ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Automatic ontology extraction in the educational context

Doctoral Thesis

Prior to obtaining the degree:

DOCTOR IN APPLIED COMPUTER SCIENCE

Presented By:
Angel Fiallos Ordoñez

GUAYAQUIL – ECUADOR

YEAR: 2020

Promotor: Dr. Katherine Chiluiza

Co-Promotor: Dr. Xavier Ochoa

Dedication

To my mother Martha,

who taught me since childhood the research world.

She lives in my memory and in her legacy to science.

To my wife Erika and my children: Mikael and Guillermo

for their love, sacrifice and patience on this journey.

Acknowledgement

I want to express my gratitude to my tutors,

Dr. Katherine Chiluiza and Dr. Xavier Ochoa,

for their wise guidance during my research work.

Content Table

Figures

Tables

1 Introduction

The vast majority of the world's text data has been created in the last years, originating from sources like the World Wide Web, social media, news, forums, digital repositories, mails, medical records, and databases. Much of these data inherently lacks coherent structure. There is far too much data for human users to process and categorize manually. Therefore, exists an enormous need to design methods that can effectively process a wide variety of text applications in different knowledge areas [1].

The educational context is no exception; large volumes of information are available in several ways. This enormous amount of text data is continuously generated by different types of users in informal and formal settings such as tutorials, textbooks, forums, blogs, papers, and slides. These data appear in different formats and structures that need to be understood and evaluated to be further used. However, to understand data on a large scale, it is needed to analyze semantics and have a higher degree of reasoning and understanding of language [2]. These objectives cannot be done with traditional text processing techniques but using more challenging techniques such as Natural Language Processing and Machine Learning.

Natural Language Processing (NLP) is an artificial intelligence field focused on allowing computers to have the ability to understand, process, and analyze the human language that includes the production of an infinite amount of data [3]. NLP includes algorithms that collect human-produced text as input and algorithms that produce natural-looking text as outputs. At the same time, Machine Learning (ML) is another branch of artificial intelligence based on the idea that computational systems can learn from data, make decisions and identify patterns with minimal human intervention [4]. These areas of artificial intelligence are the most promising avenue for providing tools, methods, and algorithms that extract meaningful information and insights from large volumes of data from multiple sources and languages in various scientific fields [5].

The data can be found in some formats. Structured data is easily organizable and follows a rigid format. This form of data can be found in databases and collections of text documents. On the other hand, unstructured data comprises most data found in all organizations. It's often found in highly unstructured environments like web pages, chats and other web-centric platforms.

Beyond structured and unstructured data, there is a third schema, which basically is a mix between them. The type of data defined as semi-structured data has some defining or consistent characteristics but does not form a structure as rigid as it is expected with a relational database. Both unstructured and semi-structured data require text preprocessing techniques to convert their formatting into a structured, multi-dimensional representation.

In education, multiple learning contents are presented as structured and semi-structured data. Some of these educational materials, created by experts, can be considered useful by teachers who are looking for resources for those who need to learn about a course. The extensive collection of online learning materials helps them not to have to create the content themselves because they can use the best of what is available [6]. Tutors regularly take advantage of textbooks with hierarchical structures (table of contents, links, sections, references) that explain the semantic relationship between different topics and subtopics, as a guide in the curricula-design process that cover content knowledge [7] to be transmitted to students.

Social networks are a growing source of unstructured data, with information continuously generated by users in informal environments. Commonly, the information is highly dynamic and involves interaction among several participants in a wide range of topics. In the educational context, it is observed that students share information about events, activities, opinions, and experiences at the university on social networks, so that the networks concerning educational topics are populated with unstructured data.

Consequently, many educational resources available for the same domain do not offer quality content. Both students and teachers can be lost in this large volume of data, having to go through a great deal of non-relevant content before reaching the suitable materials. Thus, it becomes necessary to know the most relevant content to facilitate a better understanding of the domain and for the reuse, transfer, and creation of knowledge [8].. Ontologies are the best way to share a common understanding of information structure among people or software agents [9] [10]. They can be useful tools for visual representation and conceptual structuring of domain knowledge because they function as a cognitive tool that can facilitate both communication and teacher/student interaction.

Ontologies can be defined as abstract models focused on capturing domain knowledge in a general way and providing a commonly accepted understanding of a domain that can be reused and shared between applications and groups [11]. However, the generation of educational ontologies is a complex and high-cost process, usually carried out by semantics experts and domain knowledge experts. Given these factors, the generation of automatic educational ontologies, becomes key to solving the ontology-building cost barrier. Therefore, these ontologies can be used by both teachers and systems to recommend materials, provide feedback to students or to monitor discussion forums. Considering also other potential uses of ontologies in the field of education, it is possible to enumerate the following contributions of this doctoral thesis, through two approaches followed in this work:

A first approach to (semi-)automatically generate educational ontologies extracted from existing learning materials such as digital books or web content. Which is possible through a series of natural language processing steps and, the semi-structured information present in existing content is transformed into a concept-graph. This work also evaluates the proposed approach by applying it to learning content for different courses and measuring the quality of the extracted ontologies against manually generated ones. Then, an assistant system that uses generated educational ontologies to support teachers in the unit design of a course was built to prove that course-based ontologies can be readily used to provide domain information to other educational and learning analytics tools. In order to ensure the quality of the curricula generated, the use of functionalities implemented by Natural Language Processing and Machine Learning techniques has been considered, so educators can validate if the content of their documents was covered by curriculum standards, such as the Computer Science curricula, CS2013.

The second approach is related to ontologies generation from posts related to education topics in social networks through data mining, machine learning, deep learning, and natural language processing techniques. This work collects, analyses, and processes publications from students and other users on social networks. From these unstructured data collections, domain ontologies are constructed from relevant concepts identified in this content related to user interests, course experiences, university life, and other academic aspects.

1.1 Justification

During the last years, visual knowledge representation has become one of the critical considerations in knowledge engineering methodology, and ontologies are considered as the most universal and shareable forms of such representation and modeling. For this reason, the definition and visualization of conceptual models are strongly associated with ontology design and development [12]. These ontologies, which form a conceptual skeleton of the modeled domain, might serve various purposes such as better understanding and knowledge creation.

Ontologies are also widely and effectively used in education, and many learning ontologies have been developed for several disciplines [13] [14] and other related approaches such as collaborative learning [15] and adaptive educational systems [16] and intelligent curriculum design [17]. A new research field related to educational research is learning analytics. It focuses on the measurement, collection, analysis, and reporting of data about learners and their contexts for understanding and optimizing learning and the environments in which it occurs [18]. In addition to educational research, this area also converges educational technologies analysis techniques from Artificial Intelligence, Data Science and Human-Centered Design.

Another clear usage of ontologies in the education field is an Intelligent Curriculum, which can be defined as the representation of the domain knowledge usually taught in a course in a way that is amenable to be understood and processed by a computational system [17].

However, in the educational field, automatically extracting ontologies from existing semi-structured text applied to education, and particularly for the creation of intelligent curriculum, is a field with scarce research studies. At the same time, this field is in the need of many application possibilities within the fields of learning analytics, where the most successful approaches have been proposed by Guerra at al. [19] and Taker [20] to create links between textbooks chapters and subchapters. These links formed networks of connected textbooks sections that were used to recommend those learning materials. As successful as it is, these attempts fail short of creating a fully functional ontology that goes beyond linking part of educational materials and represents the domain knowledge of a course.

The proposed approach focuses first on the generation of educational ontologies that could perform the concept of intelligent curricula using digital books and then on the improvement of the quality of the curricula of Computer Science courses. This can be achieved by evaluating its knowledge domain concerning those established in curricular standards chosen for each discipline. These approaches can contribute to the professional student training through the usage of the system for curricular design, which leads to the enhancement of decision making by educators and academic authorities; and, in the quality improvement of the curricula content, created by teachers.

Moreover, developing a domain knowledge ontology repository is an expensive, time-consuming task, and static knowledge ontologies are difficult to maintain. For this reason, this approach also contributes to resolve the ontology-building cost barrier that limits the use of the Intelligent Curriculum for educational and Learning Analytics applications. Besides, it allows us to automatically obtain adaptive and flexible domain ontologies, easy to maintain by users. In this way, attributes such as book sources, the topics hierarchy levels, the ontology size, the depth of the topics, and even the language are easy to configure.

To conclude, this research tries to demonstrate that once the concept hierarchy issue is resolved, the approach center on the generation of course-ontologies through digital books can be adjusted to achieve ontologies creation processes from unstructured text. The documents used in this part of the research were social network posts that referred to academic topics related to educational institutions. The resulting ontologies provide a comprehensive visualization that fosters the understanding of the extracted knowledge content and could give tutors and educational authorities the possibility to obtain valuable and useful information and discover insights to improve academic issues and student life.

1.2 Research questions and objectives

This doctoral thesis aims to answer the following research questions:

RQ1: In a semi-automatic way and with the support of Machine Learning and Natural Language Processing techniques, is it possible to generate domain ontologies from the following sources?

a) Semi-structured data such as texts of books, tutorials and courses from different academic disciplines
b) Unstructured data, such as social networks publications related to education topics.

RQ2: Can the automatically generated domain course ontologies be useful in the fields of educational and learning analytics?

Figure 1-1 shows the connection between the research questions and the research papers associated with each of them.

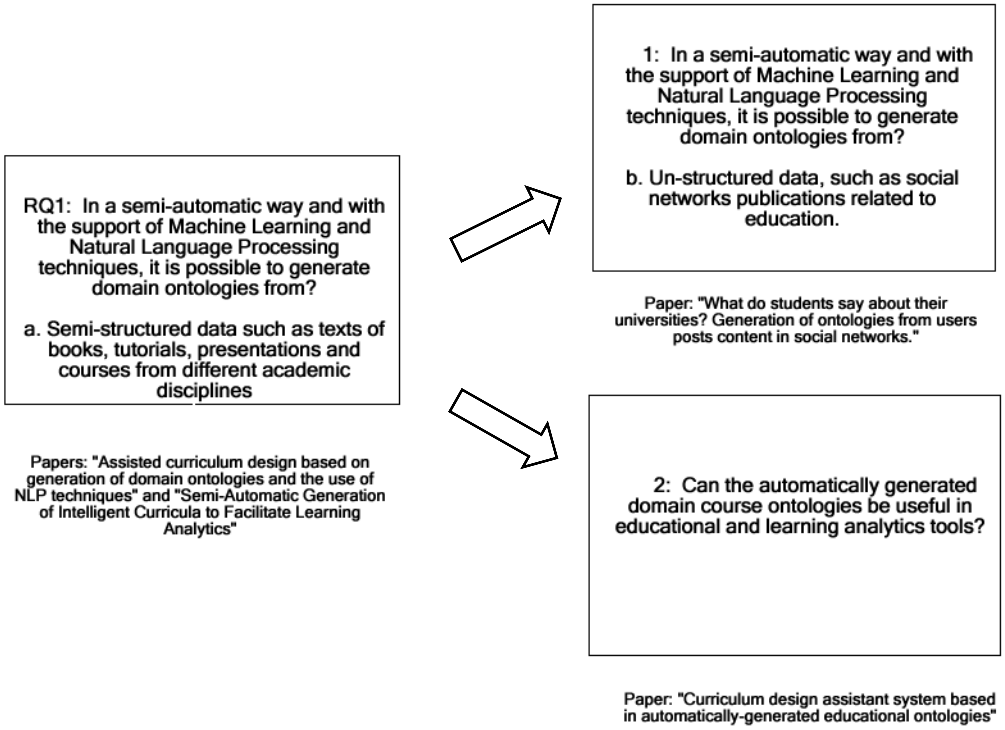

RQ1: In a semi-automatic way and with the support of Machine Learning and Natural Language Processing techniques, it is possible to generate domain ontologies from?
a. Semi-structured data such as texts of books, tutorials, presentations and courses from different academic disciplines
Papers: "Assisted curriculum design based on generation of domain ontologies and the use of NLP techniques" and "Semi-Automatic Generation of Intelligent Curricula to Facilitate Learning Analytics"
1: In a semi-automatic way and with the support of Machine Learning and Natural Language Processing techniques, it is possible to generate domain ontologies from?
b. Un-structured data, such as social networks publications related to education.
Paper: "What do students say about their universities? Generation of ontologies from users posts content in social networks."
2: Can the automatically generated domain course ontologies be useful in educational and learning analytics tools?
Paper: "Curriculum design assistant system based in automatically-generated educational ontologies"

Figure 1-1. Connections between the research questions and papers

The answer to research question 1a is presented on the research works described in the articles: "Assisted curriculum design based on the generation of domain ontologies and the use of NLP techniques" and "Semi-Automatic Generation of Intelligent Curricula to Facilitate Learning Analytics. " The former was published in IEEE Second Ecuador Technical Chapters Meeting (ETCM) proceedings and the latter was nominated as best paper in the Short Paper category at the 9th International Learning Analytics and Knowledge (LAK) Conference in 2019. The aim of both works, was the automatic generation of ontologies from semi-structured digital books and educational resources, through an original combination of NLP and machine learning methods.

The answer to research question 1bis found on the work described in the article: "What do students say about their universities? Generation of ontologies from user posts content in social networks", published in edition 23 of the Iberian Journal of Information Systems and Technologies (RISTI). This paper presents an approach for the ontologies processing from unstructured text, which is derived from the framework for ontologies automatic generation from digital books.

Finally, the second research question is answered in the paper: "Curriculum design assistant system based in automatically-generated educational ontologies", sent to the IEEE-RITA (Revista Iberoamericana de Tecnologías de Aprendizaje) Journal. This work seeks to verify the usefulness of educational ontologies, through the construction and evaluation of an assistant system for curriculum design available for teachers and tutors.

1.3 Objectives

This research thesis includes the following objectives

1. Generate an automatic way, course-based ontologies from texts and digital publications, to model the knowledge of specific Computer Science courses.
2. Apply the resulting educational ontologies, in a useful assistant system, to help the educators to design curricular academic programs.
3. Assure quality contents and references published in the curricula, through the comparison between the curricula registered in the system by teachers and curricular course standards, such as CS2013 of ACM/IEEE.
4. Build domain ontologies related to specific topics from unstructured data such as social network publications.

1.4 Document Organization

Chapter 1 presents the organization of this doctoral thesis; Chapter 2 contains the theoretical foundations of the proposed approaches, while chapter 3 deals with a summary of previous research on ontologies for education and its applications in the field of Learning Analytics.

Chapter 4 presents and approach for the automatic generation of educational ontologies from digital textbooks, followed by experiments and evaluations with teachers. Chapter 5 presents the design and implementation of a functional assistant system that uses automatically generated ontologies from textbooks to support instructors to design or re-design the curriculum of their courses. Additionally, it includes the evaluation of the perception of teachers about the usefulness, easiness, engagement, and other aspects related to the course-domain ontologies.

Chapter 6 deals with a derived approach used for the generation of ontologies from social media publications related to education, followed by experiments and evaluations. Finally, chapter 7 includes the general conclusions of this work.

2 Theoretical framework

This chapter will introduce the key concepts that have been used in the development of this work. First, ontology definitions and its usages in different fields will be described; second, selected tools for developing ontologies will be presented. Next, NLP techniques will be explained; fourth, document classification algorithms will be described, and finally, the chapter ends with an explanation about similarity measures and topic modeling techniques.

2.1 Ontologies

Ontology is a declarative representation of a precise domain specification, including the glossary of the domain terms and the logical expressions describing the meanings and the relationships of these terms; thus, it allows structured sharing of knowledge related to the domain [10]. The concepts and relationships are universal for a particular class of objects in a subject area. In contrast, the relationships between concepts in ontologies can be of different types e.g., "is," "has part," "has a property of."

Ontologies can be considered the explicit and abstract model representation of finite sets of terms and concepts already defined, including knowledge management, artificial intelligence, knowledge engineering, systems engineering, and intelligent information integration [21]. In information sciences, ontologies refer to an engineering artifact, composed by a vocabulary used to describe a particular reality, plus a set of explicit assumptions concerning the intended meaning of the vocabulary words [22]. Commonly, these ontologies provide us with the tools (conceptual modeling grammar) to validate the schemas and conceptual models we use against reality.

Ontologies aim to capture the domain knowledge and provide a commonly agreed understanding of a domain that may be reused and shared across applications and groups [11]. The creation of ontologies and explanation of the processes can indicate the extent and nature of the knowledge and understanding. Knowledge entities that represent the static knowledge of the domain are stored in hierarchical order in the knowledge repository and can be reutilized by more users. At the same time, those knowledge entities could be reused in descriptions of the properties or a methodological approach applied in the context of another related knowledge entity [13].

Meta-ontology provides a more general description dealing with higher-level abstractions like mind maps [23] and concept maps [24]. Based on Kudryavtsev and Gavrilova's work [25], figure 2-1 illustrates different ontology classifications in the form of a mind map.

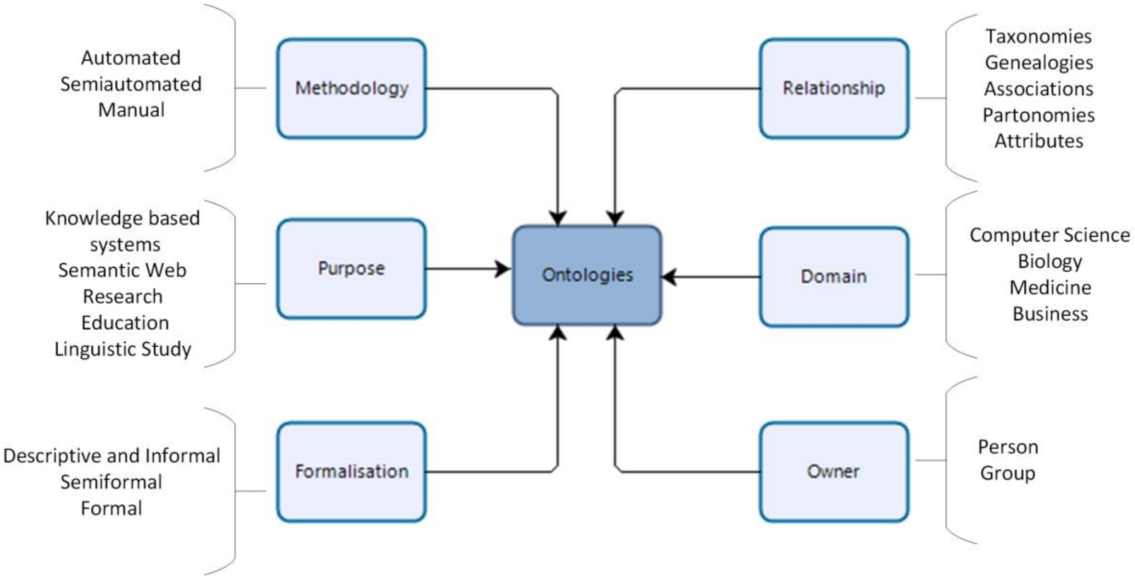


Figure 2-1. Ontology classifications summary in a mind-map

This representation may be identified as the knowledge map. Such maps are graphical tools for organizing and describing knowledge. Knowledge maps are now extensively used for visualizing ontologies at the design stage, while ontology editors promote the development stage. People recall more central ideas when they learn from a concept map than when they learn from text, and those with low verbal capacity or low prior knowledge often benefit the most. It seems that knowledge maps reduce cognitive load. The use of knowledge maps also looks to amplify the benefits associated with scripted cooperation [26]. Learning from maps and communication via maps are enhanced by active processing strategies such as summarization or annotation and by designing maps according to Gestalt principles of organization [27].

Ontologies are formal, explicit specification of a shared conceptualization and give a shared vocabulary, which can be applied to model a domain that is, the type of objects, and concepts that exist, and their properties and relations [12]. They can be represented as a graph where every node of this graph represents a domain concept. Nodes also have other information attached, like attributes, relationships, and rules (or axioms). A relationship is a link that points from one concept to another concept. It expresses how the two concepts relate to each other.

So, we try to envisage the knowledge of domain L and to find a group of terms representing relevant concepts in L. The result of this process is a list of terms.

$$L = (C1, C2, C3)$$

L is a conceptualization of *K* (Knowledge), and *C1, C2, and C3* are ontology concepts.

Although it is possible to identify a wide variety of ontologies that model context under different approaches, it is not yet possible to speak of a consensus model that can be widely used for context modeling in multiple applications. In addition to the existing ontological models, only a few are available to be studied in detail and especially to be reused. Particularly, about ontology-based context modeling, the following ontologies can be highlighted:

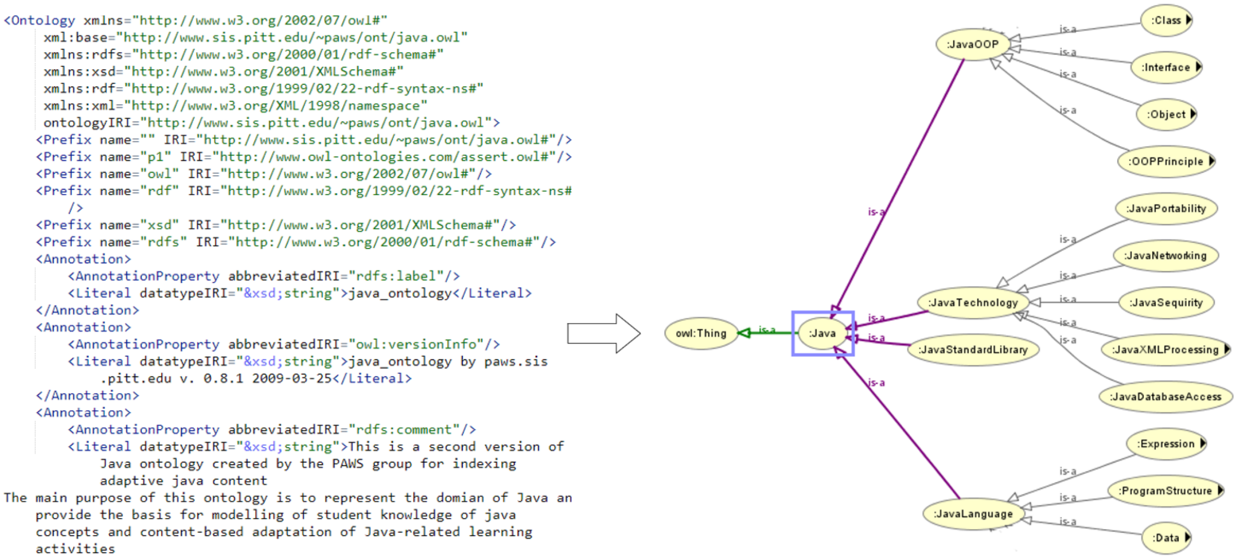
- AIISO - Academic Institution Internal Structure Ontology [28], implements classes and properties to determine the internal organizational structure of an academic institute. AIISO is designed to work in cooperation with Participation Schema FOAF [29].
- Bologna Ontology - This ontology [30] originates from a lexicon defining terms related to the Bologna Reform. It focuses mainly on study tracking, student mobility, and applications for end-users at universities such as a faceted search and browsing system for course information.
- SUMO - Suggested Upper Merged Ontology [31], listed as a base ontology for a variety of information processing computer systems. It was originally oriented to meta-level concepts (general entities that do not belong to a specific problem domain) and would naturally lead to an encyclopedia's categorization scheme. It has now expanded considerably to include a mid-level ontology and dozens of domain ontologies.
- DOLCE - Descriptive Ontology for Linguistic and Cognitive Engineering [32]. It has a clear cognitive bias, as it aims to capture the ontological categories underlying natural language and human common sense. The categories it introduces are thought of as cognitive artifacts, which ultimately depend on human perception, cultural footprints, and social conventions. In this sense, the ontologies categories

claim to be only descriptive notions that help to make explicit the conceptualizations already formed.

- COBRA-ONT - Ontology for context-aware pervasive computing environments [33], is a collection of ontologies expressed in OWL to describe context information in intelligent spaces. It is categorized into four related topics: i) ontology on physical sites, ii) ontology on agents, iii) ontology on the context of agent location, and iv) ontology to describe the activities of agents.
- CONON – Ontology-based Context Modeling and Reasoning using OWL [34] consists of a higher context ontology that captures the general concepts of a basic context, extends new ontologies, and adds the concepts of a specific domain of ubiquitous computing. This ontology defines a vocabulary based on four contextual entities: people, location, computational entities, and activities.
- MOD - stands for Metadata for Ontology Description and publication [35]. MOD proposes a set of metadata elements that can be used to describe the ontologies, for instance, in ontology libraries and repositories. Like any other resource ontologies also need to be described.
- Rei Policy Ontology [36]. This ontology specifies a generic vocabulary to model the context related to the concepts: people, security and privacy policies, actions, agents, beliefs-desires-intentions, time, space, and events.
- Soupa ontology [37] was a collaborative effort to build a generic context ontology or ubiquitous systems. The design of this ontology is driven by uses cases and relies on FOAF (Friend-Of-A-Friend Ontology) [29], DAML Time [38], OpenCyc Spatial Ontologies [39], RCC (Regional Connection Calculus) [40], MoGATU and BDI Ontology, [41]

2.2 Tools for developing ontologies

Multiple tools for the development of ontologies have been created, and multiple review works have been published comparing different aspects of these systems [42]. The most popular line of tools for ontology is the Protégé family of products developed at the Stanford University School of Medicine [43]. The Web Ontology Language (OWL) is designed to be used in applications that need to process information content instead of only representing information for humans. OWL facilitates a better mechanism for interpreting Web content than the mechanisms supported by XML, RDF, and RDF scheme (RDF-S) by providing additional vocabulary and formal semantics. An example of an OWL ontology for Java language learning is shown in figure 2-2.



```
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.sis.pitt.edu/~paws/ont/java.owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     ontologyIRI="http://www.sis.pitt.edu/~paws/ont/java.owl">
    <Prefix name="" IRI="http://www.sis.pitt.edu/~paws/ont/java.owl#"/>
    <Prefix name="p1" IRI="http://www.owl-ontologies.com/assert.owl#"/>
    <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
    <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#
        />
    <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
    <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
    <Annotation>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <Literal datatypeIRI="&xsd;string">java_ontology</Literal>
    </Annotation>
    <Annotation>
        <AnnotationProperty abbreviatedIRI="owl:versionInfo"/>
        <Literal datatypeIRI="&xsd;string">java_ontology by paws.sis
            .pitt.edu v. 0.8.1 2009-03-25</Literal>
    </Annotation>
    <Annotation>
        <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
        <Literal datatypeIRI="&xsd;string">This is a second version of
            Java ontology created by the PAWS group for indexing
            adaptive java content
The main purpose of this ontology is to represent the domian of Java an
    provide the basis for modelling of student knowledge of java
    concepts and content-based adaptation of Java-related learning
    activities
```

Ontology Web Language (OWL)

Java Course Ontology

Figure 2-2.OWL java language course ontology

The techniques and tools developed in the domain of ontology engineering can be applied successfully in the field of knowledge structuring and design [44] [45] [46] and semantic web applications [47]. During the last years, concept mapping has been used to compile maps and mental models that support the process of knowledge sharing. In this way, the idea of the use of ontologies and visual structures in research description is not new. They use to improve the quality of understanding and mentalization among researchers and has been discussed in many works [48] [49] [50] [51] and is being implemented in several research approaches, projects, and software tools [52] [53].

2.3 Natural Language Processing

Natural Language Processing (NLP) is an artificial intelligence field that allows computers to understand, process, and analyze human language. [54] NLP is widely used in the technology industry and serves as the backbone of search engines, spam filters, language translation, and much more. However, computers are still far from being able to understand natural language. Deep Natural Language processing involves common sense knowledge and inferences [2], for this reason, it works mainly in minimal domains and is feasible for large-scale text mining, but with many computational resources [55].

Text Mining refers to turn text into data for analysis discovery. This field has received much attention due to its wide application as a multi-purpose tool, borrowing techniques from Natural Language Processing, Data Mining (DM), Machine Learning (ML), and Information Retrieval (IR) [56]. Text mining techniques are based on statistical methods, are generally superficial, and can be done on a large scale. They have the advantage to be applied to any data text referring to any subject. Text mining tasks cover text categorization, text clustering, entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modeling. However, they do not contribute to a deeper understanding [2]. Text mining techniques do not discuss the standard text structure in detail, and they do not use external knowledge to deal with the semantic gap in the text representation [57]. For example, the tag "Japan Earthquake" does not contain any words or phrases related to the "Nuclear Crisis" while we learn in the news.

Robust and general NLP through text mining techniques tend to be shallow, while a deep understanding tends not to scale up well. In practical applications that demanded a deeper natural language analysis, NLP, combined with machine learning techniques, could focus on a more in-depth analysis of natural language but require a human effort to collect many examples to train ML models [58]. Several studies explore the combination of methods to obtain better results, using general NLP statistical techniques and machine learning algorithms as the basis to analyze text data more precisely, in some diverse domains and applications, such as social media [59, 60], biomedical [61], newsgroup filtering [62], opinion mining [63], and document organization [64].

2.3.1 Representation of Documents

Every day we face a growing volume of documents. The abundant texts on the Internet, vast collections, digital libraries, and repositories, pose challenges for the effective and efficient organization of documents. However, machines are better at understanding numbers that text.

The process of converting textual information into numbers is called vectorization [65]. Among the common ways using vectors for the representation of documents are the following models:

- One-hot encoding model, each unique word with an index in this vector. To represent a unique word, we set the vector's component to be one and zero out all the other components [66].
- Bag of words model (BOW) [65] use the tokenized words for each observation and find out the frequency of each token, disregarding grammar and even word order but keeping multiplicity. This model allows us to compare documents and gauge their similarities for applications like search, document classification, and topic modeling. Though a better approach is to create a vocabulary of grouped words, called the n-gram model. N-gram model changes the scope of the vocabulary and allows the bag-of-words to capture a little bit more meaning from the document. However, this model does not capture the distance between individual words. Also, the closeness between pairs of similar words is not considered by the distance based on the bag of words method.
- Vector space model [67], an algebraic model used for representing documents as vectors. From the given bag of words, it is possible to create a feature document vector where each feature is a word, and its value is term weight. TF-IDF [68] is term weight, which is represented in the Vector space model. It is based mainly on term frequency (TF) and inverse document frequency (IDF). The importance rises proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the whole corpus.

Most word representation approaches depend in one way or another on the distributional hypothesis, which states that words that appear in identical contexts share semantic meaning. The different approaches that take advantage of this principle can be divided into two categories: counting-based methods (e.g., Latent Semantic Analysis) and predictive methods (e.g., neural probabilistic language models).

Counting-based methods calculate statistics on how often a word matches its neighboring words in a large text corpus. The latent semantic analysis models address the problem by learning a latent, low-dimensional representation of documents, analyzing the relationships between documents and the terms they contain, establishing that two words are similar if they occur in similar fragments of text [69].

A matrix, including word counts per document, is constructed from a large piece of text. A mathematical method called singular value decomposition (SVD) is utilized to decrease the number of rows while preserving the similarity structure among columns. Then, documents are contrasted by using the cosine of the angle between the two vectors projected in a multi-dimensional space (or the dot product between the normalizations of the two vectors), using a matrix with the word count of the words in each document. The values close to one describe very similar documents, while values close to zero describe very different documents.

Among LSA models, we have latent semantic indexing (LSI) [70] and latent Dirichlet assignment (LDA) [71]. The latent semantic indexing (LSI) is based on a spectral analysis of the term-document matrix. It seeks to decompose the characteristic space in the word bag model (BOW). Simultaneously, the Latent Dirichlet assignment (LDA) probabilistically groups similar words into topics and represents documents as distribution on these topics. The prediction models try directly to infer a word from their neighbors, where the words are represented as real vectors of characteristics learned from data.

Instead, prediction methods for word vectorization seem to have better performance across various NLP domains such as machine translation, named entity recognition, and role labeling. These methods tend to have lower dimensions leading to a better dense word vector representation that was capturing the meaning and the relationships between words. Some of the predictive methods are Recursive neural network and Neural probabilistic language models (NPLM).

A recursive neural network is a type of deep neural network designed using the same set of weights recursively over a structured input, to give a structured prediction over variable-size input structures, or a scalar prediction on it crossing a given structure in topological order. Whereas Neural probabilistic language models were presented by Bengio et al. [72] introduced the idea that a neural network maps a context (a sequence of word characteristics vectors) to distribution, and thereby, it predicts the most likely next word.

2.3.2 Part of Speech Tagging

Part of Speech Tagging refers to marking up a word in a corpus to a corresponding part of speech tag, based on its definition and context [73] . This task is not straightforward, as a word may represent a different part of speech based on the context in which it is used.

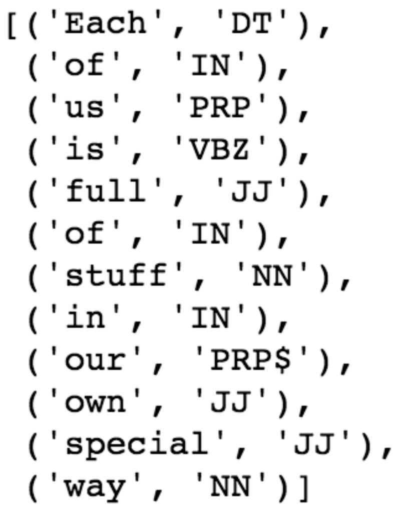
On the other hand, a Part-Of-Speech Tagger (POS Tagger) [74] is an NLP software that reads documents in some language and assigns parts of speech to each word as a noun, verb, adjective among others. Knowing whether a specific word is a verb or a noun tells us a lot about likely neighboring words (determiners and adjectives precede nouns, nouns precede verbs) and about the syntactic structure around the word (nouns are generally part of noun phrases), which makes part of speech tagging an important component of syntactic parsing [75]. POS tagging is also indispensable for building lemmatizers, tools that are used to reduce a word to its root form and minimize text ambiguity [76]. It is useful to decrease the word density in the given text and helps in preparing the accurate features for the training machine.

The Penn Treebank [77] is an inventory of POS tags and uses 48 tags: 36 for part-of-speech, and 12 for punctuation and currency symbols. This increment in the number of tags of traditional linguist's word categories is partly due to more precise granularity. Table 2-1 shows an excerpt from the Penn Treebank tags and descriptions that includes unique tags for determiners, articles, modal verbs, cardinal numbers, foreign words, among others.

Table 2-1 Overview of the Penn Treebank tagset.

Tag	**Description**	**Tag**	**Description**
JJ	Adjective	RBR	Comparative adverb
JJR	Comparative adjective	RBS	Superlative adverb
JJS	Superlative adjective	VB	Verb, base form
LS	List item marker	VBD	Ver, past tense
NN	Noun, singular or mass	VBG	Verb, gerund, or present participle
NNS/NNP	Noun, plural noun, singular	VBN	Ver, past participle
NNPS	Proper noun, plural	VBP	Verb, non-3rd-person singular
RB	Adverb	VBZ	Verb, non-3rd-person singular present

There are many tools relating to Parts of Speech, like Python NLTK [78], Stanford NLP [79], Google Cloud Natural Language [80], among others. Figure 2-3 shows an example of the output provided by a tagger using the Penn Treebank tag set for the *sentence "Each of us is full of stuff in our own special way,"* where parts-of-speech is represented by placing the tag after each word.



```
[('Each', 'DT'),
 ('of', 'IN'),
 ('us', 'PRP'),
 ('is', 'VBZ'),
 ('full', 'JJ'),
 ('of', 'IN'),
 ('stuff', 'NN'),
 ('in', 'IN'),
 ('our', 'PRP$'),
 ('own', 'JJ'),
 ('special', 'JJ'),
 ('way', 'NN')]
```

Figure 2-3. Part of Speech Tagging in a sentence.

2.3.3 Chunking Process

Chunking is an NLP process of analyzing the structure of a sentence to break it down into its smallest constituents (which are tokens such as words) and group them into higher-level phrases. Chunking works on top of POS tagging; it uses pos-tags as input and provides chunks as output. Similar to POS tags, there is a standard set of Chunk tags like Noun Phrase (NP), Verb Phrase (VP), among others [81].

Chunking is critical to extract information from text such as locations and person names in an NLP technique called Named Entity Extraction (NER). Many libraries give phrases out-of-box such as Spacy [82], TextBlob [83], or Natural Language Tool Kit (NLTK) [78] just provides a mechanism using regular expressions to generate chunks. Figure 2-4 shows results from the chunking process in a sentence. The whole sentence, *“the little yellow dog barked at the cat”* is divided into chunks and represented in tree-like structures. Based on defined grammar, an internally tree-like structure is created.

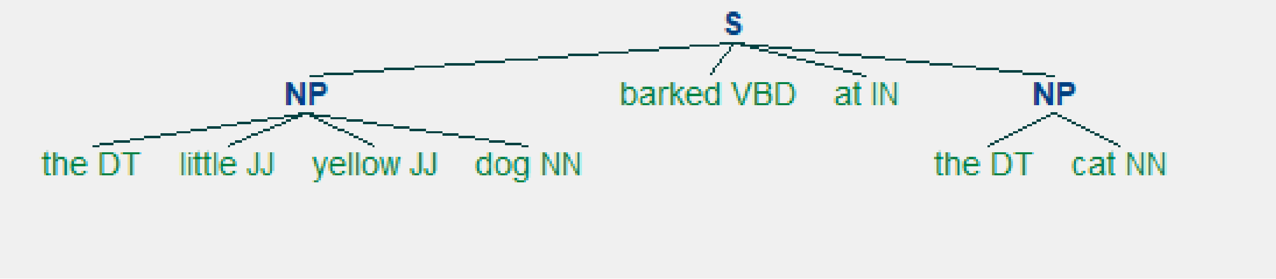


Figure 2-4. NLP Chunking in a sentence

2.3.4 Named Entity Recognition

Named entity recognition is reasonably the most significant task in information extraction. Extraction of more complex structures such as events and relations depend on certain named entity recognition as a preprocessing step. The main NER objective is to recognize named entities from free-form text and to classify them into a set of predefined types such as a person, organization, and location [84]. For example, "JFK" may refer to the president "John F. Kennedy," the place "JFK International Airport," or any other entity receiving the same abbreviation. For this reason, to determine the entity type for "JFK" happening in a document, its context must be considered.

NER systems have been created that utilize linguistic grammar-based techniques as well as statistical models such as machine learning algorithms. Many statistical learning-based named entity recognition algorithms treat the task as a sequence labeling problem-based in several approaches like hidden Markov models [85], support vector machines [86], or conditional random fields [87]. To map named entity recognition to a sequence labeling problem, we treat each word in a sentence as an observation. The class labels must clearly indicate both the boundaries and the types of named entities within the sequence.

2.3.5 Open Information Extraction (OpenIE)

Open Information Extraction (Open IE) [88] refers to the extraction of relation tuples, typically binary relations, from plain text. Unlike previous relationship extraction methods, one needs not to provide a relational schema in advance. For instance, the system first splits each sentence into a set of entailed clauses. Each clause is then maximally shortened, producing a set of entailed shorter sentence fragments. These fragments are then segmented into OpenIE triples, representing a subject, a relation, and the object of the relation. Figure 2-5 shows the OpenIE approach, wherefrom left to right, the sentence "Born in a small town, she took the midnight train going anywhere" yields some independent clauses (e.g., "she Born in a small town"). Next, each clause of the sentence produces a set of shorter sentences and segments those that coincide with an atomic pattern in a relational triple.

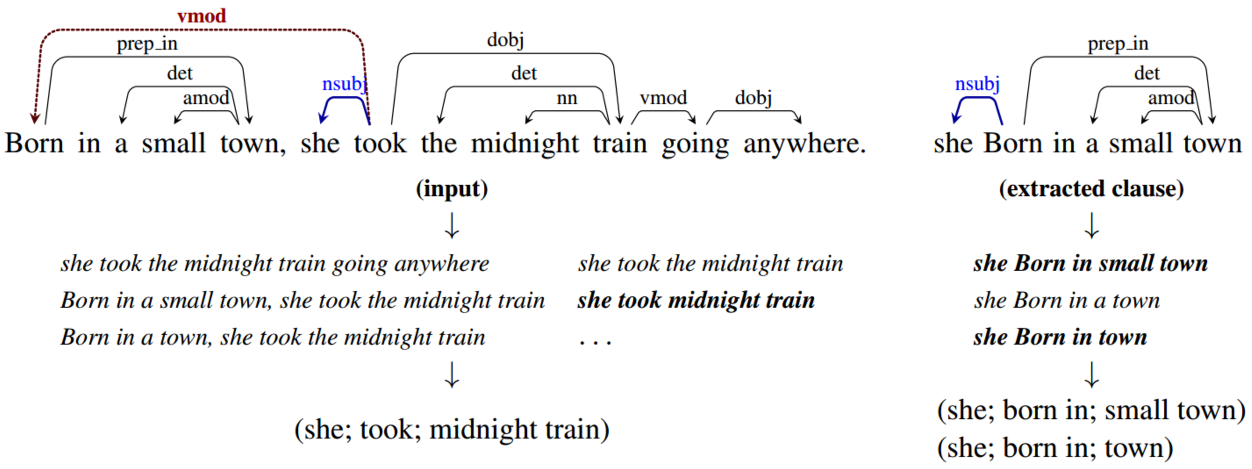

vmod
prep_in
det
amod
nsubj
dobj
det
nn
vmod
dobj
Born in a small town, she took the midnight train going anywhere.
(input)
she took the midnight train going anywhere
Born in a small town, she took the midnight train
Born in a town, she took the midnight train
she took the midnight train
she took midnight train
. . .
(she; took; midnight train)
nsubj
prep_in
det
amod
she Born in a small town
(extracted clause)
she Born in small town
she Born in a town
she Born in town
(she; born in; small town)
(she; born in; town)

Figure 2-5. OpenIE approach and triples extraction process

2.4 Documents Classification

Natural language processing, Data Mining, and Machine Learning techniques can be used to automatically classify and discover electronic document patterns. By classifying text, we focus on assigning one or more categories to a document, making it easier to manage and sort. It is especially practical for publishers, news sites, blogs, and applications such as spam detection, auto-tagging customer queries, and sentiment analysis [2] [89]. For the classification of documents, it is possible to explore unsupervised, supervised, and deep learning approaches.

2.4.1 Unsupervised learning methods

- In unsupervised learning, documents can be considered as feature vectors for the execution of clustering algorithms that suggest groups based on data patterns., such as K-Means clustering [90], spectral clustering [91], Hierarchical Clustering and Density Based Spatial Clustering of Applications with Noise (DBSCAN).

- **K-means Clustering:** K-means follows a simple way to classify a given dataset through a certain k number of clusters. The idea is to define one centroid for each cluster. These centroids should be placed, as much as possible, far away from each other. The following step is to take each point belonging to a dataset and associate it to the nearest centroid value. When there is no pending point, the next step is completed, and an early grouping is performed. At this point, we need to recalculate the k new centroids as barycenters of the clusters resulting from the previous step. After having these new centroids, a new binding must be made between the same data setpoints and the new nearest centroid [90].
- **Hierarchical Clustering**: The method of agglomerative hierarchical clustering is particularly useful to support a variety of search methods because it naturally creates a tree-like hierarchy that can be leveraged for the search process. Almost all the hierarchical clustering algorithms successively merge groups based on the best pairwise similarity between these groups of documents [92].
- **Spectral clustering:** It is an algorithm derives from graph theory, where the approach is utilized to identify node communities in a graph based on the edges that connect them. The method is adaptable and allows us to cluster non-graph data as well. It uses information from the eigenvalues of specific matrices built from the graph or the dataset [91]

2.4.2 Supervised Learning methods

The classifiers learn the characteristics of the categories of a set of previously classified documents, and after this process, they can classify documents in the predefined categories [93]. Their first steps towards training a classifier with machine learning are feature extraction: a method that is applied to transform each text into a numerical representation using vectors. The most used approach is a bag of words [65]. Once trained with enough training samples, the machine learning model can begin to make accurate predictions.

Supervised classification techniques usually included in the following components [2, 94]:

- Training text: It refers to the input text through which our supervised learning model can learn and predict the specified class.

- Feature Vector: It refers to a vector that contains information describing the input data features. These input sets are then forwarding to the model, which generates expected labels.
- Labels: These are the predefined categories or classes that the model will predict
- ML Algorithm: It is the algorithm through which our model can deal with text classification.
- Predictive Model: It refers to a model trained on the historical dataset which can perform label predictions.

Figure 2-6 shows a supervised classification model schema based on the training corpus containing the correct label for each input.

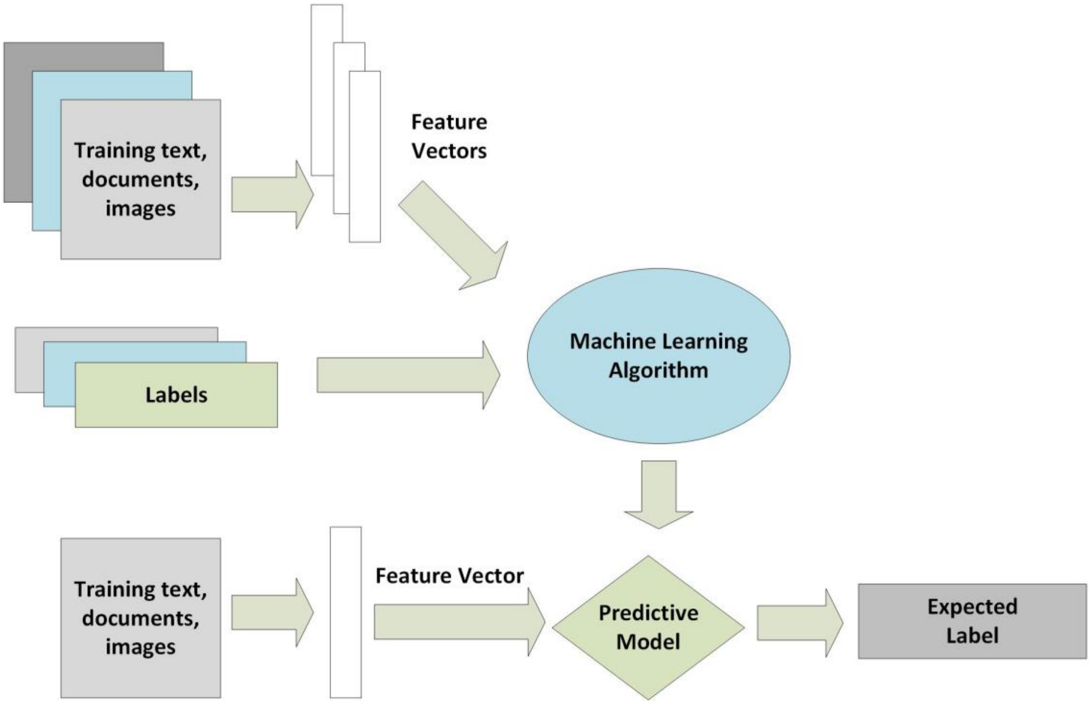

Training text, documents, images
Feature Vectors
Labels
Machine Learning Algorithm
Training text, documents, images
Feature Vector
Predictive Model
Expected Label

Figure 2-6. Supervised classification model approach

Some popular supervised learning algorithms for creating text classification models include: Naïve Bayes [95], K-nearest neighbors (KNN) [96], support vector machines (SVM) [97], decision trees [98], and neural networks [99] .

- **Naïve Bayes:** The algorithm is one of the members of the Bayesian family based on Bayes' Theorem, which can provide accurate results without much training data and helps to calculate the conditional probabilities of occurrence of two events based on the probabilities of occurrence of each event [95].

- **K nearest neighbors (KNN):** This algorithm is utilized to classify by finding the K nearest matches in training data and then using the labels of closest matches to predict clusters. Usually, distances metrics functions such as Euclidean is used calculate the distance to determine what the nearest is. It is, however, computationally expensive, and data must be normalized so that all data points are brought into the same range [96].
- **Support Vector Machines:** SVM traces a hyperplane that splits space into two subspaces: one first subspace that contains vectors that be a member of a group and a second subspace that contains vectors that do not be included in that group. The hyperplane dimension is dependent on how many features there are; i.e. if there are two input features, the hyperplane will be nothing more than a line but, if there are three features, it is a 2D plane [97].

  The SVM model does not need much training data to start providing accurate results.
- **Decision trees:** The goal of this algorithm is to find the descriptive features which contain the most information regarding the target feature and then split the dataset along the values of these features such that the target feature values for the resulting subgroups are as pure as possible. The process of finding the most informative feature is done until we meet the detention criteria, where we finally end up in the so-called leaf nodes. The leaf nodes contain the predictions we will make for the new query instances presented to our trained model [98].
- **Neural Networks:** These models are used to recognize complex patterns and relationships that exist within a labeled data. A neural network is an assemblage of neurons with weights which connects them, and they process documents one at a time and learning by comparing their classification with the actual classification. The neural network takes input values and weights from the input layer as input, and then it goes to the hidden layer in which a function sums the weights and maps the results to the corresponding output layer units [99].

2.4.3 Multi-label classification models

The classification assumes that each sample is assigned to one and only one label. On the other hand, multi-label classification algorithms assign each sample a set of target labels or classes. It is can be thought of as predicting data-point properties that are not mutually exclusive [100].

The multi-label text classification has been applied to several tasks and applications such as categorizing users, indexing of documents collections, and detecting sentiment analysis in the text. Some classification algorithms have been adapted to the multi-label task, such as k-nearest neighbors, decision trees, kernel methods for vector output, and neural networks, among others. However, many of these methods ignore word order, opting to use word bag models or Term Frequency–Inverse Document Frequency (TF-IDF) weighting to create document vectors. New approaches based on convolutional neural networks and “Word Embeddings”, can be used to improve multi-label learning.

2.4.4 Deep Learning

Deep Learning is a promising part of artificial intelligence that reproduces human brain functions in processing information and generating patterns to be used in decision-making. The field of deep learning uses a hierarchical degree of neural networks with the ability to learn from unstructured information without supervision [101].

Deep Learning, when used for NLP, can be viewed as pattern recognition applied to sentences, words, and paragraphs; the same way, deep learning for images can be viewed as pattern recognition applied to image pixels. Neural networks never take in data in their raw form; instead, all data to be fed in a neural network has to be first converted into tensors [102]. A tensor is a generalization of vectors and matrices and is represented using n-dimensional arrays.

Vectorization refers to all the processes involved in the transformation of data received as text into numeric tensors. This transformation can be carried out in different ways: by trying to segment the text received into words, then transforming every word to a vector; by trying to segment the text received into characters, then transforming every character to a vector or by extracting n-grams of words and characters, then transforming every n-gram to a vector. The various units in which text is divided are known as tokens, and these are linking to numerical vectors. All these vectors joined with chain tensors, are fed to deep learning systems.

The two central deep learning architectures applied in text classification are Convolutional Neural Networks (CNN) [103] and Recurrent Neural Networks (RNN) [104]. Deep learning algorithms are used to achieve better vector representations for words and increase the accuracy of classifiers trained with conventional machine learning algorithms.

2.4.5 Recurrent Neural Networks

A recurrent neural network (RNN) is a type of artificial neural networks; they have connections that have loops, which adds feedback and memory to the networks over time [104]. This memory recognizes this type of network to learn and generalize through input sequences instead of individual patterns. A recurrent neural network can be considered of as multiple copies of the same network, each passing a message to a successor.

Applying the knowledge from an external embedding can enhance the precision of RNN because it integrates new information (lexical and semantic) about the words, a piece of information that has been trained and distilled on an extensive corpus of data. The recurrent neural networks have been applied with massive success in several types of deep learning problems, including language modeling, speech recognition, captioning images, and language translation. It has been shown that a dominant type of recurrent neural network called Long short-term memory network (LSTM) [105], is particularly useful when stacked in a deep configuration and is perfectly able to learn long-term dependencies. They will remember information for a long time, not having to learn it over time.

Cell states are one crucial factor in LSTM. They are a transport highway that transfers relevant information down the sequence chain. LSTMs can add information, and they can remove it from the cell state, which is done in a regulated and structured way using gates. Gates are another optional way of allowing information to go through, and they are made of a single multiplication operation and a sigmoid neural layer.

2.4.6 Word Embeddings

The term "word embedding" was first studied by Bengio [72] and referred to the collective name of a set of language modeling and representation learning techniques, in natural language processing (NLP), where vocabulary words or phrases are assigned to vectors of real numbers and semantically similar words have similar or "close" representations. This approach opens a new dimension of possibilities for finding patterns or other insights in the data. Figure 2-7 shows a word embedding model representation where the vectors of words estimated as similar by context, such “pepper” and “salt” or “baking” and “boiling” are placed closer (together). On the other hand, words like "cake" and "salad" are placed further.

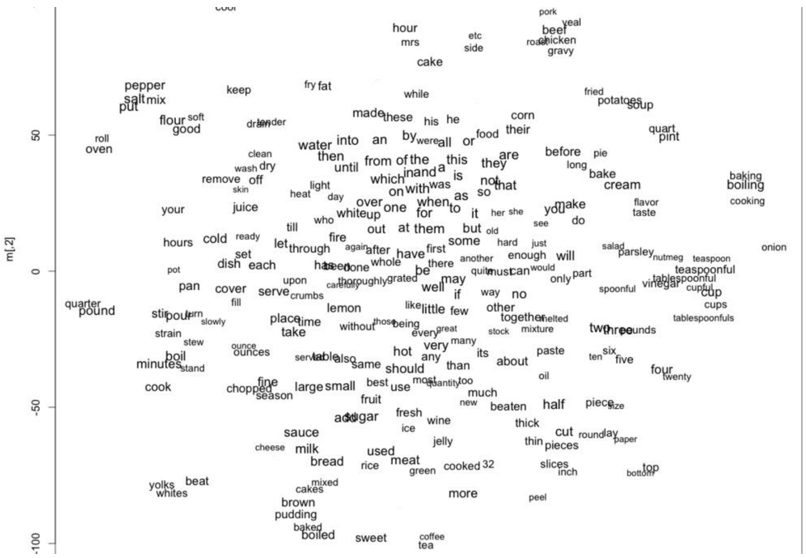


Figure 2-7. Word embedding model representation

Word embedding approaches could help feature generation, document clustering, text classification, and other natural language processing tasks. They can also be used to suggest words to the word being subjected to the prediction model or for semantic grouping, which groups things of similar characteristics and dissimilar far away.

Among the methods applied to calculate the vector representations of words, we have GloVe [106], Dependency-based word embeddings [107] and Random Indexing [108], however, Word2vec is considered state of the art in this type of models applied to natural language processing (NLP) [109]. Word2vec is a predictive model based on two-layer surface neural networks that are trained to reconstruct linguistic contexts of words. This model is very efficient in learning word embeddings from raw text. Word2Vec takes a large text corpus and produces a vector space, typically of several hundred dimensions, where each individual word in the corpus is assigned a corresponding vector in space.

The Word2vec implementation has two methods; the continuous word bag model (CBOW) [110] and the Skip-Gram model [111]. Both models are algorithmically similar, except that CBOW predicts the target words from the context words, while the Skip-gram model does the reverse and predicts the context words from the target words. The selection of one or the other method might seem an arbitrary choice. However, statistically, the effect of the CBOW method significantly softens the distributed information, by creating a whole context as an observation, which turns out to be useful for smaller data sets. However, Skip-gram treats each pair (context-objective) as a new observation, and this tends to get better results when you have more extensive data sets.

When the feature vector assigned to a word cannot be used to predict the context of the word accurately, the vector components are adjusted. The context of each word in the corpus, if applicable, sends error signals to adjust the feature vector. The vectors of words estimated as similar by context are placed closer (together), adjusting the numbers in the vector. A well-trained set of word vectors will place similar words close to each other, in that space. A trained Word2Vec model can gauge relations between words of one language and map them to another. Figure 2-8 shows (in two dimensions, reduced from hundreds) the relative positions of various words in vector spaces representing English and Spanish.

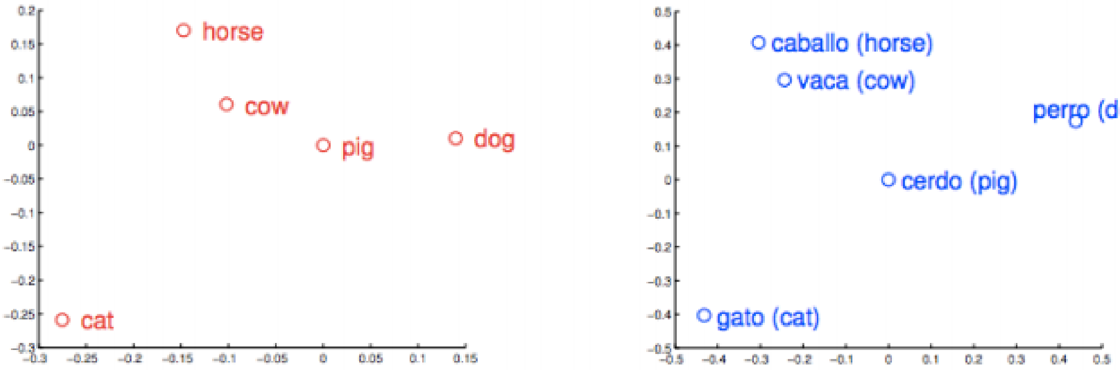


Figure 2-8 A mapping between words belonging to different vector spaces

2.4.7 Convolutional Neural Networks

Convolutional neural networks (CNN) use layers with convolving filters applied to local features [103]. Convolution is the application of a filter to an input that results in an activation. Repeated application of the same filter to an input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input [112].

Initially designed for computer vision approaches, CNN models have subsequently been shown to be useful for NLP and have reached excellent results in semantic parsing. Kim [113] and Berger [114] demonstrated that CNN models using semantic word embeddings significantly outperform the Binary Relevance method with bag-of-words features on a large scale multi-label.

Kim proposes that instead of image pixels, the inputs are sentences or documents represented as a matrix. Sentences are mapped to embedding vectors, such as Word2vec and GloVe, and are available as a matrix input to the model. It means that for a 10-word sentence using a 100-dimensional embedding, we would have a 10×100 matrix as our input.

Kim's CNN architecture consists of an input layer composed of a sentence, which is comprised of concatenated word embeddings, followed by a convolutional layer with multiple filters. Convolutions are conducted across the input word-wise using varying sized kernels, such as 2 or 3 words at a time. The resulting feature maps are then processed using a max-pooling layer to condense or summarize the extracted features.

Figure 2-9 shows a representation of convolutional Neural Networks model architecture for sentence classification. It is composed of an input layer with different ngrams window sizes, followed by convolutional and pooling layers for use across various fundamental natural language processing problems.

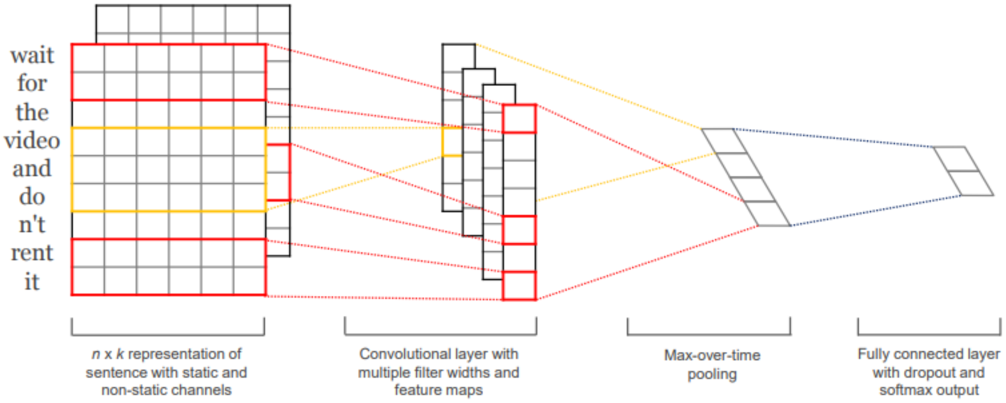

wait
for
the
video
and
do
n't
rent
it
n x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 2-9 Kim's Convolutional Neural Networks model architecture

Kim also works a model with two different channels in the form of dynamic and static word embeddings, where one channel is modified during the training task, and the other is not. A near but more complex architecture was previously proposed by Kalchbrenner and Wang [115] [116]. This proposed architecture adds a layer that fulfills "semantic clustering" to this network architecture. Semantic clustering refers to groups semantically equivalent words, phrases and sentences — into clusters based on meaning.

CNN Models for text, regularly, use filters that slide over full rows of the matrix (words). Thus, the width of the filters is usually the same as the width of the input matrix. The height, or region size, may vary, but sliding windows over 2-5 words at a time is typical.

CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons typically mean fully connected networks; that is, each neuron in one layer is connected to all neurons in the following layer. The "fully-connectedness" of these network types makes them prone to overfitting data. Common ways of regularization include adding some form of magnitude measurement of weights to the loss function [117]. However, CNN models take a distinct approach towards regularization: they take advantage of the hierarchical pattern in data and set up more complex patterns using smaller and simpler patterns [118].

2.5 Similarity Measures

Similarity measure in text mining context is usually described as a distance with dimensions describing objects features. If this distance is short, it will be a high degree of similarity where a considerable distance will be the low similarity degree [119]. There are several similarity measures between documents, such as Euclidean distance, Manhattan distance, Jaccard Similarity, Cosine Similarity, Kullback-Leibler divergence, among others [120].

The Euclidean distance between two points is the length of the path connecting them. The Pythagorean theorem gives this distance between two points. Manhattan distance is a popular metric in which the distance connecting two points is the sum of the absolute differences of their Cartesian coordinates.

The Jaccard similarity measures the similarity between finite sample sets and is defined as the cardinality of the junction of sets divided by the cardinality of the union of the sample sets [121]

The cosine similarity is a measure of similarity between two vectors in a space that has defined an inner product that evaluates the value of the cosine of the angle between them [122]. Another approach for similarity measurements is the Kullback-Leibler divergence [123], which measures the divergence or distance between two probability distributions.

The use of Word2vec's word embeddings properties is one of the features of Word Movement distance (WMD) [124], which can be calculated as the sum of the distances between word pairs in the texts. The WMD measure will be lower, while the sentence pairs are closer in similarity.

The distance depends on the vector space and, therefore, on the features used to calculate the vectors. Given the definition (0 = no similarity, 1 = identical), a similarity between documents above 0.5 might be a good starting point. The calculated similarity should be evaluated (for example, if we know which instances should be similar) to verify how good the set of characteristics is. Therefore, the resulting vectors reproduce the expected similarity. Finding an appropriate threshold value depends on the situation and is an evaluation task that must be adapted to the data.

2.6 Topic Modeling

In topic modeling, we use a probabilistic model in order to determine a soft clustering, in which each document has a membership probability of the cluster, as opposed to a hard segmentation of the documents. Each topic can be considered a probability distribution over words, with the representative words having the highest probability. Each document can be expressed as a probabilistic combination of these different topics. These approaches are LDA (Latent Dirichlet Analysis), and NMF (Non-negative Matrix factorization).

2.6.1 Latent Dirichlet Allocation

One of the most commonly used techniques for topic modeling is Latent Dirichlet Allocation (LDA) [71], a generative model representing individual documents as mixtures of topics, wherein a particular topic generates each word in the document.

In the LDA model, documents θ are not directly linked to the words w; rather, this relationship is governed by the additional latent variables, z, introduced to represent the responsibility of a particular topic in the use of that word in the document; in other words, topic(s) in which document is focused. Also, by introducing the previous Dirichlet distributions, α, and β, on the distributions of documents and topics, respectively.

Hyperparameter α can be interpreted as a prior observation count for the number of times topic i, is sampled in a document, and hyperparameter β can be interpreted as the prior observation count on the number of times words are sampled from a topic before any word from the corpus is observed.

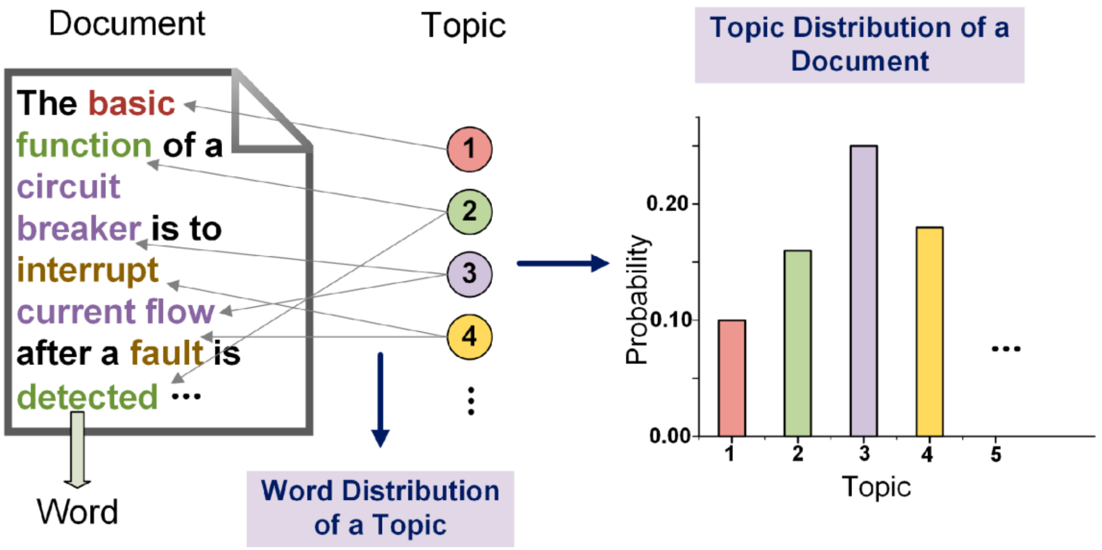
The generative model of LDA is complete and capable of processing unseen documents and is as follows:

1. Select K multinomials φ_i φ_k from the previous Dirichlet distribution β, one for each topic k.
2. Select D multinomials θ_d from the previous Dirichlet distribution α, one for each document d.
3. For each document d in the corpus, and for each word w_{di} in the document the following steps are applied:
   a) To select a topic z_i of multinomial θ_d;$(z_i|\alpha)$,
   b) To select a word w_i`from multinomial φ_z;p(w_i | $(z_i$, β).

As well as, the hidden structure of topics in LDA is described by the posterior distribution of the hidden variables given by the documents D:

$$\rho(\theta, z|w, \alpha, \beta) = \frac{p(\theta, z|w, \alpha, \beta)}{p(w|\alpha, \beta)}$$

Figure 2-10 shows a representation of Latent Dirichlet Allocation (LDA) model and document topic distribution, where each document is represented as a random mixture of latent topics, and each topic is characterized by a distribution of words


Document
Topic
Topic Distribution of a Document
The basic function of a circuit breaker is to interrupt current flow after a fault is detected ···
1
2
3
4
⋮
Word
Word Distribution of a Topic
Probability
0.20
0.10
0.00
1
2
3
4
5
···
Topic

-

Figure 2-10. Topic Modeling using LDA

Unlike latent Dirichlet allocation, this topic model infers the number of topics from the data. Hierarchical Latent Dirichlet Allocation (hLDA) model [125] extends Latent Dirichlet Allocation to infer a hierarchy of topics from a corpus of documents and makes it possible to assume that topics are arranged in a treelike structure where the tree has L levels, and every node is a topic.

With LDA, we selected topics using a mixture model with K random mixing proportions *(where *K* is the number of possible topics), denoted by the K-dimensional vector θ. With hLDA, we instead choose a path in the *L*-level tree from root to leaf, choose a vector θ of topic proportions from a Dirichlet distribution of L dimensions, next use a topics mixture from root to leaf to generate the words that compose each document, using mixing proportions θ.

Hierarchical LDA considers a dataset composed of a corpus of documents, where each document is a collection of words, and each word is an item in a vocabulary. It is possible to assume that the topics in a document are represented as a mixture model, where the mixing proportions are multinomial random and document-specific [71]. These topics are the basic mixture components in hLDA.

2.6.2 Non-Negative Matrix Factorization

Non-Negative Matrix Factorization (NMF) [126] is a linear algebra optimization algorithm that extracts significant topics from the decomposition of the matrix of document terms. The vectors in the basis system directly correspond to cluster topics. Therefore, the cluster membership for a document may be determined by examining the most significant component of the document, along with any of the vectors. The coordinate of any document along a vector is always non-negative. The expression of each document as an additive combination of the underlying semantics makes sense from an intuitive perspective.

Because it gives a semantically significant result and provides an intuitive understanding of the basic system in terms of the clusters, NMF is used as a method of classification, especially for document data and as a method for modeling topics [127].

2.7 Ontology Metrics

In order to evaluate the performance of ontology matching algorithms it is necessary to confront them with test ontologies and to compare the results. The most prominent criteria are precision and recall originating from information retrieval [128, 129] .

Where precision is term proportions in the learned ontology, included in terms identified in reference ontology and Recall refers to term proportions identified in reference ontology and which are included in the terms collected in the learned ontology.

$$Precision = \frac{RdO \ \cap drO}{drO}$$

$$Recall = \frac{RdO \ \cap drO}{RrO}$$

The set RdO represents the set of all the elements given in the manually constructed reference ontology, and drO is the set of elements contained in the learned ontology given by the ontological extraction process.

The F-measure score is the harmonic average of the precision and recall, where it reaches its best value at 1. The F-measure that can be defined as follows:

$$F\beta = \frac{(1 + \square 2) \cdot (\square\square\square\square\square\square\square\square\square \cdot \square\square\square\square\square\square)}{\square 2 \cdot \square\square\square\square\square\square\square\square\square + \square\square\square\square\square\square}$$

Where β is a real positive number between 0 and 1.

For the comparison of conceptual hierarchies [130], the measures have two categories: local and global. The local measure compares the similarity of the positions of two concepts in the learned and reference ontology. The overall measured measure is used to compare two hierarchies of complete concepts. It is calculated by averaging the local measurement results for the reference concept pairs and the learned ontology.

The local taxonomic precision tp ($c1$, $c2$, Oc, Or) compares the position of the learned concept c1 in its conceptual hierarchy with the position of the reference concept c2. It is defined as follows [131]:

$$tp(c1, c2, Oc, Or) = \frac{sub(c1{,}0c) \ \cap sub(c2{,}0r)}{sub(c1{,}0c)}$$

The overall precision TP (Oc, Or) compares the complete hierarchy of the ontology learned *Oc* concerning the ontology of the reference Or. The local taxonomic Recall is defined as follows [131]:

$$tr(c1, c2, Oc, Or) = \frac{sub(c1{,}0c) \ \cap sub(c2{,}0r)}{sub(c2{,}0c)}$$

Other quality aspects to consider ontologies are as follows: [132]

- Consistency: Describes that the ontologies do not include or allow any contradiction. Reviewers should verify whether the learned ontology presents semantic consistency in the definitions, in the meanings of the formal and informal definitions, and in the sentences that can be inferred by using other definitions and axioms that may or may not belong to the same ontology. How metric is determined to count the number of terms with the inconsistent meaning
- Concision: This principle is related to whether all the information collected in an ontology is useful and accurate. Counts will be made regarding redundant terms in the ontology.
- Computational Efficiency: It is related to the speed at which the tools can work with the ontology

- Clarity measures: how effectively the ontology communicates the intended meaning of the defined terms. Definitions should be objective and independent of the context.

3 Ontologies for Education and Learning Analytics

This chapter will introduce the importance and use of ontologies in education and its relationship with learning analytics area, in particular, with the processing and visualization of representations of the course's domain knowledge, needed for the generation of Intelligent Curricula. First, ontologies research, objectives, and ontologies generation approaches focus on the education field will be described; second, an introduction about learning analytics concepts and goals. Finally, the chapter ends with an explanation about Intelligent Curricula description and structure.

3.1 Ontologies in Education Field

During the last years, visual knowledge representation has become a critical consideration in knowledge engineering methodology. Visualization works as a cognitive tool that facilitates communication in the teacher/student interaction and the research communities. A particular interest can be observed in such graphic forms of knowledge coding in the educational sciences, especially in learning. Students participate in group processes of knowledge exchange and co-creation with continuous feedback.

Knowledge visual representations are strongly associated with ontology design and development. Ontologies are useful structuring tools; in that, they provide an organizing axis along which teachers or students can mentally mark their vision in the information hyper-space of domain knowledge. Many tutors and scholars, especially those who teach science courses, operate as knowledge analysts or knowledge engineers by making visible the skeleton of the studied discipline and showing the domain's conceptual structure [133]. An ontology frequently represents this structure.

Previous works have already identified the importance of developing rich ontologies in the education field. Ontologies, which form a conceptual skeleton of the modeled domain, could serve to several educational purposes such as better understanding, knowledge creation, knowledge sharing and reusing, collaborative learning, intelligent tutorials, problem-solving, seeking advice, developing competencies and intelligent systems to support teaching and learning by offering automated services like syllabus semantic searching, matching and interlinking syllabus recommendation and evolution [14] [134] [135].

However, ontologies are inevitably subjective to a certain extent, as knowledge includes a component of personal subjective perception; however, using the ontologies developed by others is a convenient and compact means of acquiring new knowledge. These domain ontologies are also widely and effectively used for learning in several disciplines [15] [136] [137].

Curriculum management and development could be improved using educational ontologies in curriculum tasks like comparing, aligning, and matching between universities, educational systems, or relevant disciplines. Following this approach, several learning analytics applications also could use Intelligent Curriculum represented as ontologies, to recommend learning materials [138], to automatically sequence learning activities [137], to evaluate the quality of contributions in online forums [139]. Also, it monitors the authoring process and prevents and solves inconsistencies [140] or to provide visual feedback to students about their progress [141].

Ontologies also can be used in adaptive educational systems [16]. In these systems, student models computed in terms of ontologies they have concepts that mapped to topic models extracted from educational resources can perform recommendations of learning materials to students.

3.1.1 Automatic Ontologies generation

Automatically extracting ontologies from existing semi-structured text is not a novel idea. Gaeta et al. [142] present an approach to generate large ontologies, from various sources, using semantic interpretation and harmonization algorithms. Zouaq et al [143] present a semi-automatic framework to produce domain concept maps from text and then to derive domain ontologies from these concept maps. Wong et al. [144] present a survey of several techniques to do it.

In the field of education, the most successful approaches have been proposed by. Guerra at al. [19] and Taker [20]. Guerra uses topic modeling to create links between textbooks chapters and subchapters. This links network of content was then used to recommend those materials back to students, depending on search queries.

Taker attempts to connect educational resources through concept-level embeddings, using Word2vec and Doc2vec Models. Despite their success, these efforts fail short of creating a fully functional ontology that goes beyond linking part of educational materials and represents the domain knowledge of a course.

Another interesting approach was the one, followed by Lau et al. [145]. They extract a domain ontology for a course based on the post on its online forum. While the techniques used worked, the quality of the domain ontology was heavily influenced by the (lack of) participation of the students in the forum. In these two examples, the links or ontologies automatically created were not designed to be changed or fixed by the domain expert, even in the case of computational error.

3.2 Learning Analytics

The widespread integration of digital technology in higher education influences teaching and learning practices, and allows access to data, mainly available in online learning environments, that can be used to improve student learning. To this end, higher education institutions are implementing Learning Analytics (LA) to understand better and support student learning [146].

Learning analytics is the measurement, collection, analysis, and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs [147]. As a research and teaching field, Learning Analytics focus at the convergence of learning (e.g., Educational Research, Educational Technology, Learning and Assessment Sciences), Analytics (e.g., Visualization, Statistics, Computer Science, Data Science, Artificial Intelligence), and Human-Centered Design (e.g., Sociotechnical, Usability, Participatory Design, Systems Thinking).

Learning Analytics (LA) and Educational Data Mining (EDM) are areas of scientific inquiry, that share since their inception a common interest in data-intensive approaches to education research. The overall purpose of LA and EDM is to understand how students learn. Based on the analysis of large-scale educational data, LA and EDM aim to support research and practice in education [148]. However, there are several distinctions between LA and EDM [149] [18]. First, one key distinction concerns the type of discovery that is prioritized: EDM has a primary focus on automated discovery, whereas LA has a stronger focus on leveraging human judgment. Second, EDM models are often used as the basis for automated adaptation, conducted by a computer system, whereas LA models are often developed to inform instructors and learners. Third, EDM researchers use reductionist frameworks: they reduce phenomena to components and focus on the analysis of individual components and the relationships between them. By contrast, LA researchers have a stronger focus on understanding complex systems as totalities.

Learning Analytics seeks to exploit the new opportunities once we capture new forms of digital data from students' learning activity and use computational analysis techniques from artificial intelligence and data science fields. The evidence from education research shows that there are productive and potent ways of using analytics for supporting teaching and learning. Some goals of learning analytics include the following points:

- Supporting student development of learning skills and strategies in achieving their study goals or in reaching their potential [150].
- Provision of personalized and timely feedback to students regarding their learning [151].
- Supporting development of metacognitive awareness and important skills such as collaboration, critical thinking, communication and creativity [152].
- Develop student awareness by supporting self-reflection [152].
- Support quality learning and teaching by providing empirical evidence on the success of pedagogical innovations.

LA provides researchers with new tools to study teaching and learning. Moreover, as data infrastructures improve, they offer a variety of tools, from data capture and analysis to visualization and recommendation we can close the feedback loop to students [151], offering more timely, precise, actionable feedback. Besides, educators, instructional designers, and institutional leaders gain new insights once the learning process is persistent and visible [150].

3.3 Intelligent Curriculum

Intelligent Curriculum has been identified since the inception of Learning Analytics [153] as one of the enablers of a data-informed decision support systems in education. This can be defined as the representation of the domain knowledge usually taught in a course in a way that is amenable to be understood and processed by a computational system. The most common representation that fulfills this requirement is an ontology [17].

The Intelligent Curriculum could allow computers systems support teaching and learning and to provide personalized content to students. Students' activity and their evolving profile can be constantly coinciding with the knowledge architecture of a particular domain and learning resources provided to fill any knowledge gaps.

In the course ontology structure, a term can be the title of a chapter and a section in the course, or the key concept of the course contents. The granularity of the course ontology is determined by instructor or a domain expert. The course ontology can be represented by a directed graph. The node set in the graph represents the terms. If term^1 and term^2 satisfy IS A (term^1; term^2) it means that term^2 is a (part of, or component of) term^1, thus there is a direct edge from term^1 to term^2 in the directed graph. Figure 3-1 shows a course Intelligent Curriculum represented as an ontology.

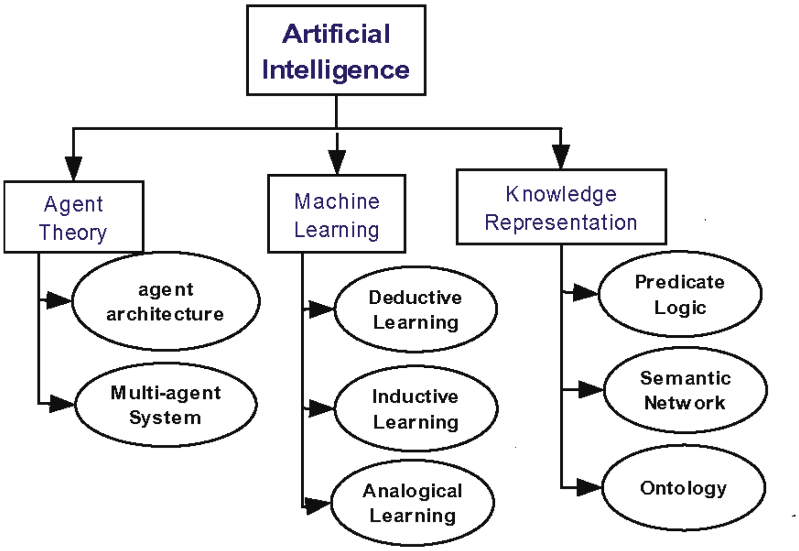


Figure 3-1. Intelligent Curriculum for AI Course

Despite the usefulness of Intelligent Curriculum, the cost of its manually creating course-oriented ontologies is high [154]. Domain experts are rarely experts in semantic technologies and vice versa. Moreover, the cost of maintaining these ontologies up-to-date to the natural changes of the courses and important topics is not trivial [155]. It could be argued that the cost of creation and maintenance of small-scale ontologies has limited their use in the field of Learning Analytics. This lack of course ontologies, however, is opposed by the abundance of semi-structured information in the form of learning materials.

4 Extraction of educational ontologies from semi-structured data

This chapter is based on the works described in the papers “Assisted curricula design based on the generation of domain ontologies and the use of NLP techniques” and “Semi-Automatic Generation of Intelligent Curricula to Facilitate Learning Analytics” and covers a novel approach to (semi-)automatically generate Intelligent Curriculum through ontologies extracted from existing learning materials such as digital books or web content. Through a series of natural language processing steps, the semi-structured information present in existing content is transformed into a concept-graph. This section also evaluates the proposed methodology by applying it to learning content for different courses and measuring the quality of the extracted ontologies against manually generated ones. The results obtained suggest that the technique can be readily used to provide domain information to other Learning Analytics tools.

4.1 Introduction

An Intelligent Curriculum can be defined as the representation of the domain knowledge usually taught in a course in a way that is amenable to be understood and processed by a computational system. The most common representation that fulfills this requirement is an ontology. Once the curriculum is represented as an ontology, several existing Learning Analytics applications could use this information to recommend learning materials, to automatically recommend learning materials [138], to automatically sequence learning activities [137], to evaluate the quality of contributions in online forums [139]

As useful as having an Intelligent Curriculum could be, the cost of its manually create these course-oriented ontologies is high [8]. Domain experts are rarely experts in semantic technologies and vice versa. Moreover, the cost of maintaining these ontologies up-to-date to the natural changes of the courses and important topics is not trivial [9]. It could be argued that the cost of creation and maintenance of small-scale ontologies has limited their use in the field of Learning Analytics.

This lack of course ontologies, however, is opposed by the abundance of semi-structured information in the form of learning materials. For most disciplines, it is easy to find learning materials created by domain experts that contain relevant content and structure (table of contents, links, sections, references) that make explicit the semantic relation between different topics and subtopics. This work proposes and tests the idea of using existing learning resources such as digital books, web-based tutorials or existing syllabus text in digital format as sources to (semi-)automatically build and maintain course-based ontologies that could realize the concept of Intelligent Curriculum.

The main contribution of this work is the creation and evaluation of an automatic algorithm to extract course-centric ontologies from authoritative sources (digital textbooks, web tutorials or syllabus as digital text) that could be used by Learning Analytics tools and could be easily modified by the domain expert without the need to know about semantic technologies.

4.2 Related Work

Automatically extracting ontologies from existing semi-structured text is not a novel idea. Wong et al. [10] present a survey of several techniques to do it. In the educational field, the most successful approach has been the one proposed by Guerra at al. [11] to use topic modeling to create links between textbooks chapters and subchapters. This network of content was then used to recommend those materials back to students, depending on search queries. As successful as it is, this attempt fails short of creating a fully functional ontology that goes beyond linking part of educational materials but also represents the domain knowledge of a course. Another interesting approach was the one, followed by Lau et al. [12]. They extract a domain ontology for a course based on the post on its online forum. While the techniques used worked, the quality of the domain ontology was heavily influenced by the (lack of) participation of the students in the forum. In these two examples, the links or ontologies automatically created were not designed to be changed or fixed by the domain expert, even in the case of computational error.

4.3 Methodology

Figure 4-1 shows the methodology phases for the proposed methodology, including parsing processing, topic modelling and preliminary and definitive educational ontologies processing. Next, each of the tasks is explained.

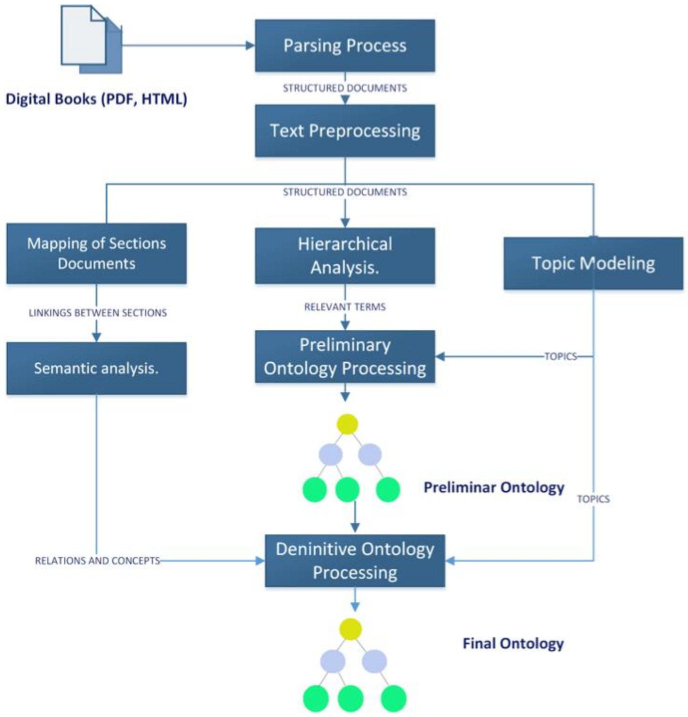
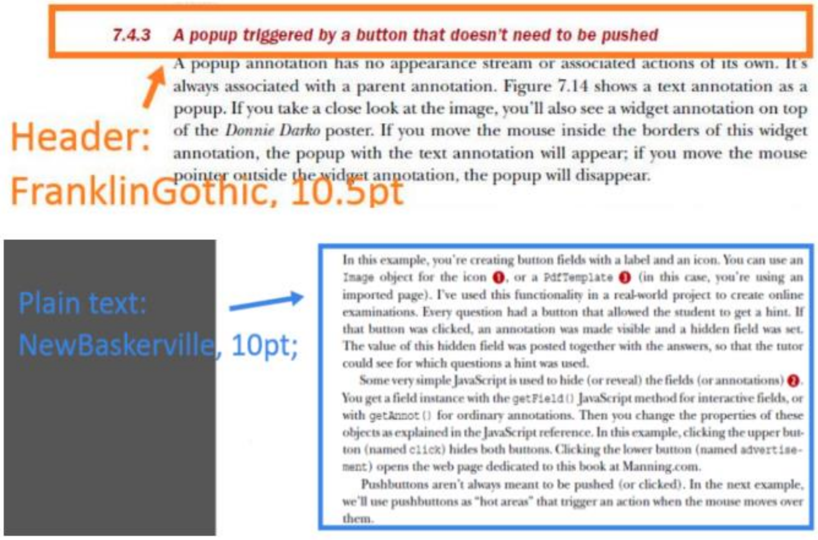


Figure 4-1. Methodology for automatically-generated educational ontologies

4.3.1 Source Selection

The selection of relevant material is a manual step. The domain expert collects documents that she or he thinks are relevant for the course. These documents could be digital textbooks, web tutorials, syllabus in digital format, among others. They could cover all or part of the topics of the course. Currently, the automatic extraction system can process Portable Document Format (PDF), Hypertext Markup Language (HTML), or plain text files. If the original document is not in one of those formats (for example, Microsoft Word), it needs to be converted. This is just a technical requirement that could be improved in the future.


Header:
FranklinGothic, 10.5pt
Plain text:
NewBaskerville, 10pt;

7.4.3 A popup triggered by a button that doesn't need to be pushed

A popup annotation has no appearance stream or associated actions of its own. It's always associated with a parent annotation. Figure 7.14 shows a text annotation as a popup. If you take a close look at the image, you'll also see a widget annotation on top of the *Donnie Darko* poster. If you move the mouse inside the borders of this widget annotation, the popup with the text annotation will appear; if you move the mouse pointer outside the widget annotation, the popup will disappear.

In this example, you're creating button fields with a label and an icon. You can use an `Image` object for the icon ❶, or a `PdfTemplate` ❶ (in this case, you're using an imported page). I've used this functionality in a real-world project to create online examinations. Every question had a button that allowed the student to get a hint. If that button was clicked, an annotation was made visible and a hidden field was set. The value of this hidden field was posted together with the answers, so that the tutor could see for which questions a hint was used.

Some very simple JavaScript is used to hide (or reveal) the fields (or annotations) ❷. You get a field instance with the `getField()` JavaScript method for interactive fields, or with `getAnnot()` for ordinary annotations. Then you change the properties of these objects as explained in the JavaScript reference. In this example, clicking the upper button (named `click`) hides both buttons. Clicking the lower button (named `advertisement`) opens the web page dedicated to this book at Manning.com.

Pushbuttons aren't always meant to be pushed (or clicked). In the next example, we'll use pushbuttons as "hot areas" that trigger an action when the mouse moves over them.

Figure 4-2. Parsing Process in PDF document

4.3.2 Parsing

After selecting documents for the course, different parsing algorithms (for each specific file format) are applied to extract the structure information and the raw text and start building the abstract representations of each original document. For HTML documents, a java process with Jsoup library is applied. This process has methods to analyze HTML content and obtain a tree structure from it, where each text section is a node. The methods expect that each chapter of a book is in a different file and that sections and subsections are labeled with specific CSS classes. For pdf documents, Java iText library is applied, to implement methods that can read the PDF sections. This library has methods that allow extracting the structure, take into consideration the style attributes (alignment, bold, italics, bullets, indentation and underlining), table of contents, and indexes present in the documents. Figure 4-2 presents how PDF parser can extract the document's structure from the style's parameters of the text. As a final product, the parser divides the document into hierarchical parts (chapters, sections, subsections).

4.3.3 Text Pre-processing

For each section of the document, the following customary text mining techniques are applied for pre-processing, to transform raw data into an understandable format for NLP models.

- Tokenization: This process breaks a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. The list of tokens becomes an input for further processing.
- Word Stemming: This process reduces the inflectional forms of each word into a common base or root. Stemmers are typically easier to implement and run faster, and the reduced accuracy may not matter for some applications.
- Stop Words: First words that are not useful (stopwords) are eliminated, then the text is tokenized and stemmed. Frequent words in textbooks and tutorials such as "exercises", "examples", "solutions", are also removed. Additionally, the domain expert can intervene to eliminate other words that could confuse the rest of the process.

4.3.4 Topic Modeling

Once the text is clean, the Latent Dirichlet Allocation (LDA) statistical modeling tool [71] is applied to each section of the document in order to determine the most relevant topics for that section. LDA generates topics based on word frequency from a set of documents and it is particularly useful for finding reasonably accurate mixtures of topics within a given document set. After the model has been built, each document is represented as a probability vector over all the topics, and each topic as a probability vector over all the words in the vocabulary. Documents are discriminated based on different concentrations of topic probabilities.

4.3.5 Hierarchical Analysis

Parallel to the topic modeling, the hierarchical structure extracted from the documents (chapters, sections, and subsections) is used to establish the first relationship between domain concepts. These concepts are extracted from the title of the sections and subsections using keyphrase extraction [156] and Name Entity Recognizer (NER) [157] methods. A concept (class) is described by one or more relevant terms extracted in the titles of the chapters and subchapters. Each concept in the structure becomes a concept within the ontology, and the parent-child relationship (HasPart) between the elements becomes a relation in the ontology.

4.3.6 Mapping of Sections

Similarly, to what Guerra at al. [19] did to link book sections, all parts of the documents are mapped to similar ones in other documents, using as a metric the similarity between the text of the two parts. For example, the content of the first section of the first document could be linked to the third section of the second document if the similarity between the text in both sections is higher than a threshold. The similarity values were calculated by a semantic similarity algorithm is applied between all the parts of all documents. The result is a weighted list of links between different document parts and a list of topics words associated with each document part. It expected that similar parts describe the same concept. In this specific system, the cosine distance [158] was used to measure the semantic similarity between two document parts. The text of each part is represented as a multidimensional vector where the value for each dimension is the frequency of a given the word in that text. The metric is the cosine of the angle between these two vectors. A threshold for the minimum value of these metrics is set to establish a similarity between each pair of document parts.

4.3.7 Preliminary Ontology Processing

The main task in this step is to create simple and compound concepts in a preliminary ontology to model the domain's knowledge. This process takes the concepts identified in the title’s sections and sub-sections during the hierarchical analysis process and checks if those concepts also appear in the list of topics associated with their corresponding text content. If a concept is identified in the title and the corresponding text, that concept is added to the ontology.

4.3.8 Semantic Analysis

To obtain more and deeper relationships between concepts than the ones present in the hierarchical analysis, a semantic analysis is run over the text content of each document part. There are several approaches to determine the conceptual relationships between words, mainly based on the assumption that concepts that are semantically related, tend to appear near one another text [159]. For the semantic analysis and the relationship extraction, the system uses Part-Of-Speech Tagging (POST) [160] and open information extraction (open IE) [19] techniques that conduct a linguistic analysis of sentences and paragraphs terms, verbs and proper names. Figure 4-3 shows an example of semantic relationships in a sentence, using POST.

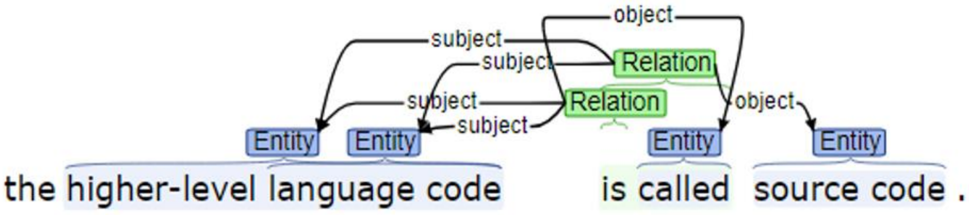


Figure 4-3. Part-of-speech tagging in a sentence

First, each sentence is divided into a set of related clauses. Each clause is reduced to its minimum, producing a set of shorter sentence fragments. These fragments are then segmented into OpenIE triplet, which is grouped and prioritized according to the concepts and terms identified in the preliminary ontology. ClausIE annotator extracts open-domain relations triples from sentences, these representing a subject, a relation, and the object of the associations.

For example, from the sentences extracted from the book “Learn to Program with Python”:

- “the interpreter translates the source code into the target machine language”
- “the interpreted machine language code is called the target code”
- It was possible generated the following triplets:
- (interpreter;translate;python code)
- (interpreter;translate;machine language code)
- (machine language code;be call;target code

The Figure No. 4-4 shows the graph between terms in structured relation triples.

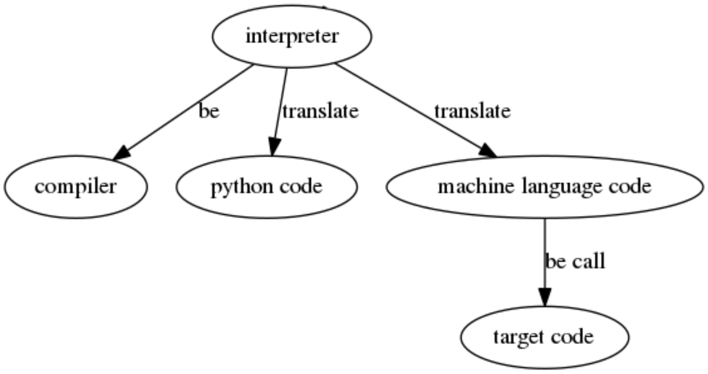


Figure 4-4. Relationships graph from triples

4.3.9 Definitive Ontology Processing

The preliminary ontology results are associated with the relationships detected in the semantic analysis process. The preliminary ontology becomes a graph structure with hierarchical levels. This new ontology is represented with a graph in which nodes are relevant concepts that belongs to the domain of interest, and edges are relations between the concepts.

This process also identifies possible problems in the ontology structure, such as the overgrowth depth of concepts in the graph and the existence of repeated terms. Finally, the definitive graph will be used to generate a domain ontology in standard Web Ontology Language (OWL) format, and document parts are converted into HTML content. This output is the normal result of any ontology building process.

4.4 System Evaluation

To evaluate the technical aspects of the proposed automatic extraction process, as well to measure the quality of the resulting ontologies, the system was applied in the first step to two different courses: Programming Fundamentals and Digital Circuit Design and next with some changes in settings, to computer organization course.

4.4.1 Exemplary Ontology Extraction

First, for each course, a professor that teaches the course was asked to select two textbooks that they use in the course, or they know to cover the content of the course. For Programming Fundamentals, the professors selected "Learn to Program with Python", and "Think Python". For Digital Circuit Design, the recommended books were "Fundamentals of

Digital Logic with Verilog Design" and "Digital Design Principles and Practices." Second, the parsing algorithms were applied to the digital version of the books to obtain several text files corresponding to chapters, sections, and subsections for each document. The subsections are considered as leaf nodes and contain raw text (the book content).

Then to prepare the data and create the documentary corpus, the text preprocessing routines were run over the individual document parts. Using Java Mallet library [1], multiples executions were run to obtain the LDA topic models. For setting the parameter K, the number of topics to extract by the LDA algorithm, the midpoint between the average number of sections and the number of subsections of each book (n = 120 to 160) was used. The number of iterations of the LDA algorithm was set from 1200 to 1500. Initially, the model was performed using the information in the subchapters, that is, the child nodes. After this model is built, a version of the collection is created, in which each chapter and section contains text content of its children nodes. Then, each aggregated document is incorporated into the built LDA model to obtain its new topic distribution.

Book1	Title	Book 2	Title	Sim.
book1	Learning to Program Python	book2	Think Python	0.71
book1_1	The Context of Software Development	book2_2_1	The Python programming language	0.73
book1_1_1	Software	book2_2_3	What is debugging?	0.22
book1_1_2	Development Tools	book2_2_1	The Python programming language	0.75
book1_1_3	Learning Programming with Python	book2_2_1	The Python programming language	0.48
book1_1_4	Writing a Python Program	book2_15_1	Persistence	0.57
book1_1_5	A Longer Python program	book2_15_1	Persistence	0.57
book1_2	Values and Variables	book2_3_5	Expressions and statements	0.66
book1_2_1	Integer Values	book2_3_1	Values and types	0.66
book1_2_2	Variables and Assignment	book2_3_2	Variables	0.54
book1_2_3	Identifiers	book2_3_3	Variable names and keywords	0.86
book1_2_4	Floating-point Types	book2_7_8	Checking types	0.30
book1_2_5	Control Codes within Strings	book2_2_5	The first program	0.29
book1_2_6	User Input	book2_6_11	Keyboard input	0.85
book1_2_7	The eval Function	book2_6_11	Keyboard input	0.29
book1_2_8	Controlling the print Function	book2_2_5	The first program	0.56
book1_3	Expressions and Arithmetic	book2_3	Variables, expressions and statements	0.54
book1_3_1	Expressions	book2_6_1	Modulus operator	0.66
book1_3_2	Operator Precedence and Associativity	book2_3_7	Order of operations	0.85
book1_3_3	Comments	book2_3_9	Comments	0.92

[1] Java Mallet text processing library - http://mallet.cs.umass.edu/

book1_3_4	Errors	book2_4_12	Why functions?	0.28
book1_3_4_1	Syntax Errors	book2_4_4	Composition	0.34
book1_3_4_2	Run-time Errors	book2_6_10	Infinite recursion	0.22
book1_3_4_3	Logic Errors	book2_4_12	Why functions?	0.22
book1_3_5	Arithmetic Examples	book2_4_4	Composition	0.32
book1_3_6	More Arithmetic Operators	book2_3_5	Expressions and statements	0.82
book1_3_7	Algorithms	book2_3_5	Expressions and statements	0.36

Table 4-1 Similarity between text subsections of two programming fundamentals books.

The number of topics words to be extracted is a configurable parameter of the topic extraction algorithm. For this demonstrative run, values between two and fifteen were used. The size of the final ontology varied according to this value. A smaller number of topic words resulted in a smaller number of entities and relationships. To obtain a value between 60 and 90 concepts in the final ontology (as suggested usually for course-based ontologies), a value of 6 was finally selected.

The similarity between different parts of documents was calculated to generate links between those with the highest similarity. As a result, a ranked list of links between documents parts was obtained, additionally to the topics associated with each link. Tables 4-1 and 4-2 show examples of the similarity detected between the sections of two Programming Fundamentals and two digital circuits design books, respectively.

book_docid1	title1	book_docid2	title2	Sim
book1_5_1	Digital Hardware	book2_5_8	Application-Specific ICs	0.67
book1_5_1_1	Standard Chips	book2_7_2	Logic Families	0.48
book1_5_1_2	Programmable Logic Devices	book2_5_8	Application-Specific ICs	0.95
book1_5_1_3	Custom-Designed Chips	book2_5_8	Application-Specific ICs	0.85
book1_5_2	The Design Process	book2_9_1_3	HDL-Based Design Flow	0.52
book1_5_3	Structure of a Computer	book2_9_1_1	Why HDLs?	0.52
book1_5_5	Digital Representation of Information	book2_6_5_7	Excess Representations	0.76
book1_5_5_1	Binary Numbers	book2_6_5_1	Signed-Magnitude Representation	0.98
book1_5_5_2	Conversion between Decimal and Binary Systems	book2_6_2	Octal and Hexadecimal Numbers	0.78
book1_6_6	Synthesis Using AND, OR, and NOT Gates	book2_8_3_3	Combinational-Circuit Minimization	0.41
book1_6_7	NAND and NOR Logic Networks	book2_7_3_6	Noninverting Gates	0.82
book1_6_8_1	Three-Way Light Control	book2_12_2_3	The Simplest Switch Debouncer	0.67
book1_6_8_2	Multiplexer Circuit	book2_5_3	Digital Devices	0.52
book1_7_1	Positional Number Representation	book2_6_5	Representation of Negative Numbers	0.98

book1_7_1_1	Unsigned Integers	book2_6_5_1	Signed-Magnitude Representation	0.96
book1_7_1_2	Octal and Hexadecimal Representations	book2_6_5_1	Signed-Magnitude Representation	0.98
book1_7_2	Addition of Unsigned Numbers	book2_10_10	Adders, Subtractors, and ALUs	0.73
book1_7_2_1	Decomposed Full-Adder	book2_7_3_6	Noninverting Gates	0.58
book1_7_2_2	Ripple-Carry Adder	book2_10_10_1	Half Adders and Full Adders	0.86
book1_7_3	Signed Numbers	book2_6_5_5	Diminished Radix-Complement Representation	0.95
book1_7_3_1	Negative Numbers	book2_6_5_3	Radix-Complement Representation	0.97
book1_7_3_2	Addition and Subtraction	book2_6_7	Ones'-Complement Addition and Subtraction	0.90
book1_7_3_3	Adder and Subtractor Unit	book2_6_7	Ones'-Complement Addition and Subtraction	0.91
book1_7_3_5	Arithmetic Overflow	book2_6_8	Binary Multiplication	0.83
book1_7_3_6	Performance Issues	book2_10_2_2	Propagation Delay	0.88

Table 4-2 Similarity between text subsections of two Digital Circuit Design books.

The hierarchical structure was constructed, taking the concepts which belong to titles and subtitles of the books sections that have the highest similarity and shared the same topics. Only concepts that were present in both books were selected. During this process, only the topic words and entities detected in the titles of the chapters and sub-chapters were used, without considering the text information of all subsections. After this process, each term becomes a concept within the ontology, and the parent-child relation between terms becomes a relation in the ontology. A similarity cut-off values of 0.54 and 0,48 were selected for the Programming Fundamentals and Digital Circuit Design ontologies.

The figure No 4-5 shows a preliminary ontology for Programing Fundamentals discipline.

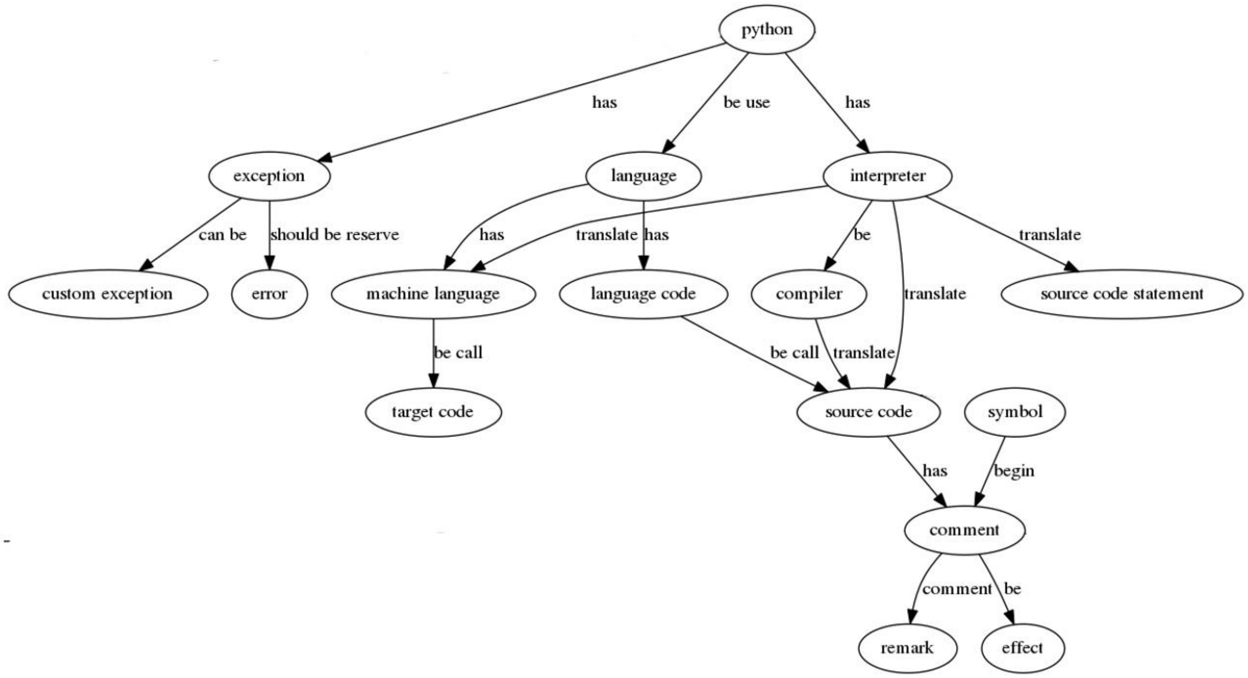


Figure 4-5. Programming Fundamentals preliminary ontology

For each of the document parts text, the Textrank library[2] was used to summarize the text, in order to find the most relevant sentences. An internal graph is constructed where the vertices represent each sentence in a document. The edges between sentences are based on the content overlap, which is the number of words that two sentences have in common. Table No. 4-3 shows the most relevant sentences extracted from two Fundamental Programming book sections.

Title Section	Sentences
Development Tools	the higher-level language code is called source code.
	the interpreted machine language code is called the target code.
	the interpreter translates the source code into the target machine language.
Expressions and statements	An expression is a combination of values, variables, and operators.
	A statement is a unit of code that the Python interpreter can execute.

Table 4-3. Identified relevant sentences from book sections

[2] Textrank text summarization library - https://nlpforhackers.io/textrank-textsummarization/

Then, during the semantic analysis process, ClausIE[3] was applied to the summarized sentences to extract relations between terms. Finally, the triples extraction process identified hundreds of triples approximately from the main sentences from the documents, but only selected those that had words present in the topic list associated with each document parts. Tables 4-4 and 4-5 show samples of triples lists for fundamental programming and digital circuit design courses.

Object	Relation	Subject
python	will display	program error message
function	require	argument
argument	be pass	value
statement	continue	iteration
program	be call	integrality constraint
set	support	operation
operation	be	search
control flow statement	cause	iteration
tuple	have	composability
runtime error	be called	program
statement	be execute	order
iteration	be use	expression
program	do	iteration
function pow	take	argument
error	be produce	runtime system
word	return	string
pass object	be	Pointer
readlines method	read	line list

Table 4-4. Triple list for fundamental programming content

Object	Relation	Subject
combinational-circuituilding	block	subcircuit multiplexer
input	have	output
decoder	evaluate	input
delay	be cause	adder circuit
circuit	be	gate delay
counter	be	clock
circuit	be	state
propagation delay	would be	circuit timing
decoder	have	input code
decoder	have	output code
state table	have	adjacency diagram

[3] ClausIE Open Information Extractor - https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/software/clausie/

hazard	be	state table
array element	can be	integer
hardware description language	signal	declaration
device	is	pld
address	provide	datum input
technology roadmap	forecast discuss	transistor
logic design principle	Chapter	example verilog
circuit	will operate	clock frequency
design technique	flip-flop	circuit

Table 4-5 Triple list for digital circuit design content

Then, to generate the definitive ontology, these terms were connected and added to the preliminary ontology. Figure 4-6 shows an extract from the final ontology for the Programming Fundamentals course. This ontology has 54 terms in a hierarchy in its first three levels and 61 terms in the first fourth levels.

For the Digital Circuit Design course, the ontology has 57 terms in the first three levels and 68 in the first four levels. An excerpt of this ontology as shown in Figure 4-7.

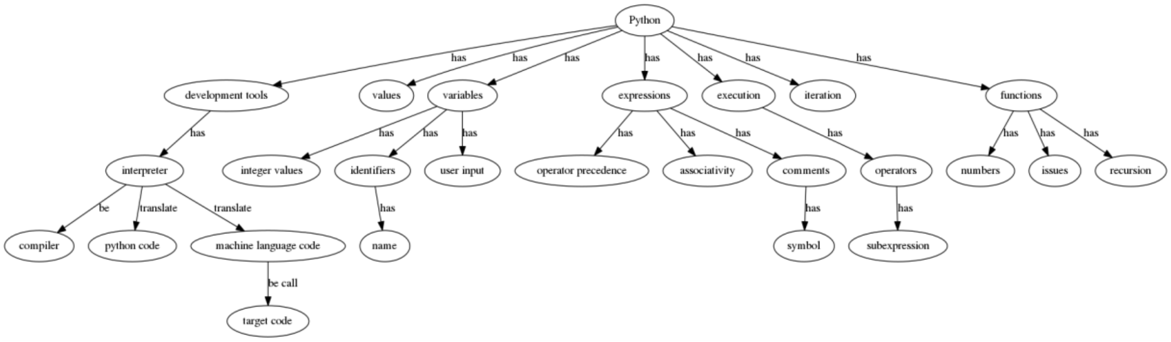


Figure 4-6. Programming Fundamentals automatically generated ontology (excerpt)

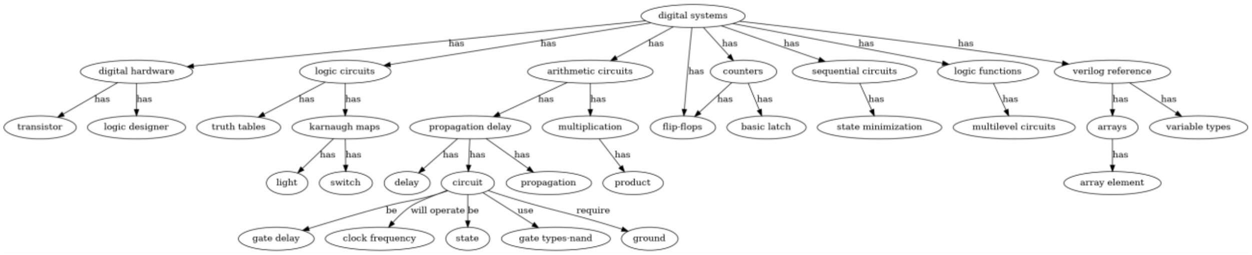
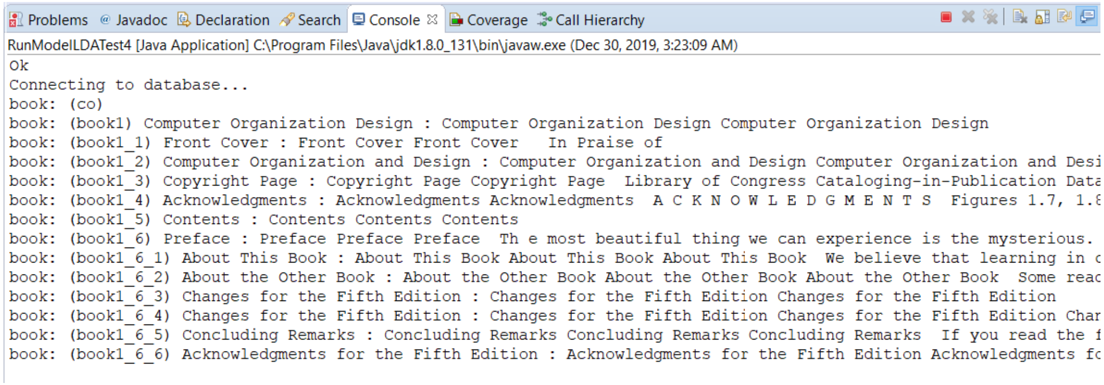


Figure 4-7. Digital design circuit automatically generated ontology (excerpt)

4.4.2 Other Ontologies.

Next, some adjustments were made in the configuration of the number of iterations of the LDA model and the selection of similarity algorithms during the generation of an ontology for the Computer Organization course. The experts were consulted and selected the following books: Computer Organization and Design: The Hardware/Software Interface (Fifth Edition.) and Computer architecture, a quantitative approach (Sixth Edition).

Both digital books in pdf format were the entry of the parsing process to obtain the corpus documental and hierarchical structure from sections and subsections; then, the LDA model was applied to the preprocessed document's collections to obtain topic probabilities. The figure 4-8 shows the java parsing process execution over computer organization digital books.



```
RunModelLDATest4 [Java Application] C:\Program Files\Java\jdk1.8.0_131\bin\javaw.exe (Dec 30, 2019, 3:23:09 AM)
Ok
Connecting to database...
book:  (co)
book:  (book1) Computer Organization Design : Computer Organization Design Computer Organization Design
book:  (book1_1) Front Cover : Front Cover Front Cover   In Praise of
book:  (book1_2) Computer Organization and Design : Computer Organization and Design Computer Organization and Desi
book:  (book1_3) Copyright Page : Copyright Page Copyright Page  Library of Congress Cataloging-in-Publication Data
book:  (book1_4) Acknowledgments : Acknowledgments Acknowledgments  A C K N O W L E D G M E N T S  Figures 1.7, 1.8
book:  (book1_5) Contents : Contents Contents Contents
book:  (book1_6) Preface : Preface Preface Preface  Th e most beautiful thing we can experience is the mysterious.
book:  (book1_6_1) About This Book : About This Book About This Book  We believe that learning in c
book:  (book1_6_2) About the Other Book : About the Other Book About the Other Book  Some reac
book:  (book1_6_3) Changes for the Fifth Edition : Changes for the Fifth Edition Changes for the Fifth Edition
book:  (book1_6_4) Changes for the Fifth Edition : Changes for the Fifth Edition Changes for the Fifth Edition Char
book:  (book1_6_5) Concluding Remarks : Concluding Remarks Concluding Remarks Concluding Remarks  If you read the f
book:  (book1_6_6) Acknowledgments for the Fifth Edition : Acknowledgments for the Fifth Edition Acknowledgments fc
```

Figure 4-8. Parsing Process Execution

LDA setup depends on the number of topics, the number of sampling iterations, the smoothing over document-topic distribution hyper-parameter α, and the smoothing over topic-word distribution hyper-parameter β. The number of iterations was set in 1000, considering the size of the documents and the collections. LDA hyper-parameters were set up in initial values of α = 0.1 and β = 0.01, and then used the fixed-point optimization for hyper-parameters. For the parameter K, the value of 125 was selected, which is equivalent to the midpoint of subsections number of each book.

Then, during the sections mapping process, for every part of the first book, each part of the second book is ranked according to the similarity of their content sections and subsections. As a similarity measure, the reciprocal symmetric KL divergence was used instead of the cosine similarity. Table 4-6 shows a sample of the similarity values detected between sections and subsections of both organization computer books.

Book_id1	Book_1 title	book_id2	Book_2 title	Sim
book1_7	Computer Abstractions and Technology	book2_9_2	Contributors to Previous Editions	0.68
book1_7_1_1	Classes of Computing Applications and Their Characteristics	book2_8_1	Why We Wrote This Book	0.65
book1_7_2_1	Design for Moore's Law	book2_10_9_4	Amdahl's Law	0.70
book1_7_2_2	Use Abstraction to Simplify Design	book2_18_2_1	Performance of Pipelines with Stalls	0.54
book1_7_2_3	Make the Common Case Fast	book2_8_7	Case Studies with Exercises	0.84
book1_7_2_6	Performance via Prediction	book2_12_2_2	Summary of the Loop Unrolling and Scheduling	0.48
book1_7_2_7	Hierarchy of Memories	book2_11_5	2.5 Crosscutting Issues: The Design of Memory Hierarchies	0.52
book1_7_2_8	Dependability via Redundancy	book2_14_2_1	What Is Multiprocessor Cache Coherence?	0.48
book1_7_4_1	Through the Looking Glass	book2_10_3_2	Genuine Computer Architecture	0.53
book1_7_4_2	Touchscreen	book2_8_10	Concluding Remarks	0.54
book1_7_4_3	Opening the Box	book2_14_3_3	A Multiprogramming and OS Workload	0.48
book1_7_4_4	A Safe Place for Data	book2_16_3_1	Interpreting Memory Addresses	0.64
book1_7_4_5	Communicating with Other Computers	book2_11_3_2	DRAM Technology	0.63
book1_7_5	Technologies for Building Processors and Memory	book2_10_6_2	Cost of an Integrated Circuit	0.70
book1_7_6	Performance	book2_10_9_5	The Processor Performance Equation	0.59
book1_7_6_2	Measuring Performance	book2_18_2_1	Performance of Pipelines with Stalls	0.56
book1_7_6_3	CPU Performance and Its Factors	book2_17_1_1	Cache Performance Review	0.90
book1_7_6_4	Instruction Performance	book2_10_9_5	The Processor Performance Equation	0.67
book1_7_6_5	The Classic CPU Performance Equation	book2_10_9_5	The Processor Performance Equation	0.59
book1_7_7	The Power Wall	book2_10_5_2	Energy and Power within a Microprocessor	0.81
book1_7_9	Real Stuff: Benchmarking the Intel Core i7	book2_18_9	C.9 Concluding Remarks	0.64

Table 4-6 Similarity between text subsections of two Computer Organization books

Then it was taken the information generated during the mapping process between the computer organization book's sections and ordered the similarity values between sections of books in descending order. They have only selected the parts that had associated the same topics and had a similarity cut-off greater than 0.65.

Topics words detected in the titles of the chapters and subchapters are used, without considering the text information of all subsections. Each item in the index becomes a concept within a preliminary ontology. Next, in order to identify more relations between concepts, a triples extraction process based in Stanford Open Information Extraction (OpenIE) java library was applied to the main sentences selected from all subsections, to extract open-domain relation triples. Finally, a process connects these terms and their associations to concepts that belong to the preliminary ontology. It was validated that the terms of triples coincide with any of the topics word lists belonging to each section and subsections. The ontology has 52 terms in a hierarchy of three levels and 56 terms until the fourth level. An excerpt from this ontology, as shown in Figure 4-9.

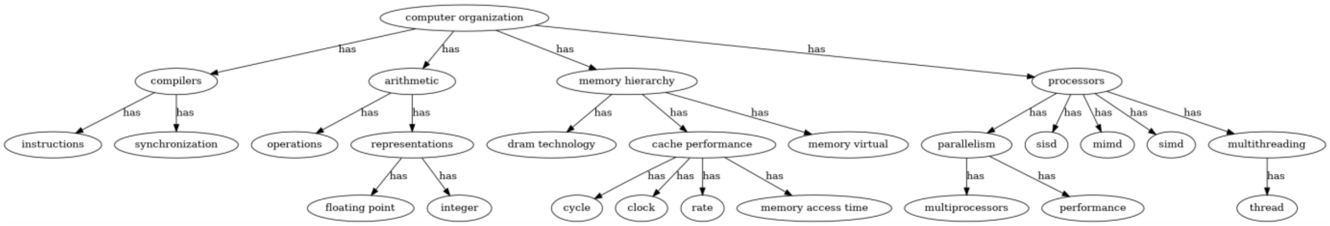


Figure 4-9 Computer Organization automatically generated ontology (excerpt)

4.4.3 Results and discussion

The graph structures formed by the concepts as child nodes and the parent-child relationship between them are processed to obtain the ontologies in OWL format, through the java libraries. This standard format for its flexibility and the ability to easily create subclasses and sub-properties based on entities in the core ontology and to search for information by queries in nodes is suitable for use by learning analytics applications. Figure 4-10 shows an OWL ontology for the programming fundamentals course.

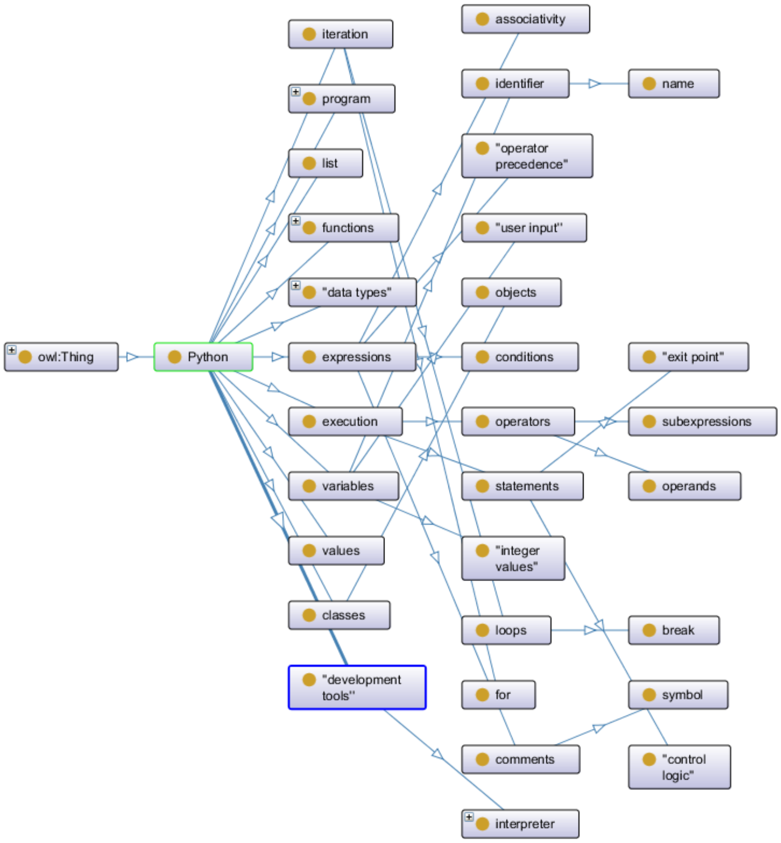


Figure 4-10. Programing Fundamentals Ontology in OWL Format

To evaluate the ontology quality, the same professors that recommended the books were asked to create an ontology manually. These manually generated ontologies were considered as the ground truth to evaluate the precision and recall of the automatically identified concepts. To compare the concepts between both ontologies, a process of stemming and similarity comparison was followed. The figure 4-11 shows programming fundamental manually generated ontology.

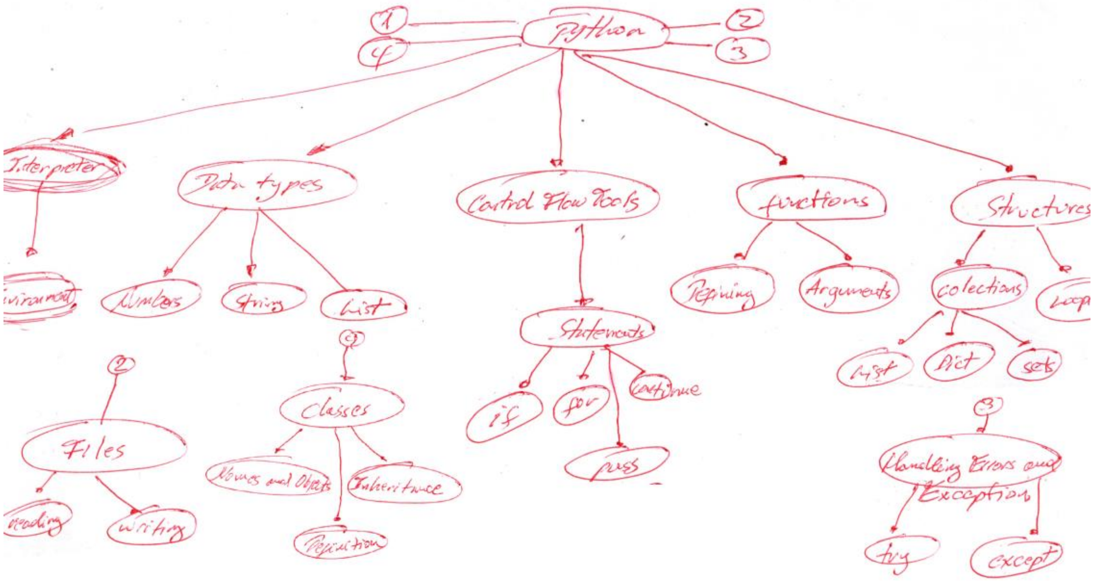


Figure 4-11. Programming fundamental manually generated ontology

The precision was calculated as the percentage of concepts that were in the automatically generated ontology that also were present in the manually generated ontology compared with the total number of concepts in the automatically generated ontology. The recall was calculated as the percentage of concepts of the manually generated ontology that also were present in the automatically generated ontology.

The results of precision and recall for the generated ontologies (Programming Fundamentals and Digital Design), in contrast with manually ontologies, show values of 72% and 67% for precision and 59% and 57% for recall, respectively. With the F-measure, a β lower than one gives more importance to precision, while a β higher than one gives more importance to recall. In our experiments, we used β =0 .5 to underline the importance of precision over recall. The results for the F-measure were 69% and 58% for each course.

On the other hand, the results of precision and recall for computer organization course were 66% and 68%, respectively, and the F-Measure was 67%. Figure 4-12 shows the Computer Organization course manually generated ontology.

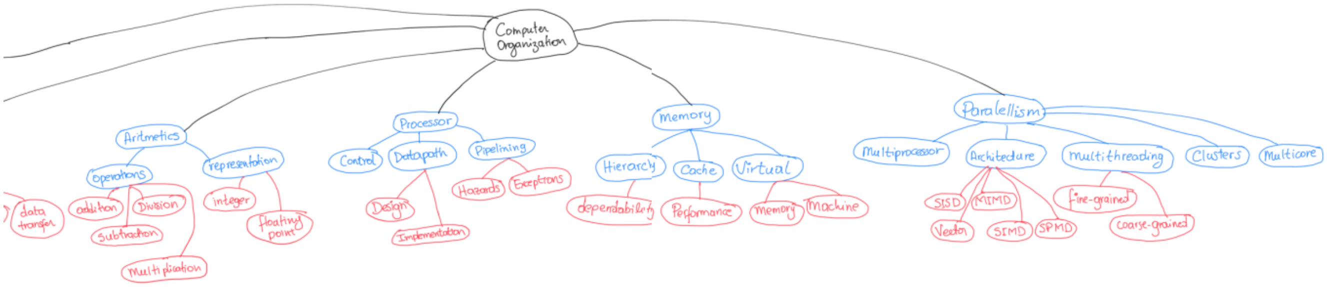


Figure 4-12. Computer Organization manually generated ontology

4.5 Conclusions

In this work, it was possible to build a framework to generate automatically and validate course-based ontologies, from digital learning resources like books and online tutorials. The proposed methodology can be applied to any discipline, regardless of the domain knowledge, as long as digital texts validated by experts are available.

In the evaluation of our methodology, two books were enough to create educational ontologies that, on average, were 70% precise and capture more than 50% of the concepts generated manually by an expert. Even the adjustment in similarity cut-off and the choice of the similarity method could influence a better automatic selection of concepts for the nodes of the ontology.

The methodology also presents several limitations that could be overcome with further work. The end-user could play with the different model parameters, especially for the LDA model, to obtain more inclusive ontologies, sacrificing some precision, or vice versa. Currently, these parameters are fixed at design time. Another limitation of this work is the lack of consideration of the links into the ontology quality. A graph-based similarity measure could be used to understand better the relationships between the automatic and manual generated ontologies.

Besides recommending educational material, these ontologies can also serve as a learning map for students who follow courses related to domain topics and want to know the most relevant concepts for study. Also, it could provide visual feedback to students about their progress in monitored courses.

5 Curriculum design assistant based in educational ontologies

This chapter is based on the work described in the paper “Curriculum design assistant system based in automatically-generated educational ontologies.” This work describes the design and implementation of a functional assistant system to support instructors to design or re-design the curriculum of their courses. The system uses automatically generated ontologies from textbooks that reflect the structure of a specific body of knowledge.

5.1 Introduction

Previous works have already identified the importance of developing rich ontologies in the education field [161] . Ontologies, which form a conceptual skeleton of the modeled domain, could serve to diverse educational purposes such as better understanding, knowledge creation, knowledge sharing, and reusing, collaborative learning, intelligent tutorials, problem-solving, seeking advice, or developing competencies [14] [134].

Curriculum management and development can be improved using ontologies in curriculum tasks like aligning, comparing, and matching between universities, educational systems, or relevant disciplines. Following this approach, we built a functional assistant system for teachers that use automatically-generated course-based ontologies, explained in the previous chapter, to help teachers in the process of creating Intelligent Curricula. The application shows a preliminary ontology, and teachers could be able to improve it via additional material or direct manipulation. Next, with the information entered by teachers, a mapping process identified the most relevant concepts using terms available or not in nodes of the domain ontology to perform learning resources suggestions.

Educators could also evaluate the Intelligent Curricula quality, validating if the content of their documents was covered in the Computer Science curricula (CS2013) [162]. Thus, the system uses a multi-level classification model implemented through a combination of Machine Learning and Natural Language Processing techniques.

The main contribution of this work is to evaluate teachers' perception about the usefulness, easiness, engagement, and other aspects related to the domain ontologies offered by our proposed system. To achieve this, we prepared an experimental test focused on teachers who preferably have not taught the programming fundamentals course since the previous semester or more.

The main questions of the study were the following:

- RQ1: Can the generated ontologies be useful in an assistant tool to design courses curriculum?
- RQ2: What is the functionality that had the highest contribution during courses curriculum design?

5.2 Related Work

Conceptual model visualization methods such as ontologies are widely and effectively used in education, and many educational ontologies have been developed for several disciplines) and other related approaches such as collaborative learning and adaptive educational systems and Intelligent Curriculum design.

An intelligent curriculum can be defined as the representation of the domain knowledge usually taught in a specific course in a way that is amenable to be understood and processed by a computational system. The most common representation that fulfills this requirement is ontology.

Many education and learning analytics applications use Intelligent Curriculum represented as courses ontologies, to recommend learning materials [163] to automatically sequence learning activities [137], to evaluate the quality of contributions in online forums [139]. Also, to monitors the authoring process and prevents and solve inconsistencies [140] or to provide visual feedback to students about their progress [141].

This study looks for shedding light into the area of Intelligent Curriculum creation exploring the use of automatically-generated course ontologies for the creation and evaluation of Intelligent Curriculum in a few steps, through an assistant system to support course design. While the instructors can design curricula and add learning content, the system recommends a preliminary ontology, and the instructor is able to improve it through direct manipulation.

5.3 Methodology

Figure 5-1 depicts the phases followed by the curricula assistant system. The pipeline initiates with data collection from digital books and continues with processes to build the course's educational ontologies that will be used by the system's functionalities. Also, multilabel classification trained models are available to validate the content of the course's curricula against CS2013 knowledge units.

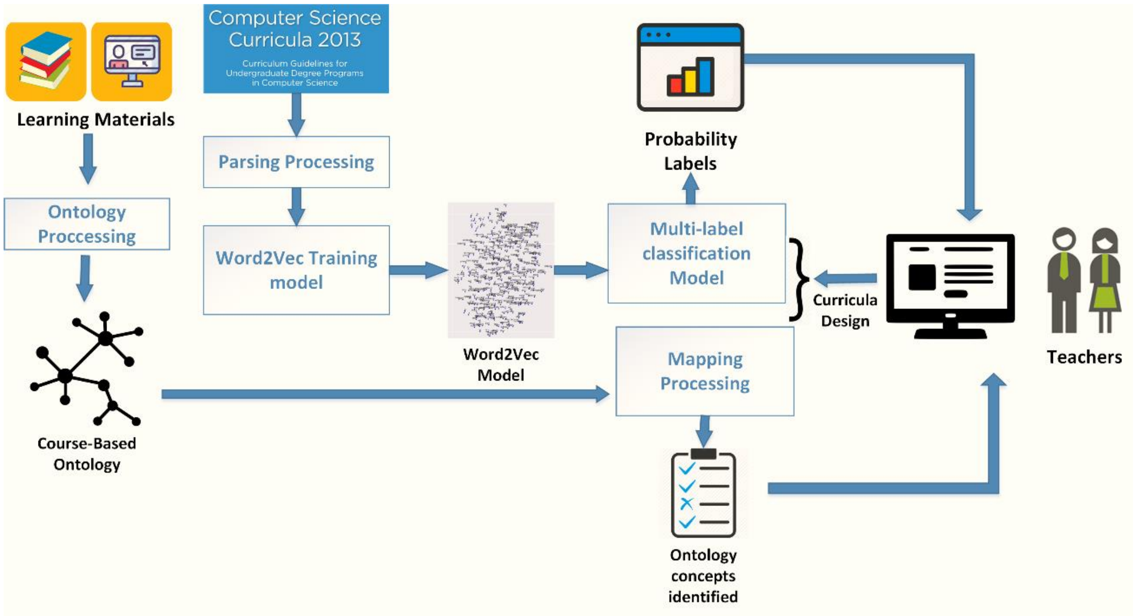


Figure 5-1. Curricula assistant system framework

The following subsections explain the functionalities of the system that use each of the mentioned processes.

5.3.1 Educational Ontology Processing

Important quality criteria for a resulting ontology are the number of relevant concepts and relations that describe the specific domain and the comprehension of its structure [164]. For this last point, the visualization of the ontology is important. For these reasons, we choose to generate ontologies with a maximum range between 50 to 100 concepts and a four-level hierarchy.

To obtain the educational ontologies, we focused on the previous approach for the semi-automatic generation from digital texts. We selected programming fundamentals digital books recommended by teachers and applied them to a parsing process to divides the documents into hierarchical parts (chapters, sections, subsections). Next, all parts of the documents are mapped to similar ones in other documents, using as a metric the similarity between the text of the two parts. Below, we used Latent Dirichlet Allocation (LDA) statistical model in order to determine the most relevant topics for each section. Then, a process connected word topics according to the semantic relations and hierarchical structure present in texts. The result was a course-based ontology, like the one in Figure 5-2, with 74 concepts for programming fundamentals course. Next, through libraries provided by Protégé editor [43], we configured it in OWL format so that it can be accessed by the system to identify the most relevant concepts available in domain ontology nodes and its hierarchical relations.

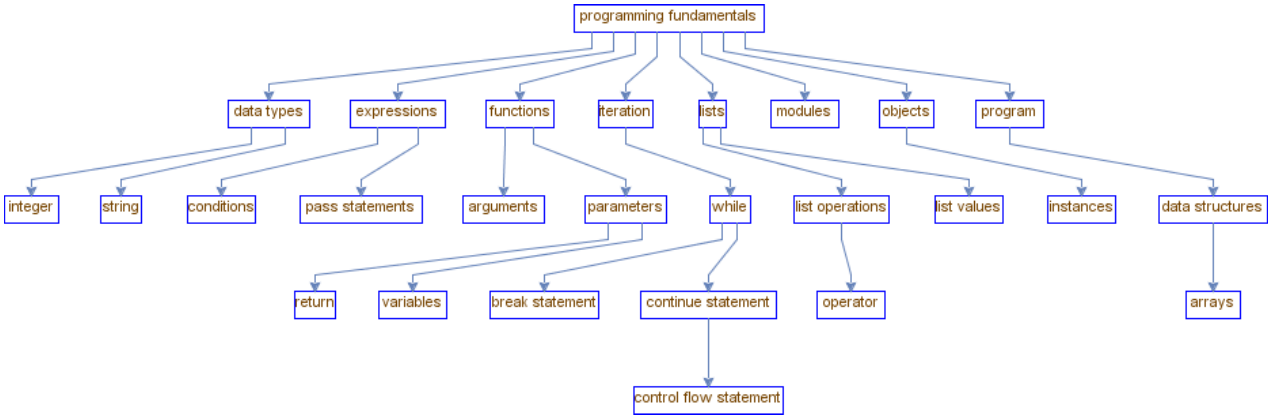
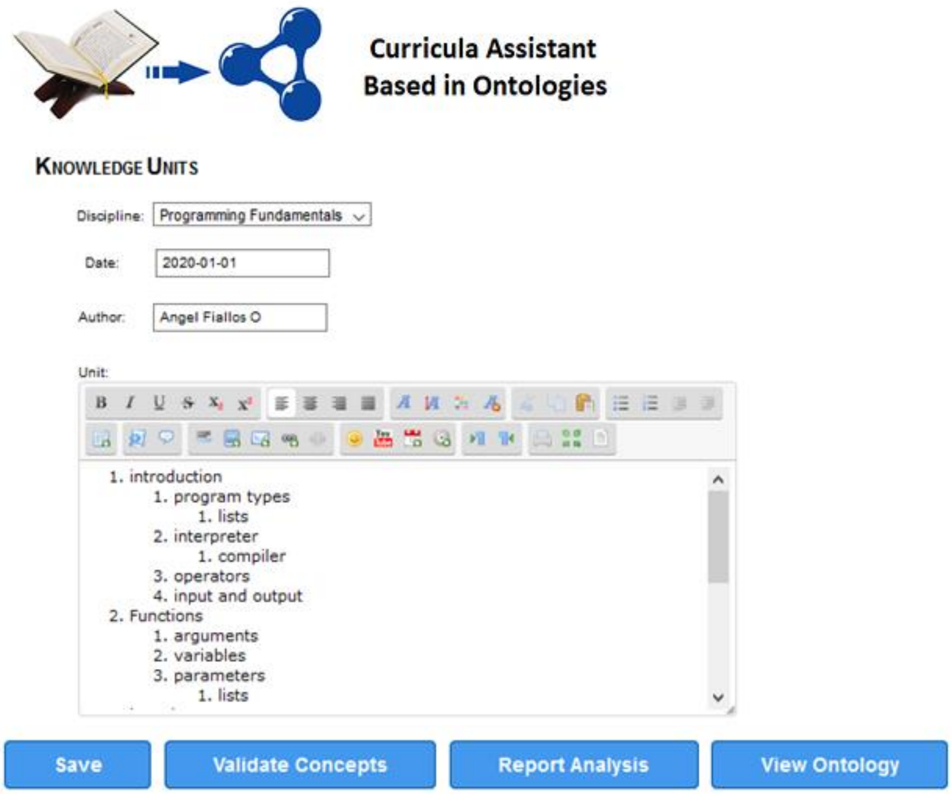


Figure 5-2 Programming Fundamentals Ontology (Excerpt)

5.3.2 Curricula assistant system.

We built and web-based system to enter, edit, and validate course curricula by teachers and tutors. The system accesses the course-based ontologies and other information related to the course content to perform guides and suggestions about the course's domain knowledge to teachers. We used Java language and Javascript to code the text editor and libraries, such as JGraphT [165]. The lightweight markup language BBCode [166] was selected because it is easy to read the text in its raw form, and it was possible to parse with Java JSOUP libraries.

Also, Stanford Core NLP libraries [79] were used for parsing algorithms and graph structures visualization.   Figure 5-3 shows an interface of the curriculum assistant system consisting of a list of pre-configured courses, a text editor for recording the course curriculum, and the functional options available to teachers and instructors.



Discipline: Programming Fundamentals

Date: 2020-01-01

Author: Angel Fiallos O

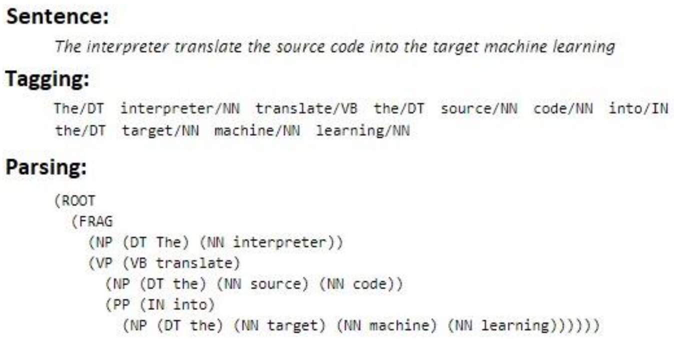
Unit:

1. introduction
    1. program types
        1. lists
    2. interpreter
        1. compiler
    3. operators
    4. input and output
2. Functions
    1. arguments
    2. variables
    3. parameters
        1. lists

Figure 5-3 Curricula Assistant System menu and editor

5.3.3 Curricula concepts validation

A parsing process was implemented to extract the text delimited by tags and to build abstract representations from course curricula. Other elements, such as bullet points, punctuation, and line breaks, were also used to identify relevant information into raw text. A linguistic analysis was then applied using Stanford NLP libraries and Part-of-speech tagging (POST) techniques to determine the nouns and key phrases available in each document and get a list of terms used by teachers. The figure 5-4 shows tagging and parsing processes in a text sentence.



Sentence:

The interpreter translate the source code into the target machine learning

Tagging:

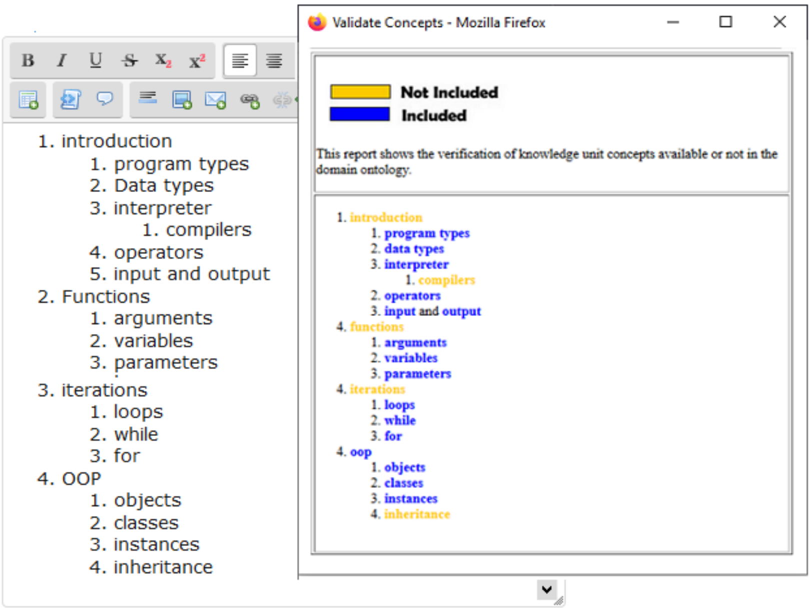
```
The/DT  interpreter/NN  translate/VB  the/DT  source/NN  code/NN  into/IN
the/DT  target/NN  machine/NN  learning/NN
```

Parsing:

```
(ROOT
  (FRAG
    (NP (DT The) (NN interpreter))
    (VP (VB translate)
      (NP (DT the) (NN source) (NN code))
      (PP (IN into)
        (NP (DT the) (NN target) (NN machine) (NN learning))))))
```

Figure 5-4. NLP tagging and parsing processes

The ontology fulfills the function of the domain model, i.e., its nodes represent the domain concepts, and the links determine the relationship between them. A mapping process implementation used these links to traverse the ontology and locate the nodes that were associated with specific concepts available in course curricula. Once the mapping is established, the identified and unidentified concepts and keywords are shown to users with a special markup to help them during the definitive selection of terms for curricula design. Figure 5-5 shows a capture of the validation curricula concepts module. Note the difference in colors, yellow for concepts not included in the ontology, and blue for concepts included.



```
1. introduction
      1. program types
      2. Data types
      3. interpreter
            1. compilers
      4. operators
      5. input and output
2. Functions
      1. arguments
      2. variables
      3. parameters
3. iterations
      1. loops
      2. while
      3. for
4. OOP
      1. objects
      2. classes
      3. instances
      4. inheritance
```

🟨 **Not Included**
🟦 **Included**

This report shows the verification of knowledge unit concepts available or not in the domain ontology.

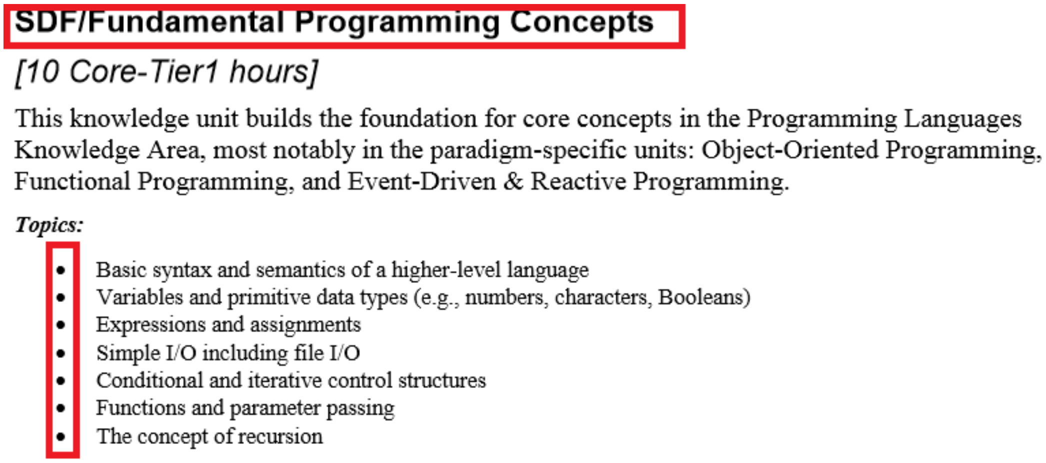
```
1. introduction
      1. program types
      2. data types
      3. interpreter
            1. compilers
      2. operators
      3. input and output
4. functions
      1. arguments
      2. variables
      3. parameters
4. iterations
      1. loops
      2. while
      3. for
4. oop
      1. objects
      2. classes
      3. instances
      4. inheritance
```

Figure 5-5. Validation Concepts Functionality

5.3.4 Report Analysis using multilabel classification

This functionality aims to assure the quality content of teachers’ course curricula through a multilabel classification model implemented using Convolutional neural networks (CNN) [103] libraries and pre-trained word vectors. First, we used semi-structured data from course knowledge units of the following courses: Programming fundamentals, architecture and organization, information management, programming languages, algorithms, operating systems, and artificial intelligence, which are available in CS-2013 digital documents.

Figure 5-6 shows a fundamental programming knowledge unit excerpt from CS-2013 digital edition.



SDF/Fundamental Programming Concepts

[10 Core-Tier1 hours]

This knowledge unit builds the foundation for core concepts in the Programming Languages Knowledge Area, most notably in the paradigm-specific units: Object-Oriented Programming, Functional Programming, and Event-Driven & Reactive Programming.

Topics:

- Basic syntax and semantics of a higher-level language
- Variables and primitive data types (e.g., numbers, characters, Booleans)
- Expressions and assignments
- Simple I/O including file I/O
- Conditional and iterative control structures
- Functions and parameter passing
- The concept of recursion

Figure 5-6. CS-2013 Fundamental programming Knowledge Unit excerpt

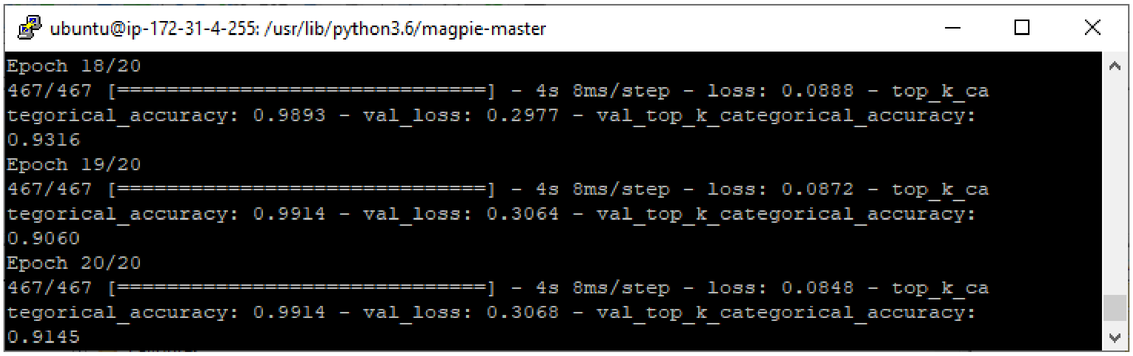
Then, to build the final corpus, a four-level hierarchical structure was designed. The sentences of the first, second, and third levels were obtained from the knowledge units of the selected courses of the CS-2013 curricular standard. The last level sentences were obtained from the Wikipedia pages, linked to the main concepts selected in the third level. The table 5-1 shows an example of sentences referring to the courses’ knowledge units of the computer science standard curricula CS-2013.

Category	Level	Sentences
Programming_Fundamentals	0	Fundamental Programming Concepts
Programming_Fundamentals	1	Variables and primitive data types
Programming_Fundamentals	1	The concept of recursion
Programming_Fundamentals	0	Fundamental Data Structures
Programming_Fundamentals	1	Arrays
Programming_Fundamentals	1	Strings and string processing
Programming_Fundamentals	1	Abstract data types and their implementation
Programming_Fundamentals	2	Stacks
Programming_Fundamentals	2	Queues
Programming_Fundamentals	2	Priority queues
Programming_Fundamentals	2	Sets
Programming_Fundamentals	2	Maps
Programming_Fundamentals	3	Queues are common in computer programs, where they are implemented as data structures coupled with access routines, as an abstract data structure.
Software_Engineering	0	Software Processes
Software_Engineering	1	Team participation. Roles and responsibilities in a software team
Software_Engineering	2	Team processes including responsibilities for tasks, meeting structure, and work schedule
Software_Engineering	2	Roles and responsibilities in a software team
Software_Engineering	2	Risks associated with virtual teams (communication, perception, structure)
Software_Engineering	3	Software requirements is a field within software engineering that deals with establishing the needs of stakeholders that are solved by software.
Artificial Intelligence	1	Basic Machine Learning
Artificial Intelligence	2	Definition and examples of broad variety of machine learning tasks, including classification
Artificial Intelligence	2	Simple statistical-based learning, such as Naive Bayesian Classifier, decision trees
Artificial Intelligence	3	Support vector machines (SVMs)
Artificial Intelligence	3	Learning decision trees,
Artificial Intelligence	4	Support vector machines (SVMs), are a set of related supervised learning methods used for classification and regression.

Table 5-1. Sentences from CS-2013 knowledge units

Following the Kim approach [113], we designed a simple CNN network composed for an input layer with five different ngrams window sizes and one layer of convolution on top of word vectors obtained from Word2Vec unsupervised neural language model [109]. These vectors representations are essentially featuring extractors that encode semantic features of words in their dimensions.

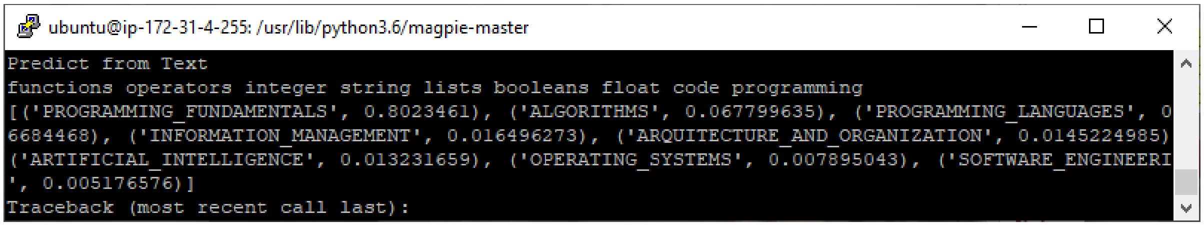
To run the experiment, first, we trained the CS2013 database using 100-dimensional word2vec embeddings. Next, we used Gensim library and Keras framework to build the convolutional neural network, using as parameters a ratio value of 0.2 and a count of epochs of 20. During the training, the model was configured to divide data into training and test datasets in a ratio of 80/20. The model evaluates itself after every epoch and adjusting parameters according to its loss function. The result is a set of parameters that have a particular ability to classify to new values, and the validation accuracy measured this ability. Fig 5-7 shows the capture of the model training process.



```
Epoch 18/20
467/467 [==============================] - 4s 8ms/step - loss: 0.0888 - top_k_ca
tegorical_accuracy: 0.9893 - val_loss: 0.2977 - val_top_k_categorical_accuracy:
0.9316
Epoch 19/20
467/467 [==============================] - 4s 8ms/step - loss: 0.0872 - top_k_ca
tegorical_accuracy: 0.9914 - val_loss: 0.3064 - val_top_k_categorical_accuracy:
0.9060
Epoch 20/20
467/467 [==============================] - 4s 8ms/step - loss: 0.0848 - top_k_ca
tegorical_accuracy: 0.9914 - val_loss: 0.3068 - val_top_k_categorical_accuracy:
0.9145
```

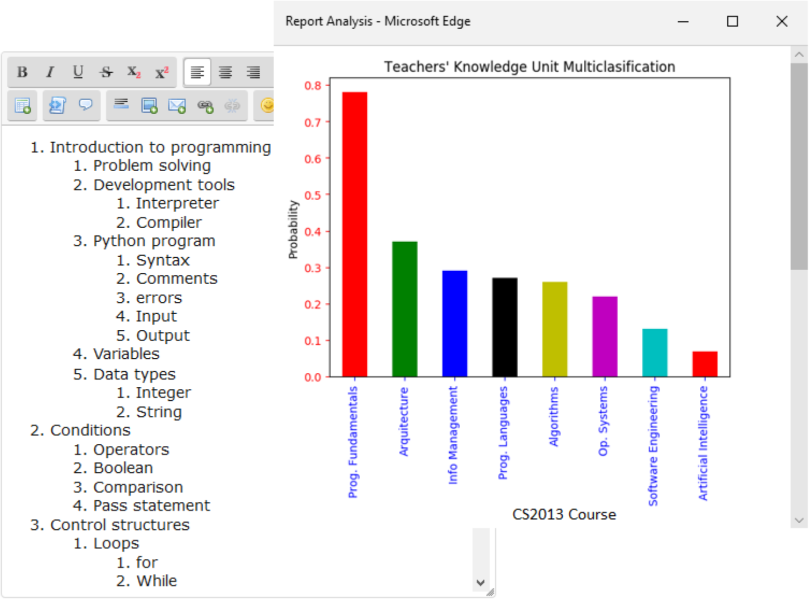
Figure 5-7. CNN Model Training Process

The networks try to predict 0 or 1 values on every label, and the model uses the confidence values to produce a ranking. In the end, we used a sigmoid activation function to treat the labels independently. Next, the multi-label text classification trained model was applied to concepts and key phrases identified using part of speech tagging over the Intelligent Curricula text generated by teachers from our web application. As a result, the model returns in JSON format a set of probabilities labels related to CS-2013 areas, similar to Figure 5-8, and the report analysis module takes these data and shows a bar chart. Figure 5-9 depicts an example in which the curriculum's content presents a probability greater than 70% of being labeled as Programming Fundamentals course within the standard curriculum CS-2013.



```
Predict from Text
functions operators integer string lists booleans float code programming
[('PROGRAMMING_FUNDAMENTALS', 0.8023461), ('ALGORITHMS', 0.067799635), ('PROGRAMMING_LANGUAGES', 0
6684468), ('INFORMATION_MANAGEMENT', 0.016496273), ('ARQUITECTURE_AND_ORGANIZATION', 0.0145224985)
('ARTIFICIAL_INTELLIGENCE', 0.013231659), ('OPERATING_SYSTEMS', 0.007895043), ('SOFTWARE_ENGINEERI
', 0.005176576)]
Traceback (most recent call last):
```

Figure 5-8. Multi-label text classification probabilities labels



1. Introduction to programming
    1. Problem solving
    2. Development tools
        1. Interpreter
        2. Compiler
    3. Python program
        1. Syntax
        2. Comments
        3. errors
        4. Input
        5. Output
    4. Variables
    5. Data types
        1. Integer
        2. String
2. Conditions
    1. Operators
    2. Boolean
    3. Comparison
    4. Pass statement
3. Control structures
    1. Loops
        1. for
        2. While

Figure 5-9. Report Analysis Functionality

5.4 Experimental Design

Next, we present the participants' profile and the experiment context carried out with the assistant system to try to answer the following research questions.

- RQ1: Can the generated ontologies be useful in an assistant tool to design courses curriculum?
- RQ2: What is the functionality that had the highest contribution during courses curriculum design?

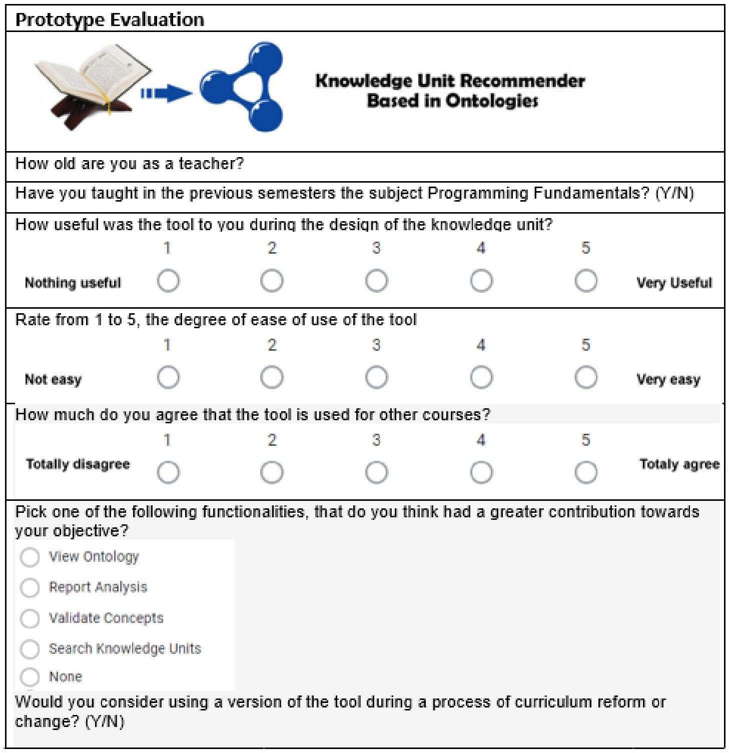
5.4.1 Participants

Twelve teachers belonging to School of Computer Science at ESPOL University were selected for the experimental test. Teachers were young lecturers (mean= 30 years old), 75% were male, 25% were female. The average teaching experience was 6.2 years, 25% of them have never taught the programming fundamentals course, 66.7% taught this course two semesters before the experiment, and the remaining 8,3% are teaching the course in the current semester.

5.4.2 Experimental Context

Teachers that participated in this study were instructed to design curricula for programming fundamentals course with the restriction that it had to be between 30 and 50 concepts in a hierarchical structure. Teachers had access to an online version of the curricula assistant system and its functionalities during the experiment. Also, each teacher was informed that the maximum time for the task was 30 minutes.

The teachers were also able to access the assistant system and a digital copy of programming books. They received a short description via mail of how to do the task. For instance, they were instructed that in the first iteration, they could use the option "Ontology View" to visualize a course-based programming fundamentals ontology as a content reference. Moreover, the explanation indicated that the remaining system options ("Validate Concepts" and "Report Analysis") could be enabled once the first version of their Intelligent Curricula was registered in the system. Next, a questionnaire similar to the one in Figure 5-10 with multiple-choice questions was implemented using Google forms to explore teachers' perceptions about system functionalities during the curricula design. The link for the questionnaire was sent to the participants once the experiment finished.



Prototype Evaluation

Knowledge Unit Recommender
Based in Ontologies

How old are you as a teacher?

Have you taught in the previous semesters the subject Programming Fundamentals? (Y/N)

How useful was the tool to you during the design of the knowledge unit?

	1	2	3	4	5	
Nothing useful	○	○	○	○	○	Very Useful

Rate from 1 to 5, the degree of ease of use of the tool

	1	2	3	4	5	
Not easy	○	○	○	○	○	Very easy

How much do you agree that the tool is used for other courses?

	1	2	3	4	5	
Totally disagree	○	○	○	○	○	Totaly agree

Pick one of the following functionalities, that do you think had a greater contribution towards your objective?

○ View Ontology
○ Report Analysis
○ Validate Concepts
○ Search Knowledge Units
○ None

Would you consider using a version of the tool during a process of curriculum reform or change? (Y/N)

Figure 5-10 Teachers evaluation questionnaire

For this analysis, data were divided into two parts. The first part is related to the first research question and included questions aimed at gauging teachers' perceptions. The second part is related to the second research question and corresponds to questions focused on teachers' opinions about the functionalities of the assistant system. Figure 7 shows screenshots of teachers using the assistant system during the experiment.

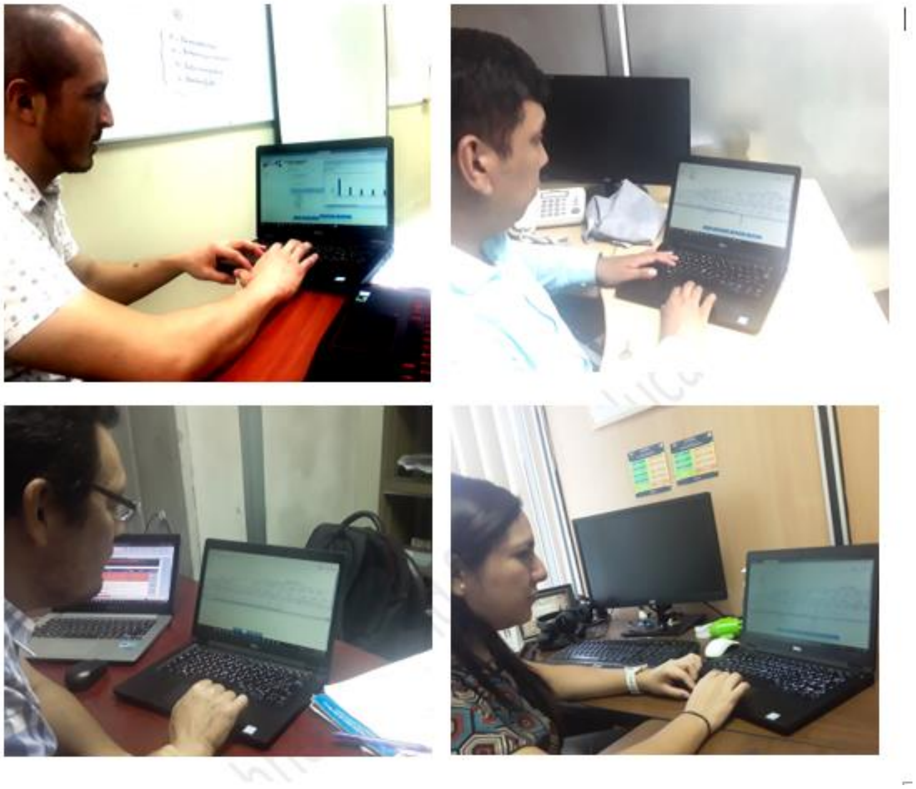


Figure 5-11. Teachers using the curricula assistant system

We measure the perception of the teachers about usefulness, easiness, and recommendation about the usage of the tool for other courses, with a five-point Likert scale (1 equals strongly disagree, and 5 equals strongly agree).

5.5 Results and discussion

To answer the research questions posed above, we present first the descriptive statistics of the usefulness, easiness, and recommendations about the usage of the tools. This information is presented in Table No. 5-2.

Question Objective	N	Mean	Std Deviation	Variance	Skewness
Usefulness of tool	12	3.9	0.70	0.491	0.123
Easiness of tool	12	3.8	0.75	0.564	0.329
Agreement of use the tool to other courses	12	4.3	0.67	0.455	-0.593

Table 5-2 Descriptive Statistic: Usefulness, Easiness, Recommendation of usage

As can be seen from Table 5-2, the system is perceived as useful for curriculum design (mean= 3.9). As for the easiness of the tool, the results were similar to the usefulness (mean= 3.8). It indicates that most teachers perceived that the tool and its functionalities are relatively easy to use. Teachers indicated they would recommend to others the use of the tool (mean=4.3). These values allow us to answer question RQ1, stating that knowledge ontologies are useful for the design of course curricula, from the point of view of teachers.

The second part of the questionnaire contained questions related to the functionality of the system. The first question aims at investigating what system functionality had a more significant contribution to teachers during the curricula design task. The results were the following: "View ontology" with 55%, "Report Analysis" with 36%, and "Validation concepts" with 9%. These results denote that the ontology is the functionality that most helped teachers during the test. These results let us answer RQ2, stating that the functionality system that contributes the most to the curriculum design is the view of the ontology.

The second question aims at investigating if teachers would use the tool during a curriculum reform. The results were positive, with 91% of teachers indicating they would use this tool for this activity.

Moreover, the participants made some comments about their experience in the test. Most were proposed improvements to the curricula assistant system, such as improving the editing process, the visualization of ontologies, and information structure. Table 5-3 shows the most relevant comments by teachers about their experience with the curricula assistant system.

Question Objective	**Comment**
Usefulness of tool	It would seem like a good idea that the ontology can be grouped and opened so that the user when seeing the information, does not feel stressed from seeing so much text.
	The system could suggest the list of topics, and the teacher chooses the subset of topics that he wants to include and the order in which he wishes to dictate them.
Easiness of tool	I would like the information presented by the ontology can be dragged towards the system editor, preserving its hierarchy.
	A text editor with formatting tools may not be the best alternative, but a smaller interface where content is entered, and then it indicates that parts of the document are nested within the text.
Usage of the tool in other courses	The tool speeds up the design of a syllabus and offers immediate feedback on topics that have not been included.
	It seems to me that it should be used in other courses as support for the creation of syllabus. The presented ontology serves as a guide for the creation of knowledge units.

Table 5-3. Teachers most relevant comments about curricula assistant system

Finally, a precision measure was calculated through as the percentage of concepts that were in each teacher curricula content, that also were included in the CS2013' programming fundamentals knowledge units, available in digital documents. The average precision for all teachers' curricula courses was 73%.

The idea of using visual structuring of information to improve the quality of understanding and mentalization among research colleagues is not new. Ontologies are useful structuring tools and provide an organizing axis along which teachers or students can mentally mark their vision in the information hyper-space of domain knowledge. These investigations could explain because most participants pointed out that the ontology was the system functionality that most helped them during tests.

This study is not without limitations. It was conducted on a limited number of participants, and only one-course ontology was tested in the assistant system for experiments; however, in general, the tool had proper usefulness and easiness evaluation. User comments highlight the few steps that must be taken to achieve the objectives but also request improvements in flexibility for the use of the ontology.

5.6 Conclusions

This study investigated how useful educational ontologies can be for the generation of course curricula. The results indicated that teachers believed which the system helped them to develop the curriculum for programming fundamentals course, and the functionality that contributed most to the design of it was the preliminary visualization of the domain ontology. The average precision of the curricula courses gives an insight into the quality of the content created by each tutor with the help of the assistant system tools.

These results are significant because by helping teachers to improve the curricular content of courses, with the support of ontologies based on digital books recommended by experts and content automatic validations against curricular standards such as CS2013, students can get better educational material at the time of development of the courses.

As future work, we envision the use and evaluation of educational ontologies on the curricula design of more courses and other learning analytics applications that consume educational ontologies.

6 Ontologies generation from unstructured data

This chapter is based on the work described in the paper “What do students say about their universities? Generation of ontologies from users posts content in social networks.”, This work defines an approach to build and test ontologies from non-structured data, such as posts in social networks related to specific topics, through datamining, machine learning, and natural language processing techniques. The results obtained suggest that the methodology can be used to provide information from unstructured data, and can also be used in analysis, understanding, and decision-making, as well as for the use of other applications in different fields.

6.1 Introduction

The quantity of textual data presented in social media, such as online forums, blogs, and social networks posts, is highly dynamic and involves interaction among various participants. There is a massive amount of text continuously generated by users in informal environments. Standard data mining techniques do not have enough resources to evaluate and understand unstructured data; therefore, linguistic techniques are necessary. Text mining and Natural Language Processes are the most promising avenues for social media data processing and for providing methods and algorithms that extract meaningful information from a large volume of data from multiple sources and languages [167]. Social media is essential because social networks have made everybody a potential author, so the language is now closer to the user than to any prescribed norms [168]. In this way, students share information about events, activities, services, opinions, and experiences at the university on social media channels [169]. This information can be used for monitoring, feedback, and to recommend actions for improvements by managers and tutors. On the other hand, ontologies are the best way to share a common understanding of information structure among people or software agents [9] [10]. These are representing a set of concepts within a domain and the relationships between those concepts. Thus, they can be considered the explicit and abstract model representation of finite sets of terms and concepts already defined [21].

We propose a methodology for ontologies generation that can express a basic understanding of the domain in unstructured data related to universities from diverse social networks. These ontologies can be used for analysis and comprehension about students’ interests, problems, behaviors, opinions, and preferences about universities and academic issues, in a simple, formal, and explicit way.

For this study, we collected social networks data that refers to the following Ecuadorian universities: Universidad Catolica Santiago de Guayaquil, Escuela Superior Politecnica del Litoral, Universidad de Especialidades Espiritu Santo, Universidad San Francisco de Quito and Universidad de Cuenca.

The rest of this work is structured as follows: Subsection 2 describes the related work, Subsection 3 presents the proposed methodology, Subsection 4 describes the results of the case study analysis, and Subsection 5 presents the conclusions and future work components, incorporating the applicable criteria that follow.

6.2 Related Work

Ruch et al. (2007) propose the use of various bag-of-words or bag-of-n-grams representations to classify sentences from abstracts in the domains of clinical trials and biomedicine in categories, such as introduction, purpose, method, results, and conclusions. Gaeta et al. [142], extract knowledge ontologies from structured documents using text mining and semantic analysis techniques, and Sun et al. [170] uses hierarchy Latent Dirichlet Allocation (hLDA) model to generate a tree structure of the topic hierarchy for a java program comprehension. Also, Sam proposes a model design that analyzes the unstructured customer reviews inside the posts on social networking websites, and Salas-Zarate et al. [171], propose a sentiment analysis method based on ontologies in the diabetes domain from Twitter. In all the approaches reviewed, something stands out. They are addressed using either rules or Machine Learning methods based on features disregarding other information, such as the syntactic dependencies. Our work differs from the above studies because the ontologies generated have common concepts found in different unstructured documents from several sources. Moreover, syntactic sequences of words were taken into consideration, as well as their order of appearance in sentences. In this way, we make sure that these ontologies, specifically designed to explain a domain representation, have only the most relevant terms and relations in a hierarchical structure.

6.3 Methodology

In figure 1, we show the methodology phases for configuration of an ontology's generation framework,

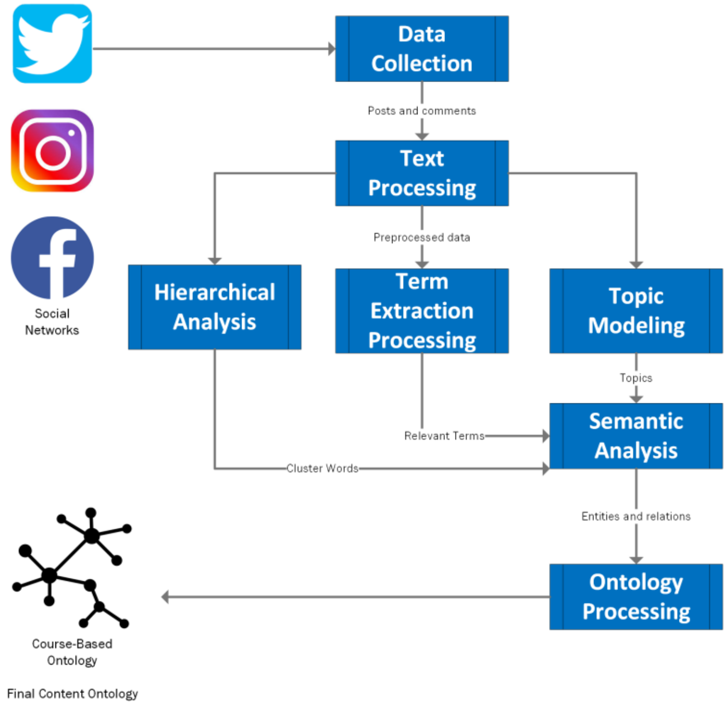


Figure 6-1. A proposed Framework for Ontologies Generation from Social Networks

6.3.1 Social Media Data in Education

Social media is essential because social networks have made everybody a potential author, so the language is now closer to the user than to any prescribed norms [172]. This way, they share information about events, activities, services, opinions, and experiences on social media channels [169].

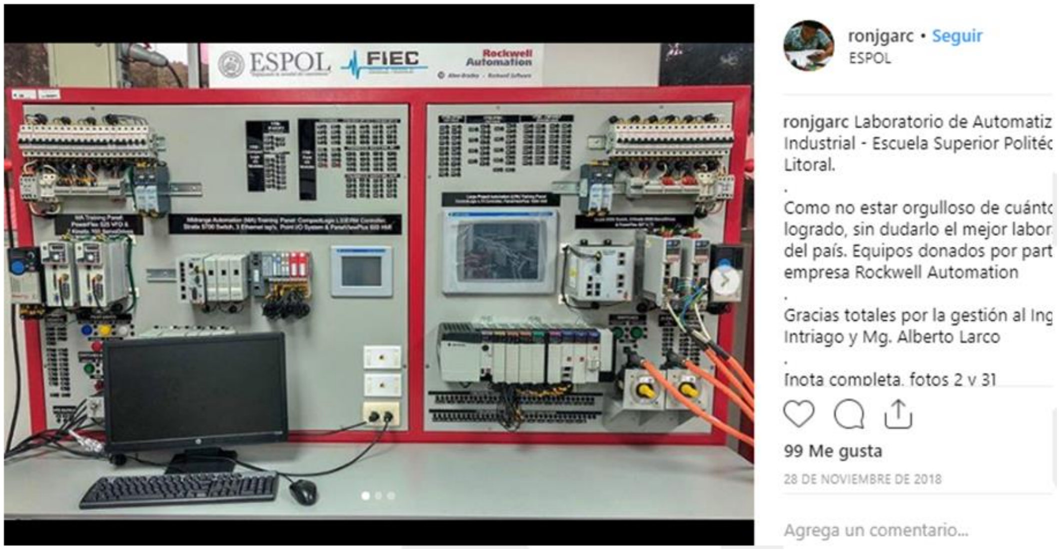
Social networks have gained credibility over the years as reliable sources of information and a platform where organizations can interact with audiences. Educational institutions adopt these developments to their systems and depend on group resources and mechanisms to improve student life [173]. The use of social networks in education gives students the ability to obtain useful information and connect with learning groups and other educational resources and applications.

Valuable knowledge can be obtained through social networks, such as analysis and insights on several topics or issues for study purposes. Social networks are also a way where students can establish beneficial connections for their careers. As an educational institution, it is crucial to be active on many possible social platforms; this helps to create better student training strategies and shapes student culture.

Also, social networks offer audience and subject monitoring tools that are useful and is one of the best platforms for extracting data. Researchers can find out how most people feel about a particular topic or how experts perceive and advise specific topics.

6.3.2 Data Collection

Information was collected from publications about activities, comments, and opinions from social networks: Facebook, Instagram, and Twitter. We selected 15.149 public posts between August and December of 2018, which presented interaction (comments and likes) among students on issues related to university life linked by hashtags referring to the selected universities (#espol, #ucsg, #uees, #ucuenca, #usfq). We only considered publications with more than two comments, and that does not refer to advertising, sales, or other issues outside academic scopes.

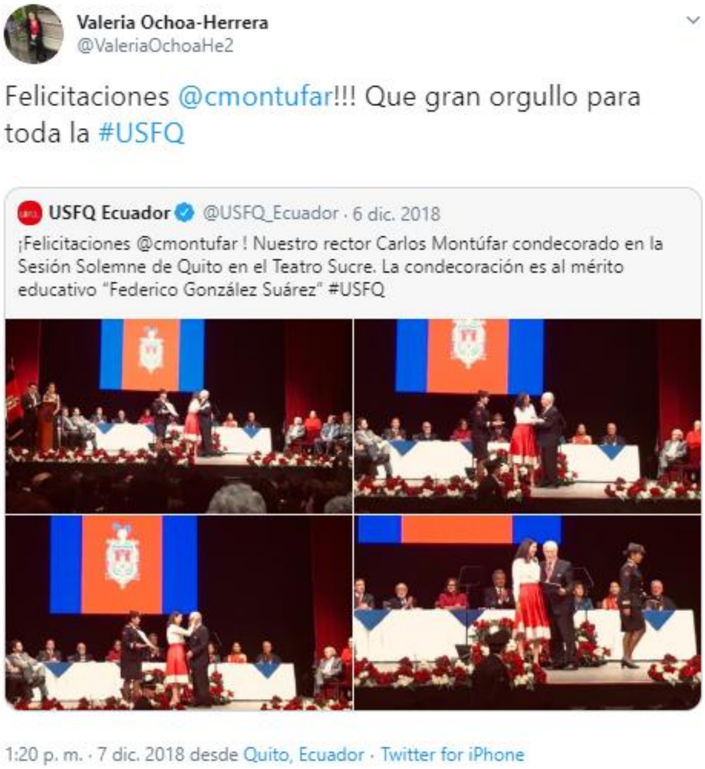


ronjgarc • Seguir
ESPOL

ronjgarc Laboratorio de Automatiz
Industrial - Escuela Superior Politéc
Litoral.
.
Como no estar orgulloso de cuántc
logrado, sin dudarlo el mejor labor
del país. Equipos donados por part
empresa Rockwell Automation
.
Gracias totales por la gestión al Ing
Intriago y Mg. Alberto Larco
.
[nota completa, fotos 2 y 3]

99 Me gusta
28 DE NOVIEMBRE DE 2018

Figure 6-2. Instagram post with #espol hashtag



Valeria Ochoa-Herrera
@ValeriaOchoaHe2

Felicitaciones @cmontufar!!! Que gran orgullo para toda la #USFQ

USFQ Ecuador @USFQ_Ecuador · 6 dic. 2018
¡Felicitaciones @cmontufar ! Nuestro rector Carlos Montúfar condecorado en la Sesión Solemne de Quito en el Teatro Sucre. La condecoración es al mérito educativo "Federico González Suárez" #USFQ

1:20 p. m. · 7 dic. 2018 desde Quito, Ecuador · Twitter for iPhone

Figure 6-3. Twitter post with #usfq hashtag

For data extraction, we used Python scraping algorithms with Selenium and BeautifulSoup libraries and wrapper functions with Twitter and Instagram API. The following features per post were gathered: post id, user id, hashtags, date, text, and comments count. Next, each document was processed, considering the following steps:

1. Lexical analysis where each document was transformed into words to describe the content,

2. Elimination of empty words, generally composed of articles, prepositions, marks, conjunctions, numbers, punctuation marks and words that did not describe the content semantically and are of little interest for the analysis.

3. Stemming process or recognition of stems consisted of the automatic elimination of non-essential parts of the terms (suffixes, prefixes) to reduce them to their essential part. The percentages of publications by social networks during this period were as follows. Twitter 30.60%, Facebook Forum Pages, 8.17%, and Instagram 61.23%, indicating the

popularity that Instagram has on students. Figure 6-2 and figure 6-3 shows an Instagram and twitter posts associated with the hashtag “espol”.

6.3.3 Terms extraction process

Text mining processes such as TF-IDF [65], were applied over documents in order to determine the most relevant terms in user posts. TF-IDF is a technique for term extraction that allows the identification of terms in a document. Besides term frequency, this technique also considers the relevance of the terms in the document. This process requires a corpus to distinguish standard terms from the relevant ones, which appear in the analyzed document but are not frequently used in any other document. After extracting all posts and their user comments, the term frequencies for the whole corpus was calculated. Then, we applied a Part-Of-Speech Tagging process using Stanford Parser NLP library [74], over terms, to identify only the nouns. Finally, these terms are stored in lists for subsequent validations. In Figure 6-4, an example of part-of-speech in a sentence is shown.

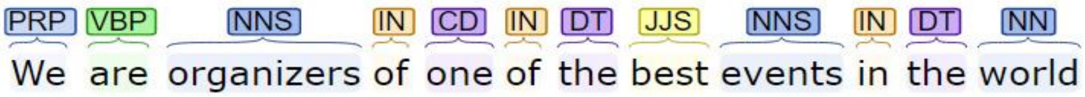


Figure 6-4. Part-of-speech from a sentence.

6.3.4 Topic Modeling

The concepts composed of one or more relevant terms extracted from the corpus was identified, using the hierarchical LDA [125]. hLDA is an adaptation of LDA that models’ topics as mixtures of a new, distinct level of topics, drawn from Dirichlet distributions and not processes. The difference is that the clustering is now hierarchical- it learns a clustering of the first set of topics themselves, giving a more general, abstract relationships between topics (and hence, words and documents). Next, to configure hLDA, it is necessary to choose a path in the L-level tree from root to leaf and a vector topic Θ of topic proportions from a Dirichlet distribution of L dimensions. Then use a mixture of the topics from root to leaf to generate the words that make up each document.

6.3.5 Semantic Analysis

For the semantic analysis and the relationship extraction, we used Part-Of-Speech Tagging (POST) and Open information extraction techniques [88] that process the linguistic analysis of sentences and paragraphs terms, verbs and proper names. First, each sentence of the documents related to its similarity is divided into a set of related clauses. Each clause is reduced to the maximum, producing a set of shorter sentence fragments. These fragments are then segmented into information extraction triples, which were grouped and prioritized according to the concepts and terms presented in selected posts that contain nouns identified in the term extraction process. The corpus and documents are almost entirely in Spanish, so the NLP tools used must support that language. Among different approaches, we chose an implementation of LinguaKit [174]. Linguakit is a Natural Language Processing tool that supports Spanish language and includes relation extraction methods that return triples: SUBJECT - RELATION - OBJECT using algorithms based on Open Information Extraction.

6.4 Results

Once data was collected, it was preprocessed to remove Unicode text emojis, stop words, and transform text in a sequence of tokens. Next, data was organized into five datasets: (ESPOL, USFQ, UCSG, UCUENCA, UEES). Table 6-1 shows a sample of terms and the number of words per each dataset.

Dataset	Words Count	Sample Terms
ESPOL	12916	students, friends, congratulations, espae, yosoyespol, event, sporadic, datajamec, campus, week, iweek
USFQ	8316	friends, congratulations, life, architecture, students, dragonsporsiempre, donation
UCSG	4393	design, research, professional, marketing, medicine, career, congress, art, life, graduation
UCuenca	6105	faculty, students, sciences, hospital, architecture, education, graduates, arts, carnetization
UEES	9029	tourism, congress, success, goals, congratulations, degree, conference, project, architecture

Table 6-1. Datasets Summary

Next, terms extraction process was used to detect the most relevant terms. For ESPOL dataset was considered 12802 words (99.12%) from 4127 documents. The most common terms with higher TF-IDF weights were: “yosoyespol”, “iweek”, "engineering", "campus", "datajamec", "congratulations", "students", "espae", "mountainbike”, among others. Figure 6-5 shows terms that have a TF-IDF weight greater than 500 and suggest that most of the posts are related to faculties, friendship, students’ activities, sports, and academic events.

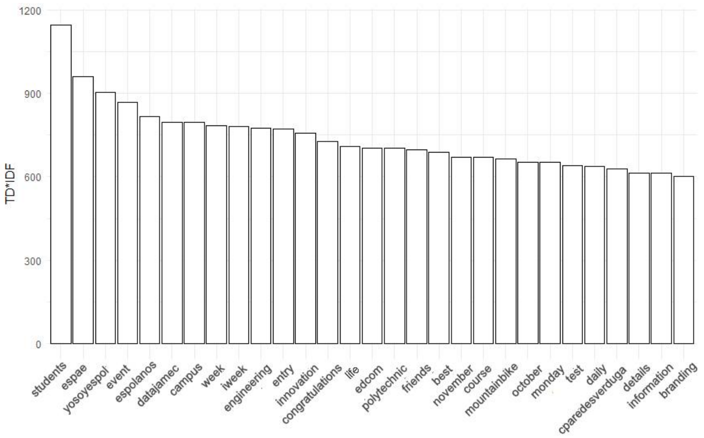


Figure 6-5. Most Frequent terms in five ESPOL dataset.

For the dataset of the five universities posts, 31.799 unique terms (99.24%) were conserved for the following analysis. Figure 5. shows the most frequent terms with higher TF-IDF weights. They were: “students,” “semester,” “lessons,” “faculty,” “life,” “thesis,” “partial,” “events,” “career,” “overcoming” among others. These terms have a TF-IDF weight greater than 1.500 and suggest that most of the posts are related to semester tests, university life, graduations, friendship, and faculty activities.

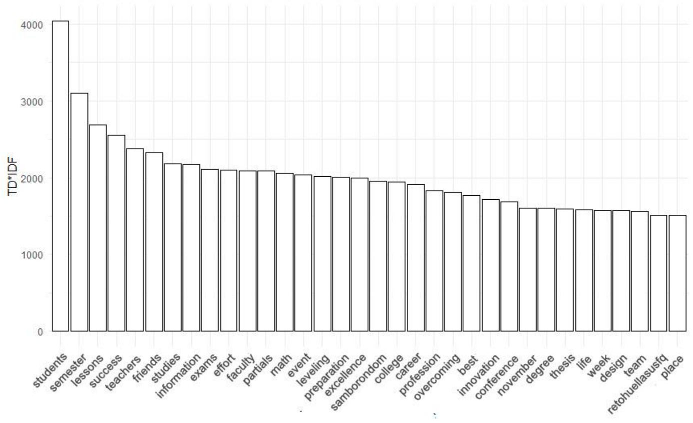


Figure 6-6. Most Frequent terms in five Universities dataset

6.4.1 Ontologies Processing

To prepare the data and create the corpus, text preprocessing routines, such as Tokenization, elimination of Stop words, and Stemming, were run over datasets. Next, using java mallet, the following parameters were configured to hLDA topic models: 500 as iterations number, a hyperparameter α of 0.1 as a value of smoothing over level distributions, and three as the number of levels in the tree. Finally, once the process run and the number of topics returned by hLDA model were 223 for the ESPOL dataset and 689 for the universities' dataset. Table 6-2 shows a sample of words for subsets of nodes (topics) in a label tree that was learned via hLDA using ESPOL and Universities datasets.

Level	**Topic Terms for ESPOL dataset**	**Topic Terms for Universities dataset**
0	i3week, yosoyespol, innovation, student, event, university, espolano, project, datajamec, friend, campus	University, student, project, faculty, career, event, best, friend, life, information, work, science
1	nation, app, life, best, training, thesis, bank, solution, edcom, section, music, academic, tech	retohuellausfq, challenge, plastic, carbon, footprint, ecological, environmental, reduction, engineering,
1	study, article, entailment, espolbike, crosscountrymtb, sunset school, cultural, datajamecuador,	induction, production, faculty, relevance, biodiversity, educational, entrepreneur, architectal, freshman
1	education, mentor, kenrobinson, degree, speed, datajam, scientific, nature, worldwide, competition	student, memory, support, generation, polytechnic, natural, project, event, community, marketing
1	information, reencuentroespol, south, dynamic, tourism, science, experience	degree, prototype, music, presence, circuit, ninth, inscription, bigdata, brand, exchange, cohort, need
1	payment, transespol, tournament, advertisement, recognition, culture, means, world, initiative,	natural, friends, engine, nature, course, creation, allyouneedisecuador, design, education, development, tourism

Table 6-2. A sample of hLDA topics from ESPOL and Universities datasets

Then, only posts that contain main concepts detected by hLDA (in topic level 0 and level 1) in their sentences were selected to apply topic modeling. Table 6-3 shows example topics per post and their distributions of words. After executing the process, the nouns extracted from topics with probability values between 0.40 and 1, were taken and connected to the main concepts. Each term becomes a concept within the ontology, and the parent-child relationship between terms becomes a relation.

Post	Prob	Topic Words
Meet our Laboratory of Experimental Economics and Behavior at @Espol	0.08	Lessons, business, english, teaching, behavior, speakers, links
	0.12	laboratory, insurance, graduate, future, unit, month, marketinglife
	0.14	share, knowledge, wonderful, speakers, doctors, work, grow, end
	0.24	entry, exam, test, teaser, pagoda, explosives, aptitude, knowledge, laboratory
	0.42	degree economics research, ethical, fortunate, crisis, counseling, learn

Table 6-3. Example of topics extracted from an Instagram Post

The number of topics words to be extracted per post is a configurable parameter k of the topic extraction algorithm. During experiments, values between four and fifteen were evaluated. The size of the final ontology varied according to this value, i.e., a smaller number of topic words resulted in a smaller number of concepts and relationships. In this way, we selected a k value of 4, to obtain a count of entities between 60 and 70 in the resulting ontologies. Figure 6-7 shows the relation between the number of topic words per post and the number of terms of the final ontology, detected during the experiments.

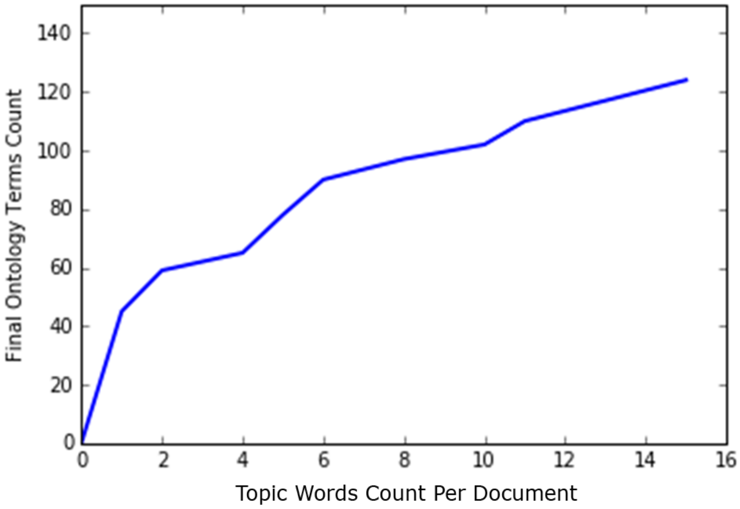


Figure 6-7. Relation between k and Ontology Terms count

The hierarchical structure is constructed, taking the terms which belong to the zero and first level, to build an index of terms. To avoid words with similar meaning, it was verified that the words are nouns using POST algorithms (Part-of-Speech Tagging) provided by Stanford Spanish Parser Tools [175] did not have the same stem.

The connections between concepts, by default, can be interpreted as a “Has Related” relation and are considered as axioms in the final ontology. However, they can be replaced by the verbs identified in Spanish sentences during semantic analysis and triples extraction processes. During the semantic analysis process, in order to extract more profound relations between terms, a triples extraction process using Linguakit Perl tool was applied to Spanish sentences extracted from social networks posts. Table 6-4 shows terms in structured relation triples.

Subject	**Relation**	**Object**
Students	share	professors and administrative staff from the faculty of natural sciences
Tomorrow,	will be participating	the Basketball women's team
This project	tries to bring	an effective communication culture of ESPOL to the city
Students from different schools	do	an educational journey
The activities	develop	the student clubs of ESPOL

Table 6-4. Triples extraction process for sentences.

The triples extraction process identified hundreds of triples in posts, but only those that had terms present in topic lists were selected. These terms were connected and added to the definitive ontology. Figure 6-8 shows an excerpt from ESPOL University posts ontology graph, obtained from the following main concepts: ’yosoyespol’, ’students’, ’i3week’, ’project’, ’event’ and ’innovation’. This ontology has 54 terms in a hierarchy in its first three levels and 68 terms in the first fourth levels.

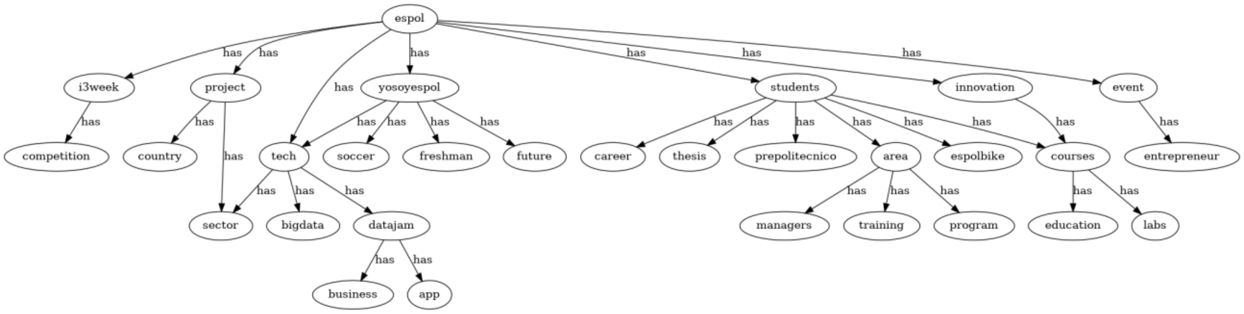


Figure 6-8. Final Ontology Excerpt for ESPOL dataset.

Figure 6-9 shows an excerpt of the ontology graph for universities dataset, obtained from six main concepts (student, faculty, project, event, work, life). This ontology has 61 terms in a hierarchy in its first three levels and 72 terms in the first fourth levels.

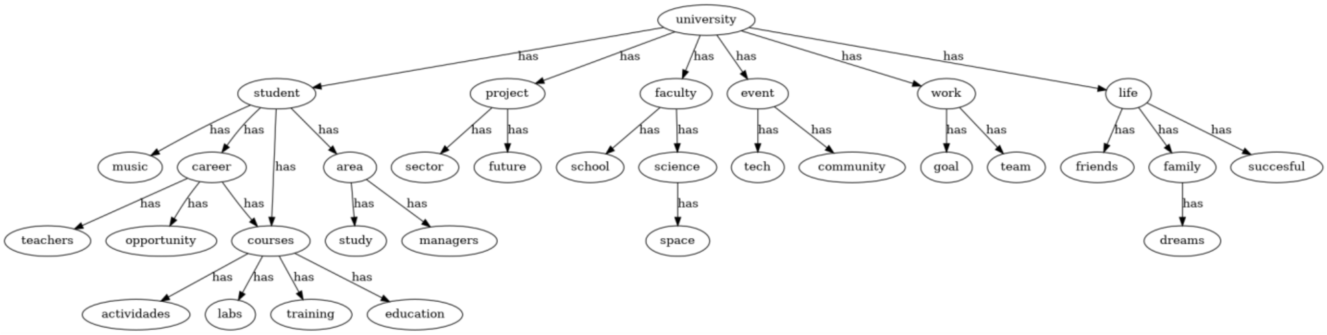
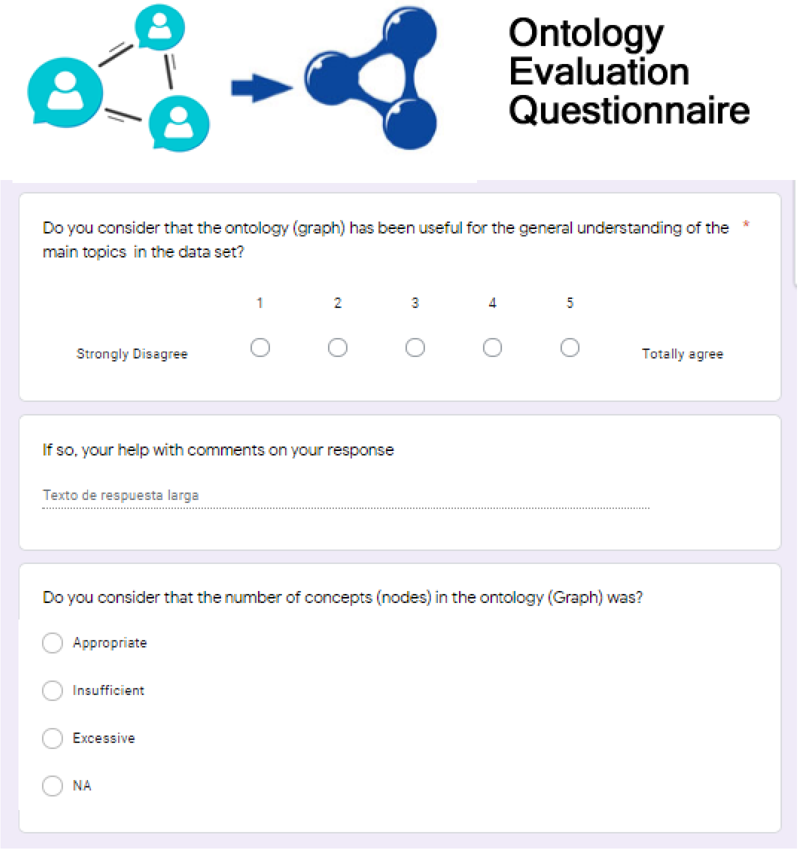


Figure 6-9. Final Ontology Excerpt for universities dataset

6.4.2 Evaluation

Our case study involved 21 participants from several universities. Three quarters are graduates or postgraduates, and the others are undergraduates. We investigated whether the approach helps the participants to understand the most relevant topics in social networks posts. To show whether the ontologies generated by our methodology is useful, the participants needed to assess whether the ontologies enable them to understand the datasets samples.

A five-point Likert scale with 1 (Strongly disagree) to 5 (very agree) assessed each participant’s view of the ontologies. According to the scores, we can see whether the model helps topic comprehension. Figure 6-10 shows the questionnaire used for the ontology evaluation process.



Ontology Evaluation Questionnaire

Do you consider that the ontology (graph) has been useful for the general understanding of the main topics in the data set? *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Totally agree

If so, your help with comments on your response

Texto de respuesta larga

Do you consider that the number of concepts (nodes) in the ontology (Graph) was?

○ Appropriate

○ Insufficient

○ Excessive

○ NA

Figure 6-10 Ontology Evaluation Questionnaire

Each participant was given a random sample of posts extracted from the Universities dataset, the ontologies generated, and a five-point Likert scale to answer questions related to the quality of the topics. The average score is around 3.95, which indicates that the participants think that the ontologies concepts are useful to understand the posts collections. Table 6-5 shows the descriptive statistics on the five-point Likert-type scale in this study.

Question Objective	**N**	**Mean**	**Std Deviation**	**Variance**	**Skewness**
Ontology Usefulness	21	3.95	0.68	0.44	0.062

Table 6-5 Five-point Likert responses descriptive statistics

Moreover, the opinion of the participants was consulted about the number of concepts of the generated ontology shown in the experiment. The results show in figure 6-11 were that 65% of the participants considered that it was appropriate, 15%, that it was insufficient, and 20%, that it was excessive.

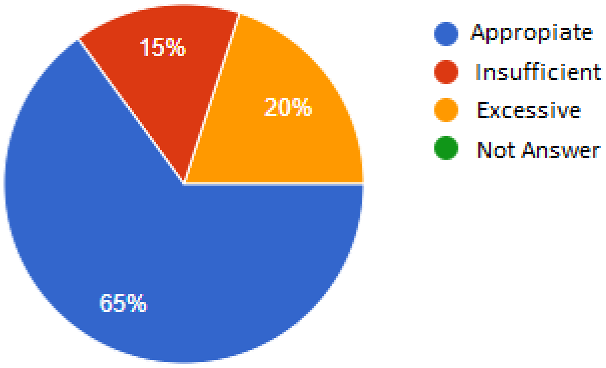


Figure 6-11. Opinion statistics about the number of ontology concepts.

Also, the participants made some comments about their test experience. Some of the reviews pointed out that the graph is an adequate representation of the topics related to the dataset. Others indicated that the classification could be improved or that specific topics were missing in the structure of the ontology. Table shows the most relevant comments about ontology evaluation from participants

Question Objective	**Comment**
Ontology's Usefulness	I consider that a large portion of the topics are covered, but there are some that do not fit the classification.
	The graph shows a good structure of the topics that are treated in the sample of the data set and allows them to have a conceptual map of the topics that are treated with respect to the university's topics.
	It can be said that each cluster/group summarizes the info of the posts, is understandable, and shows what is necessary to understand the contents of the posts.
	The first nodes after the initial node ('university') seem correct to me, but some children of those nodes may be better classified.
	I consider that some topics of conversation were missing, but it is an adequate graphic representation.

However, the ontology of the experiment, having only 61 nodes, excluded specific more in-depth topics and subtopics, which could be considered as missing or not included in the hierarchical structure of the graph, by the participants. Also, of 65% of the participants who considered that the size of the ontology was appropriate for their objective, most had a rating of 4 or 5 in the usefulness evaluation.

6.5 Conclusions

In this work, it was possible to design and built-in automatically way, an ontology in the Spanish language to discover relevant information in posts about universities. The unstructured data was extracted from several social networks such as Facebook, Instagram, Twitter, and was processed through text mining, machine learning, and natural language process techniques.

The applied methodology can be used to create ontologies based on social network publications that generate interaction between users, regardless of the language and content topics. The final ontologies can help to discover interests and conversation topics in users and provide, in a simple way, useful information for the understanding and decision-making process by managers and tutors, besides being useful for analytics applications and recommendation systems.

The topics of interest detected in the resulting ontologies had a favorable rating from an evaluation group. However, in some cases, it was hoped to find more in-depth information on other aspects of universities, such as course opinions or student problems during their academic progress. These last topics could be detectable in ontologies configured to have a larger size.

7 General Conclusions and Recommendations

The ontologies have been widely used for various purposes, for example, to support data modeling, information integration, and knowledge management. Their role is to facilitate the construction of a domain of models and provide a vocabulary of terms and relationships in various fields, including education. This research focuses on creating educational ontologies without further human intervention and then evaluating and using in potential educational applications.

First, in chapter 4, this research proposed and tested the idea of using existing learning resources such as digital books, web-based tutorials, or other educational text in digital format as sources, to (semi- automatically) build and maintain course-based ontologies. These ontologies could realize the concept of the Intelligent Curriculum. The domain expert could easily modify them without the need to know about semantic technologies. In the evaluation of our methodology, two books were enough to create ontologies that were, on average, 70% precise and capture more than 50% of the concepts generated manually by an expert. These results answered the question "In a semi-automatic way and with the support of Machine Learning and Natural Language Processing techniques, it is possible to generate domain ontologies from Semi-structured data such as texts of books, tutorials and courses from different academic disciplines?".

Thus, this research challenges the assumption that educational ontologies require expensive knowledge engineering for domain and content modeling. It also explores how to resolve the ontology-building economic barriers to entry, limiting the use of the Intelligent Curriculum for educational and learning analytics applications. With a little effort from the end-user, just recommending authoritative sources of learning materials such as digital books even in different languages, an acceptable ontology of the domain can be automatically created.

Then the research continued with the creation of a system to support the design of the Intelligent Curriculum. In this application, while the instructor added learning content, the assistant system recommended a series of concepts through an automatically generated preliminary ontology. The instructor could improve its content through additional material or direct manipulation. Also, the teacher could validate the quality of his text, either against the ontology or against curricular standards through the system's functionalities.

The evaluation results in chapter 5 answered the question, "The automatically generated domain course ontologies can be useful in the fields of educational and learning analytics? It could be determined that teachers who participated in the experiment believed that the system was useful and perceived it as relatively easy to use to develop Intelligent Curricula. Also, teachers opined that the functionality that contributed most to curricula design was the preliminary visualization of the automatically-generated educational ontology.

The system users highlighted the few steps that must be taken to achieve the objectives and requested improvements to systems functionalities, to get more flexibility at the time of using ontologies. These evaluation results allow us to envision opportunities such as the use of methods to evaluate the content of the curricula of academic courses with their respective curricular standards. Besides, they may allow the development of more sophisticated educational solutions that consume these ontologies, like tools for monitoring student learning during a course or applications that recommend courses learning materials.

Finally, the research explores the automatic building of ontologies from unstructured data extracted from publications related to academics' issues and student life in different social networks like Facebook, Instagram, and Twitter. The data was processed through Text Mining, Machine Learning, and Natural Language Process techniques to discover relevant information from post collections.

The evaluation scores from questionnaires in chapter 6, answered question “In a semi-automatic way and with the support of Machine Learning and Natural Language Processing techniques, it is possible to generate domain ontologies from unstructured data, such as social networks publications related to education topics.?”. These results show that participants think that the ontologies' structures are useful for understanding the posts collections and comprehending the main topics they address. Besides, most study users considered that the size of the generated ontologies was adequate for their objectives.

The resulting domain ontologies can help to discover interests and conversation topics in users and provide, in a simple way, useful information for the understanding and decision-making process by educational managers and tutors. Among the topics that generated interaction between users, we can mention academics events, graduations, sports activities, information requests about careers, and faculties, course experiences, among others.

The findings of this research seek to make contributions to the current state of the literature about the creation, evaluation, and exploration of the use of ontologies in the field of education. The most relevant concepts from education-related materials could be identified, processed, and linked to each other for the generation of course ontologies through a series of artificial intelligence techniques. These automatically-generated models could be presented to teachers and students as a summary of a domain knowledge and integrated with educational and learning analytics applications.

Two critical factors were identified during the design of the ontologies: the size and the language of the knowledge model. On the one hand, size as a property plays an essential role in the ontological understanding by humans and on the other hand, the domain language is key for the adequate selection and configuration of the NLP tools. These variables could be configured to generate more inclusive ontologies ideal for visual comprehension or more extensive for greater precision in computational algorithms. It was also identified that it is possible to break the language barrier to create ontologies in other languages. One of them is Spanish, a field where this kind of type of educational resource is scarce.

However, the research was not without its limitations. Textbooks and online materials are written by different authors, for different audiences, and even in several languages. Furthermore, the volume of educational resources for less popular domains can be limited, and many domains lack formal domain models or ontologies, where all concepts are listed and organized. In other words, they have no domain standards to be measured. Additionally, more participants could have been considered to receive more feedback about the content of the ontologies generated and the functionality of the assistant system. However, due to the defined profile, there were few candidates.

The use of more sophisticated techniques than topic modeling could be explored in future work, such as models to produce word embeddings with terms that share the same semantic meaning and deep learning models for text classification. Also, it would be an important asset to investigate how to improve term mismatch difficulties between documents

For technical and practical implications, ontologies generated from textbooks could be used as a component of other educational and learning analytics tools as knowledge models for recommendation systems and the feedback and implementation of pedagogical strategies in intelligent tutoring systems. Furthermore, with the changes suggested by the participants and by adding new functionalities to the assistant system, it could be used to prevent and solve inconsistencies in the curricular content and to provide visual feedback to students on their learning activities progress in courses, among other utilities.

The methods used to build ontologies from unstructured texts allow an overall visualization that supports the understanding of the knowledge extracted from publications not only in the educational field but also on topics of local and global interest. Thus, topics related to online education, social restrictions, distance learning, collaborative and remote tools, could be explored. Also, it is possible that other content sources could be investigated, such as blog pages or discussion forums like Reddit, Quora, StackOverflow, among others.

Bibliography

[1] HCM Technology Report, "Are You Ignoring Unstructured Organizational Data?," 04 2020. [Online]. Available: https://www.hcmtechnologyreport.com/are-you-ignoring-unstructured-data/.

[2] C. Aggarwal and C. Zhai, Mining Text Data, New York: Springer , 2012.

[3] Y. Goldberg and G. Hirst, Neural Network Methods in Natural Language Processing, 2017: Morgan & Claypool Publishers.

[4] SAS Institute, "Machine Learning," [Online]. Available: https://www.sas.com/en_us/insights/analytics/machine-learning.html. [Accessed April 2020].

[5] M. Johnson, How the statistical revolution changes (computational) linguistics, Association for Computational Linguistics, 2009.

[6] P. Brusilovsky and C. Peylo, "Adaptive and Intelligent Web-based Educational Systems," *Int. J. Artif. Intell. Educ,* vol. 13, 2002.

[7] Great Schools Partnership , "The Glossary of Education Reform," 2016. [Online]. Available: https://www.edglossary.org/content-knowledge/. [Accessed 04 2020].

[8] T. Gavrilova and I. Leshcheva, "Ontology design and individual cognitive peculiarities: A pilot study," *Expert Systems with Applications,* vol. 5, no. 8, 2015.

[9] M. A. Musen, "Dimensions of knowledge sharing and reuse," *Computers and Biomedical Research.,* vol. 25, no. 5, p. 435–467, 1992.

[10] T. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition,* vol. 5, no. 2, pp. 199-220, 1993.

[11] B. Chandrasekaran, J. Josephson and V. Benjamins, "What Are Ontologies, and Why Do We Need Them?," *Intelligent Systems and their Applications, IEEE,* vol. 14, 1999.

[12] D. Man, "Ontologies in computer science," *DIDACTICA MATHEMATICA,* vol. 31, no. 1, pp. pp. 43–46,, 2013.

[13] T. Gavrilova, M. Kurochkin and V. Veremiev, " Teaching Strategies and Ontologies for E-learning.," *Information Theories & Applications,* vol. 11, 2007.

[14] K.-K. Chu, C.-I. Lee and R.-S. Tsai, "Ontology technology to assist learners' navigation in the concept map learning system," *Expert Systems with Applications,* vol. 38, no. 9, pp. 11293-11299, 2011.

[15] B. Barros, F. Verdejo, T. Read and R. Mizoguchi, "Applications of a Collaborative Learning Ontology," in *roceedings of the Second Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence*, London, UK, 2002.

[16] S. Sosnovsky, I. Hsiao and P. Brusilovsky, "Adaptation "in the Wild": Ontology-Based Personalization of Open-Corpus Learning Material," in *21st Century Learning for 21st Century Skills*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2012, pp. 425-431.

[17] W. Lu and J. Wei, "Dynamic visualization of Evolutionary Curricula Model," *International Conference on Educational and Information Technology. Vol. 2. 2010, pp. V2-484-V2-488.,* 2010.

[18] G. Siemens and R. Baker, "Learning analytics and educational data mining: towards communication and collaboration," in *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge, LAK*, Vancouver, British Columbia, Canada, 2012.

[19] J. Guerra, S. Sosnovsky and P. Brusilovsky, "“When One Textbook Is Not Enough: Linking Multiple Textbooks Using Probabilistic Topic Models”," *Scaling up Learning for Sustained Impact. Ed. by Davinia Hernández-Leo et al. Berlin, Heidelberg: Springer Berlin Heidelberg,* p. 125–138, 2013.

[20] K. Thaker, P. Brusilovsky and D. He, "Concept Enhanced Content Representation for Linking Educational Resources," in *IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, 2018.

[21] D. Fensel, Ontologies: Silver Bullet for Knowledge Management., Springer Verlag, 2000.

[22] N. Guarino, "Formal Ontology and Information Systems," in *Proceedings of the First International Conference (FOIS'98)*, Trento, Italy, 1998.

[23] T. Buzan, Mind map handbook, London: Thorsons, 2002.

[24] J. Novak and A. Cañas, "The Theory Underlying Concept Maps and How to Construct Them.," in *Technical Report IHMC CmapTools* , Florida Institute for Human and Machine Cognition, 2006.

[25] D. Kudryavtsev and T. Gavrilova, "Diagrammatic knowledge modeling for managers: Ontology-based approach.," in *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, 2011.

[26] S. Stephens, "From Tree to Map: Using Cognitive Learning Theory to Suggest Alternative Ways to Visualize Macroevolution," *Evolution: Education and Outreach,* vol. 5, no. 4, 2012.

[27] S. Palmer, Vision Science: From Photons to Phenomenology., MIT Press, 1999.

[28] R. Styles and N. Shabir, "Academic Institution Internal Structure Ontology (AIISO)," 2008. [Online]. Available: https://vocab.org/aiiso/.

[29] D. Brickley and L. Miller, "FOAF Vocabulary Specification 0.99," 2014. [Online]. Available: http://xmlns.com/foaf/spec/.

[30] G. Demartini, I. Enchev, J. Gapany and P. Cudre-Mauroux, "The Bowlogna ontology: Fostering open curricula and agile knowledge bases for Europe's higher education landscape," *Semantic Web,* vol. 4, no. 1, 2012.

[31] I. Niles and A. Pease, "Towards a standard upper ontology," in *Proceedings of the International Conference on Formal Ontology in Information Systems*, Ogunquit, Maine, USA, 2000.

[32] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari and L. Schneider, "Sweetening ontologies with DOLCE," in *Proceedings of the 13th Int. Conf. on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, Siguenza, Spain, 2002.

[33] H. Chen, T. Finin and A. Joshi, "An Ontology for Context Aware Pervasive Computing Environments.," *The Knowledge Engineering Review,* vol. 18, no. 3, pp. 197 - 207, 2003.

[34] X. Wang, D. Zhang, T. Gu and H. Pung, "Ontology Based Context Modeling and Reasoning using OWL.," *Pervasive Computing and Communications Workshops, IEEE,* pp. 18-22, 2004.

[35] B. Dutta and D. G. Nandini, "MOD: Metadata for Ontology Description and Publication," in *Conference: International Conference on Dublin Core and Metadata Applications (DC-2015)*, Sao Paulo, Brazil, 2015.

[36] L. Kagal, "Rei : A Policy Specification Language," 2002. [Online]. Available: https://ebiquity.umbc.edu/project/html/id/34/Rei-A-Policy-Specification-Language.

[37] H. Chen, F. Perich, T. Finin and A. Joshi, "SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications.: Mobile and Ubiquitous Systems: Networking and Services," *MOBIQUITOUS,* pp. 258-276, 2004.

[38] DAML, "The DAML Services Coalition 2003. DAML-S (and OWL-S) 0.9 Draft Release.," 2004. [Online]. Available: OWL-S) 0.9 Draft Release..

[39] Cycorp, "Cycorp," 2005. [Online]. Available: http://www.opencyc.org/.

[40] S. Li and M. Ying, "Region connection calculus: its models and composition table," *Artificial Intelligence Journal,* 2003.

[41] F. Perich, "MoGATU BDI Ontology.," 2003. [Online]. Available: http://ebiquity.umbc.edu/resource/html/id/2/MoGATU.

[42] S. Youn, McLeod and Dennis, "Ontology Development Tools for Ontology-Based Knowledge Management," in *Encyclopedia of E-Commerce, E-Government, and Mobile Commerce*, 2006, pp. 858-864.

[43] The Board of Trustees of the Leland Stanford Junior University, "Stanford University Protégé," 2016. [Online]. Available: https://protege.stanford.edu/.

[44] D. Dicheva, "Ontologies and Semantic Web for E-Learning," *Handbook on Information Technologies for Education and Training,* pp. 47-65, 2008.

[45] C. Knight, D. Gasevic and G. Richards, "An Ontology-Based Framework for Bridging Learning Design and Learning Content," *Educational Technology & Society,* vol. 9, no. 1, pp. 23-37, 2005.

[46] G. Schreiber and H. Akkermans, Knowledge Engineering and Management: The CommonKADS Methodology, Cambridge, MA, USA: MIT Press, 2000.

[47] J. Davies, D. Fensel and F. Harmelen, Towards the Semantic Web: Ontology-Driven Knowledge Management, Wiley, 2003.

[48] F. Fonseca, C. Davis Jr and G. Câmara, " Bridging Ontologies and Conceptual Schemas in Geographic Information Integration," *GeoInformatica,* vol. 7, pp. 355-378, 2003.

[49] G. Sherlock, "Analysis of large-scale gene expression data," *Current Opinion in Immunology,* vol. 12, pp. 201-205, 2000.

[50] K. Tolle, S. Tansley and T. Hey, "The Fourth Paradigm: Data-Intensive Scientific Discovery [Point of View]," *Proceedings of the IEEE,* vol. 99, no. 8, pp. 1334-1337, 2011.

[51] M. Yudelson, T. Gavrilova and P. Brusilovsky, "Towards User Modeling Meta-ontology," in *User Modeling*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2005, pp. 448-452.

[52] J. Bard and S. Rhee, " Ontologies in biology: Design, applications and future challenges.," *Nature reviews - Genetics,* vol. 5, pp. 213-222, 2004.

[53] A. Hevner, "A Three Cycle View of Design Science Research," *Scandinavian Journal of Information Systems,* vol. 19, 2007.

[54] R. Waymond, Artificial Intelligence in a Throughput Model: Some Major Algorithms, CRC Press, 2020.

[55] R. Agerri, X. Artola, Z. Beloki, G. Rigau and A. Soroa, "Big data for Natural Language Processing," *Know.-Based Syst.,* vol. 79, p. 36–42, 2015.

[56] A. Stavrianou, P. Andritsos and N. Nicoloyannis, "Overview and semantic issues of text mining," *ACM SIGMOD Record ,* vol. 36, no. 3, pp. 23-34, 2007.

[57] Y. Chan and D. Roth, "Exploiting Background Knowledge for Relation Extraction," in *International Conference on Computational Linguistics*, Beijing, 2009.

[58] C. Zhai and S. Massung, Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining, 2016.

[59] S. Chakrabarti, B. Dom and P. Indyk, "Enhanced Hypertext Categorization using Hyperlinks," *ACM SIGMOD Record ,* vol. 27, no. 2, 1998.

[60] D. Bollegala, Y. Matsuo and M. Ishizuka, "Measuring semantic similarity between words using Web search engines," in *16th International World Wide Web Conference, WWW2007*, 2007.

[61] S. Mika and B. Rost, "Protein names precisely peeled off free text," *Bioinformatics,* vol. 20, 2004.

[62] K. Lang, "Newsweeder: Learning to filter netnews," in *Proceedings of the 12th International Machine Learning Conference*, 1995.

[63] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu and H. Lauw, "Detecting product review spammers using rating behaviors.," in *Proceedings of the 19th ACM international conference on Information and knowledge management. 939-948*, 2010.

[64] S. Chakrabarti, B. Dom, R. Agrawal and P. Raghavan, "Using Taxonomy, Discriminants, and Signatures for Navigating in Text Databases," in *Proceedings of the International Conference on Very Large Data Bases (VLDB).*, 2000.

[65] D. Jurafsky and J. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, NJ, USA: Prentice Hall PTR, Upper Saddle River, 2000.

[66] B. Lantz, Machine Learning with R, Packt, 2013.

[67] G. Salton, A. Wong and C. Yang, "A Vector Space Model for Automatic Indexing," *Association for Computing Machinery,* vol. 18, no. 11, 1975.

[68] S. Robertson, "Understanding Inverse Document Frequency: On Theoretical Arguments for IDF," *Journal of Documentation 60(5),* pp. 503-520, 2004.

[69] F. Wild and C. Stahl, "Investigating Unstructured Texts with Latent Semantic Analysis," in *Advances in Data Analysis, Proceedings of the 30th Annual Conference of the Gesellschaft für Klassifikation* , Berlin, 2006.

[70] S. Deerwester, S. Dumais, G. Furnas, T. Landauer and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science,* 1990.

[71] D. Blei, A. Ng and M. Jordan, "“Latent Dirichlet Allocation”," *Journal Machine Learning ,* p. 993–1022, 2003.

[72] Y. Bengio, R. Ducharme, P. Vincent and C. Janvin, "A Neural Probabilistic Language Model," *Journal of Machine Learning Research,* vol. 3, 2003.

[73] S. Petrov, D. Das and R. McDonald, "A Universal Part-of-Speech Tagset," *Computing Research Repository - CORR. ,* 2011.

[74] K. Toutanova, D. Klein, C. Manning and Y. Singer, "Feature-rich part of speech tagging with a cyclic dependency network," in *In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 173–180, 2003.

[75] D. Jurafsky, Speech and Language Processing. Part-of-speech Tagging, 2016.

[76] S. Eger, T. vor der Brück and A. Mehler, "Lexicon-assisted tagging and lemmatization in Latin: A comparison of six taggers and two lemmatization models," in *2015*, Beijing, Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH 2015).

[77] M. Marcus, M. A. Marcinkiewicz and B. Santorini, "Building a large annotated corpus of English: the Penn Treebank," *Comput. Linguist,* no. 19, p. 313–330, 1993.

[78] S. Bird, E. Loper and E. Klein, Natural Language Processing with Python, O'Reilly Media Inc..

[79] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard and D. McClosky, " The Stanford CoreNLP Natural Language Processing Toolkit," *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations,* pp. 55-60, 2014.

[80] Google Inc, "Cloud Natural Language," [Online]. Available: https://cloud.google.com/natural-language/. [Accessed April 2020].

[81] L. Ramshaw and M. Marcus, in *Text Chunking using Transformation-Based Learning.* , ArXiv, 1995.

[82] Explosion AI, "Industrial-Strength Natural Language Processing," 2016. [Online]. Available: https://spacy.io/. [Accessed 04 2020].

[83] S. Loria, "TextBlob: Simplified Text Processing," 2013. [Online]. Available: https://textblob.readthedocs.io/en/dev/. [Accessed 04 2020].

[84] D. Nadeau and S. Sekine, "Survey of Named Entity Recognition and Classification," *Lingvisticae Investigationes.,* vol. 30, 2006.

[85] D. Bikel, S. Miller, R. Schwartz and R. Weischedel, "Nymble: a high-performance learning name-finder," in *Proceedings of the 5th Conference on Applied Natural Language Processing, pages 194–201*, 1997.

[86] H. Isozaki and H. Kazawa, "Efficient support vector classifiers," in *Proceedings of the 19th International Conference on Computational Linguistics,*, 2002.

[87] B. Settles, "Biomedical named entity recognition using conditional," in *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, 2004.

[88] G. Angeli, M. Premkumar and C. . Manning, " Leveraging Linguistic Structure For Open Domain Information Extraction," in *Proceedings of the Association of Computational Linguistics (ACL)*, 2015.

[89] C. Fay, "Text Mining with R : A Tidy Approach," *Journal of Statistical Software.,* 2018.

[90] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman and A. Wu, "An Efficient K-Means Clustering Algorithm Analysis and Implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, pp. 881-892, 2002.

[91] U. Luxburg, "A Tutorial on Spectral Clustering," *Statistics and Computing,* vol. 17, no. 4, 2007.

[92] Y. Zhao and K. G, "Evaluation of hierarchical clustering algorithms for document data," in *CIKM Conference*, 2020.

[93] E. Ikonomakis, S. Kotsiantis and V. Tampakas, "Text Classification Using Machine Learning Techniques," *WSEAS Transactions on Computers,* vol. 4, no. 8, pp. 966-974, 2005.

[94] K. Nigam, A. Mccallum, S. Thrun and T. Mitchell, "Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning," *Machine Learning ,* vol. 39, no. 2, 2000.

[95] N. Friedman, D. Geiger and M. Goldszmidt, "Bayesian Network Classifiers," *Machine Learning - Special issue on learning with probabilistic,* vol. 29, no. 2, pp. 131-163, 1997.

[96] Y. Ko and J. Seo, "Learning with unlabeled data for text categorization using a bootstrapping and a feature projection technique," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, 2004.

[97] B. Scholkopf and A. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, Cambridge, MA, USA: MIT Press, 2001.

[98] G. Madjarov, D. Kocev, D. Gjorgjevikj and S. Deroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern Recognition,* vol. 45, no. 9, pp. 3084-3104, 2012.

[99] M.-L. Zhang and Z.-H. Zhou, "Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization," *IEEE Transactions on Knowledge and Data Engineering,* vol. 18, no. 10, pp. 1338 - 1351, 2006.

[100] scikit-learn developers, "Multiclass and multilabel algorithms," 2007. [Online]. Available: https://scikit-learn.org/stable/modules/multiclass.html.

[101] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks,* vol. 61, 201f4.

[102] J. Brownlee, Deep Learning for Natural Language Processing: Develop Deep Learning Models for your Natural Language Problems, Machine Learning Mastery, 2017.

[103] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998.

[104] P. Liu, X. Qiu and X. Huang, "Recurrent Neural Network for Text Classification with Multi-task Learning," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, New York, New York, USA, 2016.

[105] S. Hochreiter and J. Schmidhuber, "Neural computation," *Long Short-term Memory,* vol. 9, 1997.

[106] J. Pennington, R. Socher and C. Manning, "GloVe: Global Vectors for Word Representation.," in *In Proceedings of EMNLP*, 2014.

[107] L. O and Y. Goldberg, "Dependency-Based Word Embeddings," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, Maryland, 2014.

[108] M. Sahlgren, "An introduction to random indexing," Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering, 2005.

[109] N. Azam and J. Yao, "Comparison of term frequency and document frequency based feature selection metrics in text categorization," *Expert Systems with Applications, volume 39,* pp. 4760 - 4768, 2012.

[110] X. Rong, "word2vec Parameter Learning Explained.," *CoRR Vol. abs/1411.2738,* 2014.

[111] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient estimation of word representations in vector space., ICLR, 2013.

[112] J. Brownlee, Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python, Machine Learning Mastery, 2019.

[113] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* , Doha, Qatar, 2014.

[114] M. Berger, "Large Scale Multi-label Text Classification with Semantic Word Vectors," 2015.

[115] N. Kalchbrenner, E. Grefenstette and P. Blunsom, A Convolutional Neural Network for Modelling Sentences, 2014.

[116] P. Wang, J. Xu, B. Xu, C. Liu, H. Zhang, F. Wang and H. Hao, "Semantic Clustering and Convolutional Neural Network for Short Text Categorization," Association for Computational Linguistics, 2015.

[117] B. Ramsundar and R. Zadeh, TensorFlow for Deep Learning, O'Reilly Media, Inc, 2018.

[118] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Neural Computation,* vol. 29, pp. 1-98, 2017.

[119] T. Y. Lin, Y. Yao and L. Zadeh, Data Mining, Rough Sets and Granular Computing., Berlin: Springer-Verlag, 2002.

[120] M. Vijaymeena and K. Kavitha, "A Survey on Similarity Measures in Text Mining.," *Machine Learning and Applications. An International Journal,* vol. 3, pp. 19-28, 2016.

[121] S. Kosub, " A note on the triangle inequality for the Jaccard distance.," *Pattern Recognit. Lett., 120,,* pp. 36-38, 2019.

[122] P. Tan, M. Steinbach and P. Kumar, Introduction to Data Mining, Addison-Wesley, ISBN 0-321-32136-7, 2005.

[123] A. Huang, "Similarity measures for text document clustering.," in *In Proceedings of the sixth new Zealand computer science research student conference* , 49-56, 2008.

[124] M. J. Kusner, Y. Sun, N. Kolkin and Q. Kilian, "From Word Embeddings to Document Distances," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, Lille, France, 2015.

[125] D. Blei, M. Jordan, T. Griffiths and J. Tenenbaum, "Hierarchical topic models and the nested chinese restaurant process," in *In Proceedings of the 16th International Conference on Neural Information Processing Systems, NIPS'03*, Cambridge, MA, USA, 2003.

[126] I. Dhillon and S. Sra, "Generalized nonnegative matrix approximations with Bregman divergences," *Proceedings of the 18th International Conference on Neural Information Processing Systems,* pp. 283-290, 2005.

[127] W. Xu, X. Liu and Y. Gong, "Document Clustering Based On Non-negative Matrix Factorization," in *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, 2003.

[128] H. Hlomani and D. Stacey, "Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey," *Semantic Web Journal,* pp. 1-5, 2014.

[129] J. Brank, M. Grobelnik and D. Mladenić, "A survey of ontology evaluation techniques.," in *Proc. of 8th Int. multi-conf. Information Society*, 2009.

[130] K. Dellschaft and S. Staab, "On How to Perform a Gold Standard Based Evaluation," in *International Semantic Web Conference*, 2006.

[131] K. Dellschaft, "Measuring the similiarity of concept hierarchies and its influence on the evaluation of learning procedures," *In Diploma thesis, Universitat Koblenz-Landau,* 2005.

[132] J. Raad and C. Cruz, "A Survey on Ontology Evaluation Methods.," in *Conference: Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, Lisboa, Portugal, 2015.

[133] I. Kinchin, DeLeij and D. Hay, "The evolution of a collaborative concept mapping activity for undergraduate microbiology students," *Journal of Further and Higher Education,* vol. 29, no. 1, 2005.

[134] W. Yathongchai, T. Angskun and J. Angskun, "SQL Learning Object Ontology for an Intelligent Tutoring," *International Journal of e-Education, e-Business, e-Management and e-Learning,* vol. 3, no. 2, 2013.

[135] Y. Marketakis, N. Minadakis, H. Kondylakis, K. Konsolaki, G. Samaritakis, M. Theodoridou, G. Flouris and M. Doerr, "X3ML mapping framework for information integration in cultural heritage and beyond," *International Journal on Digital Libraries,* 2016.

[136] S. Miranda, M. Gaeta, F. Orciuoli, G. Mangione and V. Loia, "Unlocking serendipitous learning by means of social Semantic web," in *Proceedings of the 6th International Conference on Computer Supported Education*, 2014.

[137] Y.-L. Chi, "Ontology-based Curriculum Content Sequencing System with Semantic Rules," *Expert Syst. Appl,* vol. 36, no. 4, pp. 7838-7847, 2009.

[138] D. Gaaevic, Model Driven Architecture and Ontology Development, Berlin: Heidelberg: Springer-Verlag, 2006.

[139] A. Ezen-Can, " “Unsupervised Modeling for Understanding MOOC Discussion Forums: A Learning Analytics Approach," *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge. LAK ’15. Poughkeepsie, New York: ACM,* 2015.

[140] L. Aroyo and D. Dicheva, " Authoring Support in Concept-based Web Information Systems for Educational Applications," *International Journal on Continuing Engineering Education and Life-long Learning ,* vol. 14, pp. 297-312, 2004.

[141] E. Duval, "“Attention Please!: Learning Analytics for Visualization and Recommendation”," in *Proceedings of the 1st International Conference on Learning Analytics and Knowledge. LAK ’11*, Banff, Alberta, 2011.

[142] M. Gaeta, F. Orciuoli, S. Paolozzi and S. Salerno, "Ontology extraction for knowledge reuse: The e-learning perspective," *IEEE Transactions on Systems, Man, and Cybernetics-Part A,* 2011.

[143] Z. A and N. R, "Building Domain Ontologies from Text for Educational Purposes," *IEEE Transactions on Learning Technologies,* vol. 1, no. 1, pp. 49-62, 2008.

[144] W. Wong, W. Liu and M. Bennamoun, "“Ontology Learning from Text: A Look Back and into the Future," *ACM Comput. Surv. 44.4 ,* p. 1–20, 2012.

[145] R. Lau, "“Toward a Fuzzy Domain Ontology Extraction Method for Adaptive e-Learning”," *IEEE Trans. on Knowl. and Data Eng,* p. 800–813, 2009.

[146] C. Schumacher and D. Ifenthaler, "Features students really expect from learning analytic," *Computers in Human Behavior,* vol. 7, pp. 397-407, 2018.

[147] P. Long and G. Siemens, "Penetrating the fog: Analytics in learning and education," *Educause Review,* vol. 46, no. 5, 2011.

[148] S. G and B. R, "Learning analytics and educational data mining: Towards communication and collaboration," in *Proceedings of the second international conference on learning analytics & knowledge*, 2012.

[149] O. Viberg, M. Hatakka, O. Bälter and A. Mavroudi, "The current landscape of learning analytics in higher education," *Computers in Human Behavior,* vol. 89, pp. 98-110, 2018.

[150] Y.-S. Tsai and D. Gasevic, "Learning analytics in higher education --- challenges and policies: a review of eight learning analytics policies," in *Conference: the Seventh International Learning Analytics & Knowledge Conference*, 2017.

[151] D. Clow, "The learning analytics cycle.," in *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge - LAK*, 2012.

[152] D. Gasevic, S. Dawson and G. Siemens, "Let's not forget: Learning analytics are about learning," *TechTrends,* vol. 59, no. 1, 2015.

[153] G. Siemens., "Learning Analytics: Envisioning a Research Discipline and a Domain of Practice," *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. LAK '12. Vancouver, British Columbia, Canada: ACM,* pp. 4-8, 2012.

[154] S. Ziebarth, N. Malzahn and H. Hoppe, "“Using Data Mining Techniques to Support the Creation of Competence Ontologies”," in *Proceedings of the 2009 Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands, 2009.

[155] M. Hepp, "“Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies”," *IEEE Internet Computing 11.1,* p. 90–96, 2007.

[156] K. Hasan and V. Ng, "“Automatic Keyphrase Extraction: A Survey of the State of the Art”.," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, 2014.

[157] J. Finkel, T. Grenager and C. Manning, “Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling”, 2005.

[158] A. Huang, Similarity Measures for Text Document Clustering, 2008.

[159] A. Maedche and S. Staab, "“Mining Ontologies from Text”," in *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management. EKAW '00.*, London, UK, 2000.

[160] K. Toutanova, "“Feature-rich Part-of-speech Tagging with a Cyclic Dependency Network”," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*.

[161] G. Dall'Alba and R. Barnacle, "An Ontological Turn for Higher Education," *Studies in Higher Education 32,* 2007.

[162] ACM/IEEE-CS, Joint Task Force on Computing Curricula 2013, ACM Press and IEEE Computer Society Press, 2013.

[163] D. Gasevic, D. Djuric and V. Devedzic, Model Driven Architecture and Ontology Development, Berlin: Heidelberg: Springer-Verlag, 2006.

[164] M. Hatala, D. Gašević, M. Siadaty, J. Jovanovic and C. Torniai, ""Ontology Extraction Tools: An Empirical Study with Educators"," *IEEE Transactions on Learning Technologies,* vol. 5, no. 3, pp. 275-289, 2012.

[165] D. Michail, J. Kinable, B. Naveh and J. Sichi, "A Java library for graph data structures and algorithms," *arXiv preprint arXiv:1904.08355,* 2019.

[166] Silicon.dk ApS, "BBCode users guide," 2011. [Online]. Available: http://www.bbcode.org/.

[167] A. Farzindar and D. Inkpen, "Natural language processing for social media.," *Synthesis Lectures on Human Language Technologies,* vol. 8, no. 1, 2015.

[168] G. Beverungen and K. (. Jugal, "Evaluating Methods for Summarizing Twitter Posts.," in *Proceedings of the Fifth International Acm Conference on Web Search and Data Mining.* , 2010.

[169] B. Liu, "Sentiment analysis and subjectivity," in *In Handbook of Natural Language Processing, Second Edition*, Boca., Taylor and Francis Group, 2010.

[170] X. Sun, X. Liu, Y. Duan and B. Li, " Using hierarchical latent dirichlet allocation to construct feature tree for program comprehension," *Scientific Programming,* 2017.

[171] M. Salas-Zarate, J. Medina-Moreira, K. Lagos-Ortiz, H. Luna-Aveiga, M. Rodriguez-Garcia and R. Valencia-Garcia, "Sentiment analysis on tweets about diabetes: An aspect-level approach.," *Computational and Mathematical Methods in Medicine.,* 2017.

[172] G. Beverungen and K. Jugal, "Evaluating Methods for Summarizing Twitter Posts," in *Proceedings of the Fifth International Acm Conference on Web Search and Data Mining*, 2010.

[173] K. Dlamini, "The Role of Social Media in Education," London College of International, 2019. [Online]. Available: https://www.lcibs.co.uk/the-role-of-social-media-in-education/.

[174] P. Gamallo, M. Garcia, C. Pineiro, R. Martinez-Castano and J. Pichel, "LinguaKit: A Big Data-Based Multilingual Tool for Linguistic Analysis and Information Extraction," in *Conference: The Second International Workshop on Advances in Natural Language Processing (ANLP 2018)*, Valencia, 2018.

[175] D. Klein and C. Manning, "Accurate unlexicalized parsing," *In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics,* vol. 1, no. 3, p. 423–430, 2003.

[176] N. Noy and D. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," in *Proceeds of the 14th International Conference on Artifical Intelligence in Education 2*, 2009.

[177] CoBra, "Context Broker Architecture, An intelligent broker for context-aware systems in smart spaces," 2004 . [Online]. Available: https://cobra.umbc.edu/ontologies.html .

[178] Lexical Computing., "POS tags," [Online]. Available: https://www.sketchengine.eu/pos-tags/. [Accessed 04 2020].