Using Automatic Speech Transcriptions in Lecture Recommendation Systems

A. Pérez-González-de-Martos, J.A. Silvestre-Cerdà, M. Rihtar, A. Juan, and J. Civera

MLLP, DSIC, Universitat Politècnica de València (UPV)
{aperez,jsilvestre,ajuan,jcivera}@dsic.upv.es
Jožef Stefan Institute (IJS)
matjaz.rihtar@ijs.si

Abstract. One problem created by the success of video lecture repositories is the difficulty faced by individual users when choosing the most suitable video for their learning needs from among the vast numbers available on a given site. Recommender systems have become extremely common in recent years and are used in many areas. In the particular case of video lectures, automatic speech transcriptions can be used to zoom in on user interests at a semantic level, thereby improving the quality of the recommendations made. In this paper, we describe a video lecture recommender system that uses automatic speech transcriptions, alongside other relevant text resources, to generate semantic lecture and user models. In addition, we present a real-life implementation of this system for the VideoLectures.NET repository.

Keywords: recommender systems, automatic speech recognition, video lectures

1 Introduction

Online multimedia repositories are rapidly growing and being increasingly recognised as key knowledge assets. This is particularly true in the area of education, where large repositories of video lectures and Massive Open Online Courses (MOOCs) are becoming a permanent feature of the learning paradigm in higher education. A well-known example of this is the VideoLectures.NET repository, which currently includes more than 18,000 educational videos covering different topics of science.

These repositories are being subtitled in several languages in order to make them accessible to speakers of different languages and to people with disabilities [4, 21]. The lack of efficient solutions to meet this need is the motivation behind the European project transLectures [15, 19], which aims at developing innovative, cost-effective solutions for producing accurate transcriptions and translations for large video repositories. Transcriptions and translations of video lectures are the basis from which numerous other technologies can be derived.

For instance, digital content management applications such as lecture categorisation, summarisation, automated topic finding, plagiarism detection and lecture recommendation.

This latter has become essential due to the significant growth of video lecture repositories. Users are often overwhelmed by the amount of lectures available and may not have the time or knowledge to find the most suitable videos for their learning requirements. Up until recently, recommender systems have mainly been applied in areas such as music [8, 10], movies [2, 22], books [11] and e-commerce [3], leaving video lectures largely to one side. Only a few contributions to this particular area can be found in the literature, most of them focused on VideoLectures.NET [1]. However, none of them has explored the possibility of using lecture transcriptions to better represent lecture contents at a semantic level.

In this paper we describe a content-based lecture recommender system that uses automatic speech transcriptions, alongside lecture slides and other relevant external documents, to generate semantic lecture and user models. In Section 2 we give an overview of this system, focusing on the text extraction and information retrieval process, topic and user modeling and the recommendation process. In Section 3 we address the dynamic update of the recommender system and the required optimisations needed to maximise the scalability of the system. The integration of the system presented in Sections 2 and 3 into VideoLectures.NET, carried out as part of the PASCAL Harvest Project La Vie, is described in detail in Section 4. Finally, we close with some concluding remarks, in Section 5.

2 System Overview

Fig. 1 gives an overview of the recommender system. The left-hand side of the figure show the topic and user modeling procedure, which can be seen as the training process of the recommender system. To the right we see the recommendation process. The aim of topic and user modeling is to obtain a simplified representation of each video lecture and user. The resulting representations are stored in a recommender database. This database will be exploited later in the recommendation process in order to recommend lectures to users.

As shown in Fig. 1, every lecture in the repository goes through the topic and user modeling process, which involves three steps. The first step is carried out by the text extraction module. This module comprises three submodules: ASR (Automatic Speech Recognition), WS (Web Search) and OCR (Optical Character Recognition). As its name suggests, the ASR submodule generates an automatic speech transcription of the video lecture. The WS submodule uses the lecture title to search for related documents and publications on the web. The OCR submodule extracts text from the lecture slides, where available. The second step takes the text retrieved by the text extraction module and computes a *bag-of-words* representation. This bag-of-words representation consists of a simplified text description commonly used in natural language processing and information retrieval. More precisely, the bag-of-words representation of a given

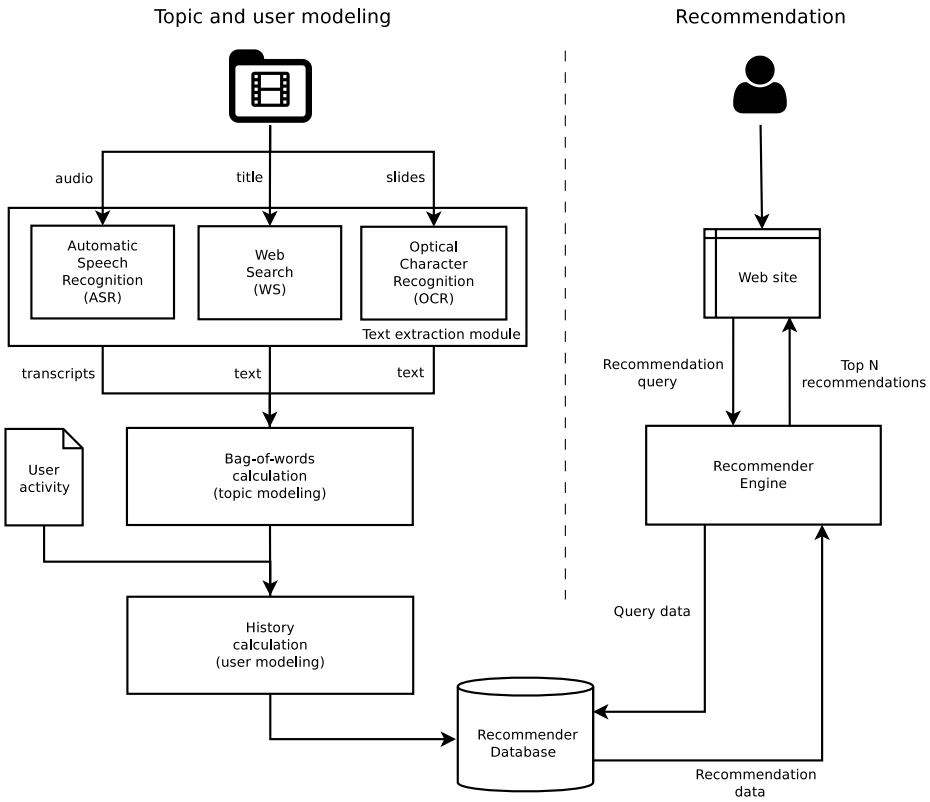

Fig. 1. System overview.

text is its vector of word counts over a fixed vocabulary. Finally, in the third step, lecture bags-of-words are used to represent the users of the system. That is, each user is represented as the bag-of-words computed over all the lectures the user has ever seen.

When the topic and user modeling process ends, the recommender database is ready for exploitation by the recommender engine (see the right-hand side of Fig. 1). This engine uses recommendation features to calculate a measure s of the suitability of the recommendation for every (u, v, r) triplet, where u refers to a particular user, v is the lecture they are currently viewing and r is a hypothetical lecture recommendation. In recommender systems, this is usually referred to as the *utility function* [13]. Specifically, it indicates how likely it is that a user u would want to watch lecture r after viewing lecture v. For instance, this utility function can be computed as a linear combination of recommendation features:

$$s(u, v, r) = \boldsymbol{w} \cdot \boldsymbol{x} = \sum_{n=1}^{N} w_n \cdot x_n \tag{1}$$

where $\boldsymbol{x}$ is a feature vector computed for the triplet (u, v, r), $\boldsymbol{w}$ is a feature weight vector and N is the number of recommendation features. In this work, the following recommendation features were considered:

1. *Lecture popularity:* number of visits to lecture r.
2. *Content similarity:* weighted dot product between the lecture bags-of-words v and r [6].
3. *Category similarity:* number of categories (from a predefined set) that v and r have in common.
4. *User content similarity:* weighted dot product between the bags-of-words u and r.
5. *User category similarity:* number of categories in common between lecture r and all the categories of lectures the user u has watched in the past.
6. *Co-visits:* number of times lectures v and r have been seen in the same browsing session.
7. *User similarity:* number of different users that have seen both v and r.

Feature weights $\boldsymbol{w}$ can be learned by training different statistical classification models, such as support vector machines (SVMs), using positive and negative (u, v, r) recommendation samples.

The most suitable recommendation $\hat{r}$ for a given u and v is computed as follows:

$$\hat{r} = \underset{r}{\operatorname{argmax}}\, s(u, v, r) \tag{2}$$

However, in recommender systems the most common practice is to provide the user the M recommendations r that achieve the highest utility values s, for instance, the first 10 lectures.

3 System Updates and Optimisation

Lecture repositories are rarely static. They may grow to include new lectures, or have outdated videos removed. Also, users' learning progress or interactions with the repository influence the user models. The recommender database must therefore be constantly updated in order to include the new lectures added to the repository and update the user models. Furthermore, the addition of new lectures to the system might lead to changes to the bag-of-words (fixed) vocabulary. Any variation to this vocabulary involves a complete regeneration of the recommender database. That said, changes to the vocabulary may not be significant until a substantial percentage of new lectures has been added to the repository.

Two different update scenarios can be defined: the incorporation of new lectures and updating the user models, on the one hand, and the redefiniton of the bag-of-words vocabulary, including the regeneration of both the lecture and user bags-of-words, on the other. We will refer to these scenarios as *regular update* and *occasional update*, respectively, after the different periodicities with which they are meant to be run.

- *Regular update*: The regular update is responsible for including the new lectures added to the repository and updating the user models with the last user activity, both in the recommender database. As its name suggests, this process is meant to be run on a daily basis, depending on the frequency with which new lectures are added to the repository, since new lectures cannot be recommended until they have been processed and included in the recommender database.
- *Occasional update*: As mentioned in Section 2, lecture bags-of-words are calculated under a fixed vocabulary. Since there is no vocabulary restriction on the text extraction process, we need to modify the bag-of-words vocabulary as new lectures are added to the system. The occasional update carries out the process of updating this vocabulary, which involves recalculating both the lecture and user bags-of-words.

In order to maximise the scalability of the system, while also reducing the response time of the recommender, the features *Content similarity*, *Category similarity*, *Co-visits* and *User similarity* described in Section 2 are precomputed for every possible lecture pair and stored in the recommender database. Then, during the recommendation process, the recommender engine loads the values of these features, leaving the computation of features *User content similarity* and *User category similarity* until runtime. The decision to calculate the features *User content similarity* and *User category similarity* at runtime was driven by the highly dynamic nature of the user models, in contrast to the lecture models, which remain constant until the bag-of-words vocabulary is changed.

4 Integration into VideoLectures.NET

The proposed recommendation system was implemented and integrated into the VideoLectures.NET repository during the PASCAL2 Harvest Project *La Vie*

(*Learning Adapted Video Information Enhancer*) [12]. Said integration is discussed here across five subsections. First, we describe the VideoLectures.NET repository, in Section 4.1. In Section 4.2 we give a brief overview of the transLectures project, as part of which transcriptions of sufficient accuracy as to be usefully deployed were generated for lectures in this repository. Next, we address topic and user modeling from video lecture transcriptions and other text resources, in Section 4.3. In Section 4.4 we describe how recommender feature weights were learned from data collected from the existing VideoLectures.NET recommender system. Finally, we present our evaluation of the system in Section 4.5.

4.1 The VideoLectures.NET Repository

VideoLectures.NET [20] is a free and open access repository of video lectures mostly filmed by people from the Jožef Stefan Institute (JSI) at major conferences, summer schools, workshops and other events from many fields of science. It collects high quality educational content, recorded to high quality, homogeneous standards. The portal is aimed at promoting science, the exchange ideas and knowledge sharing by providing high quality didactic contents not only for the scientific community, but also the general public. VideoLectures.NET has so far published more than 18,000 educational videos. Relevant details regarding the repository can be found in Table 1.

Table 1. Basic statistics on the VideoLectures.NET repository (June 2014)

Number of videos	18,824
Total number of authors	12,252
Total duration (in hours)	11,608
Average lecture duration (in minutes)	37

4.2 transLectures

The generation of accurate speech transcriptions for the VideoLectures.NET repository was carried out as part of the European research project transLectures [19]. transLectures aims to develop a set of tools for the automatic generation of quality transcriptions and translations for large video lecture repositories. At the scientific level, the goals of transLectures are to advance the state-of-the-art in model adaptation (to the domain, to the speaker, and using title searches and text data extracted from the presentation slides) and intelligent human-machine interaction, both as means of efficiently improving the end quality of the automatic transcriptions and translations generated.

The English subset of the VideoLectures.NET repository was automatically transcribed using the transLectures-UPV Toolkit [18]. The recommender system was able to access the transcriptions via the transLectures Platform API [16, 17].

4.3 Topic and User Modeling

The first step in generating lecture and user models involved collecting textual information from different sources. In particular, for VideoLectures.NET, the text extraction module gathered textual information from the following sources:

- transLectures speech transcriptions.
- Web search-based textual information from Wikipedia, DBLP and Google (abstracts and/or articles).
- Text extracted from lecture presentation slides (PPT, PDF or PNG using Optical Character Recognition (OCR)).
- VideoLectures.NET internal database metadata.

Next, the text extraction module output was used to generate lecture bags-of-words for every lecture in the repository. These bags-of-words, as mentioned in Section 2, were calculated under a fixed vocabulary that was obtained by applying a threshold to the number of different lectures in which a word must appear in order to be included. By means of this threshold, vocabulary size is significantly reduced, since uncommon and/or very specific words are disregarded. Once defined, term weights were calculated using *term frequency-inverse document frequency* (td-idf), a statistical weighting scheme commonly used in information retrieval and text mining [9]. Specifically, tf-idf weights are used to calculate the features *Content similarity* and *User content similarity*. Finally, the VideoLectures.NET user activity log was parsed in order to obtain values for the feature *Co-visits* for all possible lecture pairs, as well as a list of lectures viewed per user. This list was used together with the lectures bags-of-words to generate the users bags-of-words and categories. These, in turn, were used to calculate *User content similarity* and *User category similarity*, respectively, as well as *User similarity* for all possible lecture pairs. In a final step, all this data was stored in the recommender database in order to be exploited by the recommender engine in the recommendation process.

4.4 Learning Recommendation Feature Weights

Once the data needed to compute recommendation feature values for every possible (u, v, r) triplet in the repository was made available, the next step was to learn the optimum feature weights $\boldsymbol{w}$ for the calculation of the utility function shown in Equation 1. To this end, an SVM classifier was trained using data collected from the existing VideoLectures.NET naïve recommender system (based only on keywords extracted from the lecture titles). Specifically, every time a user clicked on any of the 10 recommendation links provided by this recommender system, 1 positive and 9 negative samples were registered. SVM training was performed using the SVM^{light} open-source software [7]. The optimum feature weights were those that obtained the minimum classification error over the recommendation data.

4.5 Evaluation

Although there are many different approaches to the evaluation of recommender systems [14, 5], it is difficult to state any firm conclusions regarding the quality of the recommendations made until they are deployed in a real-life setting. The La Vie project therefore provided an ideal evaluation framework, being deployed across the official VideoLectures.NET site. The strategy followed for the objective evaluation of the La Vie recommender was to compare it against the existing VideoLectures.NET recommender by means of a *coin-flipping* approach. Specifically, this approach consisted of logging user clicks on recommendation links provided by both systems on a 50/50 basis and comparing the total number of clicks recorded for each system.

The results did not show any significant differences between the two recommenders in terms of user behaviour. This can be explained by the fact that user-click count alone is not a legitimate point of comparison for recommendation quality. For instance, random variables not taken into account might influence how users respond to the recommendation links provided. As an alternative, we can compare the rank of the recommendations clicked by users within each system. Specifically, for each recommendation clicked by a user in either system, we can compare how the same recommendation ranked in the other system. This might be a more appropriate measure for comparing the recommendations in terms of suitability. However, additional data need to be collected in order to carry out this alternative evaluation. This data is currently being collected and future evaluation results will be obtained following this rank comparison approach.

Despite the lack of objective evidence for assessing the comparative performance of the La Vie system, subjective evaluations indicate that the proposed recommender system provides better recommendations than the existing VideoLectures.NET recommender. Fig. 2 shows recommendation examples from both systems for a new user viewing a random VideoLectures.NET lecture. Although recommendation suitability is a subjective measure, La Vie recommendations seem to be more appropriate in terms of content similarity.

5 Conclusions

In this paper we have shown how automatic speech transcriptions of video lectures can be exploited to develop a lecture recommender system that can zoom in on user interests at a semantic level. In addition, we have described how the proposed recommender system has been particularly implemented for the VideoLectures.NET repository. This implementation was later deployed in the official VideoLectures.NET site.

The proposed system could also be extended for deployment across more general video repositories, provided that video contents are well represented in the data obtained by the text extraction module.

By way of future work we intend to evaluate the recommender system using other evaluation approaches that measure the suitability of the recommendations

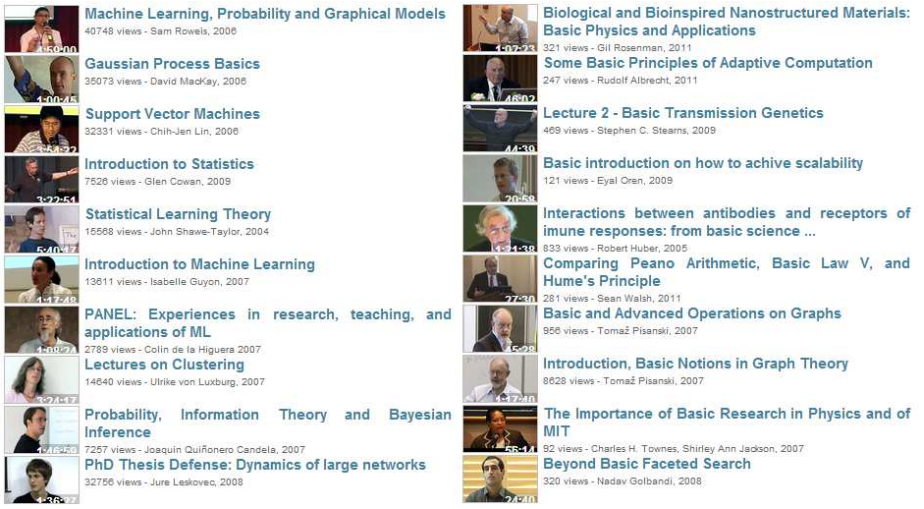

Fig. 2. On the left, La Vie system recommendations for a new user after viewing "Basics of probability and statistics" VideoLectures.NET lecture. On the right, recommendations offered by VideoLectures.NET's existing system.

more accurately, such as the aforementioned recommendation rank comparison. In addition, it is our intention to perform several analysis on the importance of the speech transcription with respect to other variables regarding recommendations quality.

Acknowledgments. The research leading to these results has received funding from the PASCAL2 Network of Excellence under the PASCAL Harvest Project La Vie, the EU 7th Framework Programme (FP7/2007-2013) under grant agreement no. 287755 (transLectures), the ICT Policy Support Programme (ICT PSP/2007-2013) as part of the Competitiveness and Innovation Framework Programme (CIP) under grant agreement no. 621030 (EMMA), the Spanish MINECO Active2Trans (TIN2012-31723) research project, and by the Spanish Government with the FPU scholarship AP2010-4349.

References

1. Antulov-Fantulin, N., Bošnjak, M., Znidaršic, M., Grcar, M.e.a.: Ecml-pkdd 2011 discovery challenge overview. Discovery Challenge (2011)
2. Carrer-Neto, W., Hernández-Alcaraz, M.L., Valencia-García, R., García-Sánchez, F.: Social knowledge-based recommender system. application to the movies domain. Expert Systems with Applications 39(12), 10990–11000 (2012)
3. Castro-Schez, J.J., Miguel, R., Vallejo, D., López-López, L.M.: A highly adaptive recommender system based on fuzzy logic for b2c e-commerce portals. Expert Systems with Applications 38(3), 2441–2454 (2011)

4. Fujii, A., Itou, K., Ishikawa, T.: Lodem: A system for on-demand video lectures. Speech Communication 48(5), 516 – 531 (2006)
5. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS) 22(1), 5–53 (2004)
6. Joachims, T.: A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Tech. rep., DTIC Document (1996)
7. Joachims, T.: Svmlight: Support vector machine. SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund 19(4) (1999)
8. Lee, S.K., Cho, Y.H., Kim, S.H.: Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations. Information Sciences 180(11), 2142–2155 (2010)
9. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval, vol. 1. Cambridge university press Cambridge (2008)
10. Nanopoulos, A., Rafailidis, D., Symeonidis, P., Manolopoulos, Y.: Musicbox: Personalized music recommendation based on cubic analysis of social tags. Audio, Speech, and Language Processing, IEEE Transactions on 18(2), 407–412 (2010)
11. Núñez-Valdéz, E.R., Cueva Lovelle, J.M., Sanjuán Martínez, O., García-Díaz, V., Ordoñez de Pablos, P., Montenegro Marín, C.E.: Implicit feedback techniques on recommender systems applied to electronic books. Computers in Human Behavior 28(4), 1186–1193 (2012)
12. PASCAL Harvest Programme. `http://www.pascal-network.org/?q=node/19`
13. Resnick, P., Varian, H.R.: Recommender systems. Communications of the ACM 40(3), 56–58 (1997)
14. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Recommender systems handbook, pp. 257–297. Springer (2011)
15. Silvestre, J.A., del Agua, M., Garcés, G., Gascó, G., Giménez-Pastor, A., Martínez, A., de Martos, A.P.G., Sánchez, I., Martínez-Santos, N.S., Spencer, R., Miró, J.D.V., Andrés-Ferrer, J., Civera, J., Sanchís, A., Juan, A.: translectures. In: Proceedings of IberSPEECH 2012 (2012)
16. Silvestre-Cerdà, J.A., Pérez, A., Jiménez, M., Turró, C., Juan, A., Civera, J.: A system architecture to support cost-effective transcription and translation of large video lecture repositories. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC) 2013. pp. 3994–3999 (2013), `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6722435`
17. The transLectures-UPV Team: The transLectures Platform (TLP). `http://translectures.eu/tlp`
18. The transLectures-UPV Team: transLectures-UPV toolkit (TLK) for Automatic Speech Recognition. `http://translectures.eu/tlk`
19. UPVLC and XEROX and JSI-K4A and RWTH and EML and DDS: translectures. https://translectures.eu/ (2012)
20. Videolectures.NET: Exchange ideas and share knowledge. `http://www.videolectures.net/`
21. Wald, M.: Creating accessible educational multimedia through editing automatic speech recognition captioning in real time. Interactive Technology and Smart Education 3(2), 131–141 (2006)
22. Winoto, P., Tang, T.Y.: The role of user mood in movie recommendations. Expert Systems with Applications 37(8), 6086–6092 (2010)