big data and cognitive computing

MDPI

Article

Semantic Hierarchical Indexing for Online Video Lessons Using Natural Language Processing

Marco Arazzi, Marco Ferretti, Antonino Nocera *

Department of Electrical, Computer and Biomedical Engineering, University of Pavia, 27100 Pavia, Italy; marco.arazzi01@universitadipavia.it (M.A.); marco.ferretti@unipv.it (M.F.)
* Correspondence: antonino.nocera@unipv.it

Abstract: Huge quantities of audio and video material are available at universities and teaching institutions, but their use can be limited because of the lack of intelligent search tools. This paper describes a possible way to set up an indexing scheme that offers a smart search modality, that combines semantic analysis of video/audio transcripts with the exact time positioning of uttered words. The proposal leverages NLP methods for topic modeling with lexical analysis of lessons' transcripts and builds a semantic hierarchical index into the corpus of lessons analyzed. Moreover, using abstracting summarization, the system can offer short summaries on the subject semantically implied by the search carried out.

Keywords: video lesson indexing; semantic indexing; topic modeling; natural language processing; abstractive summarization

check for updates

Citation: Arazzi, M.; Ferretti, M.; Nocera, A. Semantic Hierarchical Indexing for Online Video Lessons Using Natural Language Processing. *Big Data Cogn. Comput.* **2023**, *7*, 107. https://doi.org/10.3390/bdcc7020107

Academic Editor: Nik Bessis

Received: 31 March 2023
Revised: 15 May 2023
Accepted: 26 May 2023
Published: 31 May 2023

1. Introduction

Teaching institutions, public and private alike, are continuously updating their approach to delivering advanced teaching services to their students. The availability of online learning material and, specifically, video lessons has become a key feature of the offering; both regular students, enrolled in campus activities, and professionals, engaged in upskilling programs or just continuously updating their knowledge background, can benefit from video material for off-line rehearsing.

The pandemic of recent years has transformed this tendency into a widespread, generalized, and ubiquitous powerful thrust. As a consequence, all universities and high-level institutions have collected huge quantities of teaching material in the form of video lessons. Searching for and locating "interesting" sections of videos becomes an enhanced feature for otherwise static repositories, which may be difficult to use.

As an instance of this pattern, at the University of Pavia, online lesson delivery was introduced during the pandemic starting with the spring semester of 2020 and has continued for at least three semesters, well into 2021. Even if it was somehow intermingled with on-premise activity, this mode has generated much video and audio material, which might be made available, with proper authorization, both to students enrolled in an ongoing course edition and to students wishing to rehearse and to have access to lessons they could not attend.

Sequential access to the video/audio material is often unproductive, and some form of advanced access through a "quick and intelligent" search would be a most welcome feature. We call this mode *semantic hierarchical indexing* since it aims at offering intelligent search based on semantic clues while allowing delving into the videos/audio material at the actual timestamps where uttered words that match the semantic clues are located.

For the search to be really effective, it is, therefore, necessary to build two subsystems: (i) a tool to set up a list of the most-important *topics* in a single lesson and/or in a set of lessons; (ii) a *smart index* in the video/audio stream that locates the tokens (single words or short sequences of proper words, as will be discussed in the sequel) attached to the topics.

The hierarchy has the list of topics at its head; each topic is represented at a lower level by a list of tokens. This semantic hierarchy can be used in either of two modes: in a top-down fashion, by selecting one topic and retrieving all spots in a lesson (or set of lessons) where the tokens are uttered for that topic, or in a bottom-up fashion, by searching a single token and retrieving also all other tokens that are semantically akin to it, since they belong to the same topic.

The combined availability of these two facilities, namely a "smart video index" and a "smart topics list", can really benefit students, offering them a tool for effectively rehearsing learning material attached to lessons that they have attended possibly months earlier or that they never attended at all.

There is furthermore a third possible outcome of this approach, which is the compilation of a "summary" associated with the topics. If an *external source* of text is available that belongs in the same domain of the topics, it is possible to apply a *summarization* procedure to that text on the basis of the topic involved in the search. This can offer the student a short, but focused text on the subject implied by the topic. One might object that university courses are usually described by a syllabus, which should actually give a short summary of the main subjects covered in the lessons. However, the structure of syllabi, the detail they offer, and the length to which they extend the textual description are highly variable even within the course of a single track, not to mention the whole corpus within a university.

The work we are reporting here was developed as a testbed prototype, by choosing a Bachelor's-level course (taught by one of the authors), for which full access is available, along with other courses freely available on the web from two different domains, namely: "Theory of Computation" and "Databases". The set of lessons was used both for devising the low-level smart index and for analyzing the feasibility of extracting "meaningful" topics for the whole course.

This paper is organized as follows: Section 2 briefly summarizes the most-relevant scientific literature background. Section 3 offers a short description of the most-relevant NLP techniques deployed in setting up the various parts of the overall procedure. Section 4 covers the development of the smart index that builds the lower level of the system, which allows looking up in the corpus of lessons "elementary concepts" described by *binomials*. Quantitative measurements on the corpus of one of the testing courses are reported. Section 5 describes the approach used to derive the topic list from the corpus of targeted lessons and its relation to the lower-level smart index. Section 6 shows our approach to building targeted summaries for our semantic indexing, using a summarization approach on two external sources. Section 7 describes the experiments and comments on the results, and Section 8 concludes the paper.

2. Related Works

The combination of video indexing and speech analysis and recognition is a well-known and established field and has been exploited in many different environments. Initial manual annotations have long been abandoned.

Most of the approaches available in the literature are mainly suited to content based on slides and speech [1], also leveraging existing materials exploiting an OCR processing phase on the slides extracted from the videos [2]. Indeed, OCR is very effective when applied to typed text; however, if the lessons make heavy use of blackboards, this technique cannot be applied. That is why audio has become the best candidate when it comes to tracking the most important details of a lesson, even if it is less accurate than OCR. Anyway, this technology still presents some flaws; for example, it can still have issues in recognizing some voices or some terms producing Out-Of-Vocabulary (OOV) word errors. Furthermore, an additional problem could be caused by the fact that, usually, speeches lack the strong sentence structure that is typical of written text.

Nevertheless, voice analysis has mature applications with the primary purpose of minimizing the Word Error Rate (WER), particularly to avoid OOV word errors. For this project, a critical point is also the availability of a multi-language model, which is not always

included in applications such as Gaudi [3], PodCastle [4,5], NTU Virtual Instructor [6], and MIT Lecture Browser [7].

There have been some special applications where video/audio indexing has been explored, such as call centers [8] and broadcast news [9]. These contexts have many advanced features, such as detecting speaker changes, which are irrelevant to this research. In the related literature, several approaches have been developed focusing on the problem of topic modeling from different sources. This paper implements a topic-based indexing strategy based on topic extraction from video content.

In particular, the approach proposed in [10] uses a novel two-stream model named the multi-modal aggregation and co-attention network. The model processes audio and video separately with co-attentional interactions. This strategy exploits the pairwise relations between audio and video to better capture the semantic relations among the features, improving the interactions between two modalities. In the approach proposed in [11], instead, the authors extract features from the activations of the layers of a pre-trained CNN model, which receives the Mel spectrograms of the audio signals of the videos as the input. Following this strategy, they can extract compact video descriptors for the frames of the videos. In this way, the authors can compute the similarity between videos by calculating the similarity matrix that contains the pairwise similarity scores between the audio descriptors.

The approach proposed in [12] identifies the areas of research that best reflect the scope of a publication. This solution is integrated into an application called Smart Topic Miner, which allows editors of Springer Nature Journals to annotate papers based on topics from a large ontology of computer science related fields.

The authors of the approach proposed in [13] tried to find an answer to the following question: "in what research topics were the academic community of Computers & Education interested?". In computer and education between 1976 and 2018, 3963 articles were analyzed bibliometrically using a Structural Topic Model (STM) to identify topic hotspots.

In the context of education, students struggle to make informed decisions using content available through online reviews of academic institutions. The approach proposed in [14] was intended to address this difficulty. The paper proposes an Ensemble of Latent Dirichlet Allocation (E-LDA) topic model for automatically categorizing review statements by key features of student discussion.

A similar strategy exploiting a statistical algorithm (LDA) was used by the authors in [15] to identify the key research topics spanning the range of 32 years (1986–2017), based on the full-text corpus of one major journal in the field (*Biology and Philosophy*).

Moreover, the authors of [16] developed two algorithms to implement a summarization process by taking the URL of a video and implementing a summarization process using the video pointed out by the given URL. Additionally, the model gives the user the option of deciding what percentage of the summary is needed compared to the original lecture. The summarization is a subjective process based on two prominent methods, which are incorporated into the model. One is cosine similarity, and the other is the ROUGE score. In the former case, human-generated summaries are not needed, but in the latter case, they are. TF-IDF and Gensim can both achieve greater than 90% efficiency by using cosine similarity, while ROUGE scores can achieve 40–50% efficiency.

Another interesting strategy is the one proposed by [17], in which the authors designed a pipeline, which, starting from an audio file, is capable of performing a qualitative content-based analysis of the topics delivered in the lecture, rating the quality of the lecture content with the definition of a quality metric, and producing a summarization. The strategy proposed has an initial phase of audio processing to clean the audio track from spurious noise produced by the audience, then it is transcribed by using the Python *Speech recognition library* (https://pypi.org/project/automatic-speech-recognition/, accessed on 30 May 2023). Furthermore, a pre-processing phase is applied on the obtained text to translate it into English and to handle text anomalies, such as removing extra free spaces, periods in multi-period abbreviations and punctuations, converting plurals into singulars,

and text to lowercase. Focusing on the summarization task starting from the sanitized text, the authors obtained a vectorial representation for each sentence using a Word2Vec [18] model; so, they proceeded to generate a graph in which the sentences are the nodes and the edges denote the similarity between them. Then, similar to the *Google page rankingsystem* (https://www.google.com/search/howsearchworks/algorithms/, accessed on 30 May 2023), they produced an extractive summary of the lecture.

3. Background

This section is devoted to introducing some background knowledge on the natural language tools and solutions that we leveraged to build our solution. It is worth stressing that this section does not aim to provide a complete survey on the role of NLP in the context of the analysis of video lessons. It simply serves the purpose of providing some reference to the reader to better understand how our approach works. In particular, our strategy was configured as a pipeline in which each module performs a different task belonging to the Natural Language Processing field. Each task involved in our approach was fulfilled, when required, using technologies that represent the state-of-the-art for the domain of reference, such as transformer-based models.

In Section 3.1, we present GloVe, the word-embedding technology that we adopted to identify the most-similar terms in a text against a target word. Section 3.2 is devoted to presenting BERTopic, the topic-modeling technique that we exploited to extract the most-important topics. Finally, in Section 3.3, we present how the embeddings generated by a Sentence-BERT model can be useful to calculate the similarity between sentences.

3.1. Word Embedding: GloVe

Word embedding is a text-processing technique that represents one of the fundamentals of the Natural Language Processing field. The main objective of word embedding is to find an n-dimensional vectorial representation of each of the words present in the corpus of a text on which the word embedding algorithm has been trained. One of the main features of the vector space generated by these techniques is that vectors that are close in the space are expected to be related to each other.

In the scientific literature, many word embedding strategies can be found, among which two of the most-well-known and best-performing are Word2Vec [18,19] and GloVe [20].

In particular, GloVe, the combination of the two words Global and Vector, is an unsupervised learning algorithm that takes into account global word-to-word co-occurrence statistics from a given corpus to create representations that showcase interesting linear substructures of the word vector space. In particular, GloVe is a log-bilinear model with a weighted least-squares objective and is based on the simple observation that meaning can be encoded through ratios of word–word co-occurrence probabilities.

For our solution, we exploited GloVe vector representations to predict the most-similar words occurring in a text given a target one, by calculating the Euclidean distance or the cosine similarity between their vectorial embeddings. Cosine similarity in data analysis is the cosine of the angle between two non-zero vectors within an inner product space. This can be calculated as the "dot" product of the vectors divided by the product of their lengths, which results in the cosine of the angle between the two given vectors. Cosine similarity gives an output that lies in the interval $[-1, 1]$, where 1 means that, in terms of similarity, the two vectors are identical; -1, instead, means that the two documents are the exact opposite. In our case, since the information that a document is the opposite of another is not needed, we considered the negative similarities as zero, reducing the output interval to [0, 1].

3.2. Topic Modeling: BERTopic

The objective of topic modeling algorithms is to extrapolate the most-relevant concepts discussed inside a text provided as input. In the scientific literature, many approaches have been proposed, and one of the most-famous is the Latent Dirichlet Allo-

cation (LDA) [21] algorithm. The latent Dirichlet allocation algorithm leverages Dirichlet distributions to represent relations between documents and topics and between topics and words across documents.

In recent years, more sophisticated strategies have been proposed, among which the BERTopic framework [22] is one of the most-promising. As a matter of fact, this approach has been profitably exploited in several recent research studies [23,24].

The BERTopic approach combines the contribution of a clustering algorithm and vectorial representations of words generated by exploiting a transformer-based model, which represents a state-of-the-art solution in the text analysis field. In particular, this strategy leverages the embeddings of words generated by a modified version of the BERT model [25], developed by Google in 2019, known as Sentence-BERT (SBERT) [5], which generates sentence embeddings in multiple languages using different pre-trained models. More in detail, SBERT is a modification of the well-known BERT network that exploits Siamese and triplet network structures to infer semantically meaningful sentence embeddings.

Since the output of SBERT is a high-dimensional vector, it may adversely affect the performance of a clustering algorithm. Hence, a nonlinear algorithm, Uniform Manifold Approximation and Projection (UMAP) [26], is used to reduce the dimensions of the output vector. In particular, a high-dimensional graph representation of the data is first generated using the UMAP's graph layout algorithms. Then, a low-dimensional graph is optimized to be structurally similar to the high-dimensional graph representation.

The dimensionally reduced embedding can, then, be fed as the input to a clustering algorithm of choice. In particular, the BERTopic framework uses HDBSCAN to cluster the representations of the sentences in groups that represent the most-relevant topics. HDBSCAN [27] is a hierarchical version of the famous DBSCAN clustering algorithm. In contrast to DBSCAN, HDBSCAN relies on high-density clustering, which reduces the clustering noise issue of DBSCAN and enables hierarchical clustering based on a decision tree approach.

Finally, the most-relevant keywords are extracted from each topic. This can be performed by analyzing the distribution of the words within the topics using a modified version of TF-IDF, called c-TF-IDF, which, instead of considering the cluster as a set of documents, converts it by merging all the documents belonging to a cluster into a single big document and, then, extracts the keywords according to their frequencies using TF-IDF.

3.3. Text Similarity: Sentence-BERT

As mentioned in Sections 3.1 and 3.2, vectorial representations of text data can be exploited in tasks with different purposes.

An additional use of text embedding is the possibility to compute the similarity or distance between two sentences. This task can be performed by computing the cosine similarity between the vectorial representations of two different documents. This strategy leverages the assumption that two documents with a similar representation will occupy a close position in the vector space in which they are represented.

To guarantee continuity with the approach described in Section 3.2, we decided, also for this task, to exploit the embeddings generated by the same Sentence-BERT model.

4. Building Smart Video Indexes

To develop the initial prototype of the system, we relied on existing recordings (video and audio) of Bachelor's lessons. We selected a course (Introduction to Data Bases) that touches on basic subjects taught in all Bachelor's tracks in Informatics and Computer Engineering curricula, so that rich and reliable material can be easily gathered. The available course material in our university consists of 21 lessons, for a grand total of 2.192 GB and an average dimension and duration of 104.4 MB and 1 h 50 m, respectively. Similar numbers describe other courses in a few other Italian universities, which were considered as fallback.

The first feature of the "smart index" is indeed a very simple, yet important one: the identification of the timestamps in the lessons where a given set of "search words" is located. The cardinality of this "search set" turns out to be very small, since it is unlikely that an instructor repeatedly uses long sequences of words to name a subject or a property. Concepts tend to be easily referred to with a couple, possibly a triplet of "words". We, therefore, concentrated at the outset on "binomials" and "trinomials" (to be described later) and eventually retained "binomials" only. The index we built for the whole course consists of a set of entries, where each entry is a "binomial", along with the IDs of the lessons and the timestamps where that binomial is spoken. While apparently extremely simple, this index is indeed effective for quick concept search and, furthermore, can be obtained using fairly standard technology and text transformation services, such as speech-to-text, text lexical analysis with lemmatization, and basic database support. The ultimate goal is to come up with a scalable solution that allows processing increasing quantities of video lessons with matching computing resources. Finally, we give a note on the language issue. Initially, the project was developed by focusing on the Italian language; however, the proposed methodology was, then, adapted to work also on English lessons. The experimental results, reported in Section 7, showed the average performance of our approach for both Italian and English lessons.

4.1. Indexing Procedure

The indexing procedure consists of two phases: the analysis of each lesson and a final merge of the outcomes from all lessons. Briefly, in each lesson, the following five steps are carried out:

1. Speech-to-text conversion, which produces "tokens" and associated timestamps;
2. Token lexical analysis and Parts-Of-Speech (POS) tagging;
3. Grouping of tagged tokens by time proximity, yielding tentative "binomials" and "trinomials";
4. Binomial and trinomial filtering by selecting couples (triples) of significant POS tags;
5. Lemmatization of selected binomials (trinomials).

Once this procedure is carried out on each lesson, lesson outcomes are merged into a single course-level list of tagged and timestamped binomials (trinomials), which are the basis of the actual "index". Each step is described in what follows at a fairly high level of abstraction, namely skipping practical implementation details, such as the underlining DBMS and storage, which are really straightforward, at least when addressing a single course. Scaling to multiple course instances is feasible with current data technology.

4.2. Converting Speech-to-Text with Timestamps

While Speech-to-Text is a very well-established technology, this project has a few features that partially condition and steer the classical alternative "buy" or "pay per use". To begin with, the project is based on the availability of a correct timestamp for the selected "search" items, a feature that cannot be dispensed with, lest the overall goal of "indexing in time" is lost. Furthermore, good punctuation is also of the utmost importance, since subsequent phases of the project, namely topic extraction, benefit from text subdivision into sentences. As or the language models, of course, the project demands a good Italian language model, as any other project instantiation in any other language.

The approach to the "buy" vs. "pay per use" alternative took into account scalability, technology evolution, and financial issues. While initially developed on a small-scale problem instance, the project has to be robust to considerable scaling: the institution might be interested in widespread adoption of the indexing scheme to offer prospective and enrolled students detailed insight into online-available teaching resources. On another side, language models evolve very rapidly with the massive use of high-performance supercomputers, which help derive ever more effective and precise models.

These two facts have suggested using online cloud services to deploy the most-standard parts of the project. We analyzed tools and solutions provided by YouTube,

by the Google Cloud Platform (GCP) [28], and by Amazon Web Services, namely AWS Transcribe [29], and exploited free offers for limited workloads, with the goal of deploying a prototype. The three alternatives proved effective in a WER analysis of the transcripts of three representative lessons; Youtube ranked slightly better, but only Google and AWS services provided punctuation. This is why we used them in various phases.

4.3. POS Tagging and Lexical Analysis

This step of the procedure aims at generating meaningful short lists of spoken words, the "search set" ultimately specified by the user and looked upon within the video lessons. The speech-to-text procedure generates huge lists of tokens/timestamps, which have to be properly filtered, both in the lexical domain and in time. The former criterion calls for lexical analysis and POS tagging in the language of the speaker; the latter instead suggests selecting a "reasonable" time span of contiguous spoken words, which makes the search a realistic one.

Lexical analysis and POS tagging can leverage existing open access tools and services such as the Natural Language Toolkit (NLTK) Python library [30] and, in particular, its pos-tag module. NLTK provides native support for the English language; however, other libraries can be exploited to support other languages, such as TINT [31] for the Italian language, the only option available at the time of this project. It is worth noting that TINT is a fork of the Stanford CoreNLP [32], a Java annotation pipeline framework developed by Stanford that provides a set of Natural Language Processing tools including POS tagging. To keep a consistent approach, we leveraged Stanford CoreNLP as well in this project.

As for the time span for uttered words, a dimension of two/three significant contiguous words seems quite appropriate for a search aimed at finding relevant sentences. Independent of the knowledge domain covered in the lessons, it is likely that a short list of up to three uttered words is able to capture meaningful concepts used by the instructor when delivering the speech. Possibly, it is the role of the words (in the grammar sense, namely parts-of-speech) that can vary in the different knowledge domains. For the information technology domain we are considering in this project, nouns (N) and adjectives (A) seem to be more relevant than verbs. Table 1 shows the POS tags used by TINT and Stanford CoreNLP to categorize words in the Italian and English grammar, respectively. Therefore, we decided to use pairs of words ("binomials") matching the patterns $\{N, N\}$, $\{N, A\}$, and $\{A, N\}$. The inclusion of verbs, as well as the extensions to patterns of three words, proved to be ineffective.

Table 1. A section of POS tags in Stanford CoreNLP and TINT. POS tags that have no correspondence in one framework are shown with "-".

POS Tag Type	TINT	CoreNLP
Adjective	A	JJ
Possessive Adjective	AP	-
Comparative Adjective	-	JJR
Superlative Adjective	-	JJS
Noun	S	NN
Plural Noun	S	NNS
Proper Noun	SP	NNP
Plural Proper Noun	SP	NNPS
Foreign Noun	SW	FW

Lemmatization is the final step of this procedure. The final list of binomials retains the timestamp of the first spoken word in the binomial, for the purpose of indexing in the time domain.

Tables 2–4 offer an assessment of the main interesting quantities within the 21 lessons of the testbed course.

Table 2. Main measures on the 21 lessons testbed.

Lesson #	.mp4 (MB)	Tokens	Tokens (KB)	bi-Grams	Words
1	92.4	10,491	229	2966	1117
2	108.5	11,676	255	3150	991
3	102.9	9555	198	2538	803
4	105.4	9646	212	2624	840
5	114.3	8349	181	2210	703
6	128.7	9281	201	2559	824
7	123.2	9873	211	2613	874
8	72.2	7530	163	2191	828
9	94.2	9290	201	2435	901
10	100.0	10,830	234	2956	1032
11	85.7	10,656	233	2945	1063
12	89.5	9796	215	2722	931
13	73.1	6569	144	1641	591
14	124.2	11,553	252	2994	1052
15	104.1	11,043	241	2971	960
16	96.9	10,251	224	2768	882
17	118.3	9541	207	2442	797
18	112.4	8143	179	2097	761
19	113.2	9520	206	2469	776
20	97.6	10,927	238	2961	1050
21	135.9	10,630	230	2860	902
totals	2192.7	205,150.0	4454.0	55,112.0	18,678.0
avg	104.4	9769.0	212.1	2624.4	889.4

Table 3. Binomial entries (course level), first 10 of 18,920. English translation added.

Binomial	In Lessons	Tot Count	# Lessons
chiave esterna (foreign key)	2 3 4 5 6 8 9 10 17 20	99	10
dipendenze funzionali (functional dependencies)	17 18 19 21	97	4
chiave primaria (primary key)	2 3 4 6 8 9 15 17 18 19 20 21	92	12
basi dati (database)	1 2 4 6 10 11 12 14 15 16 18 20	63	12
modello relazionale (relational model)	1 2 3 4 6 7 8 12 14 15 16 17 18 21	61	14
target list	3 5 6 7 8 9 10 20 21	60	9
vincoli integrità (integrity constraint)	2 3 4 6 8 9 10 12 17 18 21	59	11
punto vista (view point)	1 2 3 5 6 7 8 9 10 11 12 13 14 15 17 18 19 20 21	51	19
legame associativo (associative link)	12 13 14 15 16 17	42	6
codice fiscale (personal ID)	2 4 6 12 13 15 17 18 19 21	41	10

Table 4. Top frequent terms of the obtained dictionary for the proof-of-concept.

Term	Frequency
relazione (relationship)	879
attributo (attribute)	866
chiave (key)	689
schema	649
entità (entity)	583
concetto (concept)	489
proiezione (projection)	375
vincolo (constraint)	359
esterno (foreign)	343
dbms	307

Table 2 shows the number of tokens, bi-grams, and words, along with the dimensions of the lesson they belong to. Table 3 instead lists the top ten most-frequent binomials (matching the patterns $\{N, N\}$, $\{N, A\}$, and $\{A, N\}$) and Table 4 the top ten words with tag $\{N\}$ or $\{A\}$.

The tables show that pruning extracted tokens with a lexical approach gives a reasonable distribution of terms that are likely to carry proper semantics, but this simple statistical filter is not able to cancel out "noise", that is bi-grams that carry no relevant meaning and are due to examples discussed in the lesson (such as "codice fiscale") or to the verbal habit of the speaker.

5. Towards Semantic Indexing

In the previous sections, we discussed a strategy based on the distribution of n-grams to build a basic version of an "analytical" index of the content from a set of video lessons of an online course. Although such an indexing mechanism could be useful as a starting tool to identify the desired contents inside the, possibly big, set of video materials, the classification of the concepts, which the identified indexes refer to, is still left to the final user. Note that this manual activity would require in-depth knowledge of the underlying domain, which cannot be assumed for any category of user to whom the video content could be targeted.

As a matter of fact, in many application scenarios, this could be an important limitation, as being able to group together indexes that refer to related meanings could allow for a better and more solid content retrieval strategy.

With that said, in the next section, we describe an advanced strategy for grouping together n-gram indexes based on their semantics with respect to the underlying domain.

5.1. Using Topic Modeling for Index Clustering

As stated above, the input to this part of our solution is the processed transcripts obtained from the previous steps (see Section 4).

However, because we are dealing with transcripts of video lessons, not all of their content is useful for identifying the topics covered in them. Indeed, typically, lessons include noise that can be caused by several factors, namely: interactions between professors and students, examples that deal with topics not included in the course, organization-related discussions, and so forth.

In order to address this problem, we introduced a supervised component that acts as an expert agent for filtering out such noise from the original input. In particular, our approach starts by constructing a vocabulary of domain terms related to the course of interest by incorporating words from a controlled training set. To build such a training set, several strategies could be adopted, including using domain-related existing manuscripts from the related literature or books. However, because our application context is the academic realm, we leveraged course syllabi. As a matter of fact, in such a context, courses are often presented through syllabi; therefore, in some cases, publishing a syllabus together with the introduction of a course in a study program is mandatory.

A course syllabus is a condensed description of a course, outlining its main characteristics and teaching objectives. Clearly, it contains an essential textual corpus that can be used in our approach to building the aforementioned dataset. Figure 1 shows an example of a syllabus from one of the courses taught by one of the authors of this paper.

To keep terms related to a domain by removing stop words and verbs, syllabi are subjected to a text pre-processing implemented through the Stanford CoreNLP library or derivatives for languages different from English. Then, the terms are lemmatized to disambiguate words that derive from the same primitive. At this point, given a course, C, the set of its (video) lesson transcripts L_C, and its syllabus Sy_C, the result of the previous computation represents our reference domain vocabulary, say V_{Sy_C} and can, hence, be used to feed a subsequent supervised learning task, as follows.

The idea underlying our approach is to expand the above constructed vocabulary for a target course, by adding to it the k most-similar words inside the corpus of the lessons of L_C for each term of V_{Sy_C}. To avoid the inclusion of noise, we also imposed a minimum similarity value, i.e., th_{sim}.

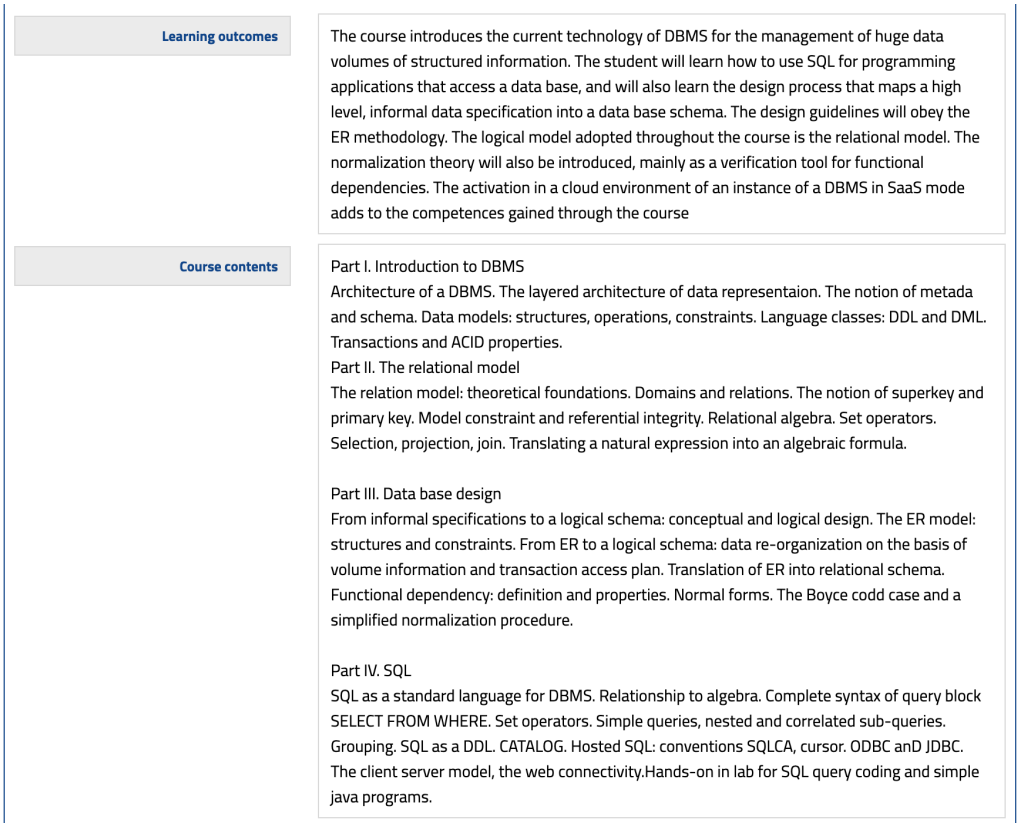

Figure 1. An excerpt from a sample syllabus for a Databases course.

To derive the set of k most-similar terms for V_{Sy_C} from the lesson transcripts, we leverage the word embedding model GloVe, described in Section 3, to generate a specific n-dimensional vector space for a target course C. Specifically, GloVe was trained with the corpus of lesson transcriptions derived from L_C and the original text of Sy_C. At this point, as explained in Section 3, the embeddings generated by GloVe are such that terms representing similar concepts will have near points in the n-dimensional space represented by such embeddings. At this point, the semantic similarity between two terms, say w_i and w_j, can be computed as the cosine similarity of the corresponding embeddings from GloVe, namely $e^{w_i} = [e_1^{w_i}, \cdots, e_n^{w_i}]$ and $e^{w_j} = [e_1^{w_j}, \cdots, e_n^{w_j}]$.

$$sim(w_i, w_j) = \frac{\sum_{t=1}^{n} e_t^{w_i} \cdot e_t^{w_j}}{\sqrt{\sum_{t=1}^{n} (e_t^{w_i})^2} \cdot \sqrt{\sum_{t=1}^{n} (e_t^{w_j})^2}} \quad (1)$$

Given the set of terms V_{L_C} included in the lessons of L_C, the extended set of terms for a vocabulary V_{Sy_C} can be, hence, obtained as follows:

$$EXV_{Sy_C,L_C} = V_{Sy_C} \bigcup V^{sim}_{Sy_C,L_C}$$

where:

$$V^{sim}_{Sy_C,L_C} = \{w_i \in V_{L_C} : \exists\, w_j \in V_{Sy_C} \wedge sim(w_i, w_j) > th_{sim} \wedge |\{w_z \in V_{L_C} : sim(w_z, w_j) > sim(w_i, w_j)\}| < k\}$$

The so-constructed extended set EXV_{Sy_C,L_C} is then used to filter out the words inside the corpus of the processed transcripts. Figure 2 shows a schematic representation of this process, while Algorithm 1 summarizes our noise removal strategy.

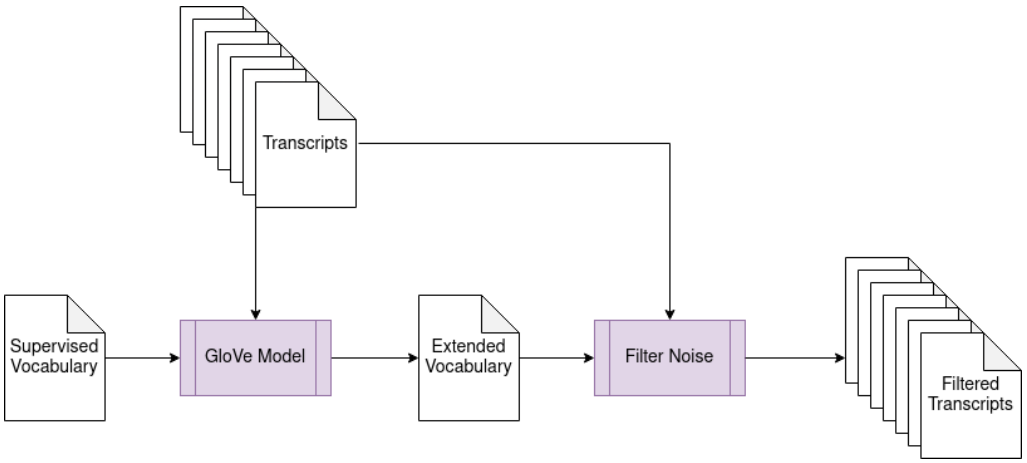

Figure 2. A schematic representation of our advanced noise removal solution.

Algorithm 1 Noise removal.

Input:
$\mathcal{T}r \leftarrow$ [transcripts] ▷ Video transcripts
$GloVe$ ▷ GloVe model trained on transcripts
$Voc \leftarrow [terms]$ ▷ Supervised set of terms specific for the course
Output:
$Ext\_Voc \leftarrow [terms]$ ▷ Extended set of supervised terms
$Fil\_Tr \leftarrow [transcripts\ without\ noise]$ ▷ Set of transcripts cleaned from noise

for each $t \in \mathcal{V}oc$ **do**
  $new\_ts \leftarrow GloVe.most\_similar(t, k)$
  $Ext\_Voc.append(t \wedge new\_ts)$
end for
for each $trs \in \mathcal{T}r$ **do**
  $fil\_trs \leftarrow$ *filter the words of trs using Ext_Voc*
  $Fil\_Tr.append(fil\_trs)$
end for
return Ext_Voc, Fil_Tr

The transformed original corpus of transcripts is now ready to be digested by a topic-modeling algorithm. A schematic representation of this solution is illustrated in Figure 3.

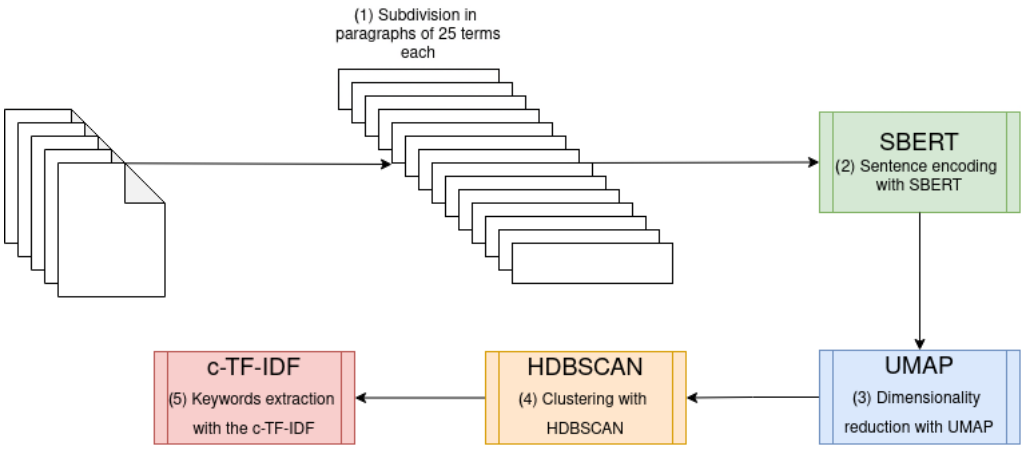

Figure 3. The architecture of the adopted topic-modeling strategy.

As a first step, each transcript is subdivided into paragraphs of 25 terms each, and then, these documents are processed according to the first step of the framework BERTopic, described in Section 3.2. It is important to highlight that this framework can work not only with the encoding of single terms, but also of n-grams of different dimensions, thus making it particularly suitable for our solution. The combination of single terms into n-grams, in our case, highly enhances the capability of the model to identify concepts derived from the composition of two or more terms. To give an example of the importance of this property, consider, for instance, a course on "Theory of Computation", in which "regular" and "expression" are already important words in the course vocabulary, but, when combined together, we obtain a bi-gram with a much more robust meaning for the reference domain.

Models derived from BERT typically generate 768-dimensional encoded vectors; therefore, as discussed in Section 3, BERTopic applies a dimensionality reduction algorithm, called UMAP, to enhance the performance of the model. A crucial aspect of this phase is the selection of the size of the reduced dimension space that can provide satisfactory performance without losing information. For this purpose, we found that reducing the original dimension to 200 allowed us to gain effective performance advantages while guaranteeing minimal information lost.

At this point, to identify the topics within the lessons as clusters of the original terms (n-grams), the HDBSCAN clustering algorithm is applied after the dimensionality reduction step. One of the parameters that can influence the algorithm is the minimum number of terms to assign to each cluster. For example, by setting a high value for this parameter, the algorithm will return stronger topics against the residual noise, but at the same time, the number of such topics will be very small, thus possibly merging near concepts together. Anyway, we obtained rather satisfactory results by adopting the default value recommended by the author of BERTopic, i.e., setting the parameter equal to 10. Furthermore, the algorithm is able to extract the keywords associated with each identified topic using the $c-TF-IDF$ algorithm. Such keywords can be used to assess the validity of the obtained results by focusing on their significance in the considered domain. Algorithm 2 refers to the steps described above to obtain our topic-modeling solution.

Algorithm 2 Topic modeling.

Input:
$Fil\_Tr \leftarrow$ [*transcripts without noise*] ▷ Video transcripts without noise
$Sb \leftarrow$ *SBERT pre-trained model*
$U \leftarrow$ *UMAP model*
$Hdbs \leftarrow$ *HDBSCAN model*
Output:
$Topics \leftarrow \{[topic]\}$ ▷ Set of topic labels
$Keywords\_Map \leftarrow \{topic : [keywords]\}$ ▷ keys: topic labels, value: set of keywords
$L\_Tr \leftarrow \{[l\_tr]\}$ ▷ Labeled transcripts with topic labels

for each $fil\_tr \in \mathcal{F}il\_Tr$ **do**
    paragraphs $\leftarrow$ *subdivision of fil_tr in paragraphs of 25 terms*
end for
$encoded\_paragraphs \leftarrow Sb(paragraphs)$
$encoded\_paragraphs \leftarrow U(encoded\_paragraphs)$
$Topics, L\_Tr \leftarrow Hdbs(encoded\_paragraphs)$
$Keywords\_Map \leftarrow c-TF-IDF(Topics\_Map)$
return $L\_Tr, Topics, Topics\_Map, Keywords\_Map$

As stated before, the obtained topics can, hence, be used to enhance the indexing mechanism described in Section 4. Indeed, the original n-grams (actually binomials) are now semantically grouped into thematic clusters, thus allowing for a hierarchical semantic-aware overall indexing strategy.

Table 5 reports an extract of the results obtained by the application of the above strategy on the 21 lessons of the testbed course analyzed in Section 4.

Table 5. The results obtained by the application of our topic modeling strategy to enhance the basic semantic indexing of the proof-of-concept.

Topic ID	Keywords
Topic 0	{dipendenza (dependency), funzionale (functional), dipendenza funzionale (functional dependency), forma (form), determinante (determinant), relazione (relationship), forma normale (normal form)}
Topic 1	{proiezione (projection), espressione (expression, attributo (attribute), sigma, restrizione (constraint), predicare (predicate), relazione (relationship)}
Topic 2	{query, tabella (table), operatore (operator), query query, cartesiano (cartesian), prodotto (product), algebra}
Topic 3	{dbms, applicazione (application), ambiente (environment), cloud, connessione (connection), sistema (system), base dato (data base), rete (network)}
Topic 4	{chiave (key), primario (primary), chiave primario (primary key), chiave esterno (foreign key), chiave chiave (key key), vincolo vincolo (constraint constraint), relazione chiave (key relationship)}
Topic 5	{tabella (table), dominio (domain), lista (list), table (table), tabella tabella (table table), esempio (example), dato (datum)}
Topic 6	{entità (entity), concetto (concept), associazione (association), identificatore (identifier), associativo (associative), associazione logica (logic association, logica (logic)}
Topic 7	{modello (model), concettuale (conceptual), progettazione (design), modello relazionare (relational model), schema, fase (phase), logico (logic)}

6. Automatic Outline Generation

After presenting our strategy to build semantic indexes for a video course, we investigated the possibility of combining them with suitable descriptions to help a user understand the intended meaning of the specific index. Roughly speaking, our indexing solution is hierarchical, with high-level indexes representing general concepts (i.e., topics) and internal ones being the actual n-grams pointing to specific locations in each video lesson composing a course. Figure 4 shows a simple representation of such a hierarchical indexing schema.

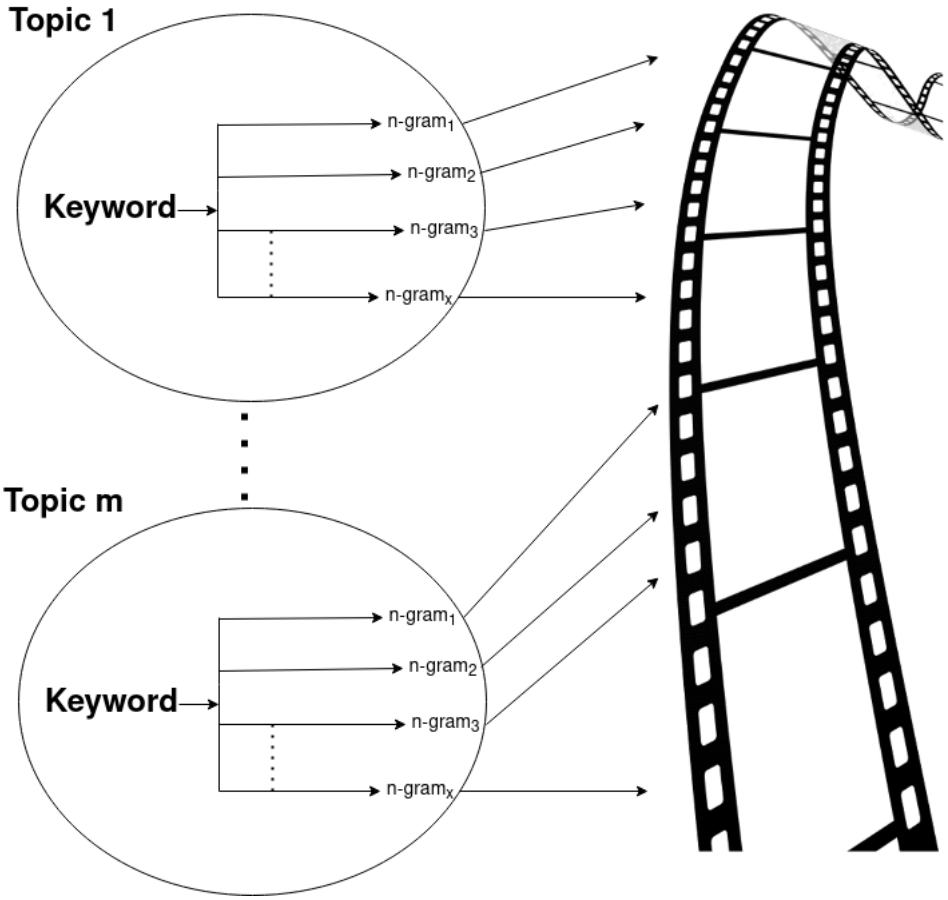

Figure 4. Our semantic hierarchical indexing schema.

From this figure, it is possible to see that, although the external indexes are mapped on specific concepts (or topics), they are still represented by keywords selected from the terms involved in the clusters. As a consequence, to produce a correct "search statement" to filter out the low-level indexes pointing to the parts of a video stream dealing with a target concept, deep knowledge of the underlying domain may still be required. Indeed, such a "search statement" will be mapped by our approach to the high-level semantic indexes, thus showing the users a set of possibly enriching tokens that are linked to the underlying video stream parts. Therefore, the user should be able to acknowledge whether the concept represented by the selected topic is related to his/her research objective. For this reason, as an additional facility, we studied the automatic generation of short descriptions in the natural language associated with each topic extracted by our solution.

To do so, we still adopted a Natural Language Processing technique to build intuitive and condensed descriptions of the concepts represented by the clusters extracted by our topic-modeling strategy.

6.1. Abstractive Summarization

In the related literature, the extraction of condensed descriptions from an article about a concept is referred to as *summarization*.

Summarization is the process of creating a summary of a text. This process can take two forms: *extractive* and *abstractive* [33]. In particular, extractive summarization selects a subset of sentences from the text to form a summary from it. On the other hand, abstractive summarization involves rewriting the entire document by creating an internal semantic representation. After that, it proceeds by creating a summary with Natural Language Processing techniques.

In our approach, we deal with abstractive summarization, and therefore, to build our solution, we started from a "guide text" and, then, generated new textual contents by leveraging the internal semantic representation of the "guide text" as a driver for the text generation. For this objective, we exploited a transformer-based model, namely `T5` [34], and we fine-tuned it to cope with our specific task.

In more detail, the `T5` model, or *Text-To-Text* Transfer Transformer, is a multi-task encoder–decoder model based on a transformer-based architecture, which has been pre-trained on a mixture of unsupervised and supervised text-to-text tasks including translation, question answering, classification, and summarization. Like SBERT, different versions of the `T5` model with different dimensions in terms of parameters are available from the scientific community. In addition, one advantage of this model is that it can be easily fine-tuned to work with texts in different languages and with different linguistic styles.

6.2. Generating Summarizations through External Sources

As explained before, summarization allows for the generation of a (abstractive) condensed description of a concept provided that a possibly long text input discussing the target concept is available. To address this task, we need a suitable source of textual information that deals with the majority of the topics that could be covered in the target course. In this way, the abstractive summarization technique, presented in Section 6.1, can be applied to it by properly selecting the document portions referring to each identified topic. In the following, we refer to this additional data source as the *external source*. In such a scenario, the keywords obtained from the previous steps, along with the n-grams associated with them, can be used combined with a suitable querying mechanism to extract, from the external source, the portion referring to a target topic. The architecture of this part of our solution is depicted in Figure 5.

Of course, there are no limitations on the adopted source of information, which can, hence, be chosen arbitrarily on the basis of the analyzed domain. In our approach, we considered two main use cases: in the first, the complete running text from the original lesson transcripts was used (it is worth pointing out that, in this case, the original transcripts were used in their original form (i.e., before the pre-processing described in Section 4)),

whereas, in the second, an external *oracle*, such as Wikipedia, which is a well-known crowd-sourced encyclopedia, was considered.

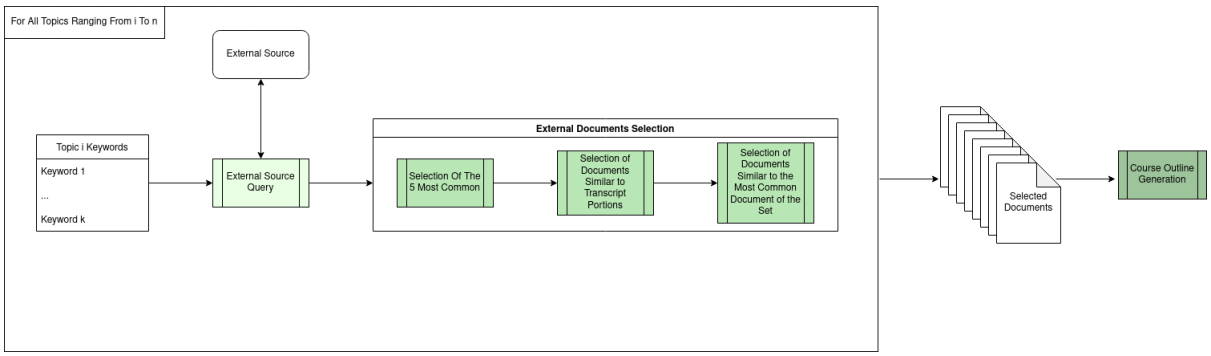

Figure 5. The architecture of our solution for the generation of condensed descriptions for the involved topics.

The idea behind the first use case is that the original transcripts can be ideally split into several parts, each focused on one of the topics derived from our topic-modeling approach. Therefore, such parts can be used as the input source to the abstractive summarization technique to derive a condensed description of the concept underlying the specific topic. To identify the parts of L_C that refer to a target topic t_x, we split each lesson transcript into paragraphs and applied the trained topic modeling algorithm to label them with topics (see Section 5.1). Therefore, given a lesson transcript, $l_i \in L_C$, the application of the trained topic modeling algorithm to it will map a set of topics T_{l_i} to each of its paragraphs. Let $\mathcal{F}(l_i, t_y)$ be a function computing for each topic t_y of T_{l_i} the number of occurrences in the paragraphs of l_i. The set of transcripts TR_{t_x} referring to a target topic t_x can be obtained by the following:

$$TR_{t_x} = \{l_i : t_x \in \arg\max_{t_y \in T_{l_i}} \mathcal{F}(l_i, t_y)\}$$

Finally, TR_{t_x} can be used as an external source to feed the abstractive summarization algorithm introduced above (see Figure 5). Algorithm 3 summarizes the previous steps.

Algorithm 3 Course outline generation (Use Case 1).

Input:
$L\_Tr \leftarrow \{[l\_tr]\}$ ▷ Labeled transcripts with topic labels
$T5 \leftarrow$ *text-to-text generative model*
Output:
$Course\_Outline \leftarrow [generated\_summaries]$ ▷ Generated course outline

1: **for each** $l\_tr \in \mathcal{L}\_Tr$ **do** ▷ Cycle through the labeled transcripts
2: $candidate\_documents \leftarrow$ *paragraphs belonging the most frequent topics in l_tr*
3: $generated\_summaries \leftarrow T5.generate\_text(candidate\_documents)$
4: $Course\_Outline.append(generated\_summaries)$
5: **end for**
6: **return** $Course\_Outline$

As to the second use case, instead, the solution requires the use of an external *oracle*, i.e., a complete source dealing with all the topics of interest in the target domain. Once again, such an *oracle* could be any existing encyclopedia, domain-related compendia, or documents produced by domain experts [35]. Without loss of generality, in our approach, we imposed that it is equipped with a search engine providing a simple means to extract all the articles matching a given query. Specifically, in our case, the query can be formulated by using the keywords returned by our topic modeling solution. More precisely, given the keywords of a single topic, we queried the *oracle* using them and extracted all the most-relevant articles returned by the platform for each topic. More formally, let $ES(Q_{t_x})$ be the

set of pages returned by the *oracle* search engine upon a query Q_{t_x}. We define a query for this search engine on a topic t_x as the set of its keywords $Q_{t_x} = \{kw_z : kw_z \; is \; keyword \; of \; t_x\}$. At this point, as done for the previous use case, the set of pages $ES(Q_{t_x})$ can be used as an external source to feed the abstractive summarization algorithm. The procedure above is sketched in Algorithm 4.

Algorithm 4 Course outline generation (Use Case 2).

Input:
$Topics \leftarrow \{[topic]\}$ ▷ Set of topic labels
$Keywords\_Map \leftarrow \{topic : [keywords]\}$ ▷ keys: topic labels, value: set of keywords
$Ext\_Sou \leftarrow External \; source \; of \; information$
$T5 \leftarrow text\text{-}to\text{-}text \; generative \; model$
Output:
$Course\_Outline \leftarrow [generated\_summaries]$ ▷ Generated course outline

1: **for each** $topic \in \mathcal{T}opics$ **do** ▷ Cycle through the keys of the map
2: $candidate\_documents \leftarrow Ext\_Sou.query(Keywords\_Map[topic])$
3: $generated\_summaries \leftarrow T5.generate\_text(candidate\_documents)$
4: $Course\_Outline.append(generated\_summaries)$
5: **end for**
6: **return** $Course\_Outline$

Finally, we report in Table 6 an example of the results obtained by the application of our approach on the testbed introduced in Section 4.

Table 6. An extract of the results obtained by our approach on the 21 lessons of the considered testbed.

Topic ID	Topic	Italian Summary	English Summary (Translated)
Topic 3	Cloud computing	il cloud computing (in italiano nuvola informatica) indica, in informatica, un paradigma di erogazione di servizi offerti su richiesta da un insieme di risorse preesistenti, configurabili e disponibili in remoto sotto forma di architettura distribuita.	cloud computing (in Italian nuvola informatica) indicates, in IT, a paradigm for the provision of services offered on request by a set of pre-existing resources, configurable and available remotely in the form of a distributed architecture.
Topic 6	Modello E-R (E-R Model)	in informatica, è un modello entità-relazione (o modello entità-relazione; più comune modello E-R) è un modello teorico per la rappresentazione concettuale e grafica dei dati a un alto livello di astrazione.	in computer science, it is an entity-relationship model (or entity-relationship model; more common E-R model) is a theoretical model for the conceptual and graphical representation of data at a high level of abstraction.
Topic 7	Modello concettuale (Conceptual model)	la modellazione o progettazione concettuale è una tecnica molto nota di progettazione dati. Chiarire il significato di vari termini spesso ambigui.	Conceptual modeling or design is a well-known data design technique. Clarify the meaning of various often ambiguous terms.
Topic 2	SQL	in informatica, è un linguaggio standardizzato per database basati sul modello relazionale (RDBMS).	in computer science, it is a standardized language for relational model-based databases (RDBMS).
Topic 2	Algebra relazionale (Relational Algebra)	in informatica l'algebra relazionale e il collegato calcolo relazionale fanno parte dell'insieme di linguaggi che permettono di esaminare le query (interrogazioni) da effettuare nella gestione/utilizzo di un database	in computer science, relational algebra and related relational calculus are part of the set of languages that allow you to examine the queries (interrogations) to be made in the management/use of a database
Topic 4	Chiave primaria (Primary key)	insieme di attributi che fa rifermiento a una chiave primaria o Primary Key.	set of attributes that refers to a primary key or Primary Key.
Topic 4	Chiave esterna (External hey)	In inglese foreign key dei database relazionali, è un vincolo di integrità referenziale tra due o più tabelle	in English foreign key of relational databases, is a referential integrity constraint between two or more tables.
Topic 4	Vincolo di integrità (Integrity constraint)	in un database relazionale, richiede che ogni valore di un altro attributo (colonna) di una relazione esista come valore di un altro attributo (colonna) di una relazione	in a relational database, requires that every value of another attribute (column) of a relationship exists as a value of another attribute (column) of a relationship
Topic 1	Normalizzazione (Normalization)	in informatica la normalizzazione è un procedimento volto all'eliminazione della ridondanza informativa e del rischio di incoerenza del database.	in computer science, normalization is a procedure aimed at eliminating information of redundancy and the risk database inconsistency.

7. Implementation and Results

This section is devoted to the description of the experiments that we carried out to validate our proposal. For these experiments, we considered a pool of courses available online for which a manually written course syllabus was available. The considered courses concerned two main domains, namely: Theory of Computation and Databases. For each domain, we split the set of courses into two parts, one for the training of our approach and

the other to test its capability to generally work as a support tool on the domains for which it was trained. To measure the performance of our solution, we adopted a strategy based on the support of human user experts of the considered domains. In particular, we designed a quality evaluation questionnaire that was administered to evaluators to estimate the perceived quality of the obtained indexes and summaries.

In the next sections, we describe some technical details of the experimental campaign concerning how we configured our solution when focusing on both cases introduced in Section 6.2. After that, we report the details of the questionnaire, as well as analyze the obtained results.

7.1. Use Case 1: Running Text from Video Transcripts

As presented in Section 6.2, for the first use case, a summary of a course is obtained using the original transcripts as a support source. Practically speaking, we used topic modeling to label each paragraph in the video transcripts. After that, we identified the most-dominant topics in each lesson. At this point, for each topic identified in the semantic indexing phase (see Section 5.1), we used the transcripts of the lessons, discussing that topic as a main objective, to generate its abstractive summarization.

As stated above, the main topic/s is/are identified by considering the distribution of topics for the single transcript. From a technical point of view, we selected as representative topics those that involve more than 50% of the text.

It is worth underlining that it is not always possible to identify a single main topic, so, in general, a transcript may actually involve a number of topics (those that altogether cover more than 50% of the involved text). Finally, given a topic, all the lessons including it (as one of the main topics) will be concatenated and considered as a source for the trained abstractive summarization model.

7.2. Use Case 2: Wikipedia as External Source

The second use case, instead, exploits an external oracle to build text sources for the summarizations. In our experiments, without loss of generality, we focused on Wikipedia as an underlying oracle. Indeed, this crowd-sourced online encyclopedia offers a rich set of application programming interfaces [36], called MediaWiki APIs [37], allowing fast access to all the articles in its collection, through a consolidated search engine [38].

From a technical point of view, starting again from the topics identified by our semantic indexing approach, we queried Wikipedia and extracted all the articles related to these topics through the MediaWiki APIs. Specifically, for each topic, we built a query for the APIs using the keywords associated with it.

As per the API functionality, an article can satisfy one or more of the keywords included in the query. Therefore, we pre-processed the list of obtained articles by focusing on their appearance frequency in the result list. Specifically, we preserved only the most-frequent articles in this list following the idea that the higher the frequency of an article in the result list, the higher the probability that it is a good representative of the target topic.

However, from the tests we carried out, we noticed that some of the keywords at this stage could still involve unrelated articles. For this reason, in our experiment with Wikipedia, we adopted two-step filtering.

First of all, for each article of the most-frequent sub-list of the results, we extracted only nouns and adjective lemmas. After that, using Sentence-BERT to extract embeddings and the cosine similarity defined in Equation (1) (see also Section 5.1 for further details on the text processing), we computed a global text *similarity* between each Wikipedia article from the list above and the transcripts of the original video lessons labeled with the target topic. Therefore, we proceeded by removing all the articles having a global similarity with the reference transcripts less than 0.5.

As for the second filtering step, this additional procedure was intended to filter out too distant articles (that could represent noise). Therefore, we considered the most-frequent article, in the most-frequent sub-list above, as a "main article", and we computed the

similarity between this and all the other articles in the list. Once again, we removed those having a similarity lower than 0.5 with respect to the "main article".

The above procedure was then repeated for all the considered topics in order to obtain the set of external sources for the summarization algorithm.

7.3. Results

As introduced above, to validate the performance of our proposal, we selected different academic online courses belonging to the two target domains. All the courses considered for testing are equipped with the respective course syllabi. It is important to underline that all the video courses collected were recordings of the corresponding in-person lessons held by the teachers.

At this point, we split the collected video courses into two sets: one for training and the other for testing. Hence, after training our solution as described in the previous sections, we applied it to the transcripts of each testing course for each domain, and we used the real original syllabi to validate the obtained results. For each course, we applied both the strategy detailed in Section 7.1 and the one in Section 7.2.

As a preliminary manual evaluation, we observed a very low general performance when the strategy of the first use case (Section 7.1) was adopted. From the obtained results, we concluded that such a strategy is not suitable for our objective, because the use of the direct running text from the transcripts is undermined by the difficulty to distinguish between parts of the lessons in which the topic is explained by the teacher and the interactive parts in which the teacher focus is engaging the students and verifying their understanding level. Moreover, we noticed that, since the video courses that we gathered were collections of recordings of in-person lessons, concepts are not always expressed exclusively through their formal definition, but also through rich examples and informal explanations. This inevitably leads to the inclusion of informal and colloquial terms or sentences related to interactions between teachers and students.

By contrast, the second approach (see the use case described in Section 7.2) produced far more promising results during the preliminary manual evaluation.

Therefore, we proceeded by performing a deeper performance analysis by formulating an evaluation questionnaire to be administered to a set of expert reviewers for the considered domains. The questionnaire was based on an evaluation grid designed to assess both the adequateness of the semantic indexing through topic modeling and the quality of the generated abstractive summarizations for such topics. The underlying evaluation grid is reported in Table 7.

Table 7. The evaluation grid adopted in the performance analysis questionnaire.

Score	Relevance of the Topic	Quality of text Generation	General Evaluation
1	Unrelated topic	Incomprehensible description	Unusable
2	Topic belonging to the field of study but unrelated to the topic of the lesson (E.g.: Topic belonging to computer science but not related to databases)	It presents numerous errors	Many important topics missing
3	Topic inherent to the domain of the course but not present among the topics of the course	It presents some errors but understandable meaning	Some missing topics
4	Correct topic	Description formulated correctly	Perfectly Usable

The received answers were, hence, processed by removing situations in which the reviewers were not in agreement. Reviewers' agreement was estimated using the Fleiss' Kappa index [39,40]. The valid results were, then, averaged and normalized between 0 and 1, and we report them in Table 8.

Table 8. Averaged scores per category.

Relevance of the Topic	Quality of Text Generation	General Evaluation
0.91	0.80	0.87

By analyzing this table, we can see that our approach was capable of reaching a score close to the maximum value of 1 both for the relevance of the identified topics and for the quality of the summarizations. This confirms that our strategy can be a valid tool to build semantic indexing of online video courses. Moreover, the high-quality descriptions that our approach could generate for the identified topics can be a key component to enable a greater awareness of the semantics behind our indexing solution.

8. Conclusions

In this paper, we proposed an approach to build semantic hierarchical indexes to improve the use of online audio and video materials. Indeed, in recent years, a huge quantity of such multimedia content has been produced by academics, teaching institutions, and domain experts. However, due also to the advent of the recent pandemic, such a production has reached an incredibly high level, leading to the construction of massive repositories of these contents. However, this material risks not being fully profitable due to the intrinsic difficulty to identify interesting parts inside them. In this paper, we tackled this problem and focused on the academic domain as a main reference scenario. In this context, video lessons have become key learning tools for students for several years; however, during the aforementioned pandemic, such an opportunity has become crucial to guaranteeing academic activities all around the world. In general, although in this reference environment course video lessons are accompanied by syllabi describing their content in a very concise way, sequential access is still the only option to identify the searched parts. Moreover, syllabi have variable structures and offer very different detail levels, spanning from thorough in-depth descriptions to extremely high-level ones. Such heterogeneity cannot guarantee consistent support to identify all the concepts included in the lessons of a course. For this reason, we proposed a solution facing the problem of limiting sequential access to video content and providing an automatic tool to generate high-quality descriptions of all the concepts included therein. To address the first requirement, we developed a semantic hierarchical indexing strategy using a combination of text mining and Natural Language Processing techniques. Such an indexing strategy allows the identification of general concepts, or topics, in the media source and associates with each of them tokens (i.e., n-grams) that are directly mapped to specific points in the video/audio streams. Moreover, through the use of a deep-learning-based approach, we addressed the second requirement by building an abstractive summarization solution and tested the quality of our proposal through two use cases. The former leverages the processed textual transcripts of the original video lessons, and the latter relies on an external *oracle*. As for the external oracle, although any existing domain-related compendia can be used, we built an implementation of our solution by using Wikipedia. The obtained results confirmed the suitability of our proposal; indeed, both the identified topics, along with their correct mapping with the original video/audio stream portions, and the produced textual summarizations appeared adequate and had a satisfactory quality level.

The research described in this paper must not be considered conclusive; indeed, several other research activities can be carried out in this context. For instance, we are planning to extend our approach by working on summarization models capable of producing not only textual descriptions, but also video or audio summaries, by suitably cutting and processing the most-adequate parts from the original streams identified by our indexing solution. Moreover, it could be possible to investigate our approach to media sources coming from environments other than academic ones, such as crowd-sourced video/audio streams from online sharing platforms.

Author Contributions: Conceptualization, A.N. and M.F.; methodology, A.N., M.A. and M.F.; software, M.A.; validation, A.N., M.A. and M.F.; investigation, A.N., M.A. and M.F.; data curation, A.N., M.A. and M.F.; writing—original draft preparation, A.N., M.A. and M.F.; writing—review and editing, A.N., M.A. and M.F.; visualization, A.N. and M.A.; supervision, A.N. and M.F.; project administration, A.N. and M.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data available on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, H.; Meinel, C. Content based lecture video retrieval using speech and video text information. *IEEE Trans. Learn. Technol.* **2014**, *7*, 142–154. [CrossRef]
2. Van Nguyen, N.; Coustaty, M.; Ogier, J.M. Multi-modal and cross-modal for lecture videos retrieval. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 2667–2672.
3. Alberti, C.; Bacchiani, M.; Bezman, A.; Chelba, C.; Drofa, A.; Liao, H.; Moreno, P.; Power, T.; Sahuguet, A.; Shugrina, M.; et al. An audio indexing system for election video material. In Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009; pp. 4873–4876.
4. Ogata, J.; Goto, M. PodCastle: A spoken document retrieval system for podcasts and its performance improvement by anonymous user contributions. In Proceedings of the Third Workshop on Searching Spontaneous Conversational Speech, Beijing, China, 23 October 2009; pp. 37–38.
5. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP 2019), Association for Computational Linguistics, Hong Kong, 3–7 November 2019.
6. Kong, S.Y.; Wu, M.R.; Lin, C.K.; Fu, Y.S.; Chung, Y.Y.; Huang, Y.; Chen, Y.N.; Shan Lee, L. NTU Virtual Instructor—A Spoken Language System Offering Services of Learning on Demand Using Video/Audio/Slides of Course Lectures. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009.
7. Chelba, C.; Hazen, T.J.; Saraclar, M. Retrieval and browsing of spoken content. *IEEE Signal Process. Mag.* **2008**, *25*, 39–49. [CrossRef]
8. Garnier-Rizet, M.; Adda, G.; Cailliau, F.; Gauvain, J.L.; Guillemin-Lanne, S.; Lamel, L.; Vanni, S.; Waast-Richard, C.; et al. CallSurf: Automatic Transcription, Indexing and Structuration of Call Center Conversational Speech for Knowledge Extraction and Query by Content. In Proceedings of the LREC, Marrakech, Morocco, 26 May–1 June 2008.
9. Makhoul, J.; Kubala, F.; Leek, T.; Liu, D.; Nguyen, L.; Schwartz, R.; Srivastava, A. Speech and language technologies for audio indexing and retrieval. *Proc. IEEE* **2000**, *88*, 1338–1353. [CrossRef]
10. Hao, X.; Zhang, W.; Wu, D.; Zhu, F.; Li, B. Listen and Look: Multi-Modal Aggregation and Co-Attention Network for Video-Audio Retrieval. In Proceedings of the 2022 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 18–22 July 2022; pp. 1–6.
11. Avgoustinakis, P.; Kordopatis-Zilos, G.; Papadopoulos, S.; Symeonidis, A.L.; Kompatsiaris, I. Audio-based near-duplicate video retrieval with audio similarity learning. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 5828–5835.
12. Salatino, A.A.; Osborne, F.; Birukou, A.; Motta, E. Improving editorial workflow and metadata quality at springer nature. In Proceedings of the Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, 26–30 October 2019; Proceedings, Part II 18; Springer: Berlin/Heidelberg, Germany, 2019; pp. 507–525.
13. Chen, X.; Zou, D.; Cheng, G.; Xie, H. Detecting latent topics and trends in educational technologies over four decades using structural topic modeling: A retrospective of all volumes of Computers & Education. *Comput. Educ.* **2020**, *151*, 103855.
14. Srinivas, S.; Rajendran, S. Topic-based knowledge mining of online student reviews for strategic planning in universities. *Comput. Ind. Eng.* **2019**, *128*, 974–984. [CrossRef]
15. Malaterre, C.; Pulizzotto, D.; Lareau, F. Revisiting three decades of Biology and Philosophy: A computational topic-modeling perspective. *Biol. Philos.* **2020**, *35*, 1–25. [CrossRef]
16. Kulkarni, K.; Padaki, R. Video Based Transcript Summarizer for Online Courses using Natural Language Processing. In Proceedings of the 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Online, 16–18 December 2021; pp. 1–5.
17. Saini, M.; Arora, V.; Singh, M.; Singh, J.; Adebayo, S.O. Artificial intelligence inspired multilanguage framework for note-taking and qualitative content-based analysis of lectures. *Educ. Inf. Technol.* **2023**, *28*, 1141–1163. [CrossRef] [PubMed]
18. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:cs.CL/1301.3781
19. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. *arXiv* **2013**, arXiv:cs.CL/1310.4546

20. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global Vectors for Word Representation. In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
21. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
22. Grootendorst, M. BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics. *Zenodo* **2020** . [CrossRef]
23. Arazzi, M.; Nicolazzo, S.; Nocera, A.; Zippo, M. The importance of the language for the evolution of online communities: An analysis based on Twitter and Reddit. *Expert Syst. Appl.* **2023**, *222*, 119847. [CrossRef]
24. Šćepanović, S.; Constantinides, M.; Quercia, D.; Kim, S. Quantifying the impact of positive stress on companies from online employee reviews. *Sci. Rep.* **2023**, *13*, 1603. [CrossRef] [PubMed]
25. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR arXiv* **2018**, arXiv:abs/1810.04805
26. McInnes, L.; Healy, J.; Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv* **2020**, arXiv:stat.ML/1802.03426
27. Campello, R.J.; Moulavi, D.; Sander, J. Density-based clustering based on hierarchical density estimates. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Gold Coast, QLD, Australia, 14–17 April 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 160–172.
28. Cloud Speech-to-Text. Available online: https://cloud.google.com/speech-to-text (accessed on 31 March 2023).
29. Amazon Transcribe. Available online: https://aws.amazon.com/it/transcribe/ (accessed on 31 March 2023).
30. Natural Language Toolkit. Available online: https://www.nltk.org/ (accessed on 31 March 2023).
31. TINT—The Italian Nlp Tool. Available online: https://dh.fbk.eu/research/tint/ (accessed on 31 March 2023).
32. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.J.; McClosky, D. The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of the Association for Computational Linguistics (ACL) System Demonstrations, Baltimore, MD, USA, 23–24 June 2014; pp. 55–60.
33. Liu, Y.; Lapata, M. Text summarization with pretrained encoders. *arXiv* **2019**, arXiv:1908.08345
34. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv* **2019**, arXiv:1910.10683
35. Nocera, A.; Ursino, D. PHIS: A system for scouting potential hubs and for favoring their "growth" in a Social Internetworking Scenario. *Knowl.-Based Syst.* **2012**, *36*, 288–299. [CrossRef]
36. Buccafurri, F.; Lax, G.; Nicolazzo, S.; Nocera, A. A model to support multi-social-network applications. In Proceedings of the On the Move to Meaningful Internet Systems: OTM 2014 Conferences: Confederated International Conferences: CoopIS, and ODBASE 2014, Amantea, Italy, 27–31 October 2014; Proceedings; Springer: Berlin/Heidelberg, Germany, 2014; pp. 639–656.
37. MediaWiki Action API. Available online: https://www.mediawiki.org/wiki/API:Main_page (accessed on 31 March 2023).
38. The Wikipedia Search Engine. Available online: https://en.wikipedia.org/wiki/Help:Searching#Under_the_hood (accessed on 31 March 2023).
39. Uebersax, J.S. A generalized kappa coefficient. *Educ. Psychol. Meas.* **1982**, *42*, 181–183. [CrossRef]
40. Quattrone, G.; Nicolazzo, S.; Nocera, A.; Quercia, D.; Capra, L. Is the sharing economy about sharing at all? A linguistic analysis of Airbnb reviews. In Proceedings of the International AAAI Conference on Web and Social Media, Palo Alto, CA, USA, 25–28 June 2018; Volume 12.