

A SEMANTIC E-LEARNING PLATFORM

Adam ALTĂR-SAMUEL^{*1}

Dragoș-Paul POP²

ABSTRACT

This papers presents ideas regarding the design and implementation of an eLearning software platform, based on a detailed study of the most important semantic web technologies, including emerging semantic eLearning standards. The users of the platform can input texts from different areas of knowledge and the infrastructure will suggest the domain of the text, based on the domain ontology especially designed for this purpose and several text processing algorithms. The platform will then suggest appropriate studying materials for the inferred domain. For this purpose, the platform features a component that continuously searches the internet for new learning materials to add to the knowledge base.

Keywords: Semantic Web, e-learning platform, text classification, domain ontology

1. Introduction

The Semantic Web is an extension of the current web that allows searching and combining data fast and effortlessly. This new approach is based on collecting, processing and publishing information interpretable by machines and metadata expressed in RDF (Resource Description Framework). At present, the content of the World Wide Web is designed to be read by humans, not reused by applications. The semantic web will complement the current web, creating an environment in which software agents will be capable of processing various sophisticated tasks, an environment in which data will have a well-defined meaning. Hence, there is hope that, in the near future, computers will not only be capable of displaying data, but of „understanding” it.

The characteristics of the semantic web, i.e. well defined meaning of concepts and automatically processable metadata, used by appropriate software agents, establish an efficient approach for satisfying the requirements of eLearning. Learning materials can be interpreted semantically and, at the request of users, reorganized in order to create new didactic modules. Based on the user's requests and preferences, learning materials and other information considered relevant can be combined in a simple and intuitive manner.

¹ * Corresponding author. Teaching Assistant, Department of Computer Science, Statistics and Mathematics, Romanian-American University, 1B, Expozitiei Blvd., district 1, code 012101, Bucharest, Romania , E-mail: adamalt@gmail.com

² Teaching Assistant, Department of Computer Science, Statistics and Mathematics, Romanian-American University, 1B, Expozitiei Blvd., district 1, code 012101, Bucharest, Romania , E-mail: pop.dragos.paul@profesor.rau.ro

This process is based on semantic queries and navigation through the learning materials and is possible through the use of ontologies, which provide exact definitions of concepts and notions.

The following table presents the benefits eLearning can bring to the classical learning systems and the way in which the semantic web can help in providing these benefits.

Characteristic	Classical Learning	e-Learning	Semantic Web
Delivery	Push Instructor determines agenda	Pull Student determines agenda	Knowledge items (learning materials) are distributed on the web, but they are linked to commonly agreed ontologies. This enables construction of a user-specific course, by semantic querying for topics of interest.
Responsiveness	Anticipatory Assumes to know the problem	Reactionary Responds to problem at hand	Software agents on the semantic web may use a commonly agreed service language, which enables co-ordination between agents and proactive delivery of learning materials in the context of actual problems. The vision is that each user has his own personalized agent that communicates with other agents.
Access	Linear Has defined progression of knowledge	Non-linear Allows direct access to knowledge in whatever sequence makes sense to the situation at hand	Users can describe the situation at hand (goal of learning, previous knowledge,...) and perform semantic querying for the suitable learning material. The user profile is also accounted for. Access to knowledge can be expanded by semantically defined navigation.
Symmetry	Asymmetric Training occurs as a separate activity	Symmetric Learning occurs as an integrated activity	The semantic web offers the potential to become an integration platform for all business processes in an organization, including learning activities.
Modality	Discrete Training takes place in dedicated chunks with	Continuous Learning runs in the parallel to business tasks and never stops	Active delivery of information (based on personalized agents) creates a dynamic

	defined starts and stops		
Authority	Centralized Content is selected from a library of materials developed by the educator	Distributed Content comes from the interaction between the participants and the educators	The semantic web will be as decentralized as possible. This allows an efficient combination of information.
Personalization	Mass produced Content must satisfy the needs of many	Personalized Content is determined by the individual user's needs and aims to satisfy the needs of every user	A user, using his/her personalized agent, searches for learning materials customized for his/her needs. The ontology is the link between user needs and characteristics of the learning material
Adaptivity	Static Content and organization/taxonomy remains in their originally authored form, without regard to environmental changes	Dynamic Content changes constantly through user input, experiences, new practices, business rules and heuristics	The semantic web enables the use of distributed knowledge provided in various forms, enabled by semantic annotation of content. The distributed nature of the semantic web enables continuous improvement of learning materials.

2. Overview

This paper proposes a semantic e-Learning platform, which uses Semantic Web technologies combined with several text-classification algorithms, as well as an ontology especially designed for this purpose.

The users of the platform can input texts from different areas of knowledge and the infrastructure will suggest the domain of the text. The platform will then suggest appropriate studying materials for the inferred domain. For this purpose, the platform features a component that continuously searches the internet for new learning materials to add to the knowledge base.

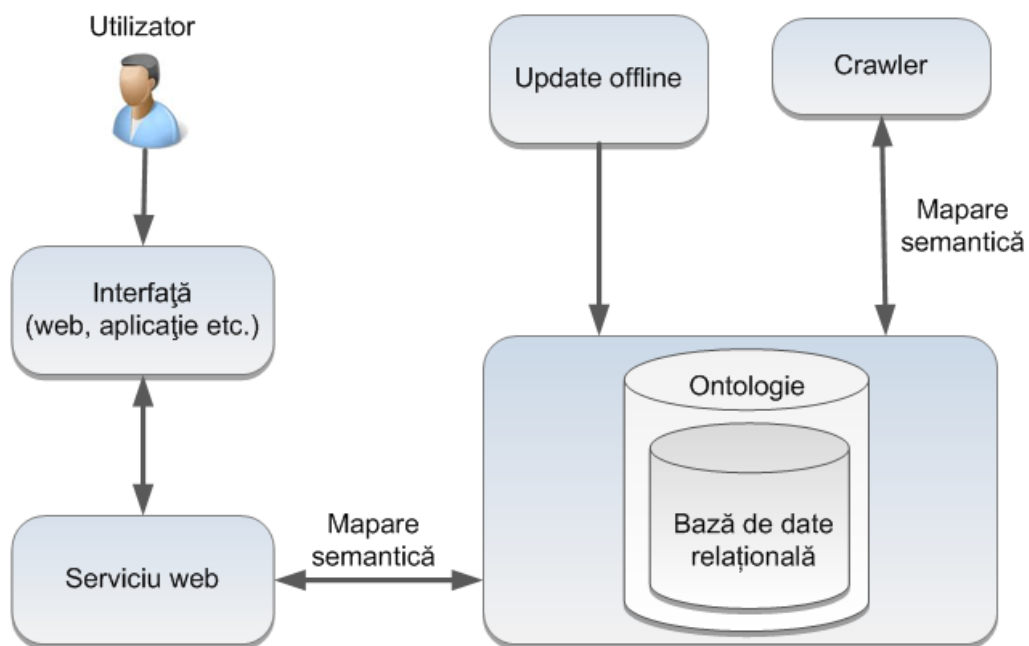


Figure 1 presents the main components of the infrastructure.

3. The Knowledge Base

The knowledge base of the infrastructure consists of a relational database, which contains information about the ontology of domains used within the platform, the learning materials provided and the users that have access to the platform.

A very important component of the infrastructure is the ontology of domains, on the basis of which the learning materials are classified. It is represented as a tree of domains and sub-domains. Each “leaf” node has its own training document, used by the classification component, in order to identify the user’s text’s domain. The training document is a text file that contains relevant information for that particular domain.

The learning materials can be of various formats, namely video, audio and text and they are associated to exactly one “leaf” domain. All users have access to any learning material provided by the infrastructure, which can also be bookmarked by users, for a later review.

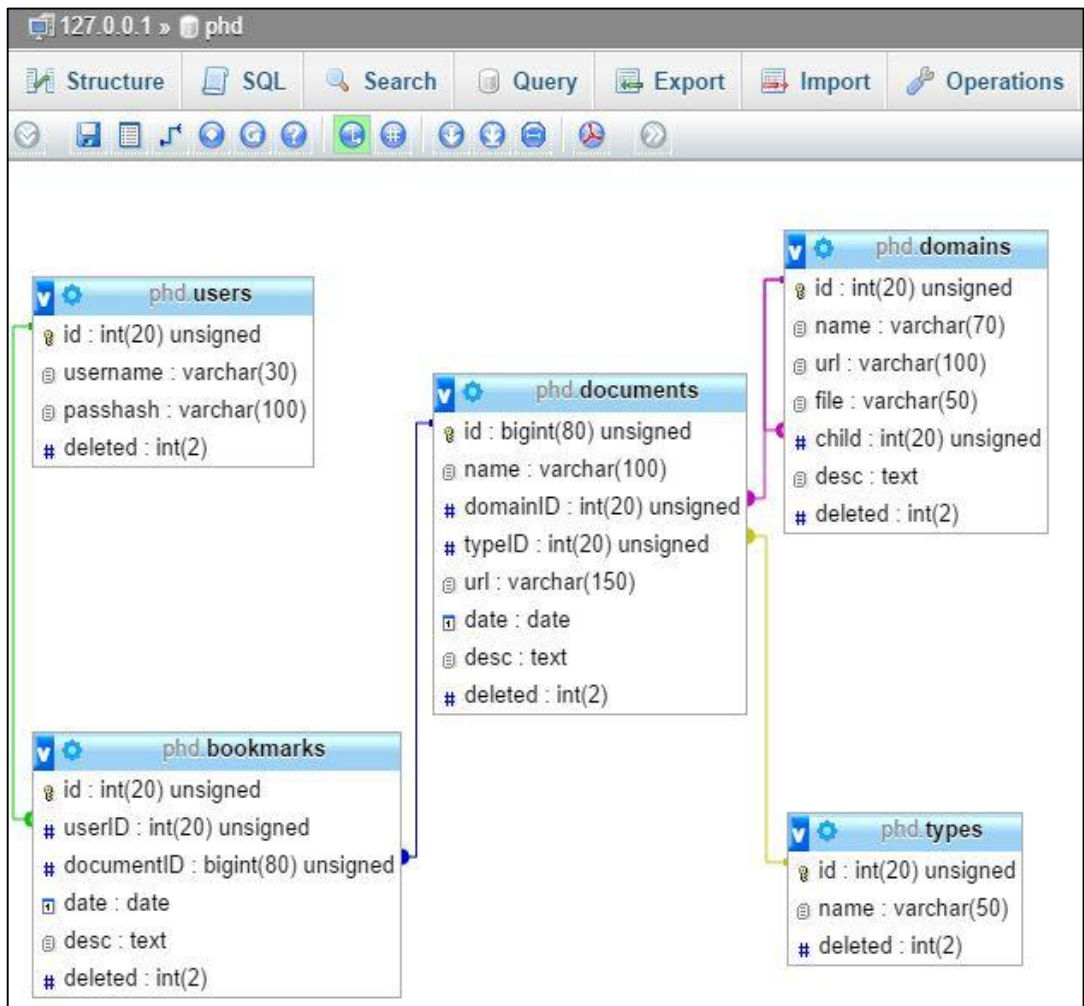


Figure 1- The figure above presents the structure of the database.

The *phd.users* table contains the necessary information required for logging in a user. The table *phd.types* stores the available formats for the learning materials, while *phd.bookmarks* is a table that allows users to keep learning materials in their accounts, for further consultation.

The domain hierarchy is represented in the table *phd.domains*, which contains information about the name of the domain, its description, as well as the path to the training document associated with the domain (if the domain is a “leaf” domain). In order to ensure the hierarchical representations of the ontology of domains, the table contains a foreign key that points to another record of the same table (*child* references *id*).

The actual URI (Uniform Resource Identifier) of the learning materials are stored within the table *phd.documents*.

4. The Crawler

Besides the initial learning materials provided by the platform, a component destined for automatic search of learning materials should also be designed and implemented. This component is called a **Crawler**, and its purpose is to browse the internet in order to find new learning materials.

The web can be considered a graph in which the pages represent the nodes of the graph, and the links contained within the pages represent the edges that connect the nodes. To browse the web means to visit every node of this graph.

The crawler does breadth-first searches (BFS) of connected, non-cyclical sub-graphs found within the graph. This means parsing a web page and analyzing its content, in order to identify the links contained within. If the visited page is in HTML format, the crawler extracts all the links contained within the page and adds them to the breadth-first search queue. Pages that contain links to other sites queued for visiting will be interior nodes of the graph, while the others will be leaf nodes. Besides analyzing the html content of the page, in order to identify the links, the crawler also classifies the content found within the html (inside the `<html>` tag), based on the domains defined in the ontology. The classification is done by the web service implemented within the infrastructure.

5. The Web Service

The web service's task is to identify the domain of the text inputted by the user, for which the platform will suggest learning materials.

For establishing the domain of interest, the text goes through two stages.

a. Preprocessing tasks

The first stage consists of pre-processing tasks, such as stemming the words and eliminating the words considered irrelevant for the classification. The most widely used algorithm for stemming words is Porter's Stemming Algorithm, proposed in the 1980 article "An Algorithm for Suffix Stripping" (Porter, 1980).

There are several methods for eliminating the components of the text considered irrelevant for classification.

One of the most common feature selection methods is the Mutual Information of a term contained in the analyzed document to a particular class (Manning, Raghavan and Schütze, 2008). This measures how much information the presence or absence of a particular term contributes to making the correct classification decision. Another common feature selection method is the Chi Square. The Chi Square test is used in statistics, among other things, to test the independence of two events. More specifically, in feature selection, we use it to test whether the occurrence of a specific term and the occurrence of a specific class are independent. High scores on χ^2 indicate that the null hypothesis (H_0) of independence should be rejected and thus that the occurrence of the term and class are dependent. If they are dependent then we select the feature for the text classification. High scores on this test indicate that the null hypothesis (H_0) of independence should be

rejected and thus the occurrence of the term and class are dependent. If they are dependent then we select the feature for the text classification. Last but not least we should note that from statistical point the Chi Square feature selection is inaccurate, due to the one degree of freedom and Yates correction should be used instead (which will make it harder to reach statistical significance). Thus we should expect that out of the total selected features, a small part of them are independent from the class). Thus we should expect that out of the total selected features, a small part of them are independent from the class. Nevertheless these noisy features do not seriously affect the overall accuracy of our classifier (Manning, Raghavan and Schütze, 2008).

b. Text classification

The second stage consists of using actual classification algorithms, in order to identify the domain of the text. Supervised classification of a document implies identifying the class it belongs to, based on a set of classes and a set of training documents associated with each class.

The eLearning platform proposed in this paper is based on three well-known text classification algorithms, k-Nearest Neighbor, k-Nearest Neighbor + TF-IDF (Term Frequency – Inverse Document Frequency) and the Naïve Bayes classifier.

K-nearest-neighbor (kNN) classification is one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data. K-nearest-neighbor classification was developed from the need to perform discriminant analysis when reliable parametric estimates of probability densities are unknown or difficult to determine.

The algorithm is based on a natural parameter $k \in N^*$. The k-nearest-neighbor classifier is commonly based on the Euclidean distance between a test sample and the specified training samples. In order to quantify the characteristics of the samples and of the analyzed text, the frequency of each word is used.

Classification is divided into two separate phases, the training phase and the actual classification phase.

The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. The training examples are vectors in a multidimensional feature space, each with a class label.

In the classification phase, k is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label, which is most frequent among the k training samples nearest to that query point.

The second classification algorithm proposed is also based on kNN, but instead of using the frequency of each word in order to quantify texts, a more sophisticated indicator is used, namely **TF-IDF** (Term Frequency – Inverse Document Frequency).

TF-IDF is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. The TF-IDF value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the

corpus, which helps to adjust for the fact that some words appear more frequently in general.

Search engines often use variations of the TF-IDF weighting scheme as a central tool in scoring and ranking a document's relevance given a user query. TF-IDF can be successfully used for stop-words filtering in various subject fields including text summarization and classification.

The statistic is calculated as a product of two distinct indicators, TF (Term Frequency) and IDF (Inverse Document Frequency). There are a few different methods for calculating the TF indicator, but the ones that are most often used are either the frequency of the word within the analyzed document, the logarithm of the frequency or, most often, the frequency of the word divided by the maximum frequency of a word within the document. The latter is the most often used and efficient, since it also takes into account the length of the document, while the others do not. The IDF indicator is calculated by dividing the number of documents in the corpus by the number of documents in the corpus that contain the analyzed word. It is straightforward to notice that the IDF indicator is inversely proportional to the number of documents that contain the analyzed word.

After the TF-IDF indicator is calculated for each word in the corpus and also for each word in the analyzed document, the rest of the algorithm is exactly the same as in the case of the classic kNN.

Another well established classification algorithm proposed in this paper is **The Naïve Bayes classifier**.

Naïve Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

The Naïve Bayes classifier is based on Bayes' Theorem regarding dependent probabilities, which states that

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where A and B are any two events.

For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

The advantage of using the Naive Bayes classifier instead of more complex classifiers is that it only requires a small amount of training data to estimate the parameters necessary for classification.

By using Bayes' theorem, one can express the probability of a document belonging to a certain class as a product of the probabilities of the words in the documents appearing in the training documents of that particular class. Hence, one can calculate the probabilities of the analyzed document belonging to each class, compare these probabilities and establish the class of the document as being the one with the highest probability.

6. Conclusions

The traditional way of learning consists of a central authority (the tutor). Learning materials are selected by the tutor and delivered to the students. This type of approach is not a student-centered approach, as there is unidirectional flow of knowledge and the learning materials are meant for satisfying the needs of many students. Also, it is a static learning process.

This is where e-Learning infrastructures come in. The field of e-Learning has gained new heights due to the popularity of the World Wide Web. Nowadays, there are a variety of different e-Learning systems that are trying to fulfill the needs of the student through web-based training. In contrast to the traditional way of learning, where the learner is restricted to a time and a place, web based training revolves around the learner.

During the past few years, a remarkable change has been noticed in the various dimensions of e-Learning. The creation of a wide variety of tools has allowed switching from an offline to an online learning environment.

The main goal of this paper was to propose a semantic e-learning platform, usable for establishing the topic of interest for the user, as well as providing learning materials. The platform also incorporates a component that continuously searches the Internet for learning materials, which are classified by using the classification component of the infrastructure and deposits the results in the database.

The platform consists of a series of interlinked components: the Knowledge Base, the Crawler, the Web Service and the user interface.

The knowledge base incorporates a relational database and the ontology of domains. The ontology is represented as a tree of domains and sub-domains, with each learning document belonging to exactly one of the leaf domains of the tree. The knowledge base is continuously updated by using another component of the platform, the Crawler.

The Crawler searches the World Wide Web for learning materials. Once a document is found, it is classified, using the classification component of the platform and the results are stored in the database.

The classification of the texts inputted by the users and of the documents found by the Crawler is the task of the web service provided by the infrastructure. In order to establish the domain of the texts, two well-known classification algorithms have been proposed, namely k-Nearest-Neighbor and the Naïve Bayes classifier. Each has its own well-documented advantages and disadvantages.

In order to minimize the probability of incorrect classification, future research includes developing a combined classifier, which will use the results of several classifiers, in order to establish the class of a document.

The issue of using multiple classification methods together (ensembles) to form a better classifier is a well-researched problem and appears in a wealth of classical Machine Learning scenarios. Many examples come to mind, from combining the same classifier on different feature sets, or different classifiers on the same feature, to algorithms like bagging and boosting that have some combiner as a final decision maker. Also, the more general learning of complex decision boundaries (e.g. not lines, nor circles), by means of

multiple classifiers, employs a combination or selection scheme to form a decision boundary shape that the individual classifiers (usually) cannot learn. (Dinu, 2010)

In order to suggest personalized learning materials, future research also includes developing intelligent software agents (user interfaces), as well as adding appropriate functionality to the web service.

REFERENCES

Allemang, D. and Hendler, J. (2011) *Semantic Web for the Working Ontology: Effective Modeling in RDFS and OWL*, Burlington, Massachussets: Morgan-Kaufmann.

Berners-Lee, T., Hendler, J. and Lassila, O. (2001) 'The Semantic Web', *Scientific American*, May, pp. 29 - 37.

Brachman, R.J. (1977) 'A Structural Paradigm for Representing Knowledge', *PhD Thesis, Harvard University*.

Crețulescu, R..M.D. (2012) *Text Mining. Tehnici de clasificare și clustering al documentelor*, Cluj - Napoca: Editura Albastră.

Dinu, L.P..R.A. (2010) 'Rank Distance Aggregation as a Fixed Classifier Combining Rule for Text Categorization', *Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science*, vol 6008, pp. 638-647.

Dinu, L. and Iuga, I. (2012) 'The Naive Bayes Classifier in Opinion Mining: In Search of the Best Feature Set', *Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science*, vol 7181, pp. 555-567.

Joachims, T. (2002) *Learning to classify text using support vector machines: methods, theory and algorithms*, Dordrecht: Kluwer Academic Publishers.

Manning, C., Raghavan, D. and Schütze, H. (2008) *Introduction to Information Retrieval*, Cambridge: Cambridge University Press.

Porter, M.F. (1980) 'An algorithm for suffix stripping', *Program*, 14(3), pp. 130-137.

Zhang, H. (2004) 'The Optimality of Naive Bayes', *Faculty of Computer Science, University of New Brunswick*.

The article has been supported by scientific research within the project entitled "PRACTICAL SCHOOL: Innovation in Higher Education and Success on the Labour Market", project identified as POSDRU/156/1.2/G/132920. The project is co-financed by the European Social Fund through the Sectorial Operational Programme for Human Resources Development 2007-2013. Investing in people!