Towards to an Automatic Semantic Annotation for Multimedia Learning Objects

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

ABSTRACT

The number of digital video recordings has increased dramatically. The idea of recording lectures, speeches, and other academic events is not new. But, the accessibility and traceability of its content for further use is rather limited. Searching multimedia data, in particular audiovisual data, is still a challenging task to overcome. We describe and evaluate a new approach to generate a semantic annotation for multimedia resources, i.e., recorded university lectures. Speech recognition is applied to create a tentative and deficient transliteration of the video recordings. We show that the imperfect transliteration is sufficient to generate semantic metadata serialized in an OWL file. The semantic annotation process based on textual material and deficient transliterations of lecture recordings are discussed and evaluated.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process, Selection process; H.3.1 [**Content Analysis and Indexing**]: Indexing methods, Abstracting methods

General Terms

Algorithms, Experimentation, Management, Reliability

Keywords

multimedia retrieval, multimedia knowledge base, speech recognition, transliteration, ontology, semantic annotation, OWL, learning object

1. INTRODUCTION

The amount of educational content in electronic form increases rapidly. At the Hasso Plattner Institut (HPI) in Potsdam 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]. Other online archives with educational content like MySchool![2], MIT Open Courseware[3], Explore e-Learning[4], and Learning Science[5] have hundreds of learning objects (e.g., animations, pictures, videos) about different topics.

Although such resources are common, it is not easy for a user to find one that corresponds best to expectations. This problem is mostly due to the fact that the content of such (multimedia) 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 is and should neither be the task of the user, nor of the creator of the learning object. 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 metadata is a subjective task and should be done with conscious. The automatic generation of reliable metadata is still a very difficult problem, and currently a hot topic in the Web 2.0 movement.

In this paper we explore a solution how to generate semantic annotations for university lectures. It is based on the extraction of metadata from two data sources — the content of the slides and the still deficient transliteration of an out-of-the-box speech recognition engine, and the mapping of natural langauge (NL) to concepts/roles in an ontology.

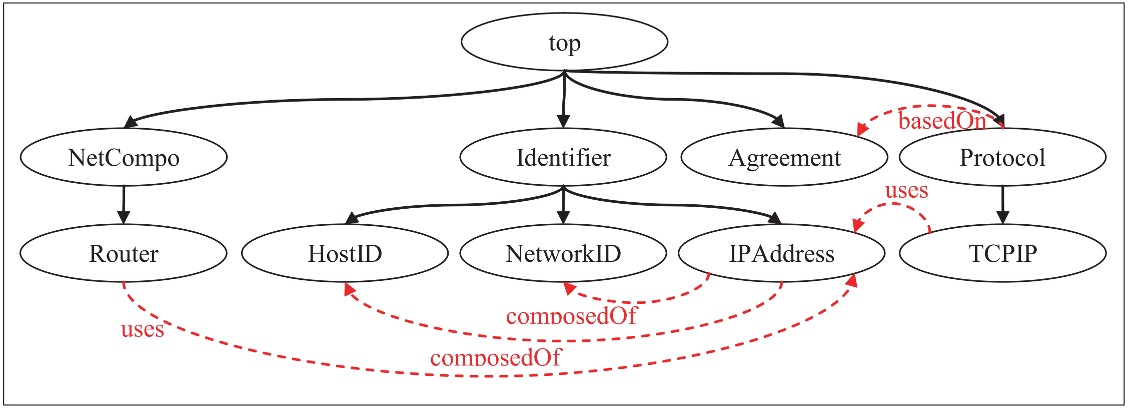
The reliability of our solution is evaluated via different benchmark tests. Firstly, we test two search engines; a keyword-based and a semantic. The later performs its retrieval over our metadata, whereas the keyword-based browses the textual content of the resources. Secondly, we run different tests to find the best configuration of our algorithm, i.e., to maximize the number of relevant documents and to minimize the overload. The outcomes are two-fold. Although the quality of the generated semantic annotation allows the semantic search engine to yield more precise results, the number of queries that can be answered correctly is not greater than the one of a classical keyword-based search engine.

The remainder of this document is structured as follows. We present in section 2 some related projects about the topic of automatically annotating resources. Our method to identify metadata from the audio data of the speaker is

MM'07, September 28, 2007, Augsburg, Germany.
Copyright 2007 ACM 1-59593-361-1/07/0003...$5.00.

[1] `http://www.tele-task.de`
[2] `http://www.education.lu`
[3] `http://ocw.mit.edu`
[4] `http://www.explorelearning.com`
[5] `http://www.learningscience.org`

Figure 1: Sample of a taxonomy about networking.



described in section 3, and evaluated in section 4. We conclude in section 5 with a short summary and some ongoing work.

2. RELATED WORK

In this section, we briefly present related projects about lecture video segmentation and indexing. Using speech recognition to annotate videos is a widely used method [5, 25, 8, 15].

Most approaches use out-of-the-box speech recognition engines, e.g., by extracting key phrases from spoken content [8]. Besides analytical approaches, an alternative approach for video annotation is described in [19]. 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. Another non-analytic approach is described in [18] for synchronizing presentation slides by maintaining a log file during the presentation to keeping track of slide changes.

Optical character recognition (OCR) for the identification and synchronization of the currently shown presentation slide within a desktop recording is studied in [18]. The design and adaptation of an automatic video browsing and retrieval system is presented in [26]. It describes a speech recognition module that recognizes speech in scenes, and an extraction module that extracts the texts from key frames. Then it constructs the textual indices for the retrieval. Unfortunately, the system is not adapted to lecture videos.

In [9] a “large vocabulary automatic speech recognition” commercial system (out-of-the-box) 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 [9] that audio retrieval can be performed with out-of-the-box speech-recognition-software. But little information can be found in 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 that extracts meta-information of the lecture from the imperfect transliteration is presented in [17]. It aims to annotate “topic-areas” with additional information, e.g., definition, summary and overview. Unfortunately a search engine based on this data is not mentioned in the paper.

A system for reasoning over multimedia e-Learning objects is described in [7]. An automatic speech recognition engine is used for keyword spotting. It extracts the taxonomy node that corresponds to the keyword, and associates it to the multimedia objects as metadata.

A domain ontology information retrieval based on speech recognition is presented in [21]. It is based on four different acoustic models, and on two different recognition processes: phonetic decoding and keyword spotting.

Two complete systems for recording, annotating, and retrieving multimedia documents are LectureLounge and MOM. LectureLounge [24] 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) [4] is a complete system that allows the creation of multimedia ontologies, supports automatic annotation and creation of extended text (and audio) commentaries of video sequences, and permits complex queries by reasoning on the ontology.

An algorithm for gesture detection in lecture videos by combining visual, speech and electronic slides is presented in [23]. It is based on modified HMM models for complete gestures to predict and recognize incomplete gestures before the whole gestures paths are observed.

Based on the assertion that information retrieval in multimedia environments actually is a combination of search and browsing in most cases, a hypermedia navigation concept for lecture recordings is presented in [13].

An experiment is described in [10], where automatically extracted audio-visual features of a video were compared to manually annotations that were created by users.

In this paper we extract from the transliteration also rules and concepts. The searching-results are analyst and evaluated.

3. EXTRACTION METHOD

We describe in this section how our solution for generating metadata works. We start with some fundamentals about ontologies and NL processing. The major part of this section is the identification of relevant keywords in the data sources and the ranking of the resulting keywords. Finally, the semantic annotation is generated and serialized as machine-readable file.

Figure 2: Examples of networking terminology.

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

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 build its own ontology that is optimized for the knowledge sources.

An ontology is basically composed of a hierarchy of concepts (*taxonomy*) and a language. As for the first issue, we created a list of semantically relevant words regarding the domain of Internetworking, and organized them hierarchical (figure 1). As for the second issue, we used *Description Logics* to formalize the semantic annotations.

Description Logics (DL) [2] 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 $\mathsf{composedOf}$. Concepts denote sets of individuals and roles denote binary relations between individuals. Complex descriptions are built inductively using concept constructors which rely on basic concept and role names. The different DL languages distinguish themselves by the kinds of constructs they allow. Examples of concept constructs are as follows:

- top-concept ($\top$) and bottom-concept ($\bot$) denoting all the individuals in the domain and the empty set respectively,
- conjunction ($\sqcap$),
- existential restriction ($\exists r.C$), e.g., $\mathsf{IPAddress} \sqcap \exists\mathsf{composedOf.HostID}$ means that an IP address is composed of a host ID.

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 $\mathsf{LO}_1 \equiv \mathsf{IPAdress} \sqcap \exists\mathsf{composedOf.HostID}$.

Figure 3: Example of terminology concerning learning objects.

$$\begin{array}{l}
\mathsf{LO}_1 \equiv \mathsf{IPAddress}\\
\mathsf{LO}_2 \equiv \mathsf{TCPIP} \sqcap\exists\mathsf{uses.IPAddress}\\
\mathsf{LO}_3 \equiv \mathsf{IPAddress} \sqcap\exists\mathsf{composedOf.HostID}\\
\mathsf{LO}_4 \equiv \mathsf{IPAddress} \sqcap\exists\mathsf{composedOf.NetworkID}\\
\mathsf{LO}_5 \equiv \mathsf{TCPIP}
\end{array}$$

Figure 2 shows the formalized taxonomy displayed in figure 1. The semantic annotation of five learning objects is shown in figure 3, describing the following content:

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,
LO$_5$: general explanation about the protocol TCP/IP.

Some advantages of using DL are the following. Firstly, DL terminologies can be serialized as OWL (*Semantic Web Ontology Language*) [22], a machine-readable and standardized format for semantically annotating resources (see section 3.5). Secondly, DL allows the definition of detailed semantic descriptions about resources (i.e., restrictions over properties), and logical inference about these descriptions [2]. Finally, the link between DL and NL has already been shown [20].

3.2 Natural Language Processing

The way our NL processing works is described in detail in [12]. To make this paper self-containing, we 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, e.g., that words such as "IP-address", "IP adresse" and "IP-Adresse" refer to the concept $\mathsf{IPAddress}$ 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., $\varphi(\text{"IP Addresse"}) = \{\mathsf{IPAddress}\}$.

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:

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

A dictionary of synonyms is used. It should contain all relevant words for the domain — in our case: networks in computer-science — that are 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. E.g., 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, each having a duration of less than 10 minutes. In the current state of our solution, this pre-processing is done manually.

For us, a learning object is composed of two data sources: the audio data and the content of the slides. As for the first issue, the audio data is analyzed with an out-of-the-box speech recognition engine. This part of our solution is already described in detail in [17, 16].

Unfortunately, most lecture recordings do not provide optimal sound quality and thus, the effectiveness of automatic speech recognition (ASR) for the extraction of spoken words suffers even if a speaker trained system is used. In fact, the raw results of an ASR applied to lecture audio streams are not suitable for indexing. The word accuracy is only about 20%-70% per transcript. But the accuracy is still sufficient to identify relevant ontology concepts/roles that the speaker is talking about at a particular time interval. In spirit of this, we tried to come upwith an additional source of data.

Today, a lecturer often uses text-based presentations such as MS PowerPoint or Portable Document Format (PDF). We used such sources to improve the automatic generation of metadata. The synchronization between speech and slides can be done immediately during the presentation or in a post-processing way; an implemented algorithm synchronizes the slides with imperfect transliteration. Our algorithm allows the time position for each slide during the lecture to be matched with an average derivation of one slide.

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

The input of our algorithm to generate metadata is the imperfect transliteration from the speech recognition engine and the content from the slides (see section 3.3). Independently of the used data source (audio only, slides only, audio and slides), the generation of the metadata always works in the same way. The relevant keywords from the data source that are 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 match all identified words to ontology concepts/roles because this will create overload. Instead, we focus on the most pertinent metadata for the particular learning object. Thus, we implement a simple ranking algorithm.

Figure 4: Example of an OWL serialization.

```
<owl:Class rdf:about="LO1">
  <rdfs:subClassOf rdf:resource="\#IPAddress" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="\#isComposedOf" />
      <owl:someValuesFrom>
        <owl:Class>
          <owl:intersectionOf
                  rdf:parseType="Collection">
            <owl:Class rdf:about="\#Host-ID" />
            <owl:Class rdf:about="\#Network-ID" />
          </owl:intersectionOf>
        </owl:Class>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

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:

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

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.

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, i.e., OWL as shown in figure 4, 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 AND DISCUSSION

In this section, we evaluate the quality of the automatic generated metadata. We refer to the empirical data made with a standard keyword-based search engine and a more sophisticated semantic search engine respectively. We also suggest further improvements for our algorithm.

4.1 Preliminaries

4.1.1 Search Engines

The standard keyword-based search engine works in a classical way by browsing the textual content of the resource. If one (relevant) keyword is found then the resource is considered as being a hit, i.e., a relevant document. The search engine is optimized in the way that it does not consider stop words.

The semantic search engine that we used is described in detail in [11]. It infers over the OWL-DL metadata and computes how much the description matches the query. In more detail, it quantifies the semantic difference between the query and the DL concept description.

4.1.2 Knowledge Source

We used the online tele-TASK archive[6] that contains hundreds of recorded university lectures, as a knowledge base. We selected the lecture on Internetworking (IPv4), which we split into 40 smaller units, i.e., multimedia learning objects.

4.1.3 Evaluation Criteria

A set of 123 NL questions on the topic Internetworking was created. We tried to work out questions as students would ask, e.g., "*What is an IP-address composed of?*", "*How does a datapacket find its way through a network?*", "*What is a switch good for?*", "*Do internetprotocols guarantee an error-free communication?*". For each question we also indicated the relevant answer(s) that should be delivered.

We call an answer from a search engine a *perfect hit* if it yields only the relevant answer (without any supplementary documents). We call an answer from the search engine a *sufficient hit* if it delivers the correct answer, but contains more information than necessary. In such a case, we call this supplementary information the *rest*.

4.2 Results

We carried out three different tests. The first with a keyword-based search engine over the textual content of the audio data and the content of the slides respectively. The second with a semantic search engine over the automatic generated metadata from the audio data and content of the slides. The third was based on random retrieval.

4.2.1 Keyword-Based Results

The aim of this first test is to evaluate the quality of the audio data processing (section 3.3), and to test if the results of keyword-bases search engines can be improved by browsing different data sources together.

We design by "txtSlides" the textual content from the slides that we have extracted with a specific tool[7]. We design by "txtAudio" the textual content identified in the imperfect transliteration of the audio data, and by "txtAudioSlides" the concatenation of "txtAudio" and "txtSlides". The results of the retrieval are the following:

	perfect hits	sufficient hits	average rest
txtAduio	8.1%	78.9%	12.1
txtSlides	7.3%	83.7%	6.1
txtAduioSlides	6.5%	89.4%	13.4

In general, the search over the textual audio data was more precise than the search over the slides (8.1% for the audio, 7.3% for the slides). However, more queries could be answered correctly by the keyword-based search engine over the slides than over the audio data (78.9% for the audio, 83.7% for the slides). The precision was far better for the slides (6.1), than for the audio (12.1) and audio & slides (13.4). The search results were best where carried out on the slides and the audio data together (89.4%).

It is normal that the precision decreases as the recall increases [3]. The more tolerant the retrieval system becomes in order to find more relevant documents, the more imprecise the yielded answers, i.e., a higher rest.

We want to emphasise that both, the full text from the slides and the text from the audio data need not necessarily have the same content. On the one hand, the speaker sometimes simply reads the content of the slides and eventually adds sentences. On the other hand, the speaker sometimes does not use the same words from the slides, but e.g., explains the topic by insisting on some major details. This explains why, in our evaluation, the search in the audio data allows to be found more precise answers, albeit not always the correct answer.

This outcome shows that search results can be improved by considering the combination of the audio data of the speaker and the content of the slides.

4.2.2 Semantic Search

The second test aims to evaluate the quality of the generated metadata. We used different configurations based on the ranking criteria (see section 3.4) and the input source (see section 3.3). The configurations are the following:

$$< source >_{\text{ranking}}$$

where $< source >$ stands for the data source (S = slides, A = audio), and $< ranking >$ stands for the ranking ration (0 is no ranking at all $d = 0$, r ranking with $d = 2$). E.g., A_0S_r means that the metadata from the audio (not ranked) and from the slides (ranked, $d = 2$) are used.

The results with a different ranking ration (d) do not affect the results very much; a smaller d causes a higher precision with a smaller recall, a greater d causes a lower precision with a higher recall. We empirically evaluated that the best results were found with $d = 2$. The following results were measured:

	perfect hits	sufficient hits	average rest
S_0	15.4%	73.2%	4.2
S_r	14.6%	51.2%	2.4
A_0	13.0%	65.0%	6.1
A_r	10.6%	48.0%	2.9
A_0S_0	12.2%	74.0%	6.4
A_0S_r	10.6%	61.8%	2.6
A_rS_0	14.6%	73.2%	4.0
A_rS_r	15.4%	58.5%	3.1

The evaluation of the keyword-based search (see section 4.2.1) showed that also using the audio data of the lecturer

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

[7] PPT2TXT, available at: http://www.linckels.lu

Figure 5: Statistical evaluation of the search results showing the number of the yielded results.

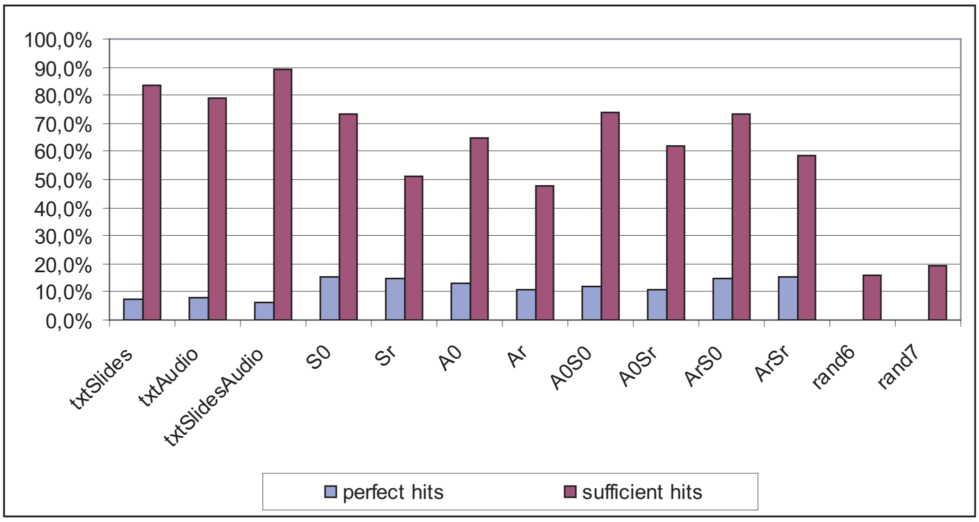
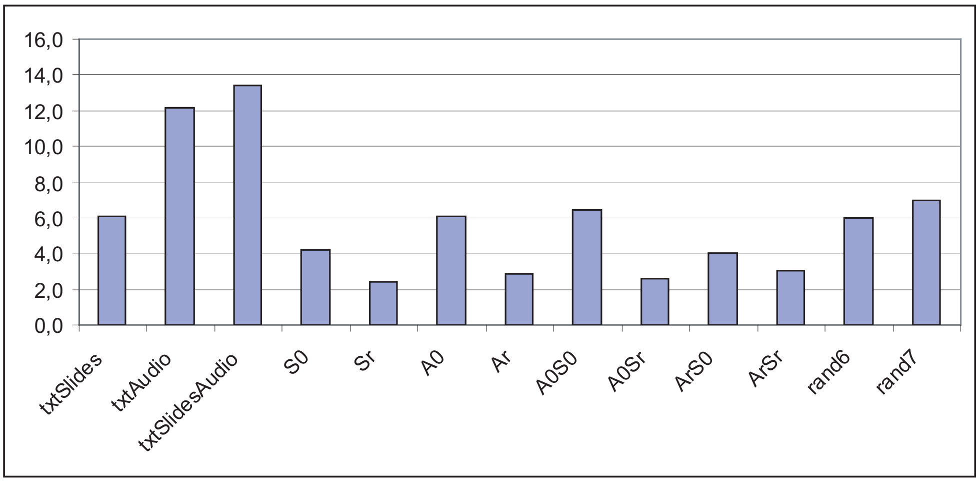


Figure 6: Average rest of the retrieval; the lower, the more precise is the result.



improve the search results. This assumption does not hold for the semantic search. We see that $S_{0,r}$ (slides only) have a much greater recall w.r.t. sufficient hits than $A_{0,r}$ (audio only). When combining slides and audio data, the results are similar to $S_{0,r}$, i.e., the non-ranked audio and slide combination (A_0S_0), and the ranked audio and the non-ranked slide (A_rS_0). This outcome must be due to the fact that the annotations for both, audio and slides, are semantically closer, so that their combination introduces generally no more additional information.

A further outcome is the fact that with a harder retrieval constraint by considering a ranking ratio, the recall decreases but the precision increases. In these cases, we have a greater value for the perfect hits as well as a small average rest, but also a smaller recall w.r.t. sufficient hits.

Finally, the semantic search has generally worse results when considering the general recall w.r.t. sufficient hits, but double, even triple the precision w.r.t. perfect hits and average rest. It would seem that the quality of our automatic generated annotations is neither good, nor precise enough for the semantic search engine to significantly improve its search results. We discuss the outcome further in section 4.3 and suggest some possible improvements.

4.2.3 Random Retrieval

We also experimented with a random retrieval; for each query the system simply yields a certain number of resources randomly. We tested two configurations: the system picked 6 and 7 random results respectively per query, called “rand6” and “rand7”, respectively.

	perfect hits	sufficient hits	average rest
rand6	0.0%	15.8%	6.0
rand7	0.0%	19.3%	7.0

Statistically, the results are not surprising. Perfect hits are quasi impossible. The score of sufficient hits compared to the keyword-based searches or the semantic searches is far worse.

4.3 Discussion and Improvements

Figure 5 shows an overview of the quality of the different retrieval strategies described in section 4.2. Figure 6 gives an overview of the average rest, i.e., the precision of the different strategies for the sufficient hits (smaller rest means higher precision).

By analyzing these figures we can draw the following conclusions. The quality of the generated metadata is not sufficient to be used efficiently by a semantic search engine. We think that the weak quality of the generated metadata has three main reasons.

The first reason is the quality of the audio data processing and, in particular, the quality of the speech recognition. One common issue in speech processing is the quality of the recordings. Here, different aspects are important and cause failures in the transliteration. These are:

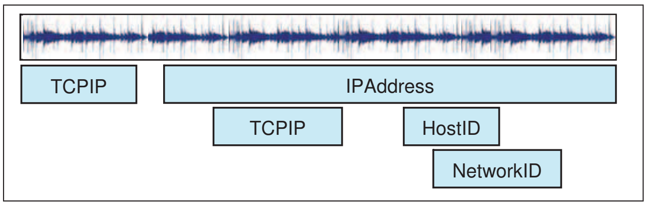
- Unappropriated acoustics in the room, e.g., background noise.
- Bad hardware, e.g., microphone.
- Training of the system, e.g., the lecturer trained the speech recognizer with domain specific words.
- Quality of the speech recognizer, e.g., word accuracy.

Secondly, one weakness of our solution is the primitive way we translate NL into DL formulas, especially the roles. Here, further work must be invested in order to better identify the arguments of the roles and their quantifiers. E.g., it is not evident for the system to translate the sentence "An IP address has a host ID" into the correct DL formula $\mathsf{IPAddress} \sqcap \exists \mathsf{composedOf.HostID}$, rather than into the formula $\mathsf{IPAddress} \sqcap \mathsf{HostID} \sqcap \exists \mathsf{composedOf}$. Some solutions and improvements are suggested in [12].

Third, NL in general is often a source of linguistic ambiguities. We had situations, where the speech recognizer created ambiguities like the German word "Mann", which can mean "a man" (German: "Mann") but it can also mean "one" (German: "man"), or a network "MAN" (Metropolitan Area Network). All three are pronounced in the same way. Such ambiguities can only be resolved by a more complex linguistic analyis of the sentence. State-of-the-art speech engines do not support such detailed linguistic information. Beside that, a lecturer does not often speak in a perfect linguistic manner. We learnt that using a *strong parser* cannot deliver much better results, due to the fact that the input is an imperfect transliteration of the audio data. In fact, the engine delivers only a stream of words without any punctuation marks and accentuation. Furthermore, there are always ambiguities that remain when dealing with NL [6, 1, 14]. In the current state of our algorithm, we tried to identify words that have same or similar pronunciations but different meanings, and gave priority to semantically relevant words. For the word "MAN", the interpretation "Metropolitan Area Network" is in our prototype the domain relevant one. Hence, all other possible interpretations were discarded from the dictionary.

We suggest two major improvements. First, by comparing the synchronized audio data with the data from the slides one can find overlapping areas, e.g., the lecturer speaks in a certain part of the presentation about host ID and shows a slide with the word host ID. Then, it is obvious that the word host ID is a relevant word in the context of this learning object, i.e., more important than a word that was only found in one of both data sources (audio or slide).

Figure 7: Example of 4 identified chains inside a lecture part about IP addressing.



Second, learning objects can be divided into cohesive areas (*chains*) of accumulated appearance of an equal word. E.g., in the space of 5 minutes the speaker uses the expression "host ID" 3 times , so this segment of 5 minutes is called a chain about the concept HostID. A chain is always about one specific word. Chains overlap when the speaker uses different relevant words several time during the same time interval (see figure 7). The overlapping is detected by comparing the start and end time of the different chains. The resulting granularity of the segmentation depends on the allowed gap (threshold) between two identical words. The length (duration) of a chain depends on the number of occurrences of the chain specific word. The higher the frequency of that word inside a chain — that is the more the speaker uses the same word during a relatively short interval — the greater is the semantic relevance of the chain. We think that by considering this measure, we can improve the semantic annotation generated by our system.

5. CONCLUSION

In this paper we have presented an algorithm for generating a semantic annotation for university lectures. It is based on two input sources: the textual content of the slides and the imperfect transliteration of the audio data from the lecturer. Our algorithm maps semantic relevant words from both sources to ontology concepts and roles. The metadata is serialized in a machine readable format, i.e., OWL.

We have shown that the metadata generated in this way can be used by a semantic search engine. Unfortunately, the quality of the annotation is not as good as if it were done by a human. We have identified different reasons for the weakness of our solution and suggested different improvements. But, the quality of the generated semantic annotation is good enough to allow the semantic search engine to yield more precise results. Furthermore, the number of perfect hits is greater than that of a classical keyword-based search engine.

Currently we are working on the synchronization of the audio data with the slide transitions. In future work, we will introduce a weight measure of the identified concept and roles in the data sources, i.e., metadata that are found in both sources (audio and slides) will have a greater relevancy than those found in one source only.

This project has been developed in the context of the Web University project[8], which aims to explore novel internet- and IT-technologies in order to enhance university teaching and research. The application of our algorithm is not limited to annotating university lectures or presentations in general. 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.

6. REFERENCES

[1] J. Allen. *Natural Language Understanding*. Addison Wesley, 1994.

[2] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[3] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[4] M. Bertini, A. D. Bimbo, C. Torniai, R. Cucchiara, and C. Grana. Mom: Multimedia ontology manager. a framework for automatic annotation and semantic retrieval of video sequences. In *ACM SIGMM*, pages 787–788, 2006.

[5] Y. Chen and W. J. Heng. Automatic synchronization of speech transcript and slides in presentation. In *International Symposium on Circuits and Systems (ISCAS)*, pages 568–571, 2003.

[6] H. S. Christopher D. Manning. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[7] M. Engelhardt, A. Hildebrand, D. Lange, and T. C. Schmidt. Reasoning about eLearning Multimedia Objects. In *International Workshop on Semantic Web Annotations for Multimedia (SWAMM)*, 2006.

[8] A. Haubold and J. R. Kender. Augmented segmentation and visualization for presentation videos, 2005.

[9] W. Hürst, T. Kreuzer, and M. Wiesenhütter. 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)*, pages 135–143, 2002.

[10] A. Jaimes, T. Nagamine, J. Liu, K. Omura, and N. Sebe. Affective meeting video analysis. In *IEEE Multimedia and Expo*, pages 1412–1415, 2005.

[11] N. Karam, S. Linckels, and C. Meinel. Semantic composition of lecture subparts for a personalized e-learning. In *European Semantic Web Conference*, volume 4519 of *Lecture Notes in Computer Science*, pages 716–728, 2007.

[12] S. Linckels and C. Meinel. Resolving ambiguities in the semantic interpretation of natural language questions. In *Intelligent Data Engineering and Automated Learning (IDEAL)*, volume 4224 of *LNCS*, pages 612–619, 2006.

[13] R. Mertens, H. Schneider, O. Mller, and O. Vornberger. Hypermedia navigation concepts for lecture recordings. In *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, pages 2480–2847, 2004.

[14] R. Mitkov, editor. *The Oxford Handbook of Computational Linguistics*. Oxford University Press, 2004.

[15] C.-W. Ngo, F. Wang, and T.-C. Pong. Structuring lecture videos for distance learning applications. In *Multimedia Software Engineering*, pages 215– 222, 2003.

[16] S. Repp and C. Meinel. Segmenting of recorded lecture videos - the algorithm voiceseg. In *Signal Processing and Multimedia Applications (SIGMAP)*, pages 317–322, 2006.

[17] S. Repp and C. Meinel. Semantic indexing for recorded educational lecture videos. In *International Conference on Pervasive Computing and Communications Workshops (PERCOMW)*, page 240, 2006.

[18] H. Sack and J. Waitelonis. Automated annotations of synchronized multimedia presentations. In *Workshop on Mastering the Gap: From Information Extraction to Semantic Representation, CEUR Workshop Proceedings*, 2006.

[19] H. Sack and J. Waitelonis. Integrating social tagging and document annotation for content-based search in multimedia data. In *Semantic Authoring and Annotation Workshop (SAAW)*, 2006.

[20] R. A. Schmidt. Terminological representation, natural language & relation algebra. In *German AI Conference (GWAI)*, volume 671 of *LNCS*, pages 357–371, 1993.

[21] J. Tejedor, R. Garca, M. Fernndez, F. J. Lpez-Colino, F. Perdrix, J. A. Macas, R. M. Gil, M. Oliva, D. Moya, J. Cols, , and P. Castells. Ontology-based retrieval of human speech. In *Workshop on Web Semantics (WebS 2007)*, 2007.

[22] W. W. W. C. W3C. *OWL Web Ontology Language*. `http://www.w3.org/TR/owl-features/`, 2004.

[23] F. Wang, C.-W. Ngo, and T.-C. Pong. Prediction-based gesture detection in lecture videos by combining visual, speech and electronic slides. In *IEEE Multimedia and Expo*, pages 653–656, 2006.

[24] P. Wolf, W. Putz, A. Stewart, A. Steinmetz, M. Hemmje, and E. Neuhold. Lecturelounge – experience education beyond the borders of the classroom. *International Journal on Digital Libraries*, 4(1):39–41, 2004.

[25] N. Yamamoto, J. Ogata, and Y. Ariki. Topic segmentation and retrieval system for lecture videos based on spontaneous speech recognition. In *European Conference on Speech Communication and Technology*, pages 961–964, 2003.

[26] Y. Zhu and D. Zhou. Video browsing and retrieval based on multimodal integration. In *Web Intelligence*, pages 650–653, 2003.

[8] `http://www.hpi.uni-potsdam.de/~meinel/research/web_university.html`