Computing Pointers Into Instructional Videos

[Extended Abstract] *

Andrew Lamb
Stanford University
andrew.lamb@stanford.edu

Jose Hernandez
Stanford University
josehdz@stanford.edu

Jeffrey Ullman
Stanford University
ullman@cs.stanford.edu

Andreas Paepcke
Stanford University
paepcke@cs.stanford.edu

ABSTRACT

We examine algorithms for creating indexes into ordered series of instructional lecture video transcripts. The goal is for students and industry practitioners to use the indexes towards review or reference. Lecture videos differ from often-examined document collections such as newspaper articles in that the transcript ordering generally reflects pedagogical intent. One challenge is therefore to identify where a concept is *primarily* introduced, and where the resulting index should thus direct students. The typically applied TF-IDF approach gets tricked in this context by artifacts such as worked examples whose associated vocabulary may dominate a lecture, but should not be included in a good index. We contrast the TF-IDF approach with algorithms that consult Wikipedia documents to vouch for term importance. This method helps filter the harmful artifacts. We measure the algorithms against three human-created indexes over the 90 lecture videos of a popular database course. We found that *(i)* humans have low inter-rater reliability, whether they are experts in the field or not, and that *(ii)* one of the examined algorithms approaches the inter-rater reliability with humans.

1. INTRODUCTION

We present comparisons of algorithms that can be applied to extract keywords from instructional videos from massively open online courses (MOOCs) in order to construct an index. We use as our raw material the closed caption files that are often available for educational video. Those files contain transcripts of the videoed instructor's words, paired with timing information at roughly sentence granularity.

Without a natural ground truth for our indexes, we evaluated our algorithms by comparing against decisions made by humans. We paid three humans to carefully index the video transcripts from a Stanford online database course. We examined how well the three resulting indexes compared to each other, and how outcomes of several algorithms compared to each of the human-generated results. We make the three reference indexes and the database course video caption files available to the public in hope of eliciting indexing approaches beyond those that we explored.

*A full version of this paper is available at `http://ilpubs.stanford.edu:8090/1140/1/indexer.pdf`

Our first experiment took a traditional approach, selecting words for the index that appeared disproportionately often in certain lectures. We then incorporated lexical information, by only considering phrases that followed certain part-of-speech patterns. Finally, we introduced external knowledge from Wikipedia into an algorithm's indexing decisions. Note that none of the algorithms included supervised learning, as we do not assume the existence of a training set for all courses.

2. EXPERIMENTS

We implemented several index term extraction algorithms and measured how closely they agreed with the gold index derived from the work of our human indexers. The following subsections introduce the algorithm (families) we applied to the lecture transcripts.

2.1 Traditional Approach: TF-IDF

Our simplest algorithm used a straight term frequency-inverse document frequency (TF-IDF) approach to identifying index terms in a lecture. TF-IDF is defined for each phrase-lecture pair as the product of the number of times the phrase appears in the lecture, divided by the logarithm of the proportion of lectures in which the phrase appears. Any phrases above a chosen threshold were marked as *in-index*. All phrases lower in the list were marked *not-in-index*. We chose the average number of keywords that the human indexers included in their indexes as the threshold value. We limited the algorithms to a maximum phrase length of four.

2.2 Leveraging Linguistic Information

We measured an algorithm that first runs a part-of-speech tagger over the lecture transcripts, and then selected only phrases that consist of a number of adjectives followed by one or more nouns. For example, "equality condition" or "XML data" were both included in the candidate set. We then ran TF-IDF over this reduced set.

2.3 Adding External Knowledge

Motivated by the intuition that phrases gain importance because of both their role in a document and their semantic meaning in the broader world, we experimented with multiple algorithms that incorporate outside knowledge. Each algorithm integrates Wikipedia as a knowledge source in different ways.

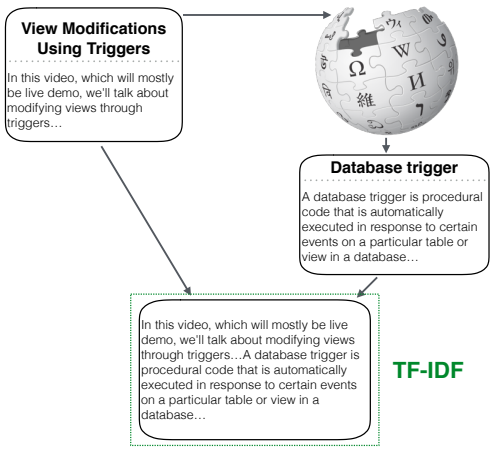

Figure 1: The Document Boosting algorithm searches for a Wikipedia page using the title of the lecture, concatenates the result to the lecture, and then runs TF-IDF over the combined document.

2.3.1 *Boosting Documents*

The first algorithm concatenates to each lecture a closely related Wikipedia page, and then uses the previously described statistical and linguistic techniques to choose phrases from the combined document. Formally, the procedure is as follows.

For example, for the lecture titled "View Modifications Using Triggers", the first page in the Wikipedia search results is "Database trigger", which is then concatenated to the transcript of the lecture. Then, using either n-grams or adjective-noun phrases as candidate keywords, the algorithm chooses phrases with TF-IDF over the combined document for the index.

2.3.2 *Boosting Phrases*

This algorithm first creates a list of candidate index terms using adjective-noun phrases. Then the candidates are ranked by their TF-IDF score summed over all Wikipedia documents.

Next, this global candidate ranking is combined with the basic TF-IDF approach described in Section 2.1 to form a final score that combines global knowledge (from Wikipedia) with local knowledge (from the specific lecture video).

We also experimented with only boosting phrases of at least two words, based on the intution that longer phrases are often meaningful, but appear infrequently and are therefore given low scores by TF-IDF. We call this alternative "Phrase Boosting N-Grams" in Figure 2.

2.4 Results

We evaluated each algorithm by computing Cohen's Kappa agreement between the algorithm and the gold set unified from two (one of the human indexes was not included in evaluation because they sometimes used words that did not appear in the lecture) of the human indexes as described in Section **??**. The intuition is a measure of inter-rater reliability, such as the Kappa, measures how closely the algorithmic index agrees with the human indexes.

Kappa values do not have a universally agreed upon interpretation, but values in the range we observe (about 0.15 to 0.3) have been interpreted as indicating "slight" to "fair" agreement. We measured an agreement of 0.325 between humans in the gold index, suggesting that the phrase extraction task is inherently subjective, and there are multiple valid interpretations of what phrases are important enough to be included in the index.

The metrics for all of the algorithms are shown in Figure 2. The Phrase Boosting N-Grams algorithm, which favors longer words, performed the best out of all algorithms, and had a Cohen's Kappa of 0.237 agreement with the gold index. This nears the lowest pairwise agreement between humans of 0.309, showing that the algorithm is close to the performance of a human.Document Boosting with adjective-noun chunks seems to yield significant improvement. Document Boosting and Phrase Boosting, the two algorithms that incorporated external knowledge, were able to make improvements on the basic algorithm. Document Boosting, which appended a Wikipedia document to each lecture, was able to improve over TF-IDF, and boosting longer phrases (Phrase Boosting N-Grams) was able to improve further.

To give a more subjective view of our results, we also show the set of keywords extracted from a lecture on 'Materialized Views' by the Phrase Boosting with N-grams algorithm, in Figure 3 . Of the top 15 keywords marked by the algorithm, 11 were included in the gold index marked by humans (for this lecture there were 18 keywords in the gold set), and the algorithm produces a ranking that is similar to the humans. Of the keywords ranked highly by the algorithm that were not in the gold index, some ('materialized', 'insert command', 'multivalued dependency') are relevant to the course, but perhaps not essential to the specific lecture. The last two keywords, 'user' and 'user query' expose a weakness of the algorithm, where it is difficult to discern phrases that are used frequently, but not essential to the lecture concept.

3. CONCLUSION

We have started to tackle the task of choosing the most important phrases from a collection of lectures, to construct a random-access index analogous to those in the back of books. Going forward we will use this capability to construct student support facilities such as automatically answering learner questions with references to relevant lecture clips, and recommendation tasks, such as finding the best study materials given a student's progress through a course. There has been little previous work on index extraction in the online education setting, and in lecture series videos in particular. After evaluating the weaknesses of TF-IDF in this educational context, we designed algorithms that incorporated linguistic information, in the form of part-of-speech tags and chunking, and external information, with the entire Wikipedia document collection used as a knowledge source. The algorithms that incorporate Wikipedia information boost performance of TF-IDF, especially on longer phrases that do not have high raw frequencies in a lecture.

4. REFERENCES

Algorithm	κ	$\mathbf{P}_{\text{pos}}$	$\mathbf{P}_{\text{neg}}$	**PABAK**
TF-IDF	0.205	0.233	0.971	0.889
TF-IDF with Adjective-Noun Chunks	0.079	0.118	0.961	0.850
Document Boosting	0.209	0.234	0.973	0.895
Document Boosting with Adjective-Noun Chunks	0.142	0.173	0.968	0.876
Phrase Boosting	0.204	0.234	0.970	0.883
Phrase Boosting N-Grams	**0.237**	**0.262**	**0.974**	**0.899**

Figure 2

Rank	Phrase
1	**view**
2	**materialized view**
3	materialized
4	**query**
5	**view query**
6	**virtual view**
7	**modify**
8	user query
9	**base table**
10	**modify command**
11	**index**
12	insert command
13	multivalued dependency
14	**database design**
15	user

Figure 3: The top 15 keywords from 'Materialized Views' by Phrase Boosting with N-grams. Phrases that also appear in the gold index are marked in bold.