Question Answering from Lecture Videos Based on Automatically-Generated Learning Objects

Stephan Repp, Serge Linckels, and Christoph Meinel

Hasso Plattner Institut (HPI), University of Potsdam
P.O. Box 900460, D-14440 Potsdam, Germany
{repp,linckels,meinel}@hpi.uni-potsdam.de

Abstract. In the past decade, we have witnessed a dramatic increase in the availability of online academic lecture videos. There are technical problems in the use of recorded lectures for learning: the problem of easy access to the multimedia lecture video content and the problem of finding the semantically appropriate information very quickly. The retrieval of audiovisual lecture recordings is a complex task comprising many objects. In our solution, speech recognition is applied to create a tentative and deficient transcription of the lecture video recordings. The transcription and the words from the power point slides are sufficient to generate semantic metadata serialized in an OWL file. Each video segment (the lecturer is speaking about one power point slide) represent a learning object. A question-answering system based on these learning objects is presented. The annotation process is discussed, evaluated and compared to a perfectly annotated OWL file and, further, to an annotation based on a corrected transcript of the lecture. Furthermore, the consideration of the chronological order of the learning objects leads to a better MRR value. Our approach out-performs the Google Desktop Search based on the question keywords.

1 Introduction

The amount of educational content in electronic form is increasing rapidly. At the Hasso Plattner Institut (HPI) alone, 25 hours of university lecture videos about computer science are produced every week. Most of them are published in the online Tele-TASK archive[1]. Although such resources are common, it is not easy for a user to find one that corresponds best to his/her expectations. This problem is mostly due to the fact that the content of such resources is often not available in machine readable form, i.e. described with metadata so that search engines, robots or agents can process them. Indeed, the creation of semantic annotation neither is nor should be the task of the user or creator of the learning objects. The user (e.g. a student) and the creator (e.g. a lecturer) are not necessarily computer-science experts who know how to create metadata in a specific formalism like XML, RDF or OWL. Furthermore, the creation of

[1] http://www.tele-task.de

F. Li et al. (Eds.): ICWL 2008, LNCS 5145, pp. 509–520, 2008.

metadata is a subjective task and should be done with care. The automatic generation of reliable metadata is still a very difficult problem and currently a hot topic in the Semantic Web movement. In this paper we will explore a solution to how to generate semantic annotations for university lectures. It is based on the extraction of metadata from two data sources — the content of the power point slides and the transliteration of an out-of-the-box speech recognition engine— and the mapping of natural language (NL) to concepts/roles in an ontology. Each time period of a power point slide represents a learning object. The reliability of our solution is evaluated via different benchmark tests.

This paper is based on the research of [13]. In addition to [13], we present an automatic generation of the learning object (the video is segmented based on the power point slide transitions), the comparison of our results with a manually-generated transcript corpus (an error free transcript), the MRR evaluation dimension and the consideration of the chronological order of the learning objects in the lecture videos. Additionally, our solution is compared to the Google Desktop Search based on the question keywords.

2 Related Work

Using speech recognition to annotate videos is a widely used method [5, 11, 14, 15, 22]. Due to the fact that the slides carried most of the information, Repp et al. synchronized the imperfect transcript from the speech recognition engine automatically with the slide streams in post-processing [16]. Most approaches use out-of-the-box speech recognition engines, e.g. by extracting key phrases from spoken content [5]. Besides analytical approaches, an alternative approach for video annotation is described in [17]. There, the user is involved in the annotation process by deploying collaborative tagging for the generation and enrichment of video metadata annotation to support content-based video retrieval.

In [6] a commercial speech recognition system is used to index recorded lectures. However, the accuracy of the speech recognition software is rather low; the recognition accuracy of the transliterations is approximately 22%-60%. It is also shown in [6] that audio retrieval can be performed with out-of-the-box speech recognition software. But little information can be found in the literature about educational systems that use a semantic search engine for finding additional (semantic) information effectively in a knowledge base of recorded lectures. A system for reasoning over multimedia e-Learning objects is described in [4]. An automatic speech recognition engine is used for keyword spotting. It extracts the taxonomic node that corresponds to the keyword and associates it to the multimedia objects as metadata.

Two complete systems for recording, annotating, and retrieving multimedia documents are LectureLounge and MOM. LectureLounge [21] is a research platform and a system to automatically and non-invasively capture, analyze, annotate, index, archive and publish live presentations. MOM (Multimedia Ontology Manager) [3] is a complete system that allows the creation of multimedia ontologies, supports automatic annotation and the creation of extended text (and audio) commentaries of video sequences, and permits complex queries by reasoning over the ontology. Based on the assertion that information retrieval in multimedia environments is actually a combination of search and browsing in most cases, a hypermedia navigation concept for lecture recordings is presented in [10]. An experiment is described in [7] where automatically-extracted audio-visual features of a video were compared to manual annotations that were created by users.

3 Extraction Method

The way our processing works is described in detail in [13]. To make this paper self-containing, we briefly summarize the major ideas.

3.1 Ontology Fundamentals

It has been realized that a digital library benefits from having its content understandable and available in a machine processable form, and it is widely agreed that ontologies will play a key role in providing a lot of the enabling infrastructure to achieve this goal. A fundamental part of our system is a common domain ontology. An existing ontology can be used or one can be built that is optimized for the knowledge sources.

An ontology is basically composed of a hierarchy of concepts (*taxonomy*) and a language. In the case of the first issue, we created a list of semantically relevant words regarding the domain of Internetworking, and organized them hierarchically. In the second case, we used *Description Logics* to formalize the semantic annotations.

Description Logics (DL) [1] are a family of knowledge representation formalisms that allow the knowledge of an application domain to be represented in a structured way and to reason about this knowledge. In DL, the conceptual knowledge of an application domain is represented in terms of *concepts* (unary predicates) such as $\mathsf{IPAddress}$, and *roles* (binary predicates) such as $\exists\mathsf{composedOf}$. Concepts denote sets of individuals and roles denote binary relations between individuals. Complex descriptions are built inductively using concept constructors which rely on basic concepts and role names. Concept descriptions are used to specify terminologies that define the intentional knowledge of an application domain. Terminologies are composed of *inclusion assertions* and *definitions*. The first impose necessary conditions for an individual to belong to a concept. E.g. to impose that a router is a network component that uses at least one IP address, one can use the inclusion assertion: $\mathsf{Router} \sqsubseteq \mathsf{NetComp} \sqcap \exists\mathsf{uses.IPAddress}$. Definitions allow us to give meaningful names to concept descriptions such as $\mathsf{LO}_1 \equiv \mathsf{IPAdress} \sqcap \exists\mathsf{composedOf.HostID}$.

The semantic annotation of five learning objects is shown in figure 3.1, describing the following content:

$$\begin{aligned}
\mathsf{Protocol} &\sqsubseteq \exists \mathsf{basedOn.Agreement}\\
\mathsf{TCPIP} &\sqsubseteq \mathsf{Protocol} \sqcap \exists \mathsf{uses.IPAddress}\\
\mathsf{Router} &\sqsubseteq \mathsf{NetComponent} \sqcap \exists \mathsf{has.IPAddress}\\
\mathsf{HostID} &\sqsubseteq \mathsf{Identifier}\\
\mathsf{NetworkID} &\sqsubseteq \mathsf{Identifier}\\
\mathsf{AddressClass} &\sqsubseteq \mathsf{Identifier}\\
\mathsf{IPAddress} &\sqsubseteq \mathsf{Identifier} \sqcap \exists \mathsf{composedOf.HostID}\\
&\quad \sqcap \exists \mathsf{composedOf.NetworkID}\\
&\quad \sqcap \exists \mathsf{partOf.AddressClass}
\end{aligned}$$

Fig. 1. Examples of networking terminology

$$\begin{aligned}
\mathsf{LO}_1 &\equiv \mathsf{IPAddress}\\
\mathsf{LO}_2 &\equiv \mathsf{TCPIP} \sqcap \exists \mathsf{uses.IPAddress}\\
\mathsf{LO}_3 &\equiv \mathsf{IPAddress} \sqcap \exists \mathsf{composedOf.HostID}\\
\mathsf{LO}_4 &\equiv \mathsf{IPAddress} \sqcap \exists \mathsf{composedOf.NetworkID}
\end{aligned}$$

Fig. 2. Example of terminology concerning learning objects

LO$_1$: general explanation about IP addresses,
LO$_2$: explanation that IP addresses are used in the protocol TCP/IP,
LO$_3$: explanation that an IP-address is composed of a host identifier,
LO$_4$: explanation that an IP-address is composed of a network identifier,

Some advantages of using DL are the following: firstly, DL terminologies can be serialized as OWL (*Semantic Web Ontology Language*) [20], a machine-readable and standardized format for semantically annotating resources (see section 3.5). Secondly, DL allow the definition of detailed semantic descriptions about resources (i.e. restrictions of properties), and logical inference from these descriptions [1]. Finally, the link between DL and NL has already been shown [18].

3.2 Natural Language Processing

The way our NL processing works is described in detail in [9]. To make this paper self-containing, we will briefly summarize the major ideas.

The system masters a domain dictionary L_H over an alphabet Σ^* so that $L_H \subseteq \Sigma^*$. The semantics are given to each word by classification in a hierarchical way w.r.t. a taxonomy. This means, for example, that words such as

"IP-address", "IP adresse" and "IP-Adresse" refer to the concept IPAddress in the taxonomy. The mapping function φ is used for the semantic interpretation of a NL word $w \in \Sigma^*$ so that $\varphi(w)$ returns a set of valid interpretations, e.g. φ("IP Addresse") ={IPAddress}.

The system allows a certain tolerance regarding spelling errors, e.g. the word "comXmon" will be considered as "common", and not as "uncommon". Both words "common" and "uncommon" will be considered for the mapping of "comXXmon". In that case the mapping function will return two possible interpretations, so that:

$$\varphi(\text{"comXXon"}) = \{\mathsf{common,uncommon}\}.$$

A dictionary of synonyms is used. It contains all relevant words for the domain — in our case: networks in computer-science — and at least all the words used by the lecturer (audio data) and in the slides.

3.3 Identification of Relevant Keywords

Normally, lectures have a length of around +/- 90 minutes, which is much too long for a simple learning object. If a student is searching for particular and precise information, (s)he might not be satisfied if a search engine yields a complete lecture. Therefore, we split such lectures in shorter learning objects. We defined that each power point slide is a learning object. The synchronization of the transcript could be done in an pre-processing with a software that is integrated in the presentation or with a post-processing algorithm [16].

For us, a learning object is composed of two data sources: the audio data and the content of the slides. In the case of the first issue, the audio data is analyzed with an out-of-the-box speech recognition engine. After a normalization preprocessing — i.e. deleting stop-words and stemming of the words — the stems are stored in a database. This part of our system has already been described in [13, 16].

Formally, the analysis of a data source is done with the function μ that returns a set of relevant words in their canonical form, written:

$$\mu(\mathsf{LO}_{source}) = \{w_i \in L_H, i \in [0..n]\} \backslash S$$

where *source* is the input source with *source* $\in$ {audio only, slides only, audio and slides}, and S is the set of stop words, e.g. S ={"the", "a", "hello", "thus"}.

3.4 Ranking of Relevant Concepts and Roles

Independent of the data source used (audio only, slides only, audio and slides), the generation of the metadata always works the same way. The relevant keywords from the data source identified by the function μ are mapped to ontology concepts/roles with the function φ as explained in section 3.2.

It is not useful to map all identified words to ontology concepts/roles because this would create to much overload. Instead, we focus on the most pertinent metadata for the particular learning object. Thus we implemented a simple ranking algorithm.

The algorithm works as follows: We compute for each identified concept/rule its hit-rate h, i.e. its frequency of occurrence inside the leaning object. Only the concepts/roles with the maximum (or d^{th} maximum) hit-rate compared to the hit-rate in the other learning objects are used as metadata. E.g. the concept Topology has the following hit-rate for the five learning objects (LO_1 to LO_5):

	LO_1	LO_2	LO_3	LO_4	LO_5
h	0	4	3	7	2

This means that the concept Topology was not mentioned in LO_1 but 4 times in LO_2, 3 times in LO_3 etc.

We now introduce the rank d of the learning object w.r.t. the hit-rate of a concept/role. For a given rank, e.g. $d = 1$, the concept Topology is relevant only in the learning object LO_4 because it has the highest hit-rate. For $d = 2$ the concept is associated to the learning objects LO_4 and LO_2, i.e. the two learning objects with the highest hit-rate.

3.5 Semantic Annotation Generation

The semantic annotation of a given learning object is the conjunction of the mappings of each relevant word in the source data written:

$$\mathsf{LO} = \bigsqcap_{i=1}^{m} rank_d \ \varphi(w_i \in \mu(\mathsf{LO}_{source}))$$

where m is the number of relevant words in the data source and d the rank of the mapped concept/role. The result of this process is a valid DL description similar to that shown in figure 3.1. In the current state of the algorithm we do not consider complex role imbrications, e.g. $\exists R.(A \sqcap \exists S.(B \sqcap A))$, where A, B are atomic concepts and R, S are roles. We also try to use a very simple DL, e.g. negations $\neg A$ are not considered.

One of the advantages of using DL is that it can be serialized in a machine readable form without losing any of its details. Logical inference is possible when using these annotations. The example shows the OWL serialization for the following DL-concept description:

$$\begin{aligned}\mathsf{LO}_1 \equiv\ & \mathsf{IPAddress} \sqcap \\ & \exists \mathsf{isComposedOf.}(\mathsf{Host\text{-}ID} \sqcap \mathsf{Network\text{-}ID})\end{aligned}$$

defining a concept name (LO_1) for the concept description saying that an IP address is composed of a host identifier and a network identifier.

4 Evaluation Criteria

4.1 Prearrangement

The speech recognition software is trained with a tool in 15 minutes and it is qualified by some domain words from the existing power point slides in 15 minutes. So the training phase for the speech recognition software is approximately 30 minutes long. A word accuracy of approximately 60% is measured. The *stemming* in the pre-processing is done by the porter stemmer [12].

We selected the lecture on Internetworking (100 Minutes) which has 62 slides, i.e. multimedia learning objects. The lecturer spoke about each slide for approximately 1.5 minutes. The synchronization between the power point slides and the erroneous transcript in a post-processing process is explored in [16], if no log file exist with the time-stamp for each slide transition. The lecture video is segmented into smaller videos — a multimedia learning object (LO). Each multimedia object represents the speech over one power point slide in the lecture. So each LO has a duration of approximately 1.5 minutes.

A set of 107 NL questions on the topic Internetworking was created. We worked out questions that students ask, e.g. "*What is an IP-address composed of?*", etc. For each question, we also indicated the relevant answer that should be delivered. For each question, only one answer existed in our corpus. Owl files from the slides (S), the transcript from the speech recognition engine (T), the transcript with error correction (PT) and the combination of these sources are automatically generated. The configurations are the following:

$$[< source >]_{ranking}$$

where $< source >$ stands for the data source (S, T, or PT), and $< ranking >$ stands for the ranking ration (0 is no ranking at all, all concepts are selected, i.e. $d = 0$, and r ranking with $d = 2$). E.g. $[\mathrm{T}+\mathrm{S}]_2$ means that the metadata from the transcript (T) and from the slides (S) are combined (set union), and that the result is ranked with $d = 2$.

Additionally, an owl file (M) is a manual annotation by the lecturer.

4.2 Search Engine and Measurement

The *semantic search engine* that we used is described in detail in [8]. It reviews the OWL-DL metadata and computes how much the description matches the query. In other words, it quantifies the semantic difference between the query and the DL concept description.

The *Google Desktop Search*[2] is used as a keyword search. The files of the transcript, of the perfect transcript and of the power point slides are used for the indexing. In three independent tests, each source is indexed by Google Desktop Search.

The *recall* (R) according to [2] is used to evaluate the approaches. The top recall R_1 (R_5 or R_{10}) analyses only the first (first five or ten) hit(s) of the result set.

The *reciprocal rank of the answer* (*MRR*) according to [19] is used. The score for an individual question was the reciprocal of the rank at which the first correct answer was returned or 0 if no correct response was returned. The score for the run was then the mean over the set of questions in the test. A *MRR* score of 0.5

[2] `http://desktop.google.com`

can be interpreted as the correct answer being, on average, the second answer by the system. The MRR is defined:

$$MRR = \frac{1}{N} \sum_{i=1}^{N} (\frac{1}{r_i})$$

N is the amount of question. r_i is the rank (position in the result-set) of the correct answer of the question i. MRR_5 means that only the first five answers of the result set are considered.

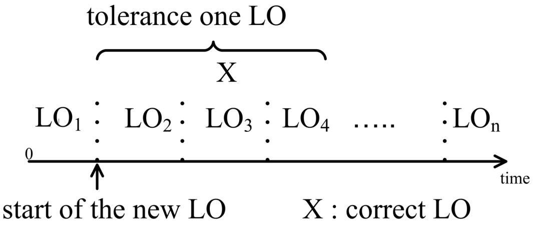

Fig. 3. Learning object (LO) for the **second test**

5 Test and Result

Two test is performed to the owl files:

The **first test** (Table 1) is to analyse which of the annotations based on the sources (S, T, PT) yields the best results from the semantic search. It is not surprising that the best search results were achieved with the manually-generated semantic description (M), with 70% of R_1 and 82% of R_5. Let us focus in this section on the completely automatically-generated semantic description ([T] and [S]). In such a configuration with a fully automated system $[T]_2$, a learner's question will be answered correctly in 14% of the cases by watching only the first result, and in 31% of the cases if the learner considers the first five results that were yielded. This score can be raised by using an improved speech recognition engine or by manually reviewing and correcting the transcripts of the audio data. In that case $[PT]_2$ allows a recall of 41% (44%) while watching the first 5 (10) returned video results. A MRR of 31% for the constellation $[PT]_2$ is measured.

In practice, 41%(44%) means that the learner has to watch at most 5 (10) learning objects before (s)he finds the pertinent answer to his/her question. Let us recall that a learning object (the lecturer speaking about one slide) has an average duration of 1.5 minutes, so the learner must spend — in the worst case — $5 * 1.5 = 7.5$ minutes (15 minutes) before (s)he gets the answer.

The **second test** (Table 2) takes into consideration that the LO (one slide after the other) are chronological in time. The topic of the neighboring learning objects (LO) are close together and we assume that answers given by the semantic search engine scatter around the correct LO. Considering this characteristic and accepting a tolerance of one preceding LO and one subsequent LO, the

Table 1. The maximum time, the recalls and MRR_5 value of the **first test** (%)

	R_1	R_2	R_3	R_4	R_5	R_{10}	MRR_5
time	1.5 min	3 min	4.5 min	6 min	7.5 min	15 min	-
LO (slides)	1 (1)	2 (2)	3 (3)	4 (4)	5 (5)	10 (10)	-
M	70	78	79	81	82	85	75
$[S]_0$	32	49	52	58	64	70	44
$[T]_2$	14	23	26	30	31	35	21
$[PT]_2$	25	33	37	40	41	44	31
$[T+S]_2$	36	42	46	50	52	64	42
$[PT+S]_2$	32	43	48	49	51	69	40

MRR value of $[PT]_2$ increased by about 21% ($[T]_2$ about 15%). Three LO are combined to make one new LO. The disadvantage of this is that the duration of the new LO object increases from 1.5 minutes to 4.5 minutes. On the other hand the questioner has the opportunity to review the answer in a specific context.

Table 2. The maximum time, the recalls and MRR_5 value of the **second test** (%)

	R_1	R_2	R_3	R_4	R_5	MRR_5
time	4.5 min	9	13.5 min	18 min	22.5 min	-
LO(slides)	1 (3)	2 (6)	3 (9)	4 (12)	5 (15)	-
$[S]_0$	42	57	62	66	70	53
$[T]_2$	22	43	50	55	56	36
$[PT]_2$	43	54	62	64	65	52
$[T+S]_2$	47	51	53	59	62	52
$[PT+S]_2$	43	54	65	66	70	53

The **third test** (Table 3) takes into consideration that the student's search is often a keyword-based search. The query consists of the important words of the question. For example, the question: "*What is an IP-address composed of?*" has got the keywords: "*IP*","*address*" and "*compose*". We extracted from the 103 questions the keywords and analysed with these the performance of Google Desktop search. It is clear that if the whole question string is taken, almost no question is answered by Google Desktop Search.

As stated in the introduction, the aim of our research is to give the user the technological means to quickly find the pertinent information. For the lecturer or the system administrator, the aim is to minimize the supplementary work a lecture may require in terms of post-production, e.g. creating the semantic description.

Let us focus in this section on the fully automated generation for semantic descriptions (T, S and its combination [T + S]) of the **second test**. In such a configuration with a fully automated system $[T + S]_2$, a learner's question will be answered correctly in 47% of the cases by reading only the first result, and in

Table 3. The maximum time, the recalls and MRR_5 value of the Google Desktop Search, **third test** (%)

	R_1	R_2	R_3	R_4	R_5	R_{10}	MRR_5
time	1.5 min	3 min	4.5 min	6 min	7.5 min	15 min	-
LO (slides)	1 (1)	2 (2)	3 (3)	4 (4)	5 (5)	10 (10)	-
S	41	44	47	48	48	50	44
T	12	22	22	23	23	24	17
PT	18	27	27	28	28	28	23

53% of the cases if the learner considers the first three results that were yielded. This score can be raised by using an improved speech recognition engine or by manually reviewing and correcting the transcripts of the audio data. In that case $[PT + S]_2$ allows a recall of 65% while reading the first 3 returned results. In practice, 65% means that the learner has to read at most 3 learning objects before he finds the pertinent answer (in 65% of cases) to his question. Let us recall that a learning object has an average duration of 4.5 minutes (**second test**), so that the learner must spend — in the worst case — $3 * 4.5 = 13.5$ minutes before (s)he gets the answer.

Comparing the Google Desktop Search (**third test**) with our semantic search (**first test**) we can point out the following:

- The search based on the power point slide yields approximately the same result for both search engines. That is due to the fact that the slide always consists of catch-words and an extraction of further semantic information is limited (especially the rules).
- The semantic search yields better results if the search is based on the transcript. Here a semantic search out-performs the Google Desktop Search (*MRR* value).
- The power point slides contain the most information compared to the speech transcripts (perfect and erroneous transcript).

6 Conclusion

In this paper we have presented an algorithm for generating a semantic annotation for university lectures. It is based on three input sources: the textual content of the slides, the imperfect transliteration and the perfect transliteration of the audio data of the lecturer. Our algorithm maps semantically relevant words from the sources to ontology concepts and roles. The metadata is serialized in a machine readable format, i.e. OWL. A fully-automatic generation of multimedia learning objects serialized in an OWL-file is presented. We have shown that the metadata generated in this way can be used by a semantic search engine and out-performs the Google Desktop Search. The influence of the chronology order of the LO is presented. Although the quality of the manually-generated metadata is still better than the automatically-generated ones, it is sufficient for use as a reliable semantic description in question-answering systems.

We are working on a more intelligent extraction of the concepts and rules from the data sources. All activity applications, e.g. newscasts, theater plays or any kind of speech being complemented by textual data, could be analyzed and annotated with the help of our proposed algorithm.

This project was developed in the context of the Web University project[3] which aims to explore novel internet and IT technologies in order to enhance university teaching and research.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
2. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval. ACM Press / Addison-Wesley (1999)
3. Bertini, M., Bimbo, A.D., Torniai, C., Cucchiara, R., Grana, C.: Mom: Multimedia ontology manager. a framework for automatic annotation and semantic retrieval of video sequences. In: Bimbo, A.D., Torniai, C., Cucchiara, R., Grana, C. (eds.) ACM SIGMM, pp. 787–788. ACM Press, New York (2006)
4. Engelhardt, M., Hildebrand, A., Lange, D., Schmidt, T.C.: Reasoning about eLearning Multimedia Objects. In: International Workshop on Semantic Web Annotations for Multimedia (SWAMM) (2006)
5. Haubold, A., Kender, J.R.: Augmented segmentation and visualization for presentation videos (2005)
6. Hürst, W., Kreuzer, T., Wiesenhütter, M.: A qualitative study towards using large vocabulary automatic speech recognition to index recorded presentations for search and access over the web. In: IADIS Internatinal Conference WWW/Internet (ICWI), pp. 135–143 (2002)
7. Jaimes, A., Nagamine, T., Liu, J., Omura, K., Sebe, N.: Affective meeting video analysis. In: IEEE Multimedia and Expo., pp. 1412–1415 (2005)
8. Karam, N., Linckels, S., Meinel, C.: Semantic composition of lecture subparts for a personalized e-learning. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 716–728. Springer, Heidelberg (2007)
9. Linckels, S., Meinel, C.: Resolving ambiguities in the semantic interpretation of natural language questions. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) IDEAL 2006. LNCS, vol. 4224, pp. 612–619. Springer, Heidelberg (2006)
10. Mertens, R., Schneider, H., Müller, O., Vornberger, O.: Hypermedia navigation concepts for lecture recordings. In: E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, pp. 2480–2847 (2004)
11. Ngo, C.-W., Wang, F., Pong, T.-C.: Structuring lecture videos for distance learning applications. In: Multimedia Software Engineering, pp. 215–222 (2003)
12. Porter, M.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
13. Repp, S., Linckels, S., Meinel, C.: Towards to an automatic semantic annotation for multimedia learning objects. In: Proceedings of the International Workshop on Educational Multimedia and Multimedia Education 2007, Augsburg, Bavaria, Germany, September 28, 2007, pp. 19–26. ACM, New York (2007)

[3] `http://www.hpi.uni-potsdam.de/meinel/research/`

14. Repp, S., Meinel, C.: Segmenting of recorded lecture videos - the algorithm voiceseg. In: Proceedings of the 1th Signal Processing and Multimedia Applications (Sigmap), Setubal, Portugal, pp. 317–322 (August 2006)
15. Repp, S., Meinel, C.: Semantic indexing for recorded educational lecture videos. In: 4th IEEE Conference on Pervasive Computing and Communications Workshops (PerCom 2006 Workshops), Pisa, Italy, March 13-17, 2006, pp. 240–245. IEEE Computer Society, Los Alamitos (2006)
16. Repp, S., Waitelonis, J., Sack, H., Meinel, C.: Segmentation and annotation of audiovisual recordings based on automated speech recognition. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 620–629. Springer, Heidelberg (2007)
17. Sack, H., Waitelonis, J.: Integrating social tagging and document annotation for content-based search in multimedia data. In: Semantic Authoring and Annotation Workshop (SAAW) (2006)
18. Schmidt, R.A.: Terminological representation, natural language & relation algebra. In: Ohlbach, H.J. (ed.) GWAI 1992. LNCS, vol. 671, pp. 357–371. Springer, Heidelberg (1993)
19. Voorhees, E.M.: The trec-8 question answering track report. In: TREC (1999)
20. W. W. W. C. W3C. OWL Web Ontology Language (2004), `http://www.w3.org/TR/owl-features/`
21. Wolf, P., Putz, W., Stewart, A., Steinmetz, A., Hemmje, M., Neuhold, E.: Lecturelounge – experience education beyond the borders of the classroom. International Journal on Digital Libraries 4(1), 39–41 (2004)
22. Yamamoto, N., Ogata, J., Ariki, Y.: Topic segmentation and retrieval system for lecture videos based on spontaneous speech recognition. In: European Conference on Speech Communication and Technology, pp. 961–964 (2003)