Portrait of an Indexer—Computing Pointers Into Instructional Videos

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

As enrollment in online courses grows, it is becoming increasingly difficult for course staff to support students on an individual basis. We are building an autonomous system that will aid instructors and learners in online education; for example, the system might link a forum post with a relevant section of lecture if it detects the poster is confused, or suggest review materials based on a student's history. As a foundation of this system we developed an indexing module that will detect phrases relevant to the key concepts of each lecture, a task often termed keyword extraction. Most research on keyword extraction has focused on heterogenous corpuses, such as abstracts from scientific journals, while our system operates on lecture videos, which are relatively homogenous. To deal with this new setting, we designed algorithms that incorporate external knowledge, using the entire Wikipedia document collection. To evaluate the keyword extraction module, we annotated an online databases course with keywords chosen by humans.

1. INTRODUCTION

Offerings of massively open online courses (MOOCs) have been expanding over the past year. Companies are betting their existence on the continuation of this trend. Universities are experimenting to find their own approaches, either to teaching open classes, or to develop courses directed to their enrolled students. Thus, while the embrace of the 'massive,' and 'open' portions of MOOCs might vary, the 'online' aspect as a tool for teaching is gaining ground, and we focus on this aspect here. Best practices for the use of the Internet to teach are still evolving, and not all voices are enthusiastic [7]. Nonetheless, given the trend it is essential to develop technologies that take maximum advantage of the online medium.

Current course offerings expose many opportunities for such technologies. Peer assessment, forum use, guided tutoring, and interventions that address dropout rates all offer such possibilities for technological improvement [20, 2, 5, 1, 10, 26].

We focus here on opportunities arising when students need to review course material before approaching assessments. Course reviews are historically offered by instructors to peers of students. In online settings, however, students may not arrive at courses on traditional term boundaries; so students' review time lines are not aligned as they are among a fixed group of peers. In the extreme, *untended* courses may not have any active teaching staff. Without support, students fend for themselves when they don't understand a concept or need to review for a test.

One important source for students to review in today's online courses are lecture videos. Typical courses include 100 or more such presentations. While the sequential nature of video may make them suited for structured, linear pedagogy during concept introductions, their sequentiality is clumsy when videos are used as reference material during course review activities. For those occasions random access as afforded by the traditional index at the end of books is much more appropriate. We are not proposing a student-facing *interface* that mimics book indexes; we are rather referring to the capabilities of book indexes—however they can best manifest in online teaching.

The problem is that human indexing is very expensive. We therefore present here comparisons of algorithms that can be applied towards the automatic production of indexes into instructional videos. 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.

The challenge in effective indexing is that keywords must reference the important elements of concepts, and must furthermore direct students to the video segments that contain the *primary* treatments of those concepts. Competent indexing into instructional videos can serve as the foundation to a number of higher-level facilities for students. In [1] we discussed how answers to forum post questions might be approached using video indexes. Other opportunities include automated advice for review when students struggle with particular assignments, and facilities that make courses suitable as reference resources for professionals after they complete a course.

Many keyword extraction systems are designed for use on large collections of loosely related documents, such as newspaper articles [24], or are directed at summarizing or indexing individual documents [18].

In contrast, the algorithms we study here are confronted with a series of documents (i.e. video transcript files) that introduce a number of related concepts in pedagogically thought-through order. Many keyword extraction algorithms leverage the fact that documents on very different topics will have mostly disjoint sets of words, which may not work as well in our setting, where very few authors (i.e. instructors) produce all the documents.

Evaluation of an algorithm's success in building a 'good' index is particularly difficult because indexing from free text is a highly subjective process. We do not in this work consider pre-defined keyword sets from which an indexer would choose when indexing a document. Instead, the task we set for our algorithms is freely to choose words from the text that should be included in the index, subject possibly to a stemming process. The task thus holds many degrees of freedom that allow for a multitude of outcomes.

Given this lack of a natural ground truth, we decided to evaluate outcomes for our algorithms by comparing against decisions made by humans. We examined how well the three human indexes compared to each other, and how outcomes of several algorithms compared to each participant. 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.

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.

In Section 2 we review some of the related literature. Section 3 offers more detail on how we created our three human-generated reference indexes. Section 4 introduces the algorithms we explored. Section 5 offers some observations around the experimental results, and we conclude in a final section.

2. RELATED WORK

Much of the research on keyword extraction has focused on leveraging statistical properties of a corpus or single document. Term frequency-inverse document frequency (TF-IDF) is a widely used method that weights phrases in a document proportionally to the number of times they appear in that document, and inversely proportionally to the number of documents in which the phrases appear at least once [14]. Statistical methods that work on a single document have also been proposed, such as selecting keywords that co-occur in sentences with frequent words unevenly [15]. The TextRank system uses sentence co-occurrence data in a graph-based approach, by forming a node for each candidate phrase and an edge between two phrases if they appear together in a sentence. The system then runs PageRank over the induced graph to select keywords [16]. An approach that uses graphs for topic clustering was presented in [18]. Other methods have combined linguistic and statistical properties of the document by first filtering the set of possible keywords (e.g. only considering noun phrases or adjectives) and then applying frequency based methods [22].

The approaches outlined so far all take an unsupervised approach to keyword extraction. There has also been work on supervised approaches, which classify each phrase by whether it should be a keyword or not. Turney trained a decision tree on different annotated corpora to choose keywords based on features such as the length of the phrase and whether it is a proper noun [25]. Hulth takes a related approach, and finds that adding part-of-speech tags as a feature leads to improved results [12]. The major downside of supervised learning is the expense associated with labeling data, as many types of documents will require a person with domain knowledge to choose keywords. As a result many of these supervised systems use academic journals with author-provided keywords as datasets. Supervised approaches are not an option in our use case.

Discerning important terms in the specific setting of instructional videos has also been investigated. This task differs from keyword extraction from a large heterogeneous document collection because of the sequential nature of the videos, the fact that they will be about closely related topics, and the additional audio-visual component. Methods have been designed to combine the statistical properties of the lecture transcripts with cues from the lecture videos, such as the introduction of a new speaker, and evaluated on governmental instruction videos [19]. This work differs from the work presented here along several dimensions. The human raters were constrained to use keywords that were pre-selected by the authors' baseline keyword extraction algorithm. Furthermore, rather than introducing Wikipedia for extraction support the cited work relies on video features. The authors report percent-agreement between their algorithm and the gold set, which can be misleading compared to indicators such as Cohen's Kappa. Nonetheless [19] is very much in the spirit of our investigation.

Keyword extraction from Khan Academy lecture videos has been experimented with, using statistical properties of the lecture transcripts [13].

Finally, there has been research into using Wikipedia's external knowledge for natural language processing tasks such as clustering documents [11] and computing semantic similarity between words [17].

3. PREPARATION OF GOLD INDEX

In order to evaluate our algorithms, we prepared a gold standard index of terms extracted from an introductory databases course by three paid human indexers. Each was presented with all 90 course closed caption video transcripts, and were asked to work through each file in the order the videos were presented in class. Videos were usually around 10 minutes long.

From each line in a file the indexers were asked to select as many keywords and phrases as they felt should appear in their index. Indexers were only allowed to use words or phrases that appeared in the text. They were asked not to generate their own original phrases. We used indexers' selected keywords and phrases in our investigation[1]. Additionally we asked indexers for two more pieces of information that we did not use, but which are included in our public copies of the indexers' results.

First, we asked indexers also to mark lines in which a particular phrase for the index appeared in the context of the primary introduction to the phrase's underlying concept. Second, for each lecture file, indexers ranked the top five most important phrases from that lecture. We allowed inclusion of fewer than five phrases.

One participant had taken the database course being indexed. A second indexer had taken at least one database course in another institution, and the third was a college-educated individual in a non-technical field.

On average indexers selected about 8 phrases per video. We evaluated agreement between the indexers' output using Fleiss' Kappa, which generalizes Cohen's Kappa to settings with more than two annotators [9]. The κ is a measure of inter-rater agreement, with range $[-1, 1]$, where 1 means the raters are in complete agreement, -1 complete disagreement, and 0 the agreement expected by chance.

For our algorithms we formulate the indexing problem as a binary classification task, where the two categories are *in-index* and *not-in-index*. The algorithms' task was to classify every phrase in all lectures into these two buckets.

Given a set of raters and a collection of lectures $\ell_1, \ldots, \ell_n$, where each lecture contains phrases $p_1, \ldots, p_n \in \ell_i$, we define the set of examples as the set of lecture-phrase pairs (ℓ, p) such that p appears in ℓ, and p is either a 1-gram or (ℓ, p) was tagged as an *in-index* phrase by at least one rater.

When not comparing against each of the three indexers individually, we combined the work of all three into a single index by computing the lecture-by-lecture three-way union. Several other approaches for combining the three ground truth examples can be formulated. We selected the union because it was the most licentious for the algorithms. However, when comparing algorithm results to individual indexers the respective indexer's actual work was used.

We computed a κ of 0.325 between the indexes produced by the three human raters. This κ is evidence of significant differences between the decisions of the three indexers. The non-expert was much more prolific in choosing *in-index* phrases than the two experts. But even the two experts frequently made different choices.

We consider this κ the upper bound on the performance we could reasonably expect from any indexing algorithm. The largest pairwise Cohen's κ between raters was 0.336 and the lowest was 0.309 (note that Fleiss' Kappa is not generally the average of the pairwise Cohen's Kappas).

4. EXPERIMENTS

We implemented different keyword extraction algorithms and measured how closely they agreed with the gold index derived from the work of our human indexers.

For each experiment we applied the Porter stemming algorithm to each word in the document and each word in phrases destined for the index. Phrases therefore matched if all of the individual stemmed tokens matched. In experiments using n-grams as candidate phrases, stopwords were removed from the document before the n-grams were formed, using the stopword list of the SMART system [23].

The following subsections introduce the algorithm (families) we applied to the lecture transcripts.

4.1 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. That is, we used the standard definition of TF-IDF, with each lecture taken as a document. We then ranked the phrases for each document by their TF-IDF score in that document. In the simplest case we used the average number of keywords per lecture as evidenced in the humans' indexes as the threshold.

4.2 Leveraging Linguistic Information

Considering all of the n-grams in the collection of lectures as candidate keywords has the potential of adding significant noise. By selecting only certain linguistic patterns for consideration as phrases, it is possible to reduce the size of the candidate set, while still covering most important phrases. We experimented with an algorithm that first runs a part-of-speech tagger over the lecture transcripts, and then selected only phrases that consist of an arbitrary 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. After this process we proceeded as in Section 4.1.

4.3 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.

4.3.1 Boosting Documents

The first algorithm attaches 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. First, for each lecture, the algorithm takes the title of the lecture, removes stopwords, and uses the result as a query to Wikipedia. Then, the best Wikipedia page match is concatenated to the lecture transcript. After attaching a Wikipedia page to each transcript, we run the previously described procedures over the collection of concatenated lecture-Wikipedia pages.

[1]From here on we use the term 'phrase' to mean n-grams of any length, including one, unless the distinction is significant.

For example, for the lecture titled "View Modifications Using Triggers", the Wikipedia query will return the Wikipedia page titled "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.

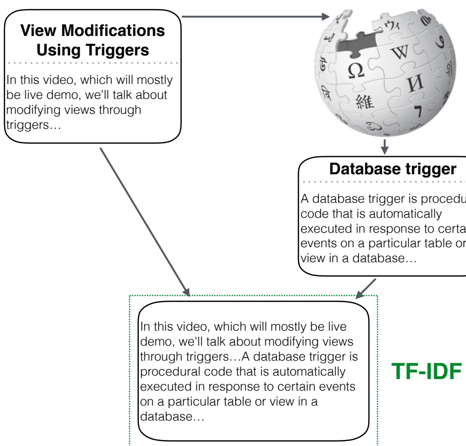

Figure 1: The Document Boosting algorithm searches for a Wikipedia page using the title of the lecture, concatenates the result to the lecture, and then runs TF-IDF over the combined document.

4.3.2 Boosting Phrases

This algorithm first creates a list of candidate index candidates using adjective-noun phrases. Then the candidates are ranked by their TF-IDF score summed over all Wikipedia documents. That is, for a phrase p, we define its term frequency $TF_w(p)$ as

$$TF_w(p) := \log(\text{number of times } p \text{ appears on Wikipedia})$$

, and its inverse document frequency $IDF_w(p, W)$ for a phrase p and Wikipedia article collection W as

$$IDF_w(p, W) := \log\left(\frac{|W|}{\text{number of documents in } W \text{ with } p} + 1\right)$$

and finally $TF\text{-}IDF_w(p, W)$ is defined as

$$TF\text{-}IDF_w(p, W) := TF_w(p) \cdot IDF_w(p, W)$$

Next, this global candidate ranking is combined with the basic TF-IDF approach described in Section 4.1. First, we create a normalized Wikipedia candidate ranking

$$TF\text{-}IDF_{w-norm}(p, W) := \frac{TF\text{-}IDF_w(p, W)}{\sum_{p'} TF\text{-}IDF_w(p', W)}$$

and normalized lecture collection ranking

$$TF\text{-}IDF_{norm}(p, l, L) := \frac{TF\text{-}IDF(p, l, L)}{\sum_{p'} TF\text{-}IDF(p', l, L)}$$

Then, we combine the two normalized rankings to form a final score

$$\begin{aligned} &TF\text{-}IDF_{combined}(p, l, L, W) := \\ &\qquad \eta TF\text{-}IDF_{w-norm}(p, W) + TF\text{-}IDF_{norm}(p, l, L) \end{aligned}$$

where η is used to determine the weight between Wikipedia scores and lecture scores. We found that a value of $\eta = 2$, meaning the Wikipedia ranking is weighted twice as much as the lecture ranking, works well in practice, and we report results of computations with that setting in place.

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 3. The approach can be thought of as multiplying $TF_w(p)$ by 0 if p consists of fewer than two tokens:

$$TF_{w-ngram}(p) := TF_w(p) \cdot \mathbf{1}\{p \text{ is at least two tokens}\}$$

A few subtleties deserve further description. First, in our simple TF-IDF approach, a score is calculated for each lecture l and phrase p, while in our Wikipedia calculation a score is only calculated for each p. That approach is chosen because in this algorithm we are not interested in keywords for every Wikipedia document, but only in obtaining a global sense of the importance of a phrase. The algorithm is equivalent to summing over all Wikipedia documents $d \in W$

$$TF_w(p) = \log\left(\sum_{d \in W} \text{number of times } p \text{ appears in } d\right)$$

Second, we take the logarithm of the phrase count in the Wikipedia ranking. This choice produces improved empirical results.

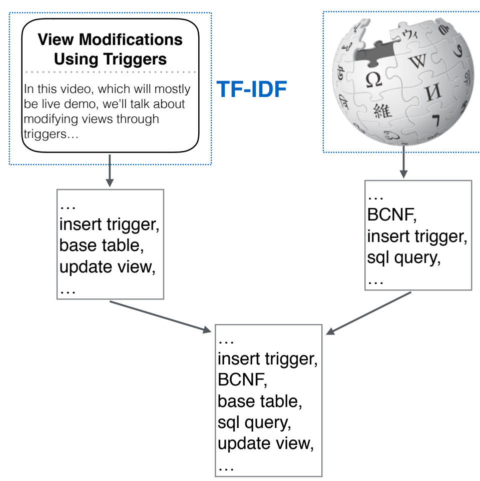

Figure 2: The Phrase Boosting algorithm runs TF-IDF over the entirety of Wikipedia and then combines the global ranking with a local ranking for a document.

4.4 Summary of Results

As stated in Section 3, we computed agreement between humans using Fleiss' Kappa. We evaluated each algorithm by computing Cohen's Kappa agreement between the algorithm and the gold set unified from the three human indexes as described in Section 3: the union of the lecture-phrase pairs in the gold indexes. That is a lecture-phrase pair is classified as *in-index* if any human indexer classified it as *in-index*, and classified as *not-in-index* otherwise. When using a single metric for agreement, such as Cohen's Kappa, paradoxes

Algorithm	κ	$\mathbf{P}_{\text{pos}}$	$\mathbf{P}_{\text{neg}}$	PABAK
TF-IDF	0.178	0.200	0.973	0.896
TF-IDF with Adjective-Noun Chunks	0.084	0.114	0.968	0.875
Document Boosting	0.180	0.199	0.974	0.901
Document Boosting with Adjective-Noun Chunks	0.131	0.154	0.971	0.890
Phrase Boosting	0.180	0.203	0.973	0.895
Phrase Boosting N-Grams	**0.204**	**0.224**	**0.975**	**0.902**

Figure 3

can arise when one category is much more prevalent than another and the chance-correction term in the agreement metric overcompensates. This effect leads to inappropriately low values of κ for high inter-rater agreement [8].

In our case, there are many more *not-in-index* phrases than phrases destined for the index. To illuminate the above-mentioned inappropriate κ values, we report three additional metrics: agreement on *in-index* decisions (P_{pos}), agreement on *not-in-index* decisions (P_{neg}) [4], and prevalence-adjusted bias-adjusted κ (PABAK) [3]. Because the algorithms produce a ranking of candidate phrases, we produce a binary classification by choosing a threshold above which candidates are labeled as *in-index*. In our reported results, we use the average number of keywords per lecture labeled by humans as the threshold.

These three metrics are defined in terms of a concordance table with two raters (the composite-human and the algorithm) and two categories (*in-index* and *not-in-index*)

	in-index	*not-in-index*	Total
Keyword	a	b	f_1
Not Keyword	c	d	f_2
Total	g_1	g_2	N

Figure 4: The concordance table used to calculate P_{pos} and P_{neg}.

Positive agreement is the number of terms both marked as *in-index* over the average number of terms marked as *in-index* and negative agreement is the number of terms both marked as *not in-index* over the average number of terms marked as *not in-index*

$$P_{\text{pos}} := \frac{a}{\frac{f_1+g_1}{2}} \qquad P_{\text{neg}} := \frac{d}{\frac{f_2+g_2}{2}}$$

PABAK is defined as the Kappa on a modified concordance table where a and d are replaced with their average and c and d are replaced with their average

	in-index	*not-in-index*
in-index	$(a+d)/2$	$(b+c)/2$
not-in-index	$(b+c)/2$	$(a+d)/2$

Figure 5: The concordance table used to calculate PABAK. a, b, c, and d are as defined in Figure 4

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. The fact that agreement between humans is only 0.336, and the lowest pairwise Kappa between humans was 0.309 suggests 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 plain TF-IDF algorithm was already able to achieve reasonable performance, with respect to the human annotators. Limiting the candidate set to adjective-noun chunks drastically hurt the performance of the algorithm, suggesting that many important phrases do not fit this linguistic pattern, and the restriction is too severe. Document Boosting and Phrase Boosting, the two algorithms that incorporated external knowledge, were able to make improvements on the basic algorithm. Phrase Boosting with the TF-IDF scores of all candidate phrases was slightly better than TF-IDF, and only boosting longer phrases (Phrase Boosting N-Grams) was able to improve further. We can see that negative agreement was high for all algorithms, suggesting that it is relatively easy to identify phrases that should not be indexed. Positive agreement was lower, implying it is indeed difficult to identify a small number of phrases to summarize the document, given that there are a large number of possible phrases, and there can be legitimate disagreement on what phrases are most important. Note that PABAK is not linearly related to Cohen's Kappa, P_{pos}, and P_{neg}, which explains how the Document Boosting algorithm can have a higher PABAK despite having a lower P_{pos} and Cohen's Kappa.

5. DISCUSSION

In order to convey intuition for how the algorithms differ, we will examine the index phrases extracted from a few lectures in depth.

In general, Phrase Boosting can be thought of as imposing a prior distribution over what phrases should be considered important. This prior is then combined with the local knowledge in a specific lecture. In practice, TF-IDF tends to give longer phrases unfairly low scores, because they do not appear frequently, even though they may be quite important to the content. Phrase Boosing corrects this by raising the value of longer phrases. For example, in the lecture "DTDs IDs and IDREFs", the phrase "Document Type Descriptors" is clearly important, but only appears 5 times in the lecture, is therefore ranked 36th if Phrase Boosting is not used. After incorporating the global Wikipedia ranking and the preference for longer phrases, the phrase is boosted to the 9th rank. There are similar occurences throughout the lecture collection of longer phrases that only appear a small number of times in the lecture but have a high $TF\text{-}IDF_w$ score on the Wikipedia ranking. In a lecture on "Multivalued Dependencies and Fourth Normal Form", the phrase "multivalued dependencies" appears only 3 times in the lecture transcript and is therefore not considered an index phrase by plain TF-IDF. The phrase appears 26 times on Wikipedia in 4 documents, and is tagged as a phrase to include in the index when the Wikipedia ranking is incorporated.

Observe the following convenient properties of Phrase Boosting from algorithmic and computational perspectives. First, we formed the combined ranking $TF\text{-}IDF_{combined}$ by giving equal weight to $TF\text{-}IDF_{w-norm}$, phrase scores in the Wikipedia ranking, and $TF\text{-}IDF_{norm}$, phrase scores in the lecture ranking. One could modify the preference of the algorithm towards favoring local or external information by weighting these two components differently. If all weight were given to $TF\text{-}IDF_{norm}$ the algorithm would output the same rankingsx as TF-IDF, and if all weight was given to $TF\text{-}IDF_{w-norm}$ the algorithm would output the ranking extracted from Wikipedia. This principle applies more generally, and Phrase Boosting can incorporate any prior belief about index phrases. Second, although there are a large number of documents in Wikipedia (the algorithm was run on 5,027,125 documents, a total size of about 50GB), the runtime complexity is linear in the number of documents and therefore quite tractable on modern hardware.

Document Boosting	**TF-IDF**
transaction	transaction
isolation level	read
read	isolation level
lock	t1
commit	t2
concurrent	commit
serialization	client
repeat read	dirty read
concurrency control	uncommit
dirty read	transaction isolation level
transaction commit	GPA

Figure 6

The Document Boosting algorithm can be thought of as amplifying the index phrases for a lecture, and therefore decreasing the scores of phrases that happen to occur frequently in a lecture but are not meaningful. This effect often occurs if a lecture has worked examples that use words from the examples frequently, as can be seen in Figure 6. The table shows the top ranked phrases for the Document Boosting algorithm and plain TF-IDF for a lecture on "Isolation Levels". In TF-IDF's index phrases we can see that the lecture included an example involving students, and the token "GPA" appears frequently, even though it is not important to the core concept of the lecture. Similarly, the instructor used examples with transactions named "T1" and "T2", which appeared frequently in this lecture and not others, in effect tricking the TF-IDF metric.

When Document Boosting was run, the Wikipedia article "Isolation (database systems)" was selected, which was able to decrease the frequency of phrases that were part of specific examples, and augment the frequency of words that were related to the concept of isolation levels, such as "concurrency control", "repeat read", and "transaction commit". We can see that conceptually the algorithm is boosting phrases in the intersection of the Wikipedia article and lecture. This allows noise that only occurs frequently in the lecture to be identified and filtered.

6. 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. We therefore began by evaluating the performance of term frequency-inverse document frequency, a well-known metric for gauging the importance of a phrase to a document. 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.

In the process of this work we paid three high-quality persons to index an internationally renowned database course. We used the three indexes to evaluate our algorithms. In an effort to allow our work to be reproduced at other institutions, and to foster additional work in this area we are making the three indexes and the course video transcripts publicly available.

In the future we will explore using the rich structure provided by the Wikipedia dataset to improve our keyword extraction algorithms further. For example, Wikipedia grants access to the link structure between its documents, and groups documents into collections, which are explored for different tasks in [11] and [17].

We are also interested in keyword extraction as a supervised learning task. As previously described, one of the main challenges is the cost of obtaining a large amount of labelled data, especially in the online education setting where annotators often need to be highly educated and devote a substantial amount of time to generate high quality indexing. One possible strategy is transfer learning, where a learning algorithm is trained on a different problem than the one on which it will make predictions. It is possible that there are features that differentiate important phrases in journal abstracts or newspapers that could also differentiate important phrases in lectures. Indeed, transfer learning for text classification has been explored previously [21], [6].

7. REFERENCES

[1] A. Agrawal, J. Venkatraman, S. Leonard, and A. Paepcke. Youedu: Addressing confusion in mooc discussion forums by recommending instructional video clips. 2015.

[2] S. P. Balfour. Assessing writing in moocs: Automated essay scoring and calibrated peer review (tm). *Research & Practice in Assessment*, 8, 2013.

[3] T. Byrt, J. Bishop, and J. B. Carlin. Bias, prevalence and kappa. *Journal of clinical epidemiology*, 46(5):423–429, 1993.

[4] D. V. Cicchetti and A. R. Feinstein. High agreement but low kappa: Ii. resolving the paradoxes. *Journal of clinical epidemiology*, 43(6):551–558, 1990.

[5] D. Coetzee, A. Fox, M. A. Hearst, and B. Hartmann. Should your mooc forum use a reputation system? In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '14, pages 1176–1187, New York, NY, USA, 2014. ACM.

[6] C. Do and A. Y. Ng. Transfer learning for text classification. In *NIPS*, pages 299–306, 2005.

[7] A. Eckerdal, P. Kinnunen, N. Thota, A. Nylén, J. Sheard, and L. Malmi. Teaching and learning with moocs: Computing academics' perspectives and engagement. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ITiCSE '14, pages 9–14, New York, NY, USA, 2014. ACM.

[8] A. R. Feinstein and D. V. Cicchetti. High agreement but low kappa: I. the problems of two paradoxes. *Journal of clinical epidemiology*, 43(6):543–549, 1990.

[9] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[10] S. Halawa, D. Greene, and J. Mitchell. Dropout prediction in moocs using learner activity features. *Experiences and best practices in and around MOOCs*, 7, 2014.

[11] X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou. Exploiting wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 389–396. ACM, 2009.

[12] A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223. Association for Computational Linguistics, 2003.

[13] A. Imran, L. Rahadianti, F. Cheikh, and S. Yayilgan. Semantic tags for lecture videos. In *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*, pages 117–120, Sept 2012.

[14] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 32 Avenue of the Americas, New York, NY 10013-2473, USA, 2008.

[15] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01):157–169, 2004.

[16] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. Association for Computational Linguistics, 2004.

[17] D. Milne. Computing semantic relatedness using wikipedia link structure. In *Proceedings of the new zealand computer science research student conference*, pages 1–8, 2007.

[18] Y. Ohsawa, N. E. Benson, and M. Yachida. Keygraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In *Research and Technology Advances in Digital Libraries, 1998. ADL 98. Proceedings. IEEE International Forum on*, pages 12–18. IEEE, 1998.

[19] Y. Park and Y. Li. Extracting salient keywords from instructional videos using joint text, audio and visual cues. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 109–112. Association for Computational Linguistics, 2006.

[20] C. Piech, J. Huang, Z. Chen, C. B. Do, A. Y. Ng, and D. Koller. Tuned models of peer assessment in moocs. *CoRR*, abs/1307.2579, 2013.

[21] R. Raina, A. Y. Ng, and D. Koller. Constructing informative priors using transfer learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 713–720. ACM, 2006.

[22] S. Rose, D. Engel, N. Cramer, and W. Cowley. Automatic keyword extraction from individual documents. *Text Mining*, pages 1–20, 2010.

[23] G. Salton. The smart retrieval systemâĂŤexperiments in automatic document processing. 1971.

[24] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, Nov. 1975.

[25] P. Turney. Learning to extract keyphrases from text. 1999.

[26] D. Yang, T. Sinha, D. Adamson, and C. P. Rose. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Proceedings of the 2013 NIPS Data-driven education workshop*, volume 11, page 14, 2013.