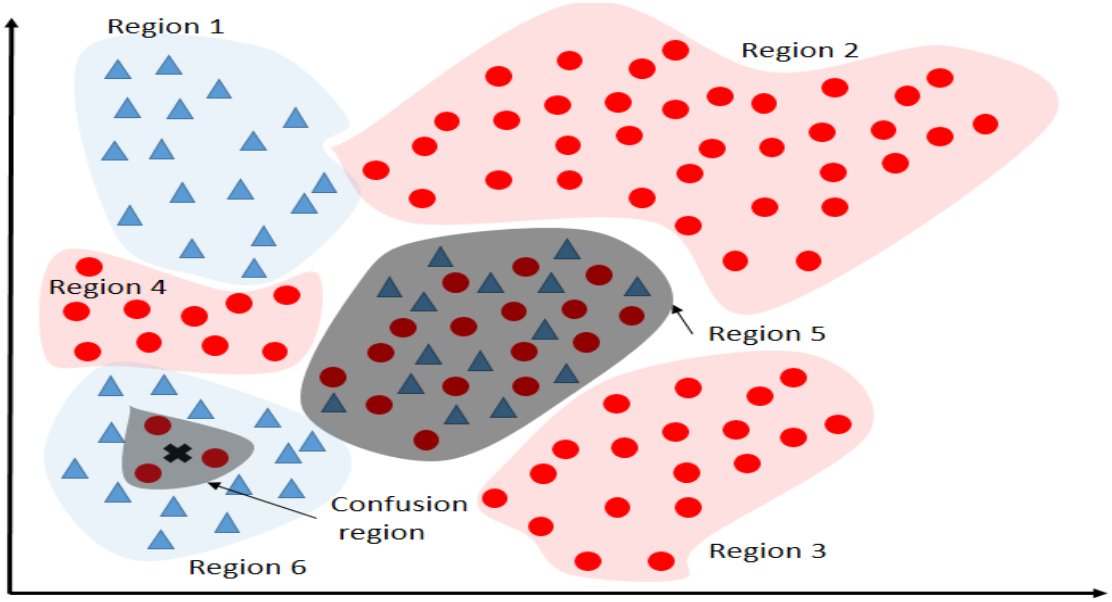


Figure 4.1: Illustration of homogeneous and heterogeneous regions

One problem is that a region may contain trajectories from different classes with different percentages. In order to predict a trajectory's class, one way is to compare it to the most similar ones. This way, there is a risk of false classification depending on the distribution of the nearest trajectories. For example, the confusion region in the figure shows how the dominant class in region 6 is class 2. Still, by comparing the trajectory to its nearest trajectories, it will be assigned to class 1.

Therefore, we aim to predict the trajectory class based on the distribution of the trajectories' classes in the region where the trajectory falls. This leads us to the next problem we aim to tackle is that some regions could be heterogeneous and contain trajectories from different classes with no dominant one.

For that reason, our objective is to extract representative trajectories only from the homogeneous regions, called patterns, that represent the classes and can help distinguish between them. For example, regions 1, 2, 3 and 4 are homogeneous and could be used to extract patterns to represent their trajectories, which have the same class. On the contrary, region 5 is heterogeneous and has no dominant class. Therefore, the extracted patterns from this region can not represent any of the two classes. Finally, we propose to predict the trajectories classes by comparing them to the extracted patterns.

We propose a trajectory analysis pipeline that first identifies the regions containing similar trajectories, followed by the extraction of patterns representing the classes and help distinguish between them. Finally, the extracted patterns will be the base of trajectory classification and

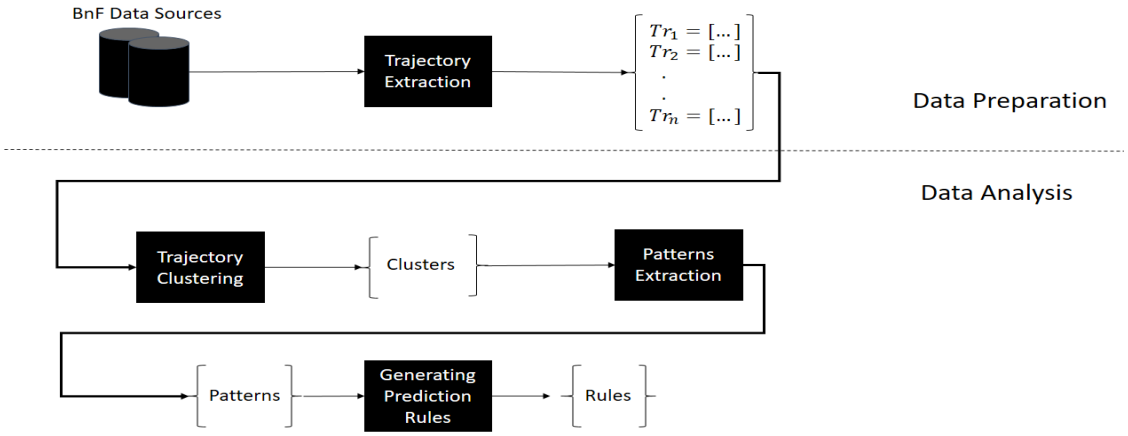


Figure 4.2: Trajectory Analysis Pipeline

the prediction rules. Figure 4.2 shows our trajectory analysis pipeline composed of four modules, and in the following we discuss the different modules, their inputs, goal and their outputs.

We propose the physical state prediction of the documents based on the analysis of the conservation–restoration histories that contain all the information related to the physical state, such as the communication, degradation and conservation–restoration process.
In chapter 3, we have proposed a representation of these histories as semantic trajectories. The first module in the pipeline aims to extract the data from the integrated database containing the relevant information and create the semantic trajectories to be the input of the next module. In our work, the trajectories are created using SQL queries on the database, and stored manually in an integrated database. This module could be automatised and abstracted in case new dimensions are added to this database, which should be part of the trajectories.

The second module in the pipeline takes as input the set of trajectories $TR = \{tr_1, tr_2, ..., tr_n\}$ and aims to find groups of similar ones. Different methods were discussed in chapter 2, some of them can not be used in our context, such as the DBSCAN clustering, and others could be used with some adaptation, such as the k-means. The output of this module will be a set of clusters $CL = cl_1, cl_2, ..., cl_m$, with $m < n$. In addition, the clusters will have different characteristics, such as the number of trajectories, each class percentage, etc.

Pattern extraction is the process of detecting a common behavior when analysing people's movements, or similar sequences when analysing semantic trajectories. In the proposed pipeline, the pattern extraction module aims to extract patterns of sequences of events representing the common characteristics of the trajectories from the same class. These patterns should help distinguishing between the two classes, available and deteriorated. The input of this module is a set of cluster $CL = cl_1, cl_2, ..., cl_m$, and the output is two sets of patterns $Pattern\_A$ and $Pattern\_D$ representing the trajectory patterns which are the most representative of the two classes available and deteriorated respectively.

The last module is to create rules to predict the documents' physical state based on the

extracted patterns. Thus, we propose to compare the trajectory of the document in which we want to predict its physical state to the patterns of the two classes. Finally, based on its similarity to the patterns, the prediction will be made. If the trajectory is similar to the patterns representing the available (resp. deteriorated) class, it will be predicted as available (resp. deteriorated).

In the following, we present in detail the three modules. In section 4.3 we present the clustering module. Section 4.4 is devoted to the pattern extraction module, and the generation of the prediction rules.

4.3 Trajectory Clustering

Trajectory clustering aims to build clusters containing similar trajectories. The clusters will be analysed later to identify common characteristics between the trajectories. In this work, we propose first to cluster the conservation–restoration trajectories to analyse each cluster separately in order to find correlations between their events and the physical state of the documents.
We have used the k-means algorithm [28] to cluster the trajectories. The optimal value of the parameter k is defined using the elbow method.

First, a set of k trajectories are selected randomly, and a cluster is initiated from each of them. Then, each trajectory is compared with the k-selected trajectories and assigned to the cluster corresponding to the most similar one. The means are recomputed for each cluster and the similarity between the trajectories and the new means is evaluated. The trajectories are re-assigned to the cluster corresponding to the nearest mean. The algorithm iterates until the clustering becomes stable.

K-medoids could be a good choice when clustering semantic trajectories with the same length where the most frequent event in each index will be selected. Still, in our context, the trajectories have different lengths. Therefore, we choose to use the k-means with some adaptation such as a customized definition of the mean of conservation–restoration trajectories. The mean of a set of conservation-restoration trajectories is defined as follows:

Definition 9 *Mean of a Cluster of Conservation–Restoration Trajectories.*
The mean m_j of a cluster cl_j with $1 \leq j \leq k$ is a conservation-restoration trajectory. The length of m_j is the average of the lengths of the conservation-restoration trajectories in cl_j. The events of m_j are the most frequent events at each position: e_{ij} in m_j is the most frequent event at position i in all the trajectories of the cluster.

Given this definition, suppose for example that 60% of the conservation-restoration trajectories in cl_j start with a $binding$ event and 40% start with $communication$ event. In that case, the first event in the mean, i.e. at index 0, is the $binding$ event. The events for the other indexes i of the mean trajectory are defined in the same way.
Figure 4.3 shows a mean calculation example. The mean calculation starts by computing the average length of trajectories in the cluster, which is equal to three in the example. Afterwards,

for each index in the trajectories, the frequencies of the events are calculated and we show the five top most frequent events in each index. In this example, 53% of the trajectories start with a low communication event, 12% of the trajectories start with a low communication refusal or medium communication etc. Therefore, the most frequent event at the beginning of the trajectories of this cluster is low communication; consequently, the first event in the mean will be low communication. The most frequent event in the second index in the trajectories is the treatment T67 with a percentage equal to 63%. And treatment T304 for the third index with a percentage equal to 53%.

[low communication ---> T: JDép_Maintenance._Dépoussiérage--> T: B2_Boîte_légère-->]

	Event 1	Event 2	Event 3
Top 1	communication,low 53%	treatment,T67 63%	treatment,T304 53%
Top 2	refuscommunication,low 12%	refuscommunication,low 10%	treatment,T308 10%
Top 3	communication,medium 12%	treatment,T304 6%	treatment,T67 10%
Top 4	communication,high 9%	communication,low 6%	refuscommunication,low 5%
Top 5	treatment,T67 5%	treatment,T3 2%	communication,low 3%

Figure 4.3: An example of a cluster's mean

The clustering process ends when the clusters become stable, and we define the clusters' stability as follows:

Definition 10 *Stability.*
We consider the clustering stable when the percentage of the trajectories that change their cluster after updating the means is less than a threshold σ.

Algorithm 1 describes the clustering process. It starts by randomly selecting k trajectories among the input. A k_means iteration is applied to the trajectories and the given means. The result of the clustering iteration is the number of trajectories which changed their cluster on this iteration, referred to as $moving\_trajectories$ (line 6). After calculating the means of the new clusters (lines 8-9), the number of the moving trajectories is computed and if this number is less than a threshold (NbChanges), the process will be considered stable and it will ends.
Algorithm 2 describes a clustering iteration. The distances with the means for each trajectory are calculated (line 3). Then, the index of the nearest mean is selected and compared to the trajectory cluster. If the closest mean is not the mean of the cluster to which the trajectory belongs, the trajectory belongs to a new cluster and is considered a moving trajectory. Finally, the means of the clusters are recalculated.

Once the clustering is performed and the clusters are stable, the clusters generated by the clustering algorithm are described by their conservation-restoration mean trajectory and by other characteristics, such as the number of trajectories in the cluster and the percentage of the ones corresponding to deteriorated documents. These mean trajectories are the representative of the clusters.

Algorithm 1 Clustering Algorithm

Input $Trajectories : TR$, $Number\ of\ iteration : NbIt$,
$Number\ of\ changes\ to\ break : NbChanges$
Output $Clusters$
$means \leftarrow Generate\_Random\_means(TR)$
$Clusters \leftarrow []$
$moving\_trajectories \leftarrow []$
$iteration \leftarrow 0$
while $iteration \leq NbIt$ **do**
  $moving\_trajectories, Clusters \leftarrow k\_means\_iteration(TR, means)$
  $means \leftarrow []$
  for $cluster \in Clusters$ **do**
    $means.append(cluster.mean)$
  end for
  if $moving\_trajectories \leq NbChanges$ **then**
    $break$
  end if
end while
$return\ Clusters$

Algorithm 2 Clustering Iteration Algorithm

Input $Trajectories : TR, Clusters' means : Means$
Output $Number\ of\ moved\ trajectories : moved, Clusters : c$
$moved \leftarrow 0$
for each $tr \in TR$ **do**
  $Distances \leftarrow Calculate\_distances\_with\_means(tr, Means)$
  $index\_new\_cluster \leftarrow index\_min\_distance(Distances)$
  if $index\_new\_cluster! = tr.cluster\_index$ **then**
    $moved \leftarrow moved + 1$
    $tr.cluster\_index \leftarrow index\_new\_cluster$
  end if
end for
$new\_means \leftarrow calculate\_new\_means(TR)$
$clusters \leftarrow create\_new\_clusters(TR, new\_means)$
$return\ moved,\ clusters$

4.4 Pattern Extraction and Prediction Rules

Pattern extraction is the process of detecting frequent sequences of events between the analysed conservation–restoration trajectories. Our goal is to find the most representative patterns of a class of documents. Recall that we consider two classes of documents, the available documents and the deteriorated ones. These representative patterns are sequences of events that are frequent among the trajectory of a given class and not in the other. In other words, we aim to detect the sequences that can help distinguish between the different classes of documents, i.e. available and deteriorated.

The input of this module is the set of clusters containing conservation–restoration trajectories from different classes. For each cluster, the percentage of trajectories corresponding to documents belonging to both classes is computed. The clusters with a percentage higher than a predefined threshold of one of the two classes will be selected, and the class with the higher percentage will be considered the dominant class. For each of the selected clusters, the conservation–restoration pattern will be extracted to represent the trajectories of this cluster. We define a conservation-restoration pattern as follows:

Definition 11 *Conservation-restoration Pattern:*
Given a cluster of conservation-restoration trajectories cl_i, the pattern P_i corresponding to this cluster is defined as the the mean trajectory of the cluster.

The clusters for which the percentage of trajectories corresponding to deteriorated documents is higher that the threshold are used for the extraction of patterns representing the class of deteriorated documents. In the same way, the clusters for which the percentage of trajectories corresponding to available documents is higher that the threshold are used to extract the patterns representing the class of available documents.
In addition, in the generated clusters, it is possible to find trajectories for which the similarity to the mean of their cluster is low. These trajectories would have been assigned to their clusters because their similarity with the means of the other clusters is even lower. In our approach, in order to identify meaningful pattern trajectories, we propose to filter the clusters by removing these outliers. It is also possible that two patterns corresponding respectively to A and D are very similar. Considering them as a good representative of their class may lead to inaccurate predictions. We propose to exclude these patterns. Consequently, we define the two following filtering rules:

Rule 1 *Outliers filtering. Consider a trajectory Tr_i in a cluster cl_j, with $0 \leq i \leq n$ and $1 \leq j \leq k$; n is the number of trajectories, k is the number of clusters and m_j is the mean of cl_j. If $d_{LCESS}(Tr_i, m_j) > \beta$ then the trajectory is deleted.*

In other words, this rule aims to delete the trajectories for which the distance with the mean of the cluster is higher than a predefined threshold β.
The second rule aims to remove the clusters that will lead to possible false predictions. The second rule depends on the percentage of deteriorated trajectories in the cluster. We determine the type of the clusters based on this percentage and we define the cluster's type and the rule for cluster filtering as follows:

Definition 12 *Cluster type. The type of a cluster cl_j, denoted $type_j$, can be either "available" or "deteriorated". Its type is deteriorated (resp. available) if the percentage of trajectories corresponding to deteriorated (resp. available) documents in the cluster in higher than a threshold γ. The type of a cluster cl_j is denoted $type_j$.*

Rule 2 *Cluster filtering. Consider two clusters cl_i and cl_j, $type_i$ and $type_j$ their respective types, m_i and m_j their respective means.*
If $d_{LCESS}(m_i, m_j) < \alpha$ and $type_i \neq type_j$, then cl_i and cl_j are not considered.

The filtering is performed on the clustering result provided by the k-means algorithm. The outliers filtering rules is first triggered. Each trajectory having a distance to the mean higher than a predefined threshold β is deleted. The means are then re-computed from the set of remaining trajectories.
After the outliers filtering rule, the cluster filtering rule is triggered. For each pair of clusters having different types and such that the distance between their respective means is less than

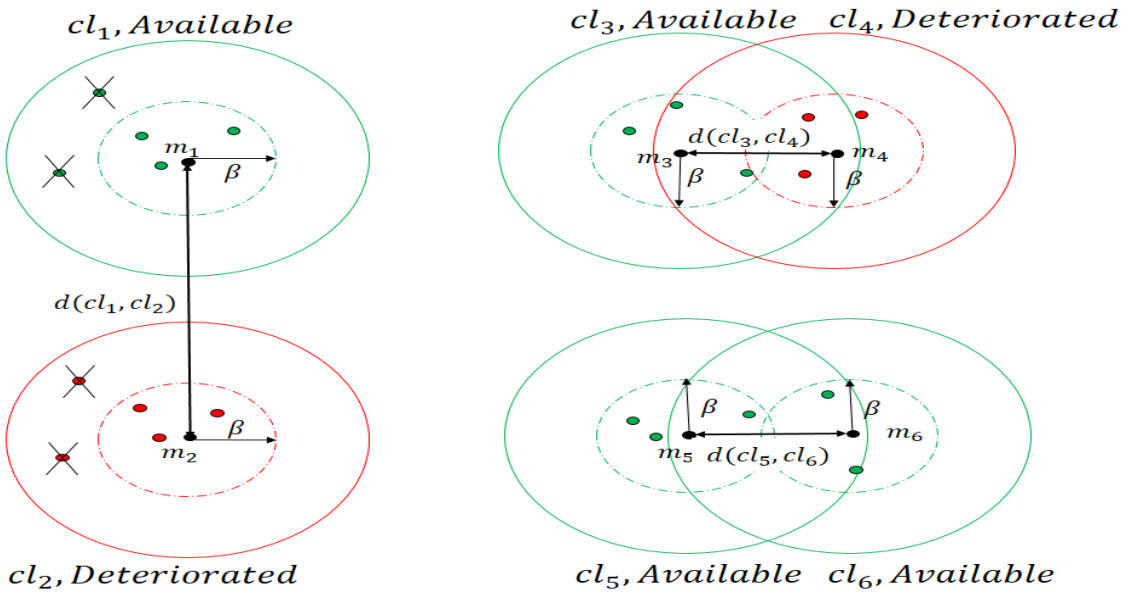


Figure 4.4: Illustration of the filtering rules

a predefined threshold α, these clusters are discarded. After the application of both filtering rules, the means of the remaining clusters will be the output of the pattern extraction module. The means of the clusters for which the type is available will be added to $Pattern\_A$, and the means of the clusters for which the type is deteriorated will be added to $Pattern\_D$.

Algorithm 3 describes the filtering process. The input is a set of clusters resulting from the application of the k-means algorithm and represented in the algorithm by C in line 1. The mean, trajectories and type of a cluster $c \in C$ are respectively denoted by c.mean, c.TR and c.TYPE. The outliers filtering checks the distance between the mean and the trajectories of each cluster and removes the trajectories having a distance greater than β with the mean (line 2 to line 8). After removing the outliers, the means are updated (line 9). The cluster filtering will result in a smaller set of clusters after removing the pairs of clusters having a distance between their means less than a threshold α and having a different type. All the remaining clusters will be added to the $Filtered\_Clusters$ set. The cluster filtering is presented from line 10 to line 26.

Figure 4.4 illustrates the use of the filtering rules using six distints clusters. The two clusters cl_1 and cl_2 illustrate the filtering of outliers, where the trajectories having a distance higher than β with the mean of the cluster were deleted. Two trajectories were deleted in both clusters cl_1 and cl_2. In order to illustrate the cluster filtering rule, we consider that $\alpha = 2\beta$. Assume that the type of two clusters cl_1 and cl_2 is available and deteriorated respectively. The distance between their respective means m_1 and m_2 is higher than the threshold α, and the two clusters should not be discarded. Now consider the two clusters cl_3 and cl_4 and assume their type is available and deteriorated respectively. The distance between their respective means m_3 and m_4 is less than α, then the two clusters should be deleted. Finally, consider the clusters cl_5 and cl_6, where the distance between their respective means m_5 and m_6 is less than α; as the

Algorithm 3 Filtering Algorithm

$C \leftarrow k - means(TR)$
for each $c \in C$ **do**
  for each $tr \in c.TR$ **do**
    if dis(tr, c.mean)$> \beta$ **then**
      c.remove(tr)
    end if
  end for
end for
$Update\_Means(C)$
$Filtered\_Clusters \leftarrow \emptyset$
$i_1 \leftarrow 0$
while $i_1 \leq len(C.means)$ **do**
  $i_2 \leftarrow 0$
  $Conflict \leftarrow False$
  while $i_2 \leq len(C.means)$ **do**
    if dis(C[i_1].mean,C[i_2].mean) $< \alpha$ AND C[i_1].TYPE $\neq$ C[i_2].TYPE **then**
      $Conflict \leftarrow True$
      $Break$
    end if
    $i_2 \leftarrow i_2 + 1$
  end while
  if $Conflict == False$ **then**
    $Filtered\_Clusters.add(C[i_1])$
  end if
  $i_1 \leftarrow i_1 + 1$
end while

type of the two clusters is available, these clusters are not deleted.
The pattern extraction and filtering process output a set of patterns representing the class of the deteriorated documents and a set of patterns representing the class of the available ones. In addition, the cluster filtering ensure that the patterns representing different classes are dissimilar.

To predict a document's physical state, we propose to compare its conservation–restoration trajectory to the patterns representing the two classes of documents, which are the available and deteriorated classes. Suppose the trajectory of a document is similar to a pattern representing deteriorated documents. In that case, it is possible to predict that the document has a chance of also being deteriorated, and it will be classified as deteriorated, the same idea for the available class. The last module in our proposed pipeline concerns defining the prediction rules. The prediction rules are defined based on the patterns representing each class.

Consider a document d_i with its conservation–restoration trajectory Tr_i, The prediction of its physical state starts by calculating the distance between Tr_i and the patterns representing each of the sets A and D corresponding to the available and deteriorated classes, respectively. If the trajectory have a similarity, higher than a predefined threshold, to at least one pattern representing the available (resp. deteriorated) class and if it is not similar to any pattern representing the deteriorated (resp. available) class then it will be predicted as available (resp. deteriorated).
Consider the set $Pattern\_A$ and $Pattern\_D$ previously defined. The rules of the prediction model for predicting the class of a document d_i are defined as follows:

1. If $\exists\ P_x \in Pattern\_D \mid dis(Tr_x, Tr_i) \leq \beta \wedge \nexists\ P_x \in Pattern\_A \mid dis(Tr_x, Tr_i) \leq \beta$ then $PS_i = Deteriorated$
2. If $\exists\ P_x \in Pattern\_A \mid dis(Tr_x, Tr_i) \leq \beta \wedge \nexists\ P_x \in Pattern\_D \mid dis(Tr_x, Tr_i) \leq \beta$ then $PS_i = Available$

Note that in some cases, the class of a document can not be predicted. For example, if Tr_i is not similar to any of the patterns representing the classes or if it is similar to more than one pattern representing different classes, it will be impossible to classify d_i.

4.5 Our Trajectory Clustering and Filtering System

In order to evaluate first the clustering and filtering approach proposed in the analysis pipeline, we have built a system that illustrates the results of these functionalities. This system will help to analyse the clusters and their homogeneity, the impact of the filtering on the clusters, and the patterns extracted.
This section presents the developed system to analyse, cluster and filter the trajectories. The data used in the analysis are extracted from the BnF databases describing real conservation–restoration histories of documents.
In the following, we show the clustering settings in the system, the clustering and analysis of the clusters, and the effect of the filtering on the final clustering results. Before the trajectory clustering, the population on which the clustering will be performed has to be selected. A population is a set of trajectories with shared characteristics such as length. The domain experts might perform the tests on different populations to analyse various aspects. For this reason, we provide the possibility to select some characteristics related to the trajectories or to set some properties for the clustering process. Figure 4.5 shows the interface of the clustering and filtering settings. We can see the parameters that can be set by the user. These parameters are described hereafter.

- Number of deteriorated documents to be used in the experiments (field n°1).
- Percentage of deteriorated documents to be included in the experiments (field n°2).
- Percentage of the testing set (field n°3).
- Minimum and maximum trajectory length (fields n°4 and fields n°5 respectively).
- The number of clusters (field n°6).
- As mentioned previously, trajectory filtering aims to remove the trajectories which are not similar to the mean of the cluster they belong to. A trajectory is considered not similar to the mean when the distance between it and the mean is more than a predefined threshold. This threshold can be set via the interface (field n°7).
- We provide in the system two windows (Canvas) to illustrate the clustering results. The window in which the results will be illustrated should be selected in the interface (radio button n°8).

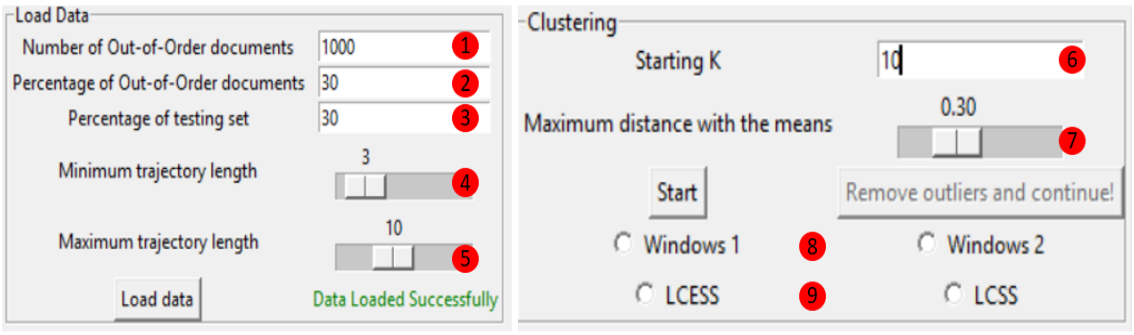


Figure 4.5: Clustering settings

- We also provide in the system to the user the possibility to choose if he wants to integrate CRM_{BnF} in the clustering process or not. In other words, the user can choose between the LCSS measure and LCESS where the former does not use the ontology, and the latter does (field n°9).

After the user has selected the parameters, the clustering is performed, and the results are illustrated in a window representing the clusters and their characteristics, such as the percentage of the deteriorated documents, i.e. out-of-order documents and the number of trajectories in the cluster etc. Figure 4.6 shows a clustering example after selecting 1000 deteriorated documents, 30% as percentage of deteriorated documents and 30% as testing set. For the clustering parameters, the k is equal to 8, the trajectories are considered as outliers f their distance with the mean is more than 0.5.

We illustrate the clustering results in a window as shown in the figure. A cluster is represented with a rectangle containing its characteristics. The clusters' characteristics are the percentage of the deteriorated documents in the cluster, the average length of the trajectories in the cluster, the average distance between the trajectories and the mean of the cluster, the number of trajectories in the cluster, and the number of trajectories which are similar to the mean (have a distance to the mean minimum than the predefined threshold).

The user has the possibility to remove the trajectories that are not similar to the mean, and analyse the characteristics of the clusters also without these trajectories. In addition, the results with and without filtering could be illustrated in separated windows (Canvas) to be compared.

Figure 4.7 shows an example of how the results with and without filtering are represented in two separated windows in the system. The clustering results are illustrated in the left window, and the results after the filtering are represented in the right window, where each rectangle represent a cluster and contains their characteristics.

Some clusters are modified by deleting the trajectories whose similarity with the mean is lower than the predefined threshold, and their mean is therefore recalculated. These clusters are the ones with a red border, and the clusters which have not been modified are represented with a greed border.. Three out of eight means are changed after the filtering.

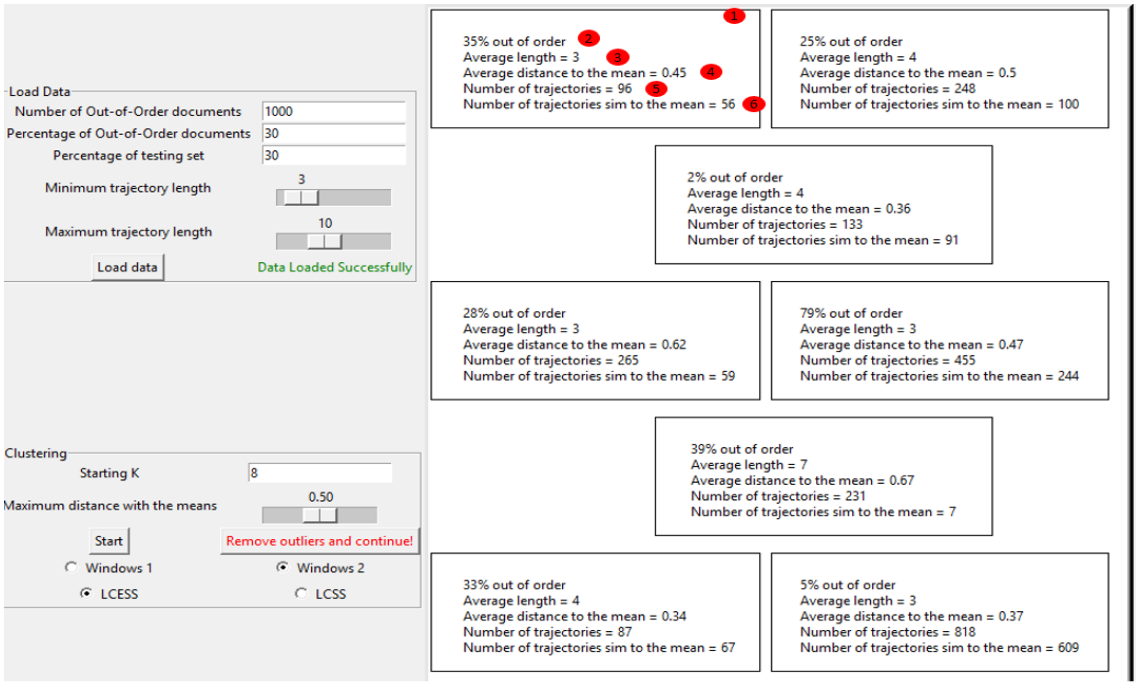


Figure 4.6: Clustering results illustration example

One of the clusters has a high percentage of deteriorated documents (79%) and contains 455 trajectories (fifth cluster). The average distance between the trajectories in this cluster with its mean is equal to 0.47. After the filtering, the remaining trajectories in this cluster are 244 with an average distance equal to 0.29, which is expected as the dissimilar trajectories are removed. The percentage of deteriorated documents is increased to 80%.

4.6 Experiments

This section evaluates the system's performance in predicting the documents' physical state. For the experiments, we have chosen 8000 documents. The length of the corresponding conservation-restoration trajectories ranges from 3 to 10. Among these documents, 25% belong to the deteriorated class. The experiment comprises two phases: training and testing. The training phase shows the clustering and pattern extraction results on 70% of the data set. Based on the generated rules using the patterns in the training phase, the testing phase shows the classification results for the remaining 30% of the data set.

4.6.1 Training Phase

The hyper-parameters of the training phase begins with selecting the appropriate number of clusters for our data set and setting the parameters for this phase. The documents' trajectories are then clustered using the k-means algorithm and the LCESS similarity metric defined in chapter 2.

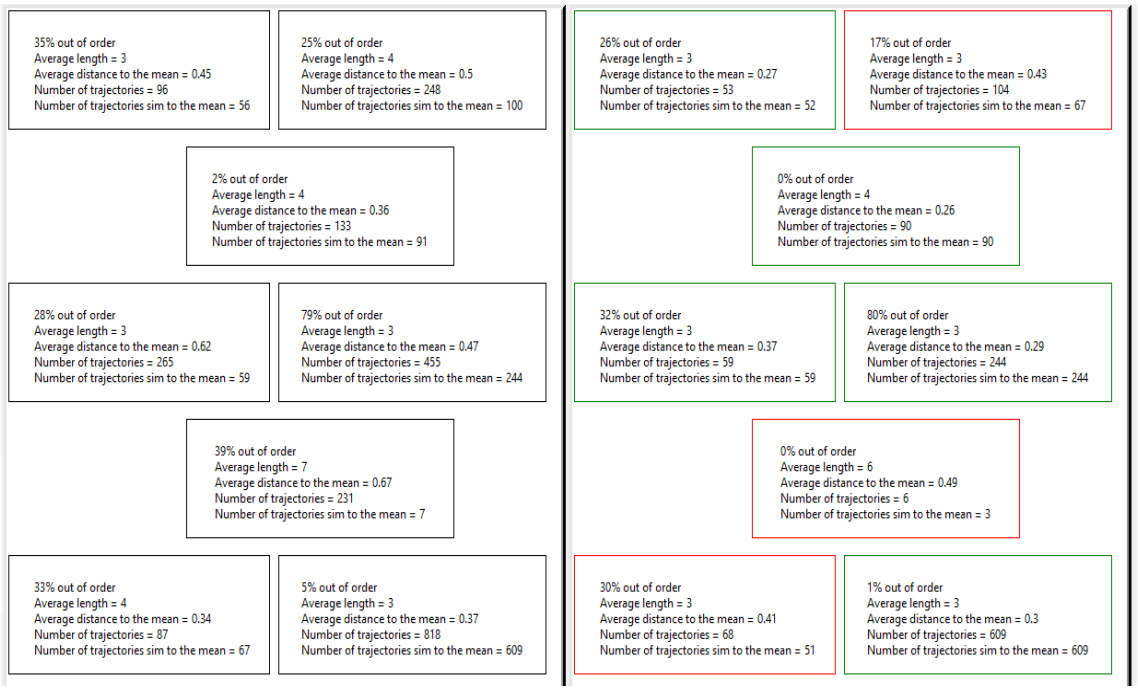


Figure 4.7: Clustering and filtering illustration example

4.6.2 Hyper-parameter Tuning

Before executing the k-means algorithm, the number k of clusters has to be determined. There are numerous techniques for calculating this value, the one used in our work is the well-known elbow method. The approach calculates the within-cluster sum of squared errors (WSS) for various values of k and chooses the k at which the WSS becomes almost stable. The result of the elbow method on the BnF documents' trajectories is shown in Figure 4.8. We set k equal to 13 based on this result. In addition, we set the parameters used in the training stages as follow:

- σ: the threshold for the clustering stability is set to 3 because, based on the tests we have run, the clustering converges to this number.
- β: the threshold for the distance with the mean is set to 0.4. This means that a trajectory should have a similarity greater than 60% with the mean of the cluster in which it belongs.
- α: the threshold for the distance between the clusters of different type is set to 0.5.
- γ: the threshold used to define the type of the clusters is set to 0.7. This means that a cluster's mean is a pattern for the deteriorated documents when the percentage of the deteriorated trajectories in the cluster is higher than 70%.

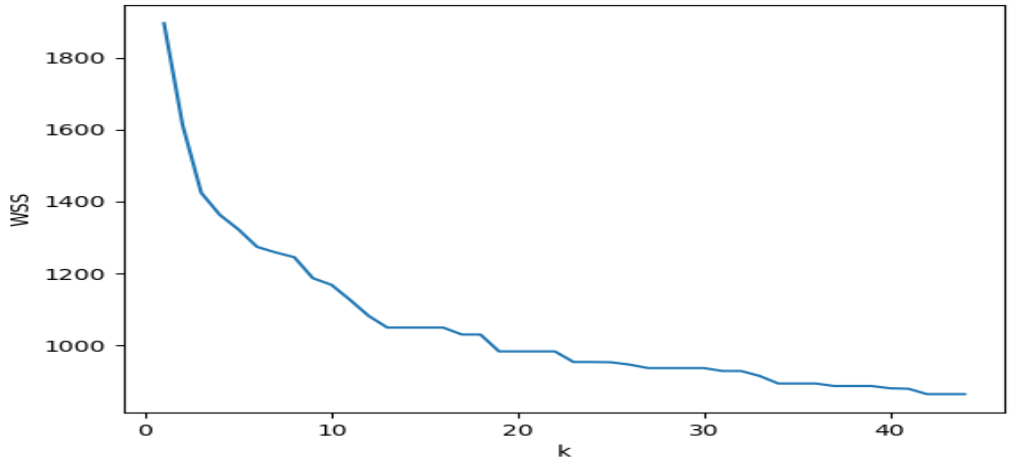


Figure 4.8: Elbow method on the documents

4.6.3 Clustering and Patterns Extraction.

On the training set, we applied the clustering method and we obtained 13 clusters. Two clusters are removed as a result of the cluster filtering because the high similarity of their mean, in addition that they represent different classes. The type of nine of the remaining clusters is available, while the type of the other two is deteriorated. The percentage of deteriorated trajectories, type, and number of trajectories for each of the 11 clusters are shown in Figure 4.9. We have extracted a pattern that represents each of the remaining clusters.

4.6.4 Testing Phase

We used the remaining 30% of the data set for testing, and based on the detected patterns, we classified the trajectories as available or deteriorated using the defined rules. First, we ran the classification on the entire testing set; the results are shown in 4.10, where 310 deteriorated trajectories out of 600 are correctly identified with a 51% accuracy. From a total of 1800 available trajectories, 1723 are correctly identified with a 95% accuracy and an accuracy equal to 84.7% on all testing sets. The results show good precision on the available trajectories and good results on the deteriorated trajectories where more than 50% of the trajectories are well classified regardless of the unbalanced distribution of the classes where only 25% of the training set belong to the deteriorated class.

By analysing the misclassified deteriorated trajectories i.e. deteriorated trajectories classified as available (false negatives), we obtained the distances of the misclassified deteriorated trajectories to their nearest pattern are shown in the figure 4.11. We can see that some of these trajectories have a high distance to their nearest pattern, indicating that they should be considered as outliers and they will be ignored during the classification. As a result, we used a threshold β to evaluate the classification of the trajectories. A trajectory can only be classified if its distance to the nearest mean is less than this threshold; otherwise, it can not be classified. Figure

Cluster ID	Percentage of out-of-order	Type	Number of documents
1	82%	Out-of-order	322
2	2%	Communicable	1166
3	21%	Communicable	516
4	71%	Out-of-order	665
5	62%	Communicable	145
6	17%	Communicable	135
7	1%	Communicable	806
8	28%	Communicable	731
9	23%	Communicable	692
10	2%	Communicable	324
11	17%	Communicable	68

Figure 4.9: Clusters characteristics

4.12 shows the classification results: 560 among 592 available documents are well classified as available and 94 among 110 deteriorated documents are well classified as deteriorated. The model's predication accuracy is equal to 93.1%. The accuracy on the deteriorated documents is equal to 85.45%, and the accuracy on the available documents is equal to 94.59%.

Figure 4.13 shows a comparison of the precision, recall and the f-score of the predictive model with and without the use of a threshold equal to 0.4. The relevant class as mentioned before is deteriorated, which mean the true positives predictions are the deteriorated trajectories that are well predicted. We are more interested to increase the recall that indicates the rate of the well classified deteriorated, which increase from 0.52 to 0.85.

4.6.5 LCSS VS LCESS

We performed another experiment to compare the clustering results using the LCSS measure without dealing with the heterogeneity of the terminology and the LCESS metric. The experiment uses 1000 trajectories with 30% deteriorated. We ran the training on 70% of the trajectories, we varied the threshold between 0.1 and 0.9 for testing, and we displayed both the percentage of successfully predicted trajectories and the recall. Figure 4.14 shows the experiment results where the predictions using LCESS outperform the predictions using LCSS. To predict the class of at least 50% of the trajectories using LCESS, we need a threshold equal to 0.5 and the precision remains high with a value equal to 0.8. Using LCSS to predict at least the communicability of 50% of the trajectories, the threshold has to be set to 0.8 and the precision will be equal to 0.3 which is significantly less than the recall using LCESS.

These results lead us to conclude that integrating the knowledge of the experts in the analysis process is very beneficial, where it succeeded in refining the computation of the simi-

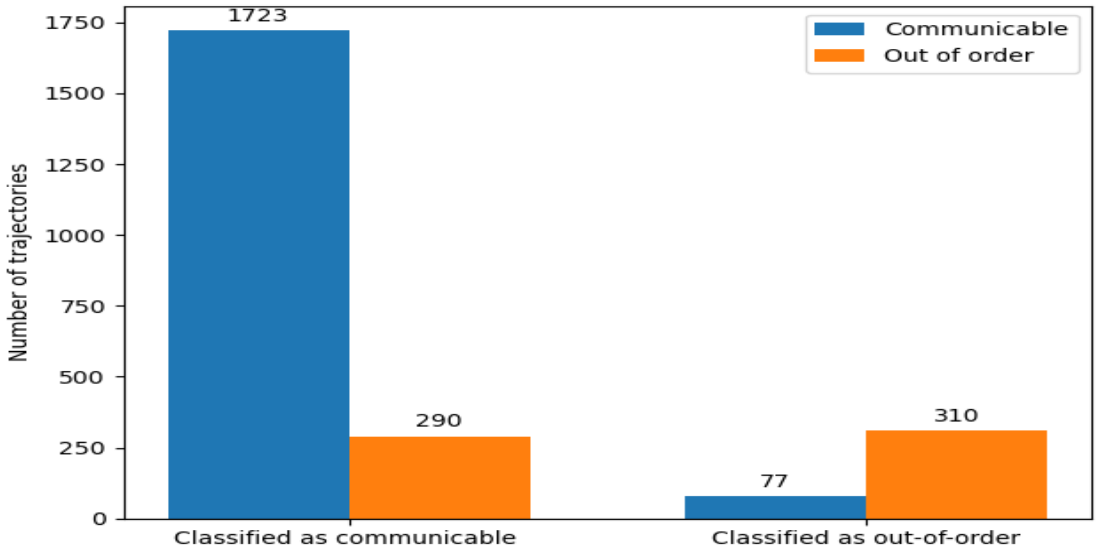


Figure 4.10: Classification on all the testing set

larity between the trajectories, which helps add more semantics to the clustering and pattern extraction processes.

4.7 Conclusion

In this chapter, we have presented a trajectory analysis pipeline that results in prediction rules to predict the documents physical state based on their conservation–restoration trajectories. The pipeline comprises four modules: trajectory extraction, which is a data preparation process, trajectory clustering, pattern extraction, and prediction rules generation.

The trajectory extraction was done manually using SQL queries, which is a weak point because the extraction process must be redefined if new document characteristics become available and must be taken into account.
More enrichment is needed for the BnF databases to consequently enrich the conservation-restoration trajectories with more dimensions to improve the prediction results. In our system, the total prediction accuracy was equal to 85%, and we believe that the accuracy could be increased with more enrichment, such as the physical characteristics of the documents. The physical characteristics could be used to create documents' profiles and analyse each profile and its corresponding documents separately. The learning modules, which are the clustering and the patterns extraction, are tested on the conservation–restoration trajectories, but they could be used for different types of elements by providing the required information, such as the similarity between the elements and their classes.

The prediction model succeeds in predicting the physical state of the documents having tra-

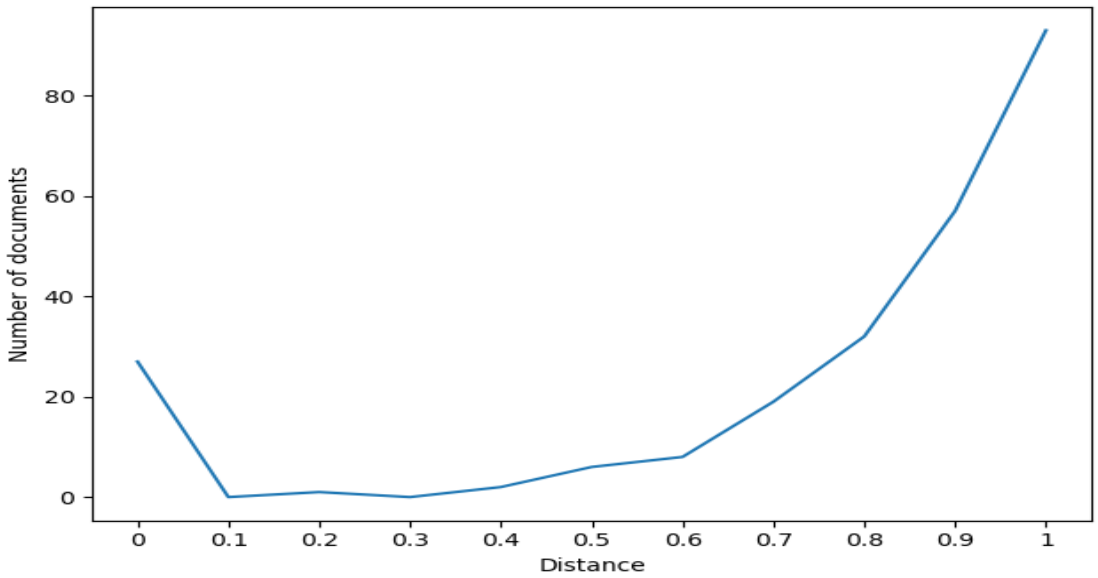


Figure 4.11: Distances of the misclassified deteriorated documents to their nearest pattern

jectories that are similar to some of the patterns. For the other documents, the prediction was impossible given the available characteristics and data. If more characteristics were provided, such as the materials used in the treatment processes, the description of the physical characteristics of the documents such as the type of paper, date of publication or type of binding, then the trajectories could be enriched and the similarity measure could be extended to take into account such new dimensions, which might improve the predictions.

Future work will include enrichment of different aspects in the analysis and prediction process such as the materials used in the conservation–events and more physical characteristics of the documents described in folders called restoration folders. Such enrichment can improve the prediction by introducing dimensions that can possibly help distinguishing between the two classes, consequently increasing the number of accurate predictions.

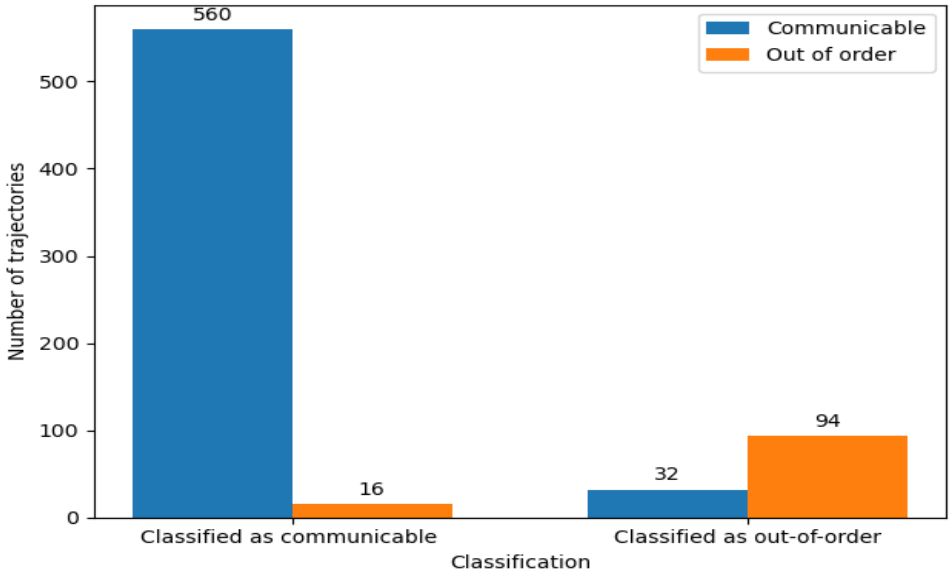


Figure 4.12: Classification with a threshold β

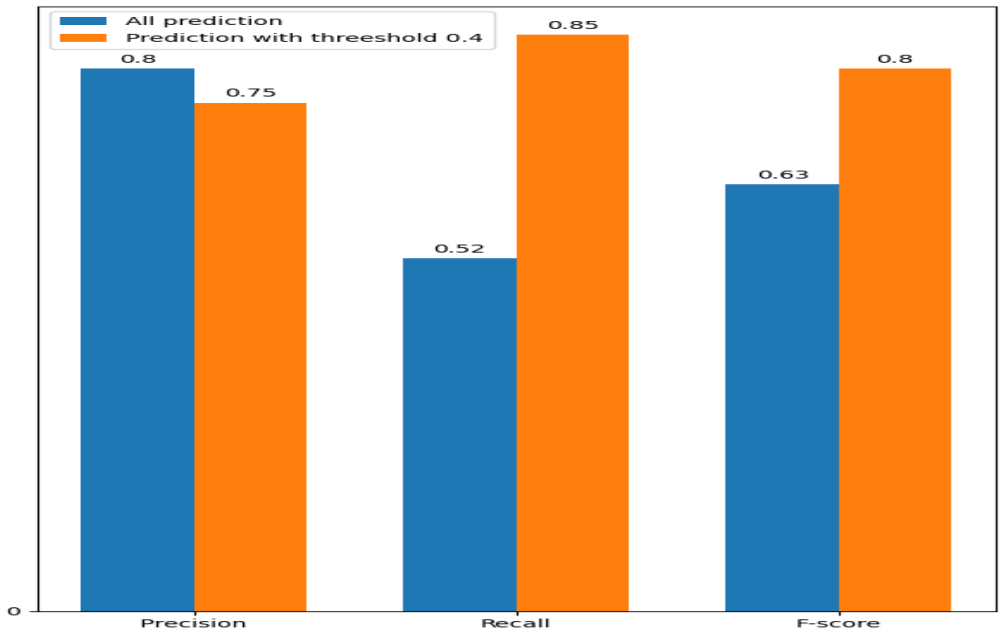


Figure 4.13: Precision, recall and f-score with and without the threshold β

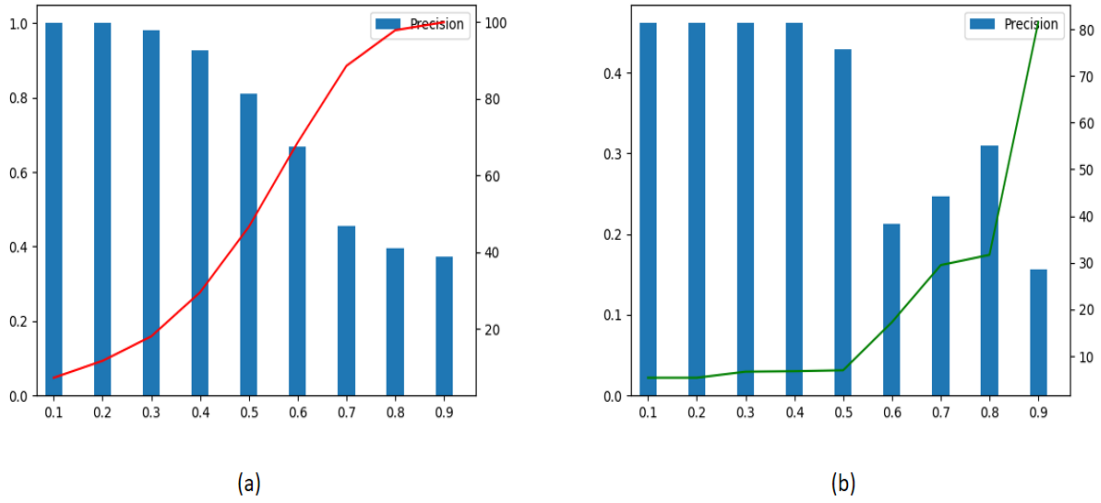


Figure 4.14: Predictions precision and prediction percentage using LCESS (a) and LCSS (b) with different values of the threshold β

5 - Trajectories' Events Weighting

Contents

5.1 Introduction

As mentioned in the previous chapters, matching the events of semantic trajectories is an important analysis task. In a trajectory classification, some events may be have more importance than others in distinguishing between the classes than others. Therefore, when analysing the trajectories, the events with high discriminative power should be given a higher weight during the analysis process. For example, such weights could be taken into account during the computation of the similarity scores between trajectories.
For example, in the healthcare field, a patient is associated with a healthcare history composed of several events related to the prognosis, treatments, and hospitalizations. Some events in this context can be more potent as an indicator of the patient's vulnerability, such as heart surgery or a brain stroke. In our context, an event is a conservation–restoration process, degradation or communication. According to domain experts, some of these events are better indicators of the documents' physical state than others. For example, a "paper acidification" event in a conservation-restoration trajectory is a stronger degradation indicator than a "document box

degradation". Therefore, we propose an approach that determines the weights of the events composing conservation-restoration trajectories considering a specific analysis task. Our goal is, given a set of trajectories and a predefined analysis task, to determine automatically the weight of the events representing their discriminative power in a specific analysis task.
In addition, it is possible that the events are not independent, where semantic relationships exist between them, which should be considered during weight learning. As presented in chapter 3, these relationships could be represented by a tree of concepts where the leaf concepts represent the trajectories' events. Therefore, we propose to weight the trajectories' events by taking into account not only their importance to distinguish between the classes, but also by taking into account the relationships between them.

Among the proposed approaches for assigning weights to the concepts in a tree or a graph, some works have focused on the information content of the concepts, and others have addressed the problem of weighting concepts represented as a hierarchical structure. [100] analyses the existing ontology-based information content computation methods, which are based either on the data and the occurrences of the concepts, or on the structure of the graph from which several characteristics can be extracted such as the concept's depth, the number of hypernyms and hyponyms, or number of leaves concepts. [95] was the first work to introduce the information content in the computation of the similarity between concepts. It has then been extended in [65, 48]. According to [19], graph weighting methods can be either extensional and intensional. The extensional methods rely on the data, while the intensional methods rely on the structure of the graph to determine concept weighting.

The existing weighting approaches assign the same weights to the concepts regardless of the analysis task, which can be inappropriate for long-lived systems which might have different analysis tasks over time. Extensional methods provides the same weights for the input data regardless of the analysis task at hand, as they rely on the data itself. The same holds for the intensional methods, which rely on the tree structure. Unless the tree structure changes, the resulting weights do not vary..
In this chapter, we introduce a novel concept weighting approach that takes into account a predefined labelling of the trajectories in the dataset, corresponding to a specific analysis requirement. The approach transforms the tree into a customized neural network where each node represents a concept from the tree, and the edges between the nodes represent the relationships between the concepts. Based on regression, the approach learns edges' weights that give the best separation of the trajectories' classes, and attribute weights for the concepts based on the edges' weights.

This chapter is organised as follows. In section 5.2, we present a statement of our problem. Section 5.3 present an overview of the weighting approach. Section 5.4 presents the transformation of trajectories into vectors. Section 5.5 is devoted to the transformation of the tree into a neural network. Section 5.6 presents the learning process that enables the definition of concept weights. Section 5.7 presents our experiments, and finally, section 5.8 provides a conclusion and some future works.

5.2 Problem Statement

Let us consider a dataset of trajectories $TR = \{Tr_1, Tr_2, ...Tr_m\}$, and a set of events $E = \{e_1, e_2, ..., e_n\}$. Each trajectory Tr_i is represented by a set of k events $Tr_i = \{e_{i_1}, e_{i_2}, ..., e_{i_k}\}$ with $e_{i_j} \in E$ for $1 \leq j \leq k$. In addition, let us consider that the trajectories of TR are classified into two disjoint classes. Each trajectory Tr_i is associated to a label l_{Tr_i} representing its class.

Given a set of predefined classes, some events may have higher discriminative power than others. The discriminative power of an event denotes its ability to distinguish between classes. For example, an event that appears in trajectories belonging to the same class has high discriminative power and should be given more importance when analysing the trajectories or predicting the class of a given trajectory than the events that characterize those belonging to different classes.
Our problem is identifying the importance of the different events for a given analysis task. This is different from the problem addressed by existing concept weighting approaches, in which weights are assigned independently from the specific task at hand. In other words, given two analysis tasks, such that the labels in the first task are l_1 and l_2, and the ones in the second task are l_3 and l_4, our weighting approach would provide two distinct sets of weights to the concepts, the first one corresponding to their ability to distinguish between l_1 and l_2, and the second one to their ability to distinguish between l_3 and l_4.
In our context, let us consider, for example, two analysis tasks, the first aiming at classifying the documents into one of the classes available and deteriorated, and the second aiming at classifying them into the class of documents requiring a binding and the class of the ones that do not require such action. By analysing the power of the events in distinguishing between the classes, we find that some events are discriminant for the first analysis task, while they may not be for the second analysis. For example, sewing in the document can help predict that the document is available but can not indicate if it needs a binding.
To address this problem, we present a novel weighting approach that assigns weights to the events based on their importance for a given partition of the considered dataset. In addition, we suppose that each event is associated to a concept in a tree representing the SubClassOf relationships between these concepts. In our work, we assume that each event correspond to a leaf concept in the tree, and the concepts and their relationships are represented by a tree structure T. Let $T.R$ be the set of edges in the tree, where each edge is of the form $< c_i, SubClassOf, c_j >$. c_i and c_j are two concepts in the tree, and the edge indicates that the concept c_j is a generalisation of c_i. The former is referred to as the child, while the latter is referred to as the parent. Each event $e_i \in E$ is represented by a leaf concept c_i in T.

Except for one concept, which we refer to as the $T.Root$ concept, we assume that all of the concepts in T have one parent. In addition, we define the set $T.Leafs$, which contains all concepts that do not have a child concept. Figure 5.1 shows a tree representing a set of concepts, and to simplify the reference to the concepts, we represent each concept with a unique id. The tree contains five leaf concepts represented by $\{K, L, N, I, J\}$. For example, the concepts I and J are sub-concepts of the concept G. K, L and M are sub-concepts of

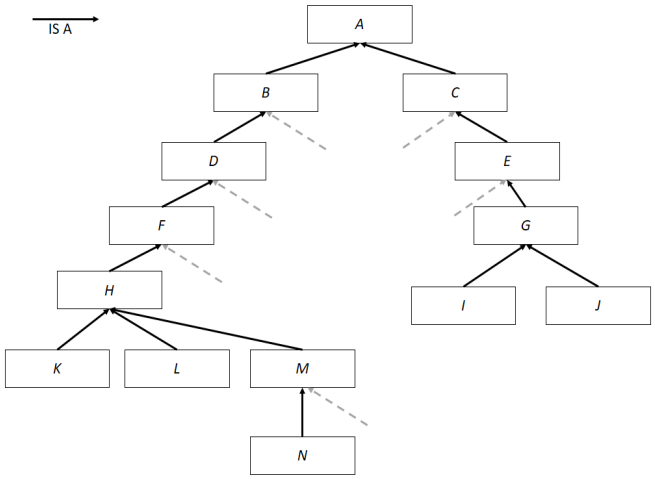


Figure 5.1: Tree representing the hierarchy of concepts

the concept H. The root of the tree is the concept A.

Several research works have proposed approaches for weighting concepts in a knowledge graph. These methods are either based on statistical computations or on the tree structure, which are fixed values that do not change if the analysis task changes, in our cases the classes' labels. Using these methods, the weights of the concepts would be the same regardless of the classes of the trajectories. The problem tackled in this chapter can be stated as follows. Let us consider:

- A set E of events that may be used in a trajectory.
- A dataset Tr composed of trajectories Tr_i, where each trajectory Tr_i is described by a set of events in E.
- A tree of concepts T where each edge between two concepts represents an SubClassOf relationship between them, and where the leaf concepts represent the events describing the trajectories in Tr.
- A partition of Tr in two subsets, where each subset corresponds to a class. Each trajectory Tr_i in Tr is labelled by its class l_{Tr_i}.

Our problem is to determine the weight of each concept in the tree based on its discriminative power to distinguish between the different classes in Tr, taking into account the hierarchical links existing between concepts provided by the input tree. The results of the weighting is a set of weights $W = \{weight_T(e_i)\}$, where $weight_T(e_i)$ is the weight of the event $e_i \in E$ regarding the tree T.

5.3 Approach Overview

In this chapter, we propose an approach that introduces a novel concept weighting principle customized for a given analysis task considering some domain knowledge represented by an SubClassOf tree of concepts and a set of labels defined for the trajectories of the considered dataset. We aim to find the ones that give the highest accuracy for predicting the class of the trajectories. The weight learning of the proposed approach is based on a customized neural network, and the importance of the concepts in the tree are learned using regression on a loss function defined according to the classes of the trajectories. Besides, our approach considers the relationships between the concepts during weight computation.

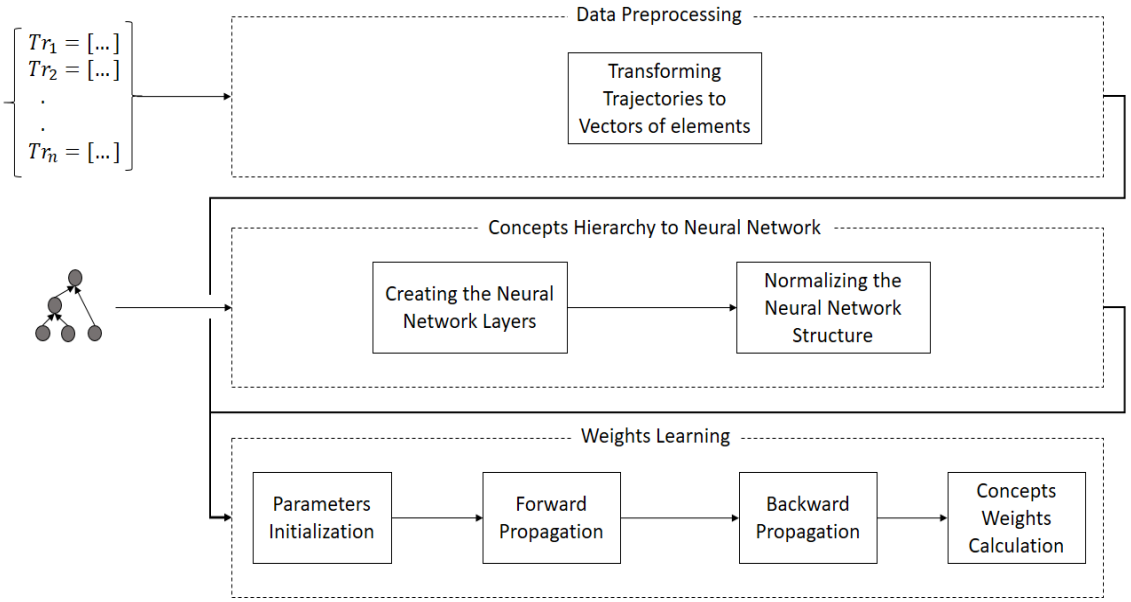


Figure 5.2: General Overview of our Concept Weighting Approach

Figure 5.2 shows the different phases of our weighting approach. The method consists of three phases. The first one is the data preparation which aims to transform the input trajectories into vectors of elements. The second phase builds a neural network based on the concepts' tree. Finally, the third one aims to learn the weights based on regression.

Data Preprocessing

The aim of data preprocessing is to transform the trajectories into a uniform representation capturing the type of events composing them as well as the number of occurrences of each type of event. Each trajectory Tr_i is represented by vector v_{Tr_i} to indicate which events exist in Tr_i and those that do not. The output of this phase is an input of the weights learning phase.

Concepts Hierarchy to Neural Network

As the concepts have relationships represented in an SubClassOf tree, we are interested in considering this information during the weights learning. In this approach, we propose transforming the tree representing the concepts and their relationships into a customized neural network. In this way, a calculation process based on the concepts relationships become possible. Therefore the second phase of this method is to transform the tree into a neural network and adapt its dimension to be normalized to facilitate the subsequent weight learning process. With this transformation, a learning process that considers semantic relationships becomes possible. In other words, performing the forward and backward propagation in the neural network with a structure, i.e. layers and nodes, representing the events and their relationships is advantageous to consider these relationships during the learning.

Weights Learning

In the final phase, the weights of the concepts are learned to indicates their power in distinguishing between the classes. The weights learning is based on the created neural network, as well as the vectors representing the trajectories. In our approach, regression is used in order to learn the weights of the links in the neural network to minimize a predefined loss function that depends on the class of the trajectories. Finally, the links weights in the neural network are considered as the contribution of each concept in distinguishing between the input classes. They are used to extract the concepts' weights.

5.4 Transforming Trajectories into Vectors of Elements

In our context, we consider semantic trajectories representing conservation-restoration histories. These trajectories are represented as sequences of elements with different lengths, each element being a conservation–restoration event. Our goal is to determine the importance of the events based on their frequency in the trajectories and their ability to discriminate between two considered classes, which correspond, in our case, to the class of trajectories corresponding to available documents and the class of the ones corresponding to deteriorated documents. Therefore, the input of the neural network is vectors representing the trajectories in addition to their labels. In order to unify the size of the inputs of the neural network, we propose to transform the trajectories into equal sized vectors.

We would like to consider not only the events in a trajectory but also their number of occurrence. For this reason, the vectors represent the number of occurrences of the events in the trajectories. The size of the vectors is equal to the number of possible events that exist in the database. For example, consider a set of possible events $E = \{e_1, e_2, e_3, e_4, e_5\}$ that contains five events. In addition, consider two trajectories Tr_1 and Tr_2 composed of events in E, where $Tr_1 = \{e_2 \rightarrow e_4 \rightarrow e_1 \rightarrow e_4\}$ and $Tr_2 = \{e_1 \rightarrow e_4 \rightarrow e_3 \rightarrow e_5 \rightarrow e_3\}$. The two trajectories have different lengths and contain different events. Using the proposed uniform representation, Tr_1 and Tr_2 are represented by the vectors v_{Tr_1} and v_{Tr_2} respectively. The size of the vectors is the number of events in E, which is five. The two vectors representing Tr_1 and Tr_2 respectively are $v_{Tr_1} = (1, 1, 0, 2, 0)$ and $v_{Tr_2} = (1, 0, 2, 1, 1)$. Each value in vectors represents the

number of occurrences of an event in the trajectory. For example, in v_{Tr_1}, the first component indicates that the event e_1 has occurred once in Tr_1. The third component indicates that the event e_3 has not occurred in Tr_1, and the forth component indicates that the event e_4 has occurred twice in Tr_1.

5.5 Transforming the Tree of Concepts to Neural Network

During the computation of the concepts' weights, it is important to consider their relationships. For this reason, We propose to build a neural network that considers the concepts and their relationships. The neural network is a transformation of the tree representing the concepts' hierarchy. The neural network structure is based on the structure of the hierarchy. For example, the number of layers in the neural network equals the size of the longest path between the root and a leaf concept in the hierarchy.

5.5.1 Neural Network Layers

The structure of the neural network, i.e., the number of layers, the size of each layer, and the links between nodes, depends on the structure of the tree representing the concepts and their relationships. Therefore in order to build the customized neural network, the following characteristics of the neural network have to be identified.

Input layer. The first layer in the neural network, i.e. the input layer, represents the leaf concepts in the tree. The number of nodes in the input layer is equal to the size of the vectors describing the trajectories. Thus, each event corresponds to a node in the input layer.

Number of layers. Considering a tree that represent the concepts' SubClassOf hierarchy as defined in section 5.2. We define the tree depth $T.depth$ as the path length between the root concept $T.Root$ and the farthest leaf concept in $T.Leafs$, which indicates the number of layers in the neural network. For example, the longest path between the root and the leaves in figure 5.1 is the path between the concepts A and N, and it is equal to six.
Once the number of layers is calculated, the root concept is represented by a node in the output layer, i.e. the last layer in the neural network, and the distance between the root and every concept in the hierarchy which is neither the root nor a leaf node is computed.

Each concept c_i in the tree is represented by a node denoted by n_{c_i} in the neural network. The layer L of a node n_{c_i} that represents the concept c_i is defined as $L = T.depth$-distance(root, c_i).

Neural Network Links. Each parent concept is a generalisation of its children concepts. Therefore if a concept exists in a trajectory's representation, then implicitly, its parent also exists.

We therefore consider that the input of a node n_{c_i} is the output of all the nodes n_{c_j} such that c_i is a specialization of c_j. In other words, the input of a node representing a concept

Algorithm 4 Transforming Concepts' Hierarchy to A Neural Network

```
   Input Concepts' Hierarchy : H
   Stack ← []
2: NN_Size ← H.depth
   AddToNN(concept = Root, layer = NN_Size)
4: for each c ∈ H.leaf_concepts do
       AddToNN(concept = c, layer = 0)
6:     Stack.push(c)
   end for
8: while Stack is NOT EMPTY do
       c ← Stack.pop()
10:    Parent ← GetParentConcept(c)
       if Parent ≠ Root And Parent Not In NN then
12:        ParentLayer ← NN_Size − distance(root, Parent)
           AddToNN(concept = Parent, layer = ParentLayer)
14:        Stack.push(Parent)
       end if
16: end while
```

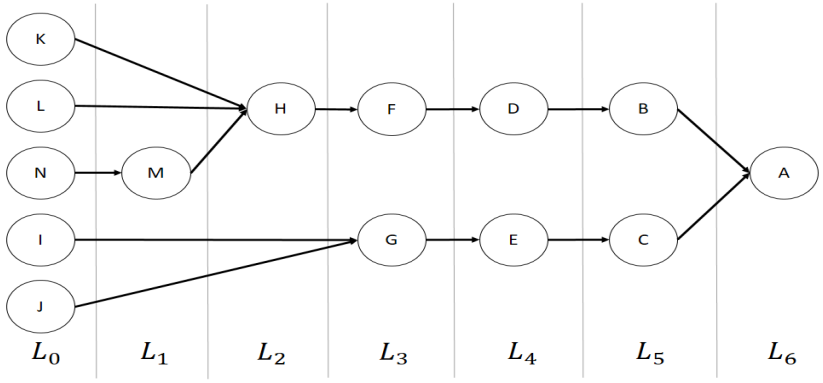


Figure 5.3: Neural network layers

c_i is the output of all the nodes representing the children of c_i. Using this principle, the core semantics of the specialization relationship is preserved. The presence of a child concept in a vector representing a trajectory implies the presence of its parent. As a result, the output value of a node representing a concept c_i is determined by the output values of its children. Therefore the links between the nodes in the different layers depend on the relationships between the concepts in the tree.

Algorithm 4 describes the transformation process. The input of the algorithm is the concepts' hierarchy, and the process starts by setting the neural network size, i.e. number of layers equal to the depth of the hierarchy (line 2). The root concept is represented by a node in the final layer (line 3). Then, each leaf concept is added to the first layer in the neural network and is added to a stack to process its parents' concepts (lines 4 to 7). Next, for each concept added to the neural network that still exists in the stack, the algorithm searches for its parent in the hierarchy, calculates its layer, and adds it to the neural network and the stack if it is not already added. The process ends when all the concepts are added to the neural network, i.e. stack is empty (lines 8 to 16).

Figure 5.3 shows the result of the transformation of the tree represented in figure 5.1 into a neural network. We can see that seven layers have been created. The input layer L_0 contains

five nodes, each of them representing a leaf concept in the hierarchy. The first hidden layer in the neural network, L_1, contains one node representing the concept M, and the input of this node is the output of the node N in the input layer.

5.5.2 Normalizing the Neural Network Structure

As mentioned before, each trajectory in the training set is represented by a vector of size l, which represents the number of leaf concepts in the hierarchy, and which also corresponds to the number of nodes in the input layer. Each value in a vector v_{Tr_i} representing a trajectory Tr_i corresponds to a specific leaf concept in the tree. This value represents the number of occurrences of the concept in Tr_i. The input of the neural network is the set of vectors representing all the trajectories in the training set that is used to learn the weights in the neural network. We recall in the sequel some of the well-known definitions related to neural networks [36]. We define an input vector as follows:

Definition 13 *Input Vector*
We define an input vector as a vector of l dimensions, where l is the number of nodes in the input layer, and we denote a single input vector as $a^{[0]}$.

For the example in figure 5.3, the input layer consists of five nodes; therefore, each trajectory Tr_i is represented by a five-dimensional vector. For instance, the vector representing the trajectory Tr_i is v_{Tr_i} = (1, 0, 0, 1, 1), which means that the concepts K, I and J exist in the trajectory Tr_i and that they occur once, and the value of each node in the input layer is equal to the corresponding value in the vector.

For each node in the neural network we define a weight vector as follows:

Definition 14 *Weight Vector*
A weight vector determines the importance of the inputs of a node n in a hidden layer or the output layer, and is used to calculate the node output. We denote a weight vector of a node n in layer L_i as w_n. The size of the weight vector is equal to the number of nodes in the layer L_{i-1}.

In our example, the node M in the first hidden layer L_1, should only take as input the third value from the previous layer because it is related only to the node representing the concept N. Therefore, the weight vector of the node M is equal to w_M=(0, 0, v_3, 0, 0) and the node's output is equal to $z_M = w_M.a^{[0]} + b_M$, where b_M is the bias term for M.
The node H in the second hidden layer L_2 takes as input values from different layers. H takes the output of the node M and of two nodes in the input layer, which are K and L. Therefore it is difficult to define the size of the weight vector w_H. In addition, the nodes in layer L_3 have input values from different layers, and the length of their weight vectors is different. Consequently, it is difficult to vectorize the calculations in each layer. As a solution, we propose to introduce empty nodes in the layers, defined as follows:

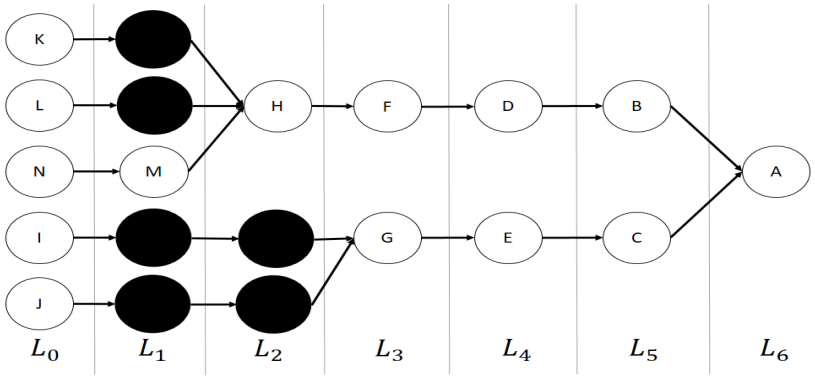


Figure 5.4: Neural network layers after adding the empty nodes

Definition 15 *Empty Node*
An empty node aims to fill a gap in the neural network where a node in a layer L_i should take an input from a node in a layer that precedes L_{i-1}. An empty node in a layer L_i takes its input from only one node in the layer L_{i-1} and provides the same value as an input to a single node in the layer L_{i+1}.

Adding the empty nodes ensures that each node in a layer L_i takes inputs only from nodes in layer L_{i-1} and the size of the weight vectors of all the nodes in a layer L_i is the same, and is equal to the number of nodes in the layer L_{i-1}.
We mention that we refer to every node that is not an empty node by actual node.

Figure 5.4 shows the neural network after adding the empty nodes. In layer L_1, four empty nodes were added to provide the value of the nodes K, L, I and J from the input layer. The gap between nodes I and J with their parent was equal to two, i.e. the number of layers between L_0 and L_3. As a result, empty nodes for these two nodes were also added to the layer L_2. Once the empty nodes are added, the computation in each layer can be vectorized, and the value vector of a layer to calculate is defined as follows:

Definition 16 *Value Vector of a Layer*
The value vector of a layer depends on the nodes' weight vectors, input vectors and the nodes' bias terms in the layer. We denote the values calculated in a layer L_i by $z^{[i]}$:

$$\underset{n\times 1}{z^{[i]}} = \underset{n\times m}{w^{[i]}}.\underset{m\times 1}{a^{[i-1]}} + \underset{n\times 1}{b^{[i]}} \tag{5.1}$$

$$\underset{n\times 1}{z^{[i]}} = \begin{pmatrix} w_1^{[i]T} \\ w_2^{[i]T} \\ \vdots \\ w_n^{[i]T} \end{pmatrix} \begin{pmatrix} a_1^{[i-1]} \\ a_2^{[i-1]} \\ \vdots \\ a_m^{[i-1]} \end{pmatrix} + \begin{pmatrix} b_1^{[i]} \\ b_2^{[i]} \\ \vdots \\ b_n^{[i]} \end{pmatrix} = \begin{pmatrix} z_1^{[i]} \\ z_2^{[i]} \\ \vdots \\ z_n^{[i]} \end{pmatrix}$$

Where:

- *$w^{[i]}$ is a matrix of $n \times m$ dimensions containing the transpose of the weight vectors of the nodes in layer L_i, n is the number of nodes in the layer L_i, and m is the number*

of nodes in the layer L_{i-1}. Each row r in the matrix represents the weight vector of a node x in the layer L_i.

- *$a^{[i-1]}$ is a matrix of $m \times 1$ dimensions containing the output values of the layer L_{i-1} with $a^{[0]}$ is the input of the neural network.*
- *$b^{[i]}$ is a matrix of $n \times 1$ matrix containing the bias terms of the nodes in the layer L_i.*

Finally, we define the output of a layer L_i as follows:

Definition 17 *Output of layer L_i*
The output of a layer depends on its values vector and the used activation function. We denote the output as act($z_{[i]}$), where act is the activation function used in the nodes of this layer.

In the example of figure 5.4, the layer L_2 contains three nodes, therefore $w^{[2]}$ is equal to:

$$\begin{pmatrix} w_{1,1}^{[2]} & w_{1,2}^{[2]} & w_{1,3}^{[2]} & 0 & 0 \\ 0 & 0 & 0 & w_{2,4}^{[2]} & 0 \\ 0 & 0 & 0 & 0 & w_{3,5}^{[2]} \end{pmatrix}$$

The values in $w^{[2]}$ indicate that the first node in the layer L_2, which is H, takes inputs only from three nodes from L_1 with the weights $w_{1,1}^{[2]}$, $w_{1,2}^{[2]}$ and $w_{1,3}^{[2]}$ corresponding to the first, second and third nodes in L_1 respectively. Each empty node in the second layer takes its input from only one node from the previous layer.

5.6 Weights Learning

The weights learning aims to assign for each concept in the hierarchy a weight indicating its power in distinguishing between the classes. The learning process is based on the neural network that depends on the structure of the hierarchy.
The weights learning starts by initializing the neural network parameters, which are, the initial weight vectors of the actual and empty nodes, and the bias terms. Then, in each iteration, forward and backward propagation [110] is performed to calculate the prediction loss and update the weights in order to decrease the loss.

Forward propagation is executed to calculate the prediction loss with the given parameters. Then backward propagation is performed to update the parameters so as to minimize the loss. Once the minimum loss is achieved, the final parameters are used to compute the weights of the concepts in the tree.

5.6.1 Parameters Initialization

Before weight computation, for each layer L_i, the bias terms $b^{[L_i]}$ and the weight vectors $w^{[L_i]}$ of the neural network nodes are initialized. $w^{[L_i]}$ is a $n \times m$ matrix where n is the number of nodes in the layer L_i and m is the number of nodes in the layer L_{i-1}. $w_s^{[L_i]}$ is the weight vector of the node at the position s in the layer L_i with $1 < s < n$, and $w_{s,p}^{[L_i]}$ represent the weight

between this node and the node at the position p in the layer L_{i-1}. $b^{[L_i]}$ is $n \times 1$ matrix and each row contains the bias term of a node.

In the initialization, we distinguish between the actual nodes and the empty nodes. For a real node representing a concept c, the weights of the neural network links between this node and all the nodes representing the direct children of c in the previous layer are initialized by assigning a random strictly positive value, and the other weights in the weight vector of this node are equal to zero. The bias term is also be initialized to a random value.
Considering an empty node e_x at layer L_i having a predecessor node x from layer L_{i-1}, the weight between e_x and x is initialized to 1 and the other weights in the weight vector of this node is equal to zero. The bias term of this node is equal to zero. For example, the weights of the layer two $w^{[2]}$ corresponding to the neural network in the figure 5.4 are the following:

$$\begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

5.6.2 Forward Propagation

As mentioned in section 5.2, our goal is to assign weights to the concepts in the tree according to their importance in distinguishing between the classes. Each trajectory Tr_i is represented by an l-dimensional vector v_{Tr_i}, where l is the number of nodes in the input layer. In addition, Tr_i has its label l_{Tr_i}. In a binary prediction, l_{Tr_i} takes one of two possible values. In our work we consider that $l_{Tr_i} \in \{0, 1\}$.
In forward propagation, the input of the neural network is a set of vectors representing the training set denoted by $A^{[0]}$. The input is a $n \times m$ matrix, where n is the number of nodes and m is the number of trajectories in the training set.

The output of each layer L_i is $A^{[L_i]} = tanh(Z^{[L_i]})$ with $1 \leq i \leq TL - 1$ where TL is the total number of layers, and $Z^{[L_i]} = w^{[L_i]} A^{[L_{i-1}]} + b^{[L_i]}$.
For the output layer, we use the sigmoid activation function, $A^{[L_{TL}]} = \frac{1}{1+e^{Z^{[L_{TL}]}}}$, and the loss is calculated using the logarithmic loss function:

$$\mathcal{L} = -\frac{1}{m} \sum_{x=1}^{m} (l_{e_x})(\log A_x^{[TL]}) + (1 - l_{e_x})(1 - \log A_x^{[TL]}) \tag{5.2}$$

5.6.3 Backward Propagation

Once the loss is calculated at the end of each forward propagation iteration, the weights and the bias terms for each layer are updated.

In the case of empty nodes, the weights should not be modified and they should always provide the same value of a node in a previous layer to a node in a next layer. Therefore, the weight vector of an empty node contains zeros except for one value equal to one, which corresponds to the node in the previous layer where the empty node provides its output value. In addition the bias term of an empty node is equal to zero. Therefore, in order to maintain

the objective of the empty nodes, which is to provide the same value from a node in a previous layer, their weight vectors and their bias terms should not be modified.

In addition, the weight vector of each actual node is initialized depending on the relationship between the corresponding concepts in the tree, i.e., given two nodes n_s and n_p in the layer L_i and L_{i-1} respectively, where n_s represents a concept c_s, and n_p represents a concept c_p, the weight between the two nodes should be always equal to zero if there is no $SubClassOf$ relationship between the two concepts in the tree.
Consequently, when updating the weight $w_{s,p}^{[L_i]}$ between the sth and the pth nodes n_s and n_p representing the concepts c_u and c_v in the layer L_i and L_{i-1} respectively, we distinguish between the following cases:

- Case 1: The nodes n_s and n_p are actual nodes and there is an $SubClassOf$ relationship between c_u and c_v. In this case, the weight $w_{s,p}^{[L_i]}$ is updated depending on the learning rate α and the derivative of the loss with respect to $w_{s,p}^{[L_i]}$.
- Case 2: The node n_s is an actual node and n_p is an empty node that should pass the value to the node n_s. Similarly to the first case, the weight $w_{s,p}^{[L_i]}$ is updated depending on the learning rate α and the derivative of the loss with respect to $w_{s,p}^{[L_i]}$.
- Case 3: The nodes n_s and n_p are actual nodes and there is no a $SubClassOf$ relationship between c_u and c_v. In this case, the weight $w_{s,p}^{[L_i]}$ is always equal to zero.
- Case 4: The node n_s is an empty node. In this case the weight vector of the node n_p should not be modified as it is an empty node.

In the first two cases, the equation to update the weight vectors is as follows:

$$w_{s,p}^{[L_i]} = w_{s,p}^{[L_i]} - \alpha \frac{\partial \mathcal{L}}{\partial w_{s,p}^{[L_i]}} \tag{5.3}$$

To generalize the weights update, we define for each layer L_i a $n \times m$ matrix called $Valid\_Input^{[L_i]}$, where n is the number of the nodes in layer L_i and m is the number of nodes in layer L_{i-1}. The definition is given hereafter.

Definition 18 *Valid Input Matrix*
The valid input matrix indicates which values in the weight vectors could be changed according to the cases defined above.

The element $Valid\_Input_{s,p}^{[L_i]}$ is equal to one if the weight between the sth node in the layer L_i and pth node in the layer L_{i-1} $w_{s,p}^{[L_i]}$ can be updated, and zero if it can not be updated.

The updated weight vector matrix $w^{[L_i]}$ of layer L_i is computed as follows:

$$w^{[L_i]} = w^{[L_i]} - \alpha \frac{\partial \mathcal{L}}{\partial w^{[L_i]}} \; Valid\_Input^{[L_i]} \tag{5.4}$$

The bias terms in the nodes are updatable in the actual nodes only. For this reason we also define, for each layer L_i a $n \times 1$ matrix called $Nodes\_Type^{[L_i]}$ where n is the number of

nodes in the layer L_i, and the element $Nodes\_Type_s^{[L_i]}$ is equal to one if the sth node in the layer is an actual node and zero if it is an empty node. The equation to update the bias terms in the layer L_i is:

$$b^{[L_i]} = b^{[L_i]} - \alpha \frac{\partial \mathcal{L}}{\partial b^{[L_i]}} \; Nodes\_Type^{[L_i]} \tag{5.5}$$

5.6.4 Concepts Weights Calculation

Once the minimum loss is reached, we assume that the influence of the nodes on the loss result represent their importance. In other words, the impact of a node representing a concept c_i on the prediction result, i.e. the neural network's final output, is the partial derivative of the final output with respect to this node's value. We use this principle to calculate the weights of the leaf concepts. The weights of the concepts at higher abstraction levels are calculated following a bottom-up approach depending on the children's weights. The impact of a node n_{c_i} representing a leaf concept c_i is defined as follows:

$$impact_{n_{c_i}} = \left| \frac{\partial A^{[TL]}}{\partial n_{c_i}} \right| \tag{5.6}$$

The weight of the concept c_i is the normalization of the impact of the corresponding node:

$$weight_{c_i} = \frac{impact_{n_{c_i}} - \min\limits_{c \in G.Leafs}(impact_{n_c})}{\max\limits_{c \in G.Leafs}(impact_{n_c}) - \min\limits_{c \in G.Leafs}(impact_{n_c})} \tag{5.7}$$

The weight of a non-leaf concept c_i is computed as the average of its children weights and defined as follows:

$$weight_{c_i} = AVG_{c \in hyponym(c_i)}(weight_c) \tag{5.8}$$

5.7 Evaluation

In the evaluation, we aim to test our weighting approach and compare it to state-of-the-art weighting methods. The goal is to assign high weights for the concepts that can help distinguish between the classes and low weights for the concepts that are not. The experiments presented in this section show that the proposed weighting method enables the accurate detection of the concepts that discriminate between the classes. High weights are assigned to these concepts. The results of our experiments show that our approach allows to capture the importance of the concepts as it derives higher weights to the concepts that discriminate between the two considered classes. The data used in the experiments is related to the Conservation-Restoration field and was extracted from the National Library of France (BnF) databases.
In the following, we will present the datasets used in our experiments. Then we will present the results of our weighting approach and compare them to the weights generated by some of the well-known weighting approaches.

5.7.1 Data

In order to test our weighting approach, we have used the CRM_{BnF} ontology introduced in [133] [134] which contains concepts representing conservation–restoration events and their generalization/specialization hierarchy.

The data used is a set of documents $D = \{d_1, d_2, ..., d_n\}$, where each document d_i is represented by a sequence of events S_i representing the conservation–restoration history of the document. Each sequence S_i is transformed into a set of events. $Concepts_{d_i}$ contains all the concepts representing the events that occurred in the conservation–restoration history of d_i. Each document d_i is associated with a label l_{d_i} representing its physical state where it can either be "$Available$" or "$out - of - order$". The evaluation is done on 11603 documents, and the number of distinct events in their conservation–restoration histories is equal to 262.

5.7.2 Concept Weighting Results

The goal of this experiment is to assess the effectiveness of our weighting approach in detecting the concepts that can help distinguish between the documents classes. After adding the empty nodes, the transformation of the tree that represent the concepts and their generalization hierarchy results in a neural network of 7 layers. The input layer contains 262 nodes representing the leaf concepts of CRM_{BnF}, and five hidden layers l_1, l_2, l_3, l_4 and l_5 containing 261, 249, 110, 26, 2 nodes respectively, and a output layer of one node representing the root concept of the ontology, i.e. $Event$. The weights are calculated then normalized, and the resulting values range in the interval [0, 100], where 0 indicates that the concept is not at all relevant for distinguishing between the two classes, and its discriminative power is null and 100 indicates that this feature characterizes documents belonging to one of the two classes only. Table 5.1

		Occurrences in classes		
ID	**Concepts**	**Available(/6962)**	**Out-of-order(/4641)**	**Weight**
3	Couture sur cahiers	687	28	77
72	Restauration reliure	59	0	100
4	Couture sur surjets	493	20	68
12	Couvrure Pleine toile	2425	123	37
165	COUV COINS Usure	15	127	37
152	MORSSUP Coupé	335	359	0.28
119	COUV PLATSUP Usure	503	577	0.12
144	COUV DOS Salissures	948	788	0.9
122	COUV Décollement	821	1227	0.65
147	COUV DOS Lacunes	800	1145	5.2

Table 5.1: Analysis-based Weighting Results

shows results that assert that the values of the weights converge to 0 or 100 depending on the distribution of the concepts in the two classes. The table contain the names the leaf concepts of the hierarchy, their occurrences in the two classes and their extracted weights. Five of them,

represented in green colour, have significant weights and the rest of the concepts, represented in red colour, have low weights. The concept "Restauration Reliur" has the highest weight, equal to 100, and by analysing its occurrences, we can see that this concept appears only in the available class. In other words, having a document containing this concept in its features indicates that the label of this document can be predicted to be "available" with a 100% accuracy. The same holds for the other concepts with high weights; we can see that their distribution among the classes is not balanced, and they can be used to accurately predict the documents' labels. If we consider the concepts with low weights, we can see that their distribution among the different classes tends to be uniform, and therefore they have low discriminative power.

Concept's ID	Our Approach	AF	CF	TD	Bayesian
3	77	27	37	95	50
72	100	51	58	95	50
4	68	31	41	95	50
12	37	14	26	72	9
165	37	47	55	76	22
152	0.28	29	40	70	21
119	0.12	25	36	99	23
144	0.9	19	31	99	23
122	0.65	18	30	99	23
147	5.2	19	31	99	23

Table 5.2: Concepts Weights Using AF, CF, TD, Bayesian and Analysis-based methods

The weights of the concepts are shown in table 5.2 using the weighting methods presented in section 2.3.2. Based on the occurrences of the concepts, the AF and CF methods give weights to the five non-relevant concepts greater than the concept "12", which should be more important. Furthermore, given that the weights are very close, the concepts are not distinguished using these extensional weighting methods. Depending on the graph structure, the TD approach provides the highest weight to three non-relevant concepts. Finally, the Bayesian method does not distinguish between the concepts, where the highest weights are equal to 50, and all the non-relevant concepts have a weight greater than the concept "12". We can see that the other weighting methods assign the weight regardless of the distribution of the concepts in the two classes. Our proposed approach successfully detects the ones that can help distinguish between the classes.

5.8 Conclusion

In this chapter, we have proposed a novel concept weighting method based on a neural network approach. The workflow corresponding to our approach starts by transforming the trajectories to vectors and the concepts' hierarchy to a neural network. Then the weights are learned using regression on a predefined loss function that depends on the classes considered for the analysis

task.

The trajectory transformation aims to unify the dimensions of the neural network input. Therefore, the vectors representing the trajectories have the same size and represent the events occurrences. The introduced process to transform the concepts' hierarchy into a customized neural network represents each concept by a node in the neural network and its relationships are respected and represented by edges between nodes. The neural network size, layers and edges depend on the structure of the concepts' hierarchy.

In the weights learning process, we have proposed different adaptation of the forward and backward propagation processes to take into account the relationships between the concepts and to distinguish between the different types of nodes, i.e. actual and empty nodes.
The experiments have shown that our method outperforms the existing weighting methods to detect the useful concepts to distinguish between the classes. The method gives high weights for the concepts that frequently appear only in one class and low weights for those that frequently appear in more than one class. The proposed approach give the possibility to give different weights for the elements that compose the trajectories when the analysis task is different.
In this approach, we considered the SubClassOf relationship between concepts, but we could also propose extensions that allow taking into account more semantic relationships of another type. In addition, we could also introduce algorithms to weight concepts that are not represented by a hierarchy but by graphs of different types, such as RDF graphs or property graphs in graph databases.

6 - Conclusion

The aim of our work is to propose contributions towards a decision support system for the conservation–restoration experts to help in predicting the documents' physical state. To this end, we have addressed different problems posed by the design of an analysis pipeline tailored to supporting the experts in the definition of their conservation-restorations policies.

The first of these problems is the identification of the relevant data in the heterogeneous distributed BnF databases to generate a conservation–restoration trajectory for each of the documents and to define a suitable representation for these trajectories. The second problem is related to the definition of an enhanced trajectory matching process that can solve the terminological heterogeneity between the trajectories by considering the heterogeneous terminology used in the elements constituting them. In addition, this work tackled different problems in the advanced analysis of these histories, such as proposing an analysis pipeline that integrates experts' knowledge to result in good predictions. Another problem we have addressed is identifying the events' weight that represents their importance in the analysis task.

6.1 Summary of the Contributions

Our first contribution, described in Chapter 3, is related to the generation, representation and matching of the semantic trajectories representing conservation–restoration histories. We have proposed a representation of these histories as semantic trajectories, which are sequences of events. In addition, in this chapter, we introduced a novel similarity measure that tackles the terminological heterogeneity of the events composing the trajectories. The proposed similarity measure uses an external knowledge source representing the domain experts' knowledge. We have initiated a conceptualization of this knowledge as an ontology where each concept represents an event, and we have defined different types of relationships between the events using this ontology.

The second contribution, presented in chapter 4, is related to the analysis of the conservation–restoration trajectories in order to predict the documents' physical state. We proposed an analysis pipeline that predicts the document's physical state based on its conservation–restoration trajectory. The proposed pipeline is formed of different modules. We have proposed to transform the conservation–restoration histories into trajectories. In addition, we have introduced an approach to identify similar trajectories based on clustering, where we have used the k-means algorithm with adaptation to our context on calculating the means. Finally, we have proposed a filtering process to identify the relevant clusters from which we extract patterns to represent the different classes. Finally, in the last module, we propose prediction rules based on the extracted patterns.

In order to capture the importance of the events for a specific analysis task, we have proposed a novel approach for weighting the types of events composing a trajectory. We have

proposed to give each element a weight representing its importance. The weights learning is based on regression using a customized neural network that we propose to build based on the concepts hierarchy representing the trajectories elements and their relationships.
Learning the weights starts with data preprocessing, where we have proposed transforming the trajectories into element vectors. In addition, we have proposed an approach to transform the concepts hierarchy into a neural network and to normalize the neural network structure. Finally, we have proposed to learn the weights using regression. The learning is based on forward and backward propagation, and the loss function is related to the class of the trajectories. Therefore by changing the goal of the analysis, i.e., trajectories' classes, the method gives different weights.

6.2 Future Works

Our work still has a high potential for future improvements. For conservation–restoration history creation, the integrated database could be further enhanced in future works by adding more relevant properties, such as the preservation materials employed and more physical characteristics of the documents detailed in restoration folders (unstructured data). Therefore, the suggested ontological model could also be enhanced in future works by adding new dimensions representing these characteristics. In addition, adding reasoning capabilities to our similarity computation would allow us to consider the knowledge explicitly supplied in the model and the knowledge that can be inferred. This would be another area of research. Moreover, the similarity computation should also be adapted to take into consideration more semantic relationships related to these new dimensions. Such enrichment can improve the prediction by introducing dimensions that can possibly help in the analysis task, consequently increasing the number of accurate predictions when predicting the documents physical state.

Regarding the proposed weighting method, it could be more automatized and generalized. For example, the is-a relationships between the concepts in the hierarchy were implicitly transformed into output and input links between the neural network nodes. When adding more complex relationships to the ontology, more complex transformation rules could be defined when transforming the ontology into a neural network.
In addition, some constraints on the trajectories could be removed. For example, a trajectory could be represented by concepts at any level of the hierarchy; therefore, the input will affect different layers in the neural network. Finally, the weights of the events could be integrated into the analysis process to analyze their impact on the prediction results. The weight could be used when matching the events and calculating the similarity between the trajectories.

A - Appendix : CRM BnF

The CRM_{BnF} ontology contains more than four hundred concepts representing the events constituting the conservation–restoration trajectories or new concepts that we add to represent the group of similar ones. In this appendix, we present some of the dimensions of this ontology.

The root of the ontology is the event concept, and its direct sub-concepts are the conservation–restoration process, i.e. treatment, and degradation. We have identified 19 groups of the conservation–restoration processes; in other words, we have identified 19 different purposes of these events.
Figure A.1 shows the different groups of the conservation–restoration processes.

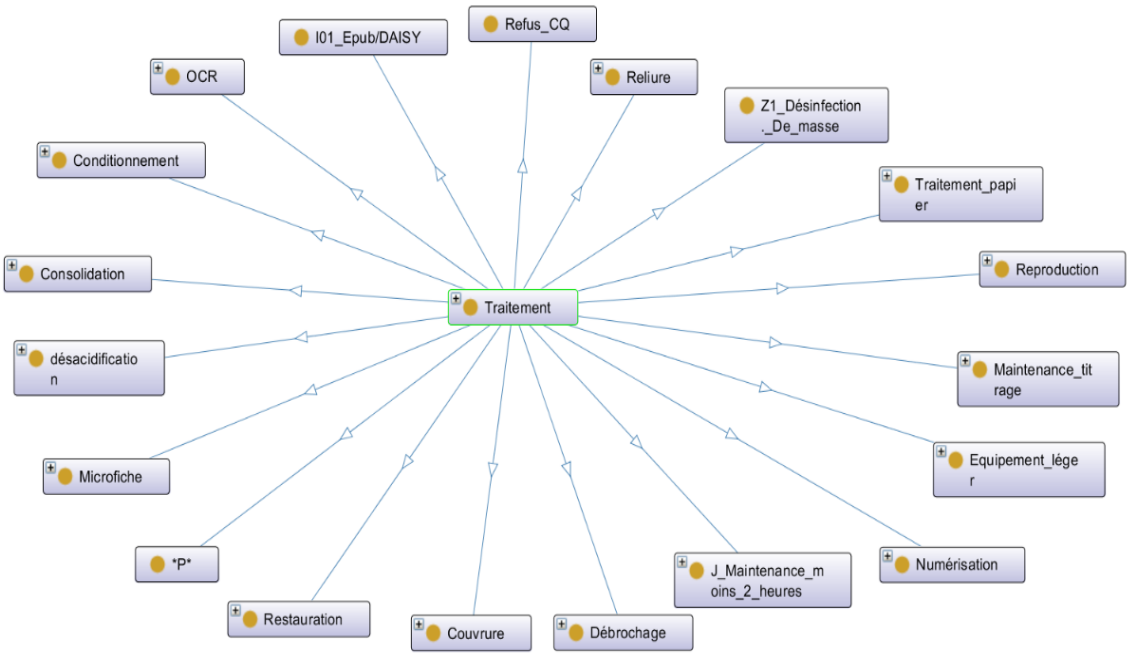


Figure A.1: Conservation–restoration processes groups

Concerning the degradations, we have identified 11 different types of these events. Figure A.2 shows the direct sub-concepts of the degradation concept.

The consolidation is a conservation–restoration group containing different types of consolidation events. Figure A.3 shows the concepts in this group. Some concepts were added to represent similar events, such as "Japon" and "cuir". A double path between the events represents the equivalence relationship. For example, the three sub-concepts of "Japon" are equivalent.

The degradation concepts was grouped depending on the degraded part in the document. Figure A.4 shows the different concepts in this group.

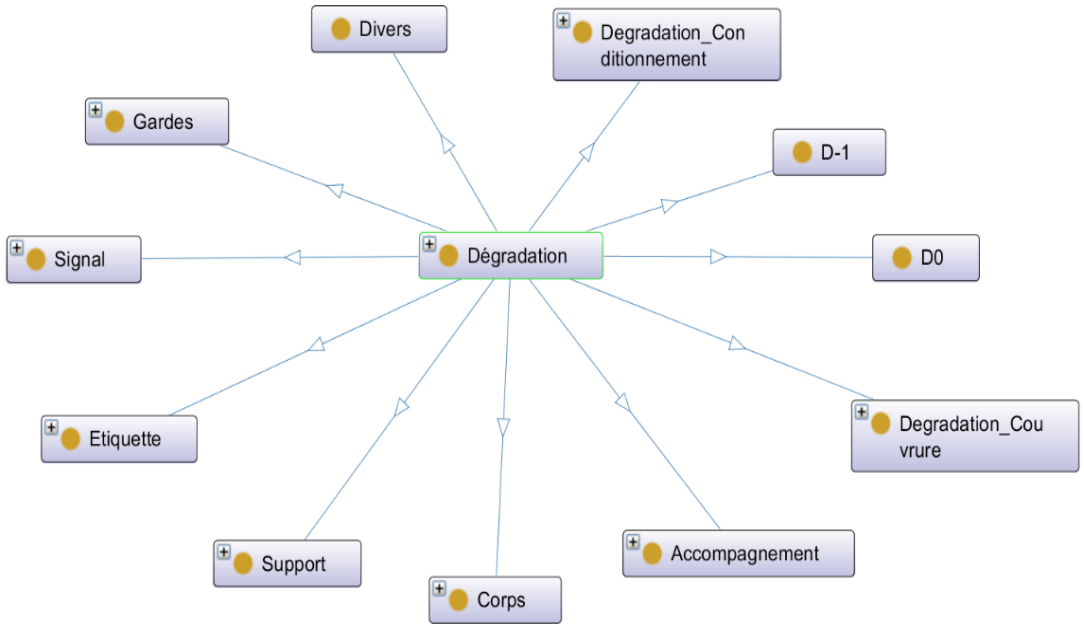


Figure A.2: Degradation types

The longest path in the ontology is between the root concept "événement" and the concept "B3 Boite à comb lage" equal to seven. Figure A.5 shows this longest path.

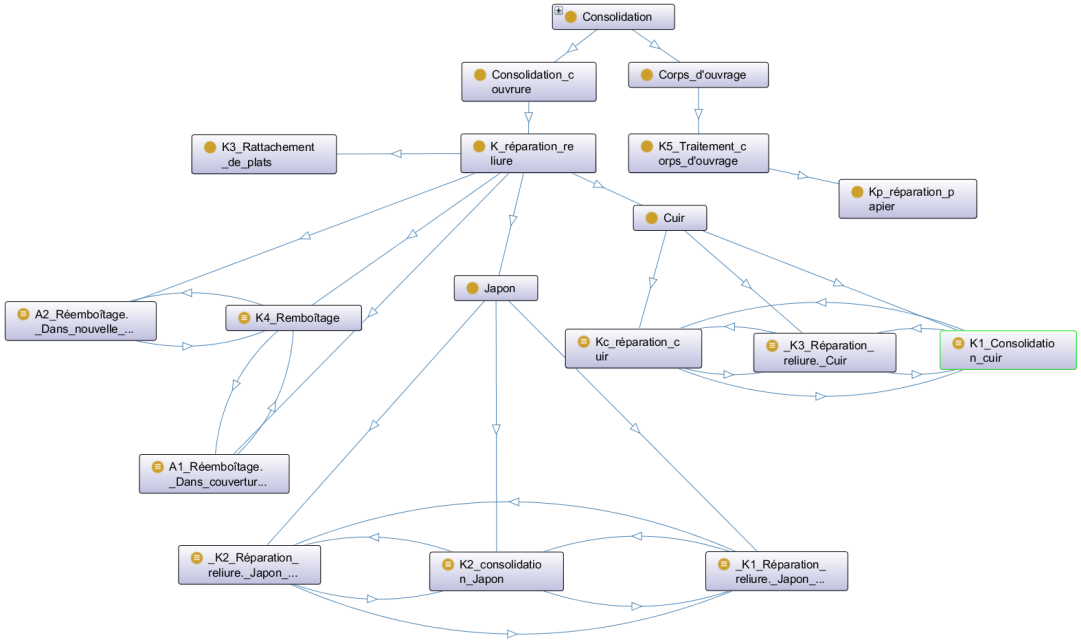


Figure A.3: Consolidation Group

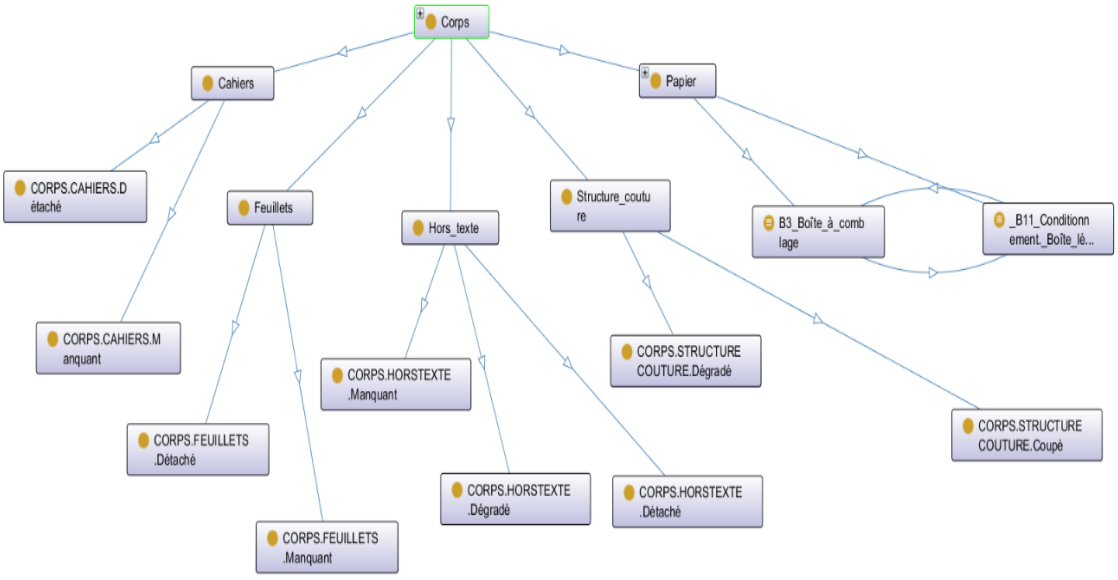


Figure A.4: Body degradation concepts

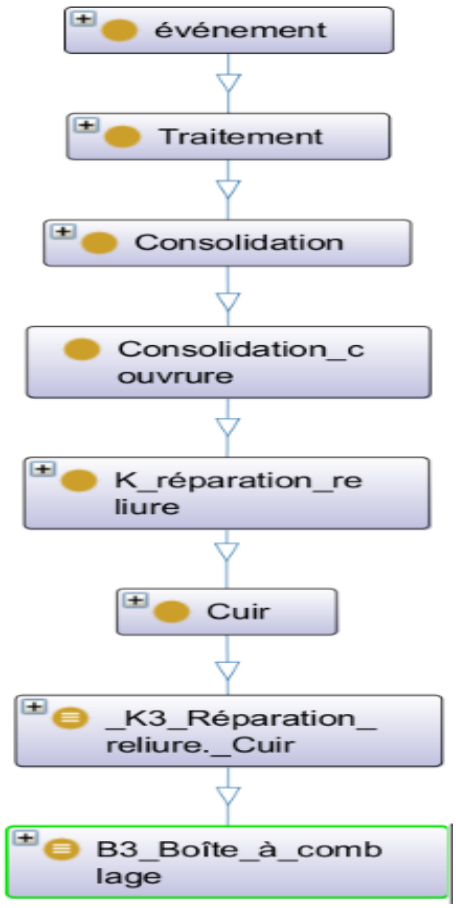


Figure A.5: Longest Path in the ontology

B - Publications and Reports

1. Alaa Zreik and Zoubida Kedad. "Matching and analysing conservation–restoration trajectories". In: Data Knowledge Engineering 139 (2022), p. 102015.
2. Alaa Zreik and Zoubida Kedad. "Matching Conservation-Restoration Trajectories: An Ontology-Based Approach." In: RCIS. 2021, pp. 230–246, Best Paper Award.
3. Customized Concept Weighting : A Neural Network Approach, Submitted Paper.
4. Analysing Conservation Data at the BnF, BDA, Poster, September 2019.
5. Towards an ontology for conservation-restoration in the libraries: CRM_{BnF}, September 2021, BnF Internal research Report.
6. CRM-BnF Dimensions, January 2020, BnF Internal Research Report.
7. Analysing the BnF databases, April 2019, BnF Internal Research Report.

Bibliography

[1] Rakesh Agrawal and Ramakrishnan Srikant. "Mining sequential patterns". In: *Proceedings of the eleventh international conference on data engineering*. IEEE. 1995, pp. 3–14.

[2] Majed A Alkhamees et al. "A semantic metric for concepts similarity in knowledge graphs". In: *Journal of Information Science* (2021), p. 01655515211020580.

[3] David Arthur and Sergei Vassilvitskii. *k-means++: The advantages of careful seeding*. Tech. rep. Stanford, 2006.

[4] Ines Bannour et al. "CRM CR-a CIDOC-CRM extension for supporting semantic interoperability in the conservation and restoration domain". In: *2018 3rd Digital Heritage International Congress (DigitalHERITAGE) held jointly with 2018 24th International Conference on Virtual Systems & Multimedia (VSMM 2018)*. IEEE. 2018, pp. 1–8.

[5] Ines Bannour et al. "CRMcr - a CIDOC-CRM extension for supporting semantic interoperability in the conservation and restoration domain". In: *Digital Heritage 2018*. San Francisco, United States, Oct. 2018. URL: https://hal.archives-ouvertes.fr/hal-01872164.

[6] Cesar Barrios et al. "Multiple model framework of adaptive extended Kalman filtering for predicting vehicle location". In: *2006 IEEE Intelligent Transportation Systems Conference*. IEEE. 2006, pp. 1053–1059.

[7] Jiang Bian et al. "A survey on trajectory clustering analysis". In: *arXiv preprint arXiv:1802.06971* (2018).

[8] Adel Bolbol et al. "Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification". In: *Computers, Environment and Urban Systems* 36.6 (2012), pp. 526–537.

[9] Robert R Bush and Frederick Mosteller. "A model for stimulus generalization and discrimination." In: *Psychological review* 58.6 (1951), p. 413.

[10] Guochen Cai, Kyungmi Lee, and Ickjai Lee. "Mining mobility patterns from geotagged photos through semantic trajectory clustering". In: *Cybernetics and Systems* 49.4 (2018), pp. 234–256.

[11] Aurelio Cappozzo et al. "Human movement analysis using stereophotogrammetry: Part 1: theoretical background". In: *Gait & posture* 21.2 (2005), pp. 186–196.

[12] Claude Cariou and Kacem Chehdi. "Unsupervised nearest neighbors clustering with application to hyperspectral images". In: *IEEE Journal of Selected Topics in Signal Processing* 9.6 (2015), pp. 1105–1116.

[13] Anil Chaturvedi, Paul E Green, and J Douglas Caroll. "K-modes clustering". In: *Journal of classification* 18.1 (2001), pp. 35–55.

[14] Lei Chen and Raymond Ng. "On the marriage of lp-norms and edit distance". In: *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. 2004, pp. 792–803.

[15] Edward Choi et al. "Multi-layer representation learning for medical concepts". In: *proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1495–1504.

[16] William W Cohen, Alexander Borgida, Haym Hirsh, et al. "Computing least common subsumers in description logics". In: *AAAI*. Vol. 1992. 1992, pp. 754–760.

[17] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297.

[18] Sina Dabiri and Kevin Heaslip. "Inferring transportation modes from GPS trajectories using a convolutional neural network". In: *Transportation research part C: emerging technologies* 86 (2018), pp. 360–371.

[19] Antonio De Nicola et al. "A Comparative Assessment of Ontology Weighting Methods in Semantic Similarity Search." In: *ICAART (2)*. 2019, pp. 506–513.

[20] Lee R Dice. "Measures of the amount of ecologic association between species". In: *Ecology* 26.3 (1945), pp. 297–302.

[21] Somayeh Dodge, Robert Weibel, and Ehsan Forootan. "Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects". In: *Computers, Environment and Urban Systems* 33.6 (2009), pp. 419–434.

[22] M Doerr, G Hiebel, and Y Kritsotaki. "The scientific observation model an extension of cidoc-crm to support scientific observation". In: *29th CRM-SIG Meeting, Heraklion, Greece*. 2013.

[23] Martin Doerr. "The CIDOC CRM - an Ontological Approach to Semantic Interoperability of Metadata". In: *Ai Magazine - AIM* 24 (Jan. 2003).

[24] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification and scene analysis*. Vol. 3. Wiley New York, 1973.

[25] Hannes Eisler and Gösta Ekman. "A mechanism of subjective similarity". In: *Nordisk psykologi* 11.1 (1959), pp. 1–10.

[26] Martin Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *kdd*. Vol. 96. 34. 1996, pp. 226–231.

[27] Mohammad Etemad, Amílcar Soares Júnior, and Stan Matwin. "Predicting transportation modes of gps trajectories using feature engineering and noise removal". In: *Canadian conference on artificial intelligence*. Springer. 2018, pp. 259–264.

[28] Vance Faber. "Clustering and the continuous k-means algorithm". In: *Los Alamos Science* 22.138144.21 (1994), p. 67.

[29] Carlos Andres Ferrero et al. "Movelets: Exploring relevant subtrajectories for robust trajectory classification". In: *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. 2018, pp. 849–856.

[30] Anna Formica et al. "Semantic Search for Enterprises Competencies Management." In: *KEOD*. 2010, pp. 183–192.

[31] Anna Formica et al. "Semantic search for matching user requests with profiled enterprises". In: *Computers in Industry* 64.3 (2013), pp. 191–202.

[32] Anna Formica et al. "Weighted ontology for semantic search". In: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer. 2008, pp. 1289–1303.

[33] Santo Fortunato and Darko Hric. "Community detection in networks: A user guide". In: *Physics reports* 659 (2016), pp. 1–44.

[34] Andre Salvaro Furtado et al. "Multidimensional similarity measuring for semantic trajectories". In: *Transactions in GIS* 20.2 (2016), pp. 280–298.

[35] Fosca Giannotti et al. "Trajectory pattern mining". In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2007, pp. 330–339.

[36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press, 2016.

[37] Robert Anthony Mills Gregson. *Psychometrics of similarity*. Academic Press, 1975.

[38] Yi Guo et al. "A survey on visual analysis of event sequence data". In: *IEEE Transactions on Visualization and Computer Graphics* (2021).

[39] Fredrik Gustafsson. "Particle filter theory and practice with positioning applications". In: *IEEE Aerospace and Electronic Systems Magazine* 25.7 (2010), pp. 53–82.

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

[41] John Hancock. "Jaccard Distance (Jaccard Index, Jaccard Similarity Coefficient)". In: Oct. 2004. ISBN: 9780471650126. DOI: 10.1002/9780471650126.dob0956.

[42] John A Hartigan and Manchek A Wong. "Algorithm AS 136: A k-means clustering algorithm". In: *Journal of the royal statistical society. series c (applied statistics)* 28.1 (1979), pp. 100–108.

[43] Jeffrey Hightower and Gaetano Borriello. "Particle filters for location estimation in ubiquitous computing: A case study". In: *International conference on ubiquitous computing*. Springer. 2004, pp. 88–106.

[44] Robert R Hoffman. "The problem of extracting the knowledge of experts from the perspective of experimental psychology". In: *AI magazine* 8.2 (1987), pp. 53–53.

[45] Shenda Hong et al. "Event2vec: Learning representations of events on temporal sequences". In: *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data*. Springer. 2017, pp. 33–47.

[46] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[47] Matthew A Jaro. "Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida". In: *Journal of the American Statistical Association* 84.406 (1989), pp. 414–420.

[48] Jay J Jiang and David W Conrath. "Semantic similarity based on corpus statistics and lexical taxonomy". In: *arXiv preprint cmp-lg/9709008* (1997).

[49] Paul E Johnson. "What kind of expert should a system be?" In: *The Journal of medicine and philosophy* 8.1 (1983), pp. 77–97.

[50] Amílcar Soares Júnior, Chiara Renso, and Stan Matwin. "Analytic: An active learning system for trajectory classification". In: *IEEE computer graphics and applications* 37.5 (2017), pp. 28–39.

[51] Amilcar Soares Junior et al. "A semi-supervised approach for the semantic segmentation of trajectories". In: *2018 19th IEEE International Conference on Mobile Data Management (MDM)*. IEEE. 2018, pp. 145–154.

[52] Antonios Karatzoglou. "Semantic Trajectories and Predicting Future Semantic Locations." PhD thesis. Karlsruhe Institute of Technology, Germany, 2019.

[53] Leonard Kaufman and Peter J Rousseeuw. "Partitioning around medoids (program pam)". In: *Finding groups in data: an introduction to cluster analysis* 344 (1990), pp. 68–125.

[54] Eamonn J Keogh and Michael J Pazzani. "Scaling up dynamic time warping for datamining applications". In: *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2000, pp. 285–289.

[55] Taehwan Kim et al. "A decision tree framework for spatiotemporal sequence prediction". In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2015, pp. 577–586.

[56] Ioannis Kontopoulos, Iraklis Varlamis, and Konstantinos Tserpes. "Uncovering hidden concepts from AIS data: A network abstraction of maritime traffic for anomaly detection". In: *International Workshop on Multiple-Aspect Analysis of Semantic Trajectories*. Springer. 2019, pp. 6–20.

[57] Claudia Leacock and Martin Chodorow. "Combining local context and WordNet similarity for word sense identification". In: *WordNet: An electronic lexical database* 49.2 (1998), pp. 265–283.

[58] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. "Trajectory clustering: a partition-and-group framework". In: *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. 2007, pp. 593–604.

[59] Jae-Gil Lee et al. "TraClass: trajectory classification using hierarchical region-based and trajectory-based clustering". In: *Proceedings of the VLDB Endowment* 1.1 (2008), pp. 1081–1094.

[60] Wang-Chien Lee and John Krumm. "Trajectory preprocessing". In: *Computing with spatial trajectories*. Springer, 2011, pp. 3–33.

[61] Yuhua Li, Zuhair A Bandar, and David McLean. "An approach for measuring semantic similarity between words using multiple information sources". In: *IEEE Transactions on knowledge and data engineering* 15.4 (2003), pp. 871–882.

[62] Zhenhui Li et al. "Movemine: Mining moving object data for discovery of animal movement patterns". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.4 (2011), pp. 1–32.

[63] Leo Liberti et al. "Euclidean distance geometry and applications". In: *SIAM review* 56.1 (2014), pp. 3–69.

[64] Bin Lin and Jianwen Su. "Shapes based trajectory queries for moving objects". In: *Proceedings of the 13th annual ACM international workshop on Geographic information systems*. 2005, pp. 21–30.

[65] Dekang Lin et al. "An information-theoretic definition of similarity." In: *Icml*. Vol. 98. 1998. 1998, pp. 296–304.

[66] Caihong Liu and Chonghui Guo. "STCCD: Semantic trajectory clustering based on community detection in networks". In: *Expert Systems with Applications* 162 (2020), p. 113689.

[67] Chuanren Liu et al. "Temporal skeletonization on sequential data: patterns, categorization, and visualization". In: *IEEE Transactions on Knowledge and Data Engineering* 28.1 (2015), pp. 211–223.

[68] Siyuan Liu, Shuhui Wang, and Qiang Qu. "Trajectory Mining". In: *Encyclopedia of GIS*. Ed. by Shashi Shekhar, Hui Xiong, and Xun Zhou. Cham: Springer International Publishing, 2017, pp. 2310–2313. ISBN: 978-3-319-17885-1. DOI: 10.1007/978-3-319-17885-1_1576. URL: https://doi.org/10.1007/978-3-319-17885-1_1576.

[69] Sana Malik et al. "High-volume hypothesis testing: Systematic exploration of event sequence comparisons". In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 6.1 (2016), pp. 1–23.

[70] Lucas May Petry et al. "MARC: a robust method for multiple-aspect trajectory classification via space, time, and semantic embeddings". In: *International Journal of Geographical Information Science* 34.7 (2020), pp. 1428–1450.

[71] Leland McInnes, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[72] Ronaldo dos Santos Mello et al. "MASTER: A multiple aspect view on trajectories". In: *Transactions in GIS* 23.4 (2019), pp. 805–822.

[73] F.P. Miller, A.F. Vandome, and J. McBrewster. *Damerau-Levenshtein Distance*. Alphascript Publishing, 2010. ISBN: 9786130889272. URL: https://books.google.fr/books?id=kyPObwAACAAJ.

[74] F.P. Miller, A.F. Vandome, and J. McBrewster. *Levenshtein Distance*. VDM Publishing, 2009. ISBN: 9786130216900. URL: https://books.google.fr/books?id=TTzhQgAACAAJ.

[75] Frederic P. Miller, Agnes F. Vandome, and John McBrewster. *Levenshtein Distance: Information Theory, Computer Science, String (Computer Science), String Metric, Damerau?Levenshtein Distance, Spell Checker, Hamming Distance*. Alpha Press, 2009. ISBN: 6130216904.

[76] Efthymia Moraitou, John Aliprantis, and George Caridakis. "Semantic Preventive Conservation of Cultural Heritage Collections." In: *SW4CH@ ESWC*. 2018.

[77] Clément Moreau et al. "A contextual edit distance for semantic trajectories". In: *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. 2020, pp. 635–637.

[78] Clement Moreau et al. "Clustering sequences of multi-dimensional sets of semantic elements". In: *Proceedings of the 36th Annual ACM Symposium on Applied Computing*. 2021, pp. 384–391.

[79] Bruno Moreno et al. "Weka-SAT: A hierarchical context-based inference engine to enrich trajectories with semantics". In: *Canadian Conference on Artificial Intelligence*. Springer. 2014, pp. 333–338.

[80] Xing Mu et al. "MOOCad: Visual Analysis of Anomalous Learning Activities in Massive Open Online Courses." In: *EuroVis (Short Papers)*. 2019, pp. 91–95.

[81] Meinard Müller. "Dynamic time warping". In: *Information retrieval for music and motion* (2007), pp. 69–84.

[82] Fionn Murtagh and Pedro Contreras. "Algorithms for hierarchical clustering: an overview". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2.1 (2012), pp. 86–97.

[83] Phong H Nguyen et al. "Vasabi: Hierarchical user profiles for interactive visual user behaviour analytics". In: *IEEE transactions on visualization and computer graphics* 26.1 (2019), pp. 77–86.

[84] Dominic Oldman and CRM Labs. "The CIDOC Conceptual Reference Model (CIDOC-CRM): PRIMER". In: (2014).

[85] Behrooz Omidvar-Tehrani, Sihem Amer-Yahia, and Laks VS Lakshmanan. "Cohort analytics: efficiency and applicability". In: *The VLDB Journal* 29.6 (2020), pp. 1527–1550.

[86] Behrooz Omidvar-Tehrani, Sihem Amer-Yahia, and Laks VS Lakshmanan. "Cohort representation and exploration". In: *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2018, pp. 169–178.

[87] M. S. Paterson and Vlado Dancik. *Longest Common Subsequences*. Tech. rep. GBR, 1994.

[88] Lokukaluge P. Perera, Paulo Oliveira, and C. Guedes Soares. "Maritime Traffic Monitoring Based on Vessel Detection, Tracking, State Estimation, and Trajectory Prediction". In: *IEEE Transactions on Intelligent Transportation Systems* 13.3 (2012), pp. 1188–1200. DOI: 10.1109/TITS.2012.2187282.

[89] Lucas May Petry et al. "Towards semantic-aware multiple-aspect trajectory similarity measuring". In: *Transactions in GIS* 23.5 (2019), pp. 960–975.

[90] Guande Qi et al. "How long a passenger waits for a vacant taxi–large-scale taxi trace mining for smart cities". In: *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*. IEEE. 2013, pp. 1029–1036.

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

[92] Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

[93] Sayan Ranu et al. "Indexing and matching trajectories under inconsistent sampling rates". In: *2015 IEEE 31st International conference on data engineering*. IEEE. 2015, pp. 999–1010.

[94] John W Ratcliff and David E Metzener. "Pattern-matching-the gestalt approach". In: *Dr Dobbs Journal* 13.7 (1988), p. 46.

[95] Philip Resnik. "Using information content to evaluate semantic similarity in a taxonomy". In: *arXiv preprint cmp-lg/9511007* (1995).

[96] Patrick Riehmann, Manfred Hanfler, and Bernd Froehlich. "Interactive sankey diagrams". In: *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* IEEE. 2005, pp. 233–240.

[97] Jose Antonio MR Rocha et al. "DB-SMoT: A direction-based spatio-temporal clustering method". In: *2010 5th IEEE international conference intelligent systems*. IEEE. 2010, pp. 114–119.

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

[99] J. Russell and R. Cohn. *Hamming Distance*. Book on Demand, 2012. ISBN: 9785512146972. URL: https://books.google.fr/books?id=m9X1MgEACAAJ.

[100] David Sánchez, Montserrat Batet, and David Isern. "Ontology-based information content computation". In: *Knowledge-based systems* 24.2 (2011), pp. 297–303.

[101] Nuno Seco, Tony Veale, and Jer Hayes. "An intrinsic information content metric for semantic similarity in WordNet". In: *Ecai*. Vol. 16. 2004, p. 1089.

[102] Lokesh K Sharma et al. "Nearest neighbour classification for trajectory data". In: *International Conference on Advances in Information and Communication Technologies*. Springer. 2010, pp. 180–185.

[103] Camila Leite da Silva, Lucas May Petry, and Vania Bogorny. "A survey and comparison of trajectory classification methods". In: *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE. 2019, pp. 788–793.

[104] Atish P Sinha and Huimin Zhao. "Incorporating domain knowledge into data mining classifiers: An application in indirect lending". In: *Decision Support Systems* 46.1 (2008), pp. 287–299.

[105] Ali Soleymani et al. "Integrating cross-scale analysis in the spatial and temporal domains for classification of behavioral movement". In: *Journal of Spatial Information Science* 8 (2014), pp. 1–25.

[106] Libo Song et al. "Evaluating location predictors with extensive Wi-Fi mobility data". In: *Ieee Infocom 2004*. Vol. 2. IEEE. 2004, pp. 1414–1424.

[107] Rudi Studer, V Richard Benjamins, and Dieter Fensel. "Knowledge engineering: principles and methods". In: *Data & knowledge engineering* 25.1-2 (1998), pp. 161–197.

[108] Han Su et al. "A survey of trajectory distance measures and performance evaluation". In: *The VLDB Journal* 29.1 (2020), pp. 3–32.

[109] Han Su et al. "Calibrating trajectory data for spatio-temporal similarity analysis". In: *The VLDB Journal* 24.1 (2015), pp. 93–116.

[110] Daniel Svozil, Vladimír Kvasnicka, and Jirí Pospichal. "Introduction to multi-layer feed-forward neural networks". In: *Chemometrics and Intelligent Laboratory Systems* 39.1 (1997), pp. 43–62. ISSN: 0169-7439.

[111] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. "Data mining introduction". In: *People's Posts and Telecommunications Publishing House, Beijing* (2006).

[112] Amos Tversky. "Features of similarity." In: *Psychological review* 84.4 (1977), p. 327.

[113] MK Vijaymeena and K Kavitha. "A survey on similarity measures in text mining". In: *Machine Learning and Applications: An International Journal* 3.2 (2016), pp. 19–28.

[114] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. "Discovering similar multidimensional trajectories". In: *Proceedings 18th international conference on data engineering*. IEEE. 2002, pp. 673–684.

[115] Katerina Vrotsou and Camilla Forsell. "A qualitative study of similarity measures in event-based data". In: *Symposium on Human Interface*. Springer. 2011, pp. 170–179.

[116] Hongjian Wang, Zhenhui Li, and Wang-Chien Lee. "PGT: Measuring mobility relationship using personal, global and temporal factors". In: *2014 IEEE International Conference on Data Mining*. IEEE. 2014, pp. 570–579.

[117] William E Winkler. "The state of record linkage and current research problems". In: *Statistical Research Division, US Census Bureau*. Citeseer. 1999.

[118] Ian H Witten et al. "Practical machine learning tools and techniques". In: *Data Mining*. Vol. 2. 4. 2005.

[119] Krist Wongsuphasawat et al. "LifeFlow: visualizing an overview of event sequences". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2011, pp. 1747–1756.

[120] Ruizhi Wu et al. "Location prediction on trajectory data: A review". In: *Big data mining and analytics* 1.2 (2018), pp. 108–127.

[121] Zhibiao Wu and Martha Palmer. "Verb semantics and lexical selection". In: *arXiv preprint cmp-lg/9406033* (1994).

[122] Zhibin Xiao et al. "Identifying different transportation modes from trajectory data using tree-based ensemble classifiers". In: *ISPRS International Journal of Geo-Information* 6.2 (2017), p. 57.

[123] Di Yao et al. "Serm: A recurrent model for next location prediction in semantic trajectories". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017, pp. 2411–2414.

[124] Ning Ye et al. "Vehicle trajectory prediction based on Hidden Markov Model". In: (2016).

[125] Huabei Yin and Ouri Wolfson. "A weight-based map matching method in moving objects databases". In: *Proceedings. 16th International Conference on Scientific and Statistical Database Management, 2004.* IEEE. 2004, pp. 437–438.

[126] Daqing Zhang et al. "Understanding taxi service strategies from taxi GPS traces". In: *IEEE Transactions on Intelligent Transportation Systems* 16.1 (2014), pp. 123–135.

[127] Fuzheng Zhang et al. "Sensing the pulse of urban refueling behavior: A perspective from taxi mobility". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 6.3 (2015), pp. 1–23.

[128] Yu Zheng. "Trajectory data mining: an overview". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 6.3 (2015), pp. 1–41.

[129] Yu Zheng and Xiaofang Zhou. *Computing with spatial trajectories*. Springer Science & Business Media, 2011.

[130] Yu Zheng et al. "Understanding mobility based on GPS data". In: *Proceedings of the 10th international conference on Ubiquitous computing*. 2008, pp. 312–321.

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

[132] Ganggao Zhu and Carlos A Iglesias. "Computing semantic similarity of concepts in knowledge graphs". In: *IEEE Transactions on Knowledge and Data Engineering* 29.1 (2016), pp. 72–85.

[133] Alaa Zreik and Zoubida Kedad. "Matching and analysing conservation–restoration trajectories". In: *Data & Knowledge Engineering* 139 (2022), p. 102015.

[134] Alaa Zreik and Zoubida Kedad. "Matching Conservation-Restoration Trajectories: An Ontology-Based Approach." In: *RCIS*. 2021, pp. 230–246.