A knowledge-based approach for keywords modeling into a semantic graph

Oumayma Chergui, Ahlame Begdouri, and Dominique Groux-Leclet

***Abstract*— Web based search for a specific problem usually returns long lists of results, which may take up a lot of time to browse until finding the exact solution, if found at all. Community Question Answering systems on the other hand offer a good alternative to solve problems in a more efficient way, by directly asking the community, or automatically extracting similar questions that have already been answered by other users. Using external knowledge bases for such similarity measures is a growing field of research, due to their rich content and semantic relations. Indeed, many research works base their semantic textual similarity measures on annotating texts or extracting specific knowledge from an external knowledge base.**

Our research aims at creating a semantic domain-specific graph of keywords using data extracted from the DBpedia knowledge base. This keywords graph will be used later, in a graph-based similarity approach inside a CQA archive in order to retrieve similar questions. In this paper, we define the structure of the semantic graph and propose our method for automatically creating it, backed with experimental results.

***Index Terms*— DBpedia, graph-based similarity, knowledge base, semantic graph**

I. INTRODUCTION

This research work falls under the broad scope of web-based knowledge and information acquisition through the use of search engines for instance, which aims at providing the best results for a given search query. Usually, a "simple" search query returns a large amount of results that are more or less relevant and from which the user can choose. However, the search results may not provide an exact solution to a specific problem and it may be time-consuming to review all of them, with no guarantee of finding the desired answer.

Community Question Answering (CQA) websites on the other hand, such as Stack Overflow [1], Yahoo Answers [2], or forums, offer a good alternative to obtain the desired knowledge in a more efficient way. When a user makes a question query, a set of questions similar to the new one, and that have already been answered by other users, are automatically retrieved. Unlike web based search engines which return long lists of results, these domain specific Question Answering systems give more exact and correct answers since they are limited to a specific community and the answers are generally provided by experts on the same topic [31][32].

With the success of social web technologies and the continuous supply of content on the Internet, big community memories consisting of large collections of thematic threads are available. Hence, there is a need for automated tools that help exploring this type of content, and more specifically, help navigate CQA websites' archives.

In this respect, our work revolves around question answering based on large discussion threads and CQA systems archives, in order to identify the most useful content for answering a new question. In other words, we need to perform similarity measures between the new question and the old ones to retrieve the most similar questions.

Different textual similarity approaches exist, depending on the type of texts and the research context. Statistical and corpus-based approaches are found to be more suitable for long texts and documents similarity, whereas semantic and knowledge-rich approaches are better suited for short texts [12][13][34][35].

Therefore, due to the nature of the questions in CQA websites (mostly short texts), we proposed in [29] a semantic approach for textual similarity using a semantic graph of keywords. These latters could be key-terms or key-phrases and comprise all the core concepts of a specific domain, which are the most likely to be used in the community discussions. The proposed graph structure reflects the relatedness between the concepts of the discussed topic and therefore the semantics of the community discussions. Later on, the question answering task will consist in computing the similarity between two questions by linking each of them to the keywords graph.

In fact, modeling the semantic organization of a specific topic is a challenging task and is usually conducted manually. We propose constructing this semantic domain-specific graph using keywords extracted from an external knowledge base. And as the discussions go on, the graph will be updated by adding more potential semantic links. The aim of this work is: i) to define the structure of the graph which will later be used for the similarity measure, and ii) propose a method for

Manuscript received December 21, 2017 for review;
O. Chergui and A. Begdouri are with the SIA laboratory, FST, University of Sidi Mohamed Ben Abdellah, Fez, Morocco
(e-mail: oumayma.chergui@usmba.ac.ma, ahlame.begdouri@usmba.ac.ma).

D. Groux-Leclet is with the MIS laboratory, University of Picardie Jules-Verne, Amiens, France (dominique.groux@u-picardie.fr).

automatically creating this graph by extracting information from an external knowledge base.

The rest of the paper is organized as follows. Section 2 gives an overview of our CQA system and the use of Case-Based Reasoning for knowledge reuse. Then, in Section 3, we present a state of the art covering the main textual similarity approaches, the use of knowledge bases for textual similarity, as well as information retrieval in the form of sub-graphs. Section 4 explains the proposed method for the graph's creation. Section 5 describes the sample data then displays and discusses the main findings. Finally, Section 6 concludes the paper.

II. Our CQA system

Our community is a Community of Practice (CoP) of students enrolled in a university academic subject along with their teacher(s). The objective behind creating this kind of CoPs is to help increasing students' motivation by enhancing peer interactions [27][33]. The CoP's domain of interest is the subject itself, and the community will last as long as this subject is taught (over the years).

Our CQA system is designed in a way allowing informal and spontaneous students interactions, with a minimum amount of teacher's supervision, just enough to be able to assess and validate the students' understanding and the correctness of the exchanged information.

The practice inside the CoP revolves around students providing mutual help to each other. When a student faces any difficulty while conducting the different learning tasks, he can rely on the community to answer his questions. Whether by directly asking his current colleagues, or based on the community's history of previous questions and answers validated by the teacher.

Therefore, the three main functionalities of our CQA system are: *i) community interactions support*, *ii) teacher's assistance* and *iii) knowledge reuse* inside the community memory. The general architecture of our system is given in Fig.1

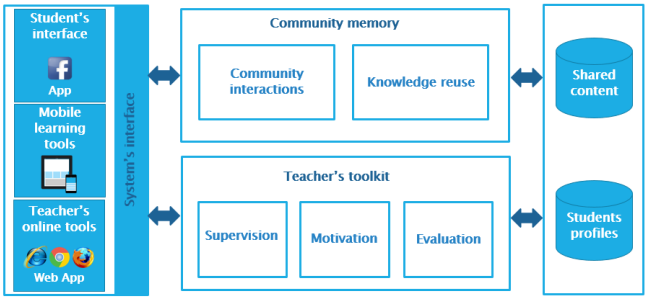

Fig. 1. General architecture of our CQA system

A. *Community interactions support*

The communication medium for the students, members of the community, should combine ease of use and fast access in order to foster their engagement. For this reason, it is recommended to provide them with the same technologies they are using in their personal daily lives [40][41]. Being a part of the Millennial Generation, they are heavily immersed in social technologies (podcasts, social network sites, video and photo sharing) as well as mobile technologies. The social networking sites in particular, have the features of social interaction and collaboration that facilitate knowledge sharing and learning [41]. Many studies have shown that using social networking tools in formal education has positive effects on students' engagement and learning, they are usually open and highly motivated to use these tools for educational purposes [40][41][42]. Therefore, we chose to build our community around Facebook, one of the leading social networking sites.

The first element of our tool is a Facebook App, similar to a Facebook "Group", a closed space for discussion with a limited number of members, and including additional features related to the knowledge transfer and reuse within the CoP.

B. *Teacher's assistance*

Our tool also provides functionalities which help the teacher fulfill his role in the community. In order to maintain the informal aspect and to let students interact freely, the teacher is not able to take part of the community interactions in the Facebook App. Instead, he performs a certain level of supervision to correct any possible mistakes or misunderstandings among the students. He is also provided with functionalities allowing him to motivate the students and evaluate their performance as community members, which is an important part of the overall evaluation. These functions form a teacher's kit, in a separate online Web App.

C. *Knowledge reuse*

The most important function of our virtual environment is to capitalize the community knowledge, both among the current members, and from one generation of students to another. For this purpose, we propose a CBR (Case Based Reasoning) based architecture.

CBR means using old experiences to understand and solve new problems. The general CBR cycle can be described by the following four processes: 1) Retrieve most similar case or cases, 2) Reuse the information and knowledge in that case to solve the problem, 3) Revise the proposed solution, and 4) Retain the parts of this experience likely to be useful for future problem solving. A new problem is solved by retrieving one or more previously experienced cases, reusing the case in one

way or another, revising the solution based on reusing a previous case, and retaining the new experience by adding it into the existing knowledge-base (Case-Base) [28].

In our solution for knowledge reuse, we propose integrating the CBR approach to use the past community knowledge at three levels:

- When a student asks a question and before moving to interact with other community members. This will prevent the duplication of previously answered questions, prevent the questions overload sent to members and therefore not demotivating them to keep interacting actively in the community,
- Along with the members interactions, based on the new information generated during the discussions (the comments), which will help finding an answer more quickly and encourage the members to keep discussing by inspiring them with new ideas,
- To help the teacher validate the students' answers, he would not have to rethink about questions that have previously been answered, or rewrite answers similar to already existing answers, which will minimize the time and effort in his supervision task.

In the CBR approach, a case is defined by two parts: a problem, and a solution for this problem. In our CQA system, a student's question can be a difficulty in executing a learning task, a need for clarification, a lack of understanding of a certain concept, etc. Therefore, our CBR case is represented as the couple made up of the question itself and its answer (Fig. 2)

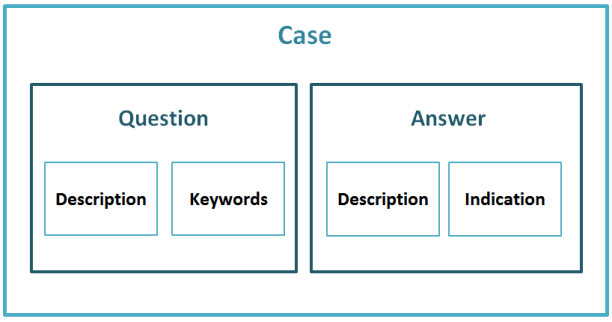

Fig. 2 Our "Case" structure

Our case structure is as follows:

The Question part

- Question description: a text description of the question.
- Keywords: a set of important words describing the question.

The Answer part

- Answer description: a text description of the answer.
- Indication: an optional additional indication or clarification regarding the answer. This is useful in the case where a question may have several answers according to the context where it is asked.

Therefore, in the « Retrieval » phase of the CBR cycle, we use a semantic similarity approach using the keywords of the new question and the old questions in the case-base. We propose modeling the CoP's domain of interest in a semantic graph of keywords and creating this graph using information extracted from an external source.

We underpin our arguments, in the following paragraph, with a state of art on textual similarity, knowledge bases and their use for similarity measures as well as extracting semantic graphs from these bases.

III. State of the art

A. Textual similarity approaches

The domain of text mining uses the techniques of machine learning and statistics in order to deal with research problems such as text representation, classification, information extraction, search for hidden patterns, etc. [5]. One of the main questions addressed in text mining is textual similarity, which aims at measuring the extent of closeness between two texts. These latters could be, depending on the context, phrases, paragraphs or full documents. Different text similarity approaches are found in literature, and can be classified into two main categories: (1) lexical similarity, and (2) semantic similarity [37].

Lexical, or string-based, similarity measures are the most basic similarity measures, they operate on string sequences and character composition, by calculating the distance metric between two text strings for approximate string matching or comparison, without taking into account the actual meaning behind words or the entire phrase context. Some of the most common metrics in this category are statistic similarity metrics [30][5] such as the Jaccard coefficient, Dice and Cosine similarity, which are Vector Space Models (VSMs) [36]. The idea behind the VSMs is to represent a text as a vector of index terms, and then carry out a comparison between vectors in order to define how close the represented texts are. These word-level approaches can become susceptible to problems caused by polysemy (ambiguous terms) and synonymy (words with similar meaning) [7]. This type of similarity is generally used for long texts and documents, and has the advantage that no additional or external resources are needed. It is used in many natural language applications, such as the automatic creation of thesauri and synonym identification.

The other category of text similarity measures is semantic similarity. Words can be semantically similar even if they are lexicographically different if they have the same meaning, are

opposite of each other, used in the same way, used in the same context or one is a type of another etc.[37]. Semantic similarity measures can be either: i) Corpus-Based, where similarity between words is determined according to information gained from analyzing a large corpus of documents; Or ii) graph-based, which quantify semantic relatedness of words using information derived from semantic networks or knowledge graphs [37][3], and thus can leverage valuable knowledge about relations between entities [7]. This category is found to be more suitable for short texts [12][13][34][35] since the short texts do not provide sufficient word occurrences, and the word frequencies are not enough to capture the semantics of the questions. It is generally used for applications such as query expansion and text classification.

B. Knowledge bases

Knowledge bases are playing an increasingly important role in solving the various problems that arise in the domain of text mining. A knowledge Base (KB) is a large collection of structured knowledge, typically an ontology, facts, rules and/or constraints [14]. Formally, a knowledge base is defined as a collection of triples, (e_s, r, e_t), where each triple expresses some relation "r" between a source entity "e_s" and a target entity "e_t". The relations "r" could be from an underlying ontology, or they could be verb phrases extracted from text, such as "belongs to". The entities "e" could be formal representations of real-world people, places, categories, or things or they could be noun phrases taken directly from text [14].

Some of the knowledge bases perform on a lexicographic level, such as the WordNet database [15] where the main relation among words is synonymy: Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.

Others usually offer cross-domain conceptual knowledge, offering structured and easily accessible data, beyond text labels and language links, usually in the form of RDF (Resource Description Framework) triples.

As example, we can cite Wikidata [17], which acts as central storage for the structured data of its Wikimedia sister projects: Wikipedia, Wikivoyage, Wikisource, etc. [18]. It is collaboratively edited by a global community, and especially well populated in fields such as "Person" and biological entities [6].

DBpedia [19] is also one of the leading projects that define semantics of data and is a cornerstone of the semantic web today [38], it allows sophisticated queries using RDF triples about relationships and properties associated with available resources. It is created from automatically-extracted information contained in the Wikipedia (e.g. infobox tables, categories, etc.) and mapped to the DBpedia ontology (manually created). It also sets RDF links pointing into various external data sources (e.g. OpenCyc, DBLP, etc.), which made it a central inter-linking hub in the Web of Linked Data [38].

YAGO (Yet Another Great Ontology) [4][20] is another largely used KB. According to its name, it is ontology, but is referred to both as knowledge base and as knowledge graph. It comprises information extracted from Wikipedia (e.g., categories, redirects, infoboxes), WordNet (e.g., synsets, hyponymy), and GeoNames (geographical database).

The two knowledge bases, YAGO and DBpedia, are connected. For example, DBpedia offers the YAGO type hierarchy as an alternative to the DBpedia ontology and sameAs links are provided in both directions [38].

The remaining knowledge bases are either monolingual (e.g. OpenCyc), private and not openly available (e.g. Google Knowledge Graph and Google Knowledge Vault [21]) or small and limited.

C. Related works

Knowledge Bases are used for entity modeling and can also be used for weighting or ranking similar concepts based on different semantic similarity metrics. Several text similarity approaches are found in literature based on knowledge bases or semantic graphs [10][11][23][25][26]. The use of a knowledge base for similarity can be either by:

Annotating/expanding texts with additional semantic information from a KB, and then using traditional similarity metrics. In fact, some works extract the necessary information from KBs as a pre-processing step on texts/documents before applying the similarity metrics. In the approach proposed in [8], the step of Semantic Document Expansion consists of annotating documents with relational knowledge from a knowledge base (DBpedia). Expanding an entity means enriching it with all information required for hierarchical similarity computation, so that it can be performed between any two expanded entities without accessing the knowledge base. Nunes et al. [9] also annotate the documents content with structured information from DBpedia, they use the categories of the extracted concepts to interlink documents through the topics they cover. In cases where two documents share the same category (dcterms:subject property), a link between them is created. Authors in [13] also enhance the texts with meta-information from external information sources such as Wikipedia and WordNet, the aim is to inflate the short text with additional information to make it appear like a large

document of text, which allows applying statistical similarity metrics for long texts.

Linking the text terms to an external KB and directly calculating the similarity based on the distance between concepts, i.e. assessing the proximity of entity nodes in its associated semantic graph (the graph representation of the KB) or ontology. Intuitively, the shorter the path from one concept to another, the more similar they are [11], either by working on the KB itself directly or by extracting a sub-graph from the KB and using it for the similarity measure:

- In relation with the first approach, a generic method is proposed in [11] for measuring the semantic similarity between concepts in any knowledge base (e.g. WordNet and DBpedia). It aims to not only rely on the distance between nodes, but give different weights to the shortest path length between concepts based on other shared information: shared parent and child nodes, statistical association between concepts based on the occurrence and co-occurrence, etc. By using this shared information between concepts to weight their path length, the similarity measure is more accurate, since two concept pairs (A,B) and (A,C) with the same path length do not necessarily reflect equal relatedness between A-B and A-C.
- By extracting and using a small portion of these data sources as an ontology or semantic graph: Authors in [10] do not work directly on the knowledge base; they extract an ontological graph for a given domain from DBpedia and propose an algorithm for finding and weighting a collection of paths connecting two terms, which will determine how similar these terms are.

In relation with the latter point, various works extract sub-graphs from large KBs for various text mining purposes, the extracted graphs are in the form of ontologies, conceptual graphs or lexical chains, etc.

In [10], authors create a "configuration ontology" of a specific domain based on the classes of the DBpedia ontology related to this domain, as well as the properties associated with these classes that are considered relevant. Then the extractor processes the configuration data and produces SPARQL queries that fetch a DBpedia sub-graph relevant for the given domain.

In [23], for a given Wikipedia article, authors develop a full-automated approach for semantic relation extraction in the form of semantic triples {Subject, Predicate, Object}, by mining the article sentences and resolving co-references between synonyms and related terms.

The approach proposed in [22] consists in creating lexical chains of given texts based on Wordnet. A lexical chain is a sequence of related words that represent the semantic content of a text, computing the lexical chains allows identification of the main topics of a document. The approach consists in disambiguating the text nouns by replacing them by multiple word senses from WordNet, then determining relations between terms also by referring to the Wordnet links. Then, based on these terms and relations, they propose an algorithm to create the lexical chain representing the given text.

A similar approach is found in [3], which consists in generating structured representations of textual content using DBpedia as the backend ontology. Given an input text document, they identify the set of concepts it contains, then words and phrases are annotated with DBpedia concepts using a document entity linking system, and finally a semantic graph representing the text is generated.

IV. OUR APPROACH: A SEMANTIC GRAPH OF KEYWORDS

A. Our proposition

In our work, we need to perform a semantic similarity measure in the retrieval phase of the CBR cycle in order to find similar questions from the existing case-base. The questions are in the form of short texts, therefore the lexical-based similarity metrics and statistical corpus-based approaches may not be the best solutions since the short texts do not provide sufficient word occurrences. Furthermore, the word frequencies are not enough to capture the semantics of the questions [12][13]. This is why we are interested in the knowledge-based approaches.

In our context, we chose the question's keywords to be the indexing terms of the cases since they hold the essence of the text and verbalize the described problem. In fact, considering two questions, the more they have semantically similar or related keywords, the more similar they are.

The first solution for obtaining the set of keywords of a given question is by asking the user to provide them himself. The risk here is to have keywords that are too broad or even irrelevant to the subject since the choice is left to the judgment of a still learning member and maybe not expert in the field. This may falsify, in some cases, the similarity results. Instead, we rather extract the keywords automatically from the question description, by simply eliminating the stop-words and keeping the other terms. Different methods exist for keywords extraction from texts in general, and short texts in particular, but since we are dealing with direct questions in the form of short texts, every word is meaningful and therefore all the terms left are keywords.

As for the similarity measure, once we have the keywords of the new question, as well as the previously answered

questions. The retrieval task will consist in computing the similarity between the questions by linking each of them to the semantic graph of keywords.

B. Knowledge base choice

As reported in the state of the art (see .III-C), the use of a knowledge base for similarity is performed by ***annotating/expanding texts*** with additional semantic information from a KB, and then use traditional similarity metrics. This could lead to the problem of dimensionality (the data set becoming too large), the accuracy of the added information may be called into question as well [13]. The use of KB could be also through ***linking the text terms to a KB*** and directly calculate the similarity based on the distance between the graph nodes, but querying such sources online poses the problem of longer time [13]; or by ***extracting and using a small portion*** of these data sources as an *ontology* or *semantic graph*.

In this respect, we chose the last approach since the texts we are dealing with (the questions) revolve around one subject only, and thus there is no need to query the whole KB each time a similarity measure is needed. Also, the specific context of our system requires a customized graph which reflects the semantics of the CoP's domain of interest.

For many of the existing knowledge bases, Wikipedia has proven to be one of the most valuable resources. A considerable number of researches on Wikipedia mining have been conducted and the fact that Wikipedia is a valuable corpus has been confirmed [23][26].

It is basically an online, collaboratively generated text-based encyclopedia and one of the largest and most consulted reference works in the world [16][3][23][26]. Even though it is text-based and written with the goal of human consumption, but it contains a certain structure which can be exploited by automated algorithms. Indeed, Wikipedia includes a dense link structure, well-structured Infoboxes, and a well-organized category tree reflecting, in some extent, the semantics of its content [16]. The inlinks and outlinks between Wikipedia articles connect the most important terms to other pages providing the users with a quick way of accessing additional information. Moreover, each article is mentioned inside different Wikipedia categories and each Wikipedia category generally contains parent and children categories. Each category can have an arbitrary number of subcategories as well as being mentioned inside an arbitrary number of supercategories.

Wikipedia's continuous expansion over a period of years makes it likely to stay useful over a number of years to come. It has led to the development of ontologies and knowledge bases such as DBPedia and YAGO. One of the main differences between them is that DBpedia tries to stay closer to Wikipedia and provide an RDF version of its content [38] while YAGO focuses on more specific entities such as people, places and events [6].

DBpedia extracts various kinds of structured information from Wikipedia, it has over 3 billion triples (facts stored using the W3C standard RDF data model) and over a million of SKOS (Simple Knowledge Organization System) concepts, arranged hierarchically and available for use by applications via SPARQL endpoints. DBpedia also uses Info-box Ontology as a new Info-box extraction method, based on hand-generated mappings of Wikipedia info-boxes. The mappings adjust weaknesses in Wikipedia's info-box system, such as using different infoboxes for the same type of thing (class) or using different property names for the same property. Therefore, the instance data within the info-box ontology is much cleaner and better structured than the Info-box Dataset [25].

Therefore, due to the fact that DBpedia is covering all of the content of Wikipedia and that it is a more structured version of it (with additional links and easier to parse), we chose to use it, jointly with the Wikipedia articles' contents, for extracting the concepts which will be added to the nodes of our semantic graph.

C. Graph structure

The indexation of cases for the retrieval phase of the CBR cycle requires an adequate representation for keywords related to the CoP's domain of interest since all students questions will revolve around it. We find semantic networks especially adapted to our needs. Other than their semantic nature, networks as a data structure, make a better representation than trees since they include more relations between nodes and provide more information. In addition, semantic networks are less complex than other representation forms (e.g. ontologies), which is enough for our needs.

Therefore, we chose using a semantic graph to model the keywords, in such a way that the main concept (subject of the course) represents the graph center, the nodes directly related to it represent the general concept categories of the subject, and as we go further in the graph, we get into more details and specific concepts.

The graph's construction takes place at the very beginning when setting up the platform by the teacher. With the aim of a greater semantic personalization, the input is either the main title of the taught subject or a set of titles and subtitles allowing to reflect the desired semantics in relation with the objectives of the teaching.

Therefore, we define our graph structure as follows (Fig. 3):

Nodes

- **Title nodes:** they are the input of the teacher. Like a table of contents, they represent his vision in relation to the semantics of the taught subject: chapters, paragraphs, etc. The number of this type of nodes is at least one title (the course name).
- **Keyword nodes:** they represent specific concepts and terms related / relevant to one or many titles. They hold the essence of the content related to a chapter/paragraph.
- **General keywords node:** A particular title node that we decided to add under the main title node defined by the teacher. This node will contain miscellaneous keywords extracted as relevant to the given subject, but which do not fit into any subtitle.

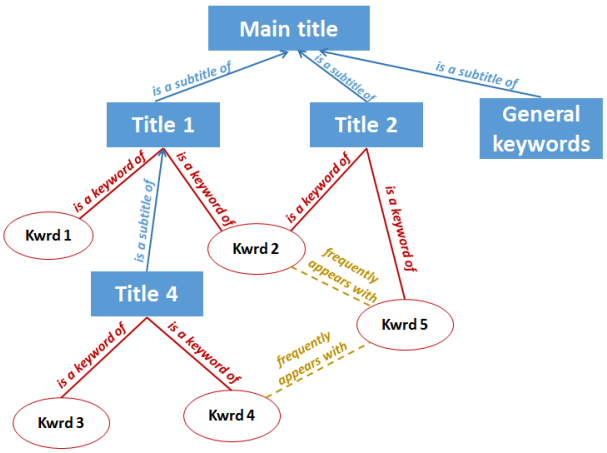

Fig. 3. Structure of the semantic graph of keywords

Arcs

- **Title X – is a subtitle of – Title Y:** This type of arcs reflects the subject hierarchy.
- **Keyword K – is a keyword of –Title X:** keyword K is related to the subject content of Title X.
- **Keyword K – frequently appears with – keyword J:** keyword K and keyword J usually appear together in members questions during the CoP discussions (co-occurrence level). This type of arcs cannot be defined from the beginning; they will later be added dynamically to the semantic graph as the community discussions go on.

D. Graph construction

This section presents the main steps of our graph construction process; we present an overview of the approach in Fig. 4. The process is composed of three steps, described as follows:

1) Preprocessing

Input: The teacher manually defines the subject organization according to his own planning in the form of a table of content: main title, chapter titles, and subtitles (with no restriction for the subtitles number and levels).

Titles preprocessing: This step allows defining the keywords contained in the titles. We start by removing stop-words. Based on the remaining words, the system generates different word combinations for each title and the teacher is required to choose the most meaningful term sets. This allows removing some very general words like "introduction", "definitions", etc. The resulting term sets are considered as keywords of the title in question. For example, from the title "Introduction to relational databases" we keep "databases" and "relational databases".

Initial nodes creation: The graph is initially in the form of a tree structure. The first node is the main title of the subject, then titles and subtitles nodes are added in the given hierarchy, including the "General keywords" title node. Then, to each title, we add the previously validated term sets of the title preprocessing step, as "keyword nodes".

At this point, we might get a few duplicated nodes due to the fact that the given table of contents could include duplicated words in different titles. This could influence the graph's consistency and the keywords' extraction in the following step. For this reason, we define three rules for eliminating nodes duplications:

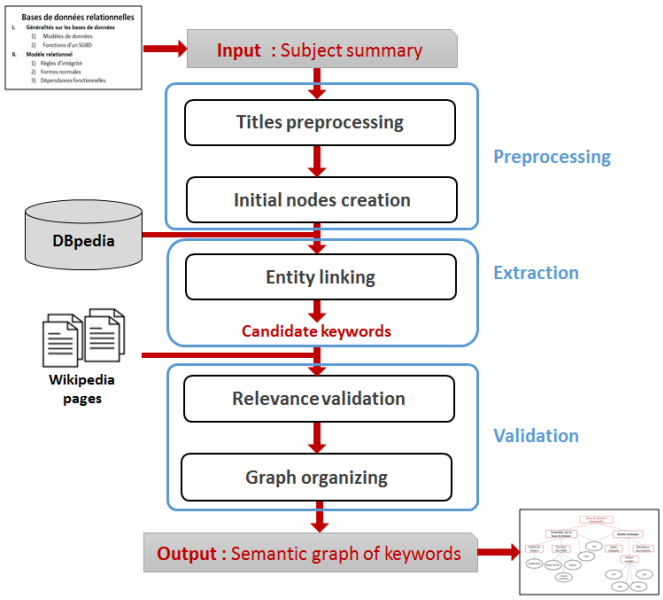

Fig. 4. The graph construction process

- ***If a keyword is duplicate in a title and its subtitle, it is removed from the title***. If we consider the example of a paragraph, this means that the term is a keyword of the

paragraph and its sub-paragraph. In order to be semantically more specific, we decided to keep it as keyword of the subtitle.

- If a keyword is duplicated in two or more subtitles of the same level under the same title, it is removed from both and added to the title. Otherwise, the duplicate term will generate duplicate keywords that are common for the subtitles; hence it is better to add them directly to the upper level title.
- ***If a keyword is duplicated in two or more subtitles under different titles***, this means that it is semantically related to them all. So, we reduce it to one node related to all subtitles.

2) Extraction: Entity linking to DBpedia

The keywords that we want to extract from DBpedia are in fact pages' titles, which are "Article titles" from the corresponding Wikipedia articles. According to Wikipedia Manual of Style "*A title should be a recognizable name or description of the topic that is natural, sufficiently precise, concise, and consistent with the titles of related articles*", which means that article titles make appropriate keywords.

In this step, for each graph node, we search for DBpedia pages with titles and labels (rdfs:label) similar to the node using SPARQL queries.

One of the informations found in a DBpedia page is "Wikilinks", hyperlinks found in the corresponding Wikipedia article and which link to other Wikipedia articles, i.e they are titles of other pages.

There are two types of Wikilinks: i) *one-way links*, random words which usually take to an irrelevant concept (e.g. some of the one-way links in the page "XML" are: "Publishing", "Human languages", clearly irrelevant to the XML language); and ii) Wikilinks that are *at the same time inlinks and outlinks* of the page, i.e. Wikilinks mentioned in the page and which also mention this page as Wikilinks in their pages. We consider that this inter-connection between two pages means that the two concepts are semantically related to a certain degree.

If a page is found using the SPARQL query, we extract the Wikilinks of the second type, they form a list of "candidate keywords". For each Wikilink, we also get the "label" of the corresponding page and store it as a synonym of the keyword, since sometimes pages' titles and labels are different words/phrases referring to the same concept.

The returned result can be a "Disambiguation list", DBpedia provides these lists when there is more than one existing page to which that word or phrase might lead. For instance, if the subject is "computer networks", by IP we mean "Internet Protocol", but when we search for "IP" page in DBpedia, we find a list containing: Intellectual Property, Imperial police, Industrial Policy, etc. In this case, we only need to extract "Internet protocol" and "IP address". So, in order to filter the found pages, we turn to each page's categories and compare them to the categories of our titles, which we extract using another SPARQL query, this allows choosing from the disambiguation list the pages that are more likely to be the same concept we are looking for. This categorization information is found in DBpedia pages as Dublin Core "subject" values (dcterms: subject) which lead to category pages, containing a more general category value as the SKOS value "broader" (skos:broader). We use this last value for our comparison.

3) Validation

At this step, we have a set of candidate keywords that need to be affected to their proper position in the graph. The question here is about their relevance to the subject. Some of the extracted keywords could be somehow far from the taught subject content.

Therefore, the obtained candidate keywords need to be filtered in order to remove irrelevant terms. This is done in the **Relevance validation**. We start by removing the keywords that do not belong to the same category as the current node, based on the skos:broader property since the DBpedia categories represent the semantic hierarchy of Wikipedia.

Our vision to measure the relevance of an extracted keyword is to determine, according to the external knowledge base, how many times this keyword is mentioned in relation with the concept of the related node, and vice versa.

For this purpose, we use the function W(k) (equation (1) below) to weight each of the remaining candidate keywords, which allows ranking them and identify the most relevant ones. To calculate this value, we turn to the original Wikipedia pages of the candidate keywords as well as the searched term set (the current node), and extract their pages contents (i.e. the full text). These contents are not stored in DBpedia.

$$W(k) = F(k) * NbOcc(T) \quad (1)$$

- **F(k)** reflects the importance of the keyword k compared to the other candidate keywords. It is calculated from the Wikipedia page of the term set as follows

$$F(k) = \frac{Number\ of\ occurences\ of\ K}{Total\ of\ occurences\ of\ all\ candidate\ keywords}$$

- **NbOcc(T)** is the number of occurrences of the term set in the Wikipedia page of the candidate keyword. It reflects the importance of the keyword k compared to the searched term set.

After applying the weight function on each of the candidate keywords, we rank them based on their weight. The purpose is

to keep the highest ranked keywords considering they are the most relevant.

In order to determine the proper percentage, we tested the function on 50 different term sets, and calculated the percentage of relevant keywords among the returned results. The average percentage based on the results is 66%, therefore we keep the highest ranked 66% of the keywords, and add them to the corresponding title nodes as keywords nodes.

At this stage, and similarly to the preprocessing step, we could get duplicated keywords nodes. The aim of the **"Graph organizing"** step is to optimize the graph by eliminating duplications. Thus, we apply on the new keywords nodes the three rules previously defined, and define two additional rules:

- keywords similar to one of the titles or generated term sets are removed,
- Keywords left under the main title or not fit to any of the titles are added to the "general keywords" node.

V. EXPERIMENTAL EVALUATION

A. Dataset and results

In this section, we present the experimental results of our approach. The evaluation aims at proving the effectiveness of our proposed method for extracting keywords from a knowledge base (DBpedia, Wikipedia) and validating the semantic relevance of the resulting graphs by: i) measuring the acceptance rate of the retrieved keywords by the teachers, and ii) determining the accuracy of the retrieved keywords against expected keywords.

With the aim to confirm the genericity of the approach, the dataset used to evaluate our approach consists of tables of contents of different courses/subjects that we collected from 8 teachers of different teaching specialties (Computer science, management, chemistry and mathematics). For validation purpose, the teachers were also asked to provide a list of important keywords related to each title in the tables of contents, i.e. keywords that represent the corresponding lesson/chapter/paragraph. We call them expected keywords and they are used to be compared with the obtained results.

In order to gather the results, we implemented an online tool integrating the graph creation algorithm:

- Teachers insert a table of content including the main title, chapters titles and subtitles,
- The system performs the titles pre-processing, and the teachers choose the semantically relevant term sets,
- They propose a list of expected keywords related to each title / subtitle
- Finally, they validate the extracted keywords by removing the keywords they find irrelevant to their subject and teaching context. We end up with a list of accepted keywords.

We received 50 tables of contents (i.e. 50 generated graphs) containing 469 titles and 404 expected keywords in total. The system retrieved a total of 1696 keywords based on the given titles. The evaluation is based on:

- The keywords that were retrieved using our algorithm and validated by the teachers, and
- The keywords that were initially proposed by the teachers as relevant for each title (expected keywords).

In the following figure (fig. 5), we present an example of the input table of contents and the obtained semantic graph. For visualization purposes the blue lines represent the arcs of type "is a keyword of".

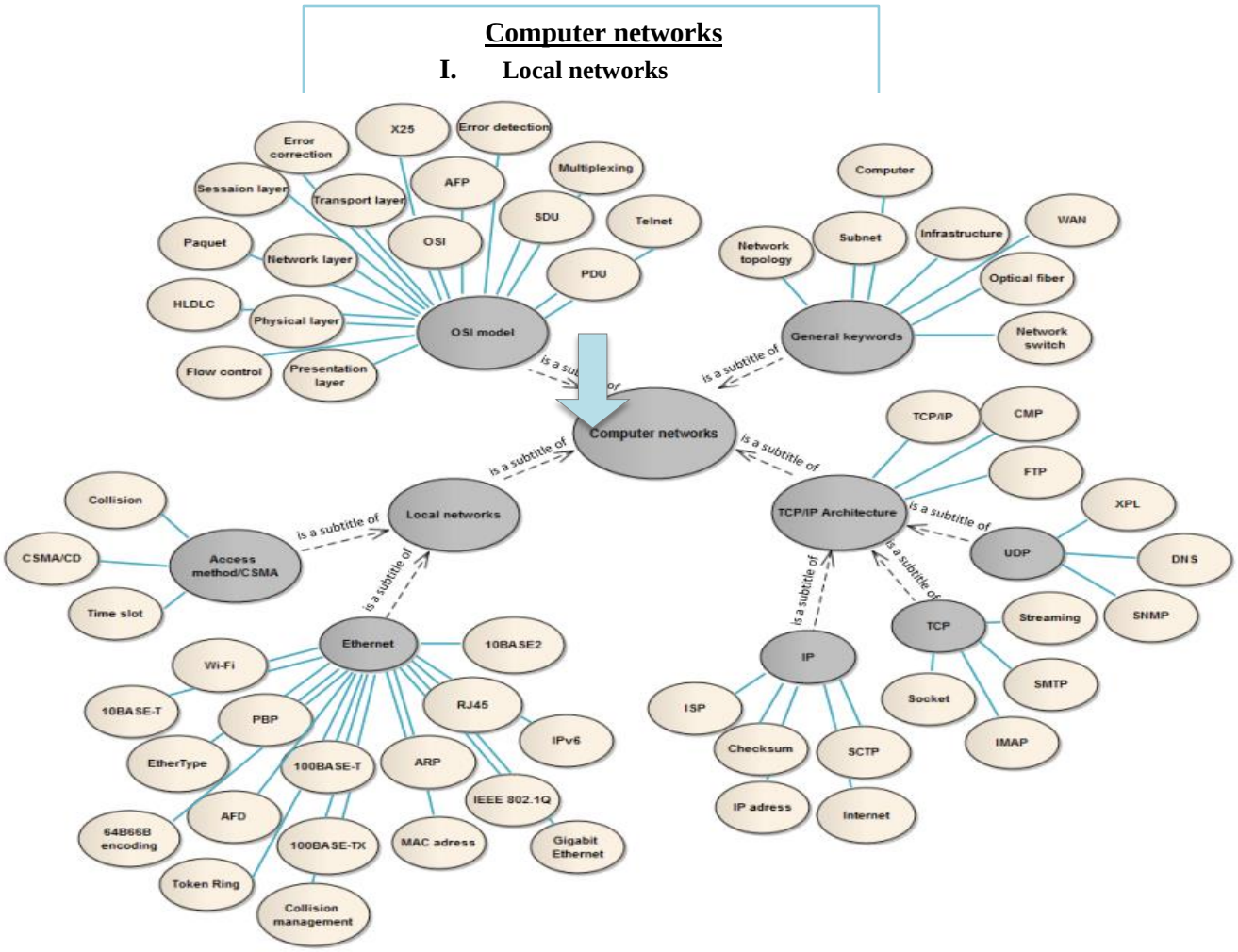

Fig. 5. The table of contents and the obtained graph of the "Computer networks" subject

B. Acceptance rate for the retrieved keywords

The Acceptance Rate (AR) that we measured is based on teachers' validation of the retrieved keywords. It aims at validating the relevance of these keywords to the specific subject's content and pedagogical objectives set by the teachers. An accepted keyword will probably be used later as a search term in the students' questions on the platform while a rejected keyword is not relevant to the subject. I.e. the more relevant the retrieved keywords are, the better the quality of the constructed semantic graph, and thus the better the accuracy of similarity measures.

We consider the average Acceptance Rate as:

$$\mathrm{AR} = \frac{Nb.of\ keywords\ accepted\ as\ relevant}{Total\ number\ of\ retrieved\ keywords} \qquad (2)$$

The average Acceptance Rate based on all the retrieved keywords is 72%, a pretty good rate considering that the keywords were retrieved from a general and external knowledge base (not teaching oriented nor structured by a pedagogical taxonomy).

We also calculated the AR per graph in order to have a detailed view over the results. Fig. 6 presents the % of graphs in each AR interval:

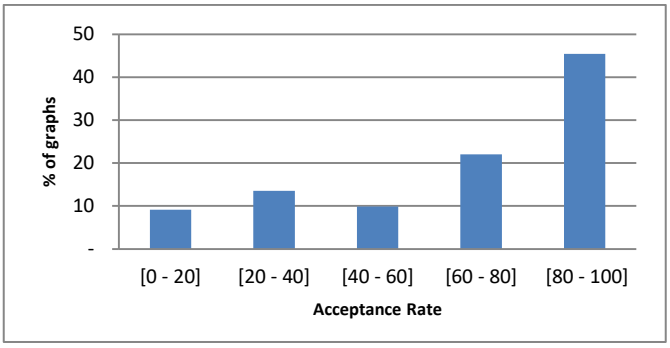

Fig. 6. % of graphs in each Acceptance Rate interval

67% of the graphs have good to excellent keywords acceptance rate greater than 60%, and 10% have average AR between 40% and 60%.

Moreover, teachers proposed a relatively small number of expected keywords in each subject. The results show that 70% of all relevant keywords that we have (expected + accepted) were automatically retrieved rather than being proposed by the teachers. This aspect proves to be useful in extracting a large number of relevant keywords to a specific subject, which is very helpful in reducing the time and mental effort required to do it manually.

On another note, we noticed that the nature of the given tables of contents influences the performance of our approach in terms of the nature of titles organization: smaller tables of contents with general titles give better results than more detailed titles (Fig. 7). By general titles, we mean the main concepts of the subject that are also general concepts, unlike detailed ones which reflect particular elements of the subject.

Therefore, we measured the AR for small tables of contents with general titles (with only one or two levels of titles), then we tested with detailed tables of contents of the same subjects. First, we added more specific subtitles to the previous general titles and we obtained the same AR (the example of "Semantic web" and "Network programming", fig. 7). Then, we used different tables of contents of the same subject containing detailed titles and subtitles (the example of "Computer science", "Basic chemistry" and "Web X.0"). In this case, the general tables of contents give better results.

The fact that general tables of contents generate better or similar results than detailed ones is related to the nature of information found in Wikipedia. It mostly provides very limited information and sometimes even contains no pages for very specific concepts, compared to more general concepts. Again, this is appropriate to the purpose of our approach, since we mainly aim at helping the teachers by minimizing manual effort. They will be advised to only provide general titles and have a semantic graph of relevant keywords.

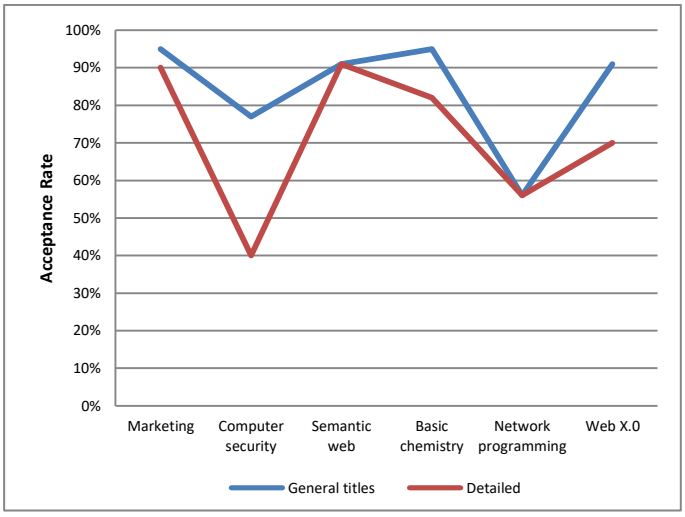

Fig. 7. Acceptance rates for general and detailed tables of contents

C. *The retrieved keywords against expected keywords*

In this step of the evaluation, automatically retrieved keywords are matched with the expected ones. Thus, the expected keywords are used as gold standard. We measure Recall and Precision against this standard. In our context (Information Retrieval) we have a binary classification (relevant or not relevant), *Recall* is the fraction of expected keywords that are retrieved and *Precision* is the fraction of retrieved keywords that were expected.

$$\text{Precision} = \frac{|\{\text{Expected keywords}\} \cap \{\text{Retrieved keywords}\}|}{|\{\text{Retrieved keywords}\}|}$$

$$\text{Recall} = \frac{|\{\text{Expected keywords}\} \cap \{\text{Retrieved keywords}\}|}{|\{\text{Expected keywords}\}|}$$

Precision and Recall do not take into account the relevant retrieved keywords that were not expected. Therefore, due to the fact that the teachers only proposed a small number of expected keywords (404) comparing to the number of retrieved keywords (1696), we obtained very low Precision and Recall by doing the calculations over the whole dataset (P=0.1, R=0.3).

In order to overcome this problem and determine the real Precision and Recall of our method, we re-tested the algorithm on 10 other tables of contents by focusing on providing, as input, all the possible expected keywords for each title, then we generated the keywords graphs and recalculated the precision and recall for each graph.

We obtained the following Precision-Recall Curve (PRC) (Fig. 8). It shows a fairly high recall. Indeed, a good fraction of expected keywords is correctly retrieved.

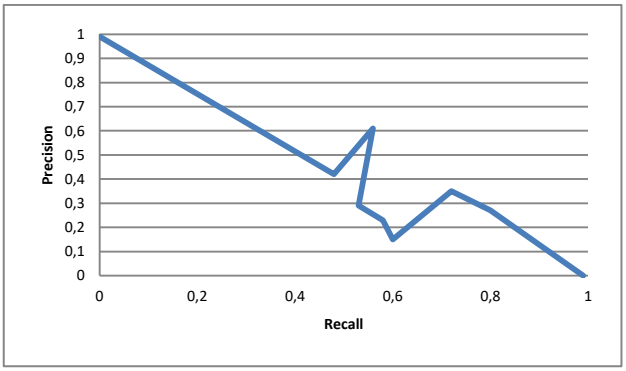

Fig. 8. Precision/Recall curve

Besides the Recall, the Precision shows that the system returns many results but only a few are accurate compared to the expected keywords. This is more of a strong point than weakness since we do obtain a large number of correct and relevant keywords that were not proposed at all as expected keywords. This is why we favor the recall (fraction of expected keywords that are retrieved) over Precision, since the purpose is to find the most relevant results while minimizing the junk that is retrieved.

VI. Conclusion

In this paper, we presented our semi-automatic approach for creating a semantic graph of keywords related to a specific domain (a teaching subject in our research context), which adapts data extracted from the DBpedia knowledge base into a domain-specific knowledge representation. Although this approach for creating the semantic network is proposed for a specific pedagogical context, it can be used for a broader range of applications that require extracting information related to a specific concept from DBpedia/Wikipedia. For instance, query expansion, keywords extraction from texts, annotating short texts, etc.

At this stage of work, we are implementing the graph-based textual similarity measure following the CBR cycle for knowledge reuse and using the proposed semantic graph of keywords, which will allow to extend the graph by adding more arcs (of the type frequently-appears-with) and eventually even more keywords based on the ongoing community discussions.

VII. References

[1] https://stackoverflow.com/
[2] https://answers.yahoo.com/
[3] M. Atif, "Utilising Wikipedia for text mining applications," Ph.D. dissertation, College of Engineering and Informatics, National University of Ireland, Galway, 2015.
[4] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 194, pp. 3161–3165, 2013.
[5] A. Hotho, A. Nürnberger, and G. Paaß, "A Brief Survey of Text Mining," *LDV Forum - Gld. J. Comput. Linguist. Lang. Technol.*, vol. 20, pp. 19–62, 2005.
[6] M. Färber, B. Ell, C. Menne, and A. Rettinger, "A Comparative Survey of DBpedia , Freebase,OpenCyc,Wikidata,And YAGO," *Semant. Web*, vol. 1, pp. 1–5, 2015.
[7] C. Paul, A. Rettinger, A. Mogadala, C. A. Knoblock, and P. Szekely, "Efficient Graph-based Document Similarity," *LNCS B. Ser. - Semant. Web. Latest Adv. New Domains, ESWC 2016*, vol. 9678, pp. 334–349, 2016.
[8] R. Thiagarajan, G. Manjunath, and M. Stumptner, "Computing Semantic Similarity Using Ontologies," in *International Semantic Web Conference (ISWC)*, 2008, Germany.
[9] B. P. Nunes, B. Fetahu, R. Kawase, S. Dietze, M. A. Casanova, and D. Maynard, "Interlinking documents based on semantic graphs with an application," *SIST B. Ser. - Knowledge-Based Inf. Syst. Pract.*, vol. 30, pp. 139–155, 2015.
[10] J. P. Leal, V. Rodrigues, and R. Queirós, "Computing Semantic Relatedness using DBPedia," *OpenAccess Ser. Informatics* -, pp. 133–147, 2012.
[11] G. Zhu and C. A. Iglesias, "Computing Semantic Similarity of Concepts in Knowledge Graphs," vol. 29, no. 1, pp. 72–85, 2017.
[12] D. Metzler, S. Dumais, and C. Meek, "Similarity Measures for Short Segments of Text," *LNCS B. Ser. - Adv. Inf. Retr.*, vol. 4425, pp. 16–27, 2007.
[13] B. Sriram, "Short text classification in Twitter to improve information filtering," MS dissertation, The Ohio State University, 2010.
[14] M. Chein and M.L. Mugnier, "Graph-based Knowledge Representation: Computational Foundations of conceptual graphs", in *Advanced Information and Knowledge Processing*, 2009
[15] https://wordnet.princeton.edu/
[16] https://www.wikipedia.org/
[17] https://www.wikidata.org
[18] https://www.wikimedia.org/
[19] https://www.dbpedia.org
[20] www.yago-knowledge.org/
[21] https://www.google.com/intl/es419/insidesearch/features/search/knowledge.html
[22] T. Wei, Y. Lu, H. Chang, Q. Zhou, and X. Bao, "A semantic approach for text clustering using WordNet and lexical chains," *Expert Syst. Appl.*, vol. 42, pp. 2264–2275, 2015.
[23] K. Nakayama, "Wikipedia Mining for Triple Extraction Enhanced by Co-reference Resolution," in *First Workshop on Social Data on the Web (SDoW2008)*, 2008.
[24] M. Schuhmacher and S. P. Ponzetto, "Knowledge-based graph document modeling," in *ACM international conference on Web search and data mining*, 2014, July, pp. 543–552.
[25] Y. I. A. Khalid and S. A. Noah, "A Framework for Integrating DBpedia in a Multi- Modality Ontology News Image Retrieval System," in *International Conference on Semantic Technology and Information Retrieval*, 2011, pp. 144–149.
[26] Z. Wu *et al.*, "An efficient Wikipedia semantic matching approach to text document classification," *Inf. Sci. (Ny).*, vol. 393, pp. 15–28, 2017.
[27] O. Chergui, A. Begdouri, and D. Groux-Leclet, "CBR approach for knowledge reuse in a Community of Practice for university students". in *the 4th IEEE Inter. Col. on Inf. Sci. and Tech. (CiSt'16)*, 2016, October, pp. 553-558.

[28] A. Aamodt and E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches,", in *AI Communications*, vol. 7, no. 1, pp. 39–59, Mar. 1994

[29] O. Chergui, A. Begdouri, and D. Groux-Leclet, "Keyword-based similarity using automatically generated semantic graph in an online Community of Practice", *LNCS Em. Tech. for Edu.*, vol. 10108, pp. 526 – 532, 2017.

[30] C. D. Manning Hinrich Schiitze, *Foundations of Statistical Natural Language Processing*. 1999.

[31] R. P. Kamdi and A. J. Agrawal, "Keywords based Closed Domain Question Answering System for Indian Penal Code Sections and Indian Amendment Laws," *I.J. Intell. Syst. Appl. Intell. Syst. Appl.*, vol. 12, no. 12, pp. 57–67, 2015.

[32] A. Baltadzhieva, "Question Quality in Community Question Answering Forums : a survey," *Sigkdd Explorations*, vol. 17, no. 1, pp. 8–13, 2015.

[33] E. Wenger, "Communities of Practice: Learning, Meaning, and Identity", *New York: Cambridge University Press*, 1998.

[34] W. Yih and C. Meek, "Improving Similarity Measures for Short Segments of Text," *Adv. Inf. Retr.*, pp. 1489–1494, 2007.

[35] A. H. Jadidinejad, F. Mahmoudi, and M. R. Meybodi, "Conceptual feature generation for textual information using a conceptual network constructed from Wikipedia," *Expert Syst.*, vol. 33, no. 1, pp. 92–106, 2016.

[36] G. Salton, A.Wong, and C. S. Yang. "A vector space model for automatic indexing". *Communications of the ACM*, vol.18, no. 11, pp. 613–620, 1975.

[37] W. H. Gomaa and A. A. Fahmy, "A Survey of Text Similarity Approaches," *Int. J. Comput. Appl.*, vol. 68, no. 13, pp. 975–8887, 2013.

[38] C. Bizer, "DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from\nWikipedia," *SemanticWeb*, vol. 1, pp. 1–29, 2012.

[39] V. Rus, M. Lintean, A. C. Graesser, and D. S. McNamara, "Text-to-Text Similarity of Sentences," *Appl. Nat. Lang. Process.*, pp. 110–121, 2012.

[40] M. A. Kadry and A. R. M. El Fadl, "A proposed model for assessement of social networking supported learning and its influence on learner behaviour," in *the Int. Conf. on Int. Mob. and Comp. Aid. Lear.* , pp. 101–108, 2012.

[41] J. Friedman, "Social Media Gains Momentum in Online Education", 2014, [Online] http://www.usnews.com/education/online-education/articles/2014/11/05/social-media-gains-momentum-in-online-education

[42] L. Deng and N.J. Tavares, "From Moodle to Facebook: Exploring students' motivation and experiences in online communities", *Computers & Education*, vol. 68, p167–176, 2013.

[43] M. Gardner, "Reading and Reasoning with Knowledge Graphs," Carnegie Mellon University, 2015.