Labeling Educational Content with Academic Learning Standards

Danish Contractor
IBM Research
New Delhi, India
dcontrac@in.ibm.com

Kashyap Popat
IBM Research
Bangalore, India
kaspopat@in.ibm.com

Shajith Ikbal
IBM Research
Bangalore, India
shajmoha@in.ibm.com

Sumit Negi
IBM Research
New Delhi, India
sumitneg@in.ibm.com

Bikram Sengupta
IBM Research
Bangalore, India
bsengupt@in.ibm.com

Mukesh K Mohania
IBM Research
New Delhi, India
mkmukesh@in.ibm.com

Abstract

Learning standards (frequently referred to as academic standards, course curriculum *etc*.) define the specific structure of an educational program. Learning standards contain a list of instructions specifying various skills that students should learn at different points during their learning progression. For example,"*calculate the area of a triangle*" is one such instruction in a 6^{th} grade geometry curriculum. Currently these instructions are imparted using prescribed textbooks or lesson plans which have been labeled with learning standard instructions. Teachers and students use this labeled learning content to identify relevant material for teaching and studying. However with an increasing amount of users as well as publisher generated content in recent days, teachers and students may want to refer to additional content apart from prescribed textbooks for their teaching/learning needs which is not labeled with learning standard instructions. Manually identifying the appropriate learning standard instruction for each learning content is time consuming and not scalable especially since learning standards frequently contain thousands of instructions, and subject to periodic revision.

In this paper, we address the problem of automatically labeling digital learning content with the learning standards. Towards this goal, we first build semantic representations of the learning standard instructions using external knowledge sources such as Wikipedia and domain text books. These semantic representations are then used in a framework which utilizes structural constraints imposed by the hierarchy of the learning standards to assign labels to the learning materials. We demonstrate the usefulness of our approach on a collection of high school learning materials that were labeled by curriculum experts from a US school district according to a publicly available learning standard. The system developed has been deployed and is in use by the school district. To the best of our knowledge we are the first to attempt this novel task and develop such a system.

1 Introduction

The education domain is witnessing an unprecedented transformation primarily driven by digitization of vast amount of educational data and its processes. As a consequence, enormous amount of user and publisher generated learning materials are being made available online. This ever increasing amount of digitized learning material is slowly changing the way students learn, plan and progress through their educational careers. A survey conducted in the United States found that 84% of teachers use the Internet weekly to find additional content for teaching [13]. Providing students access to the right content has been shown to have a positive correlation with student engagement and student performance [17]. Systems such as Desire2learn[1] (D2L), Knowillage[2] and open source learning content management systems (LCMS) such as Moodle[3] attempt to cater to these requirements. However, the widespread growth of the learning materials necessitates the development of newer systems that would efficiently manage, organize and deliver the content.

In the context of a specific educational program, such as K-12 schooling, a key unsolved problem in content management is to automatically label the learning materials with the corresponding learning standards. Learning standards specify at a granular level what set of skills a student should acquire as a result of pursuing a particular course/subject in an education program. The CCGPS (Common Core Georgia Performance Standard) and CCSS (Common Core State Standards) are examples of learning standards followed in different school districts of the United States for K-12 schooling. These learning standards are typically organized according to *Grade* $\rightarrow$ *Subject* $\rightarrow$ *Course* $\rightarrow$ *Topic* $\rightarrow$ *Instruction*. This organization exhibits a natural and explicit hi-

[1] http://www.desire2learn.com
[2] http://www.knowillage.com
[3] http://www.moodle.org

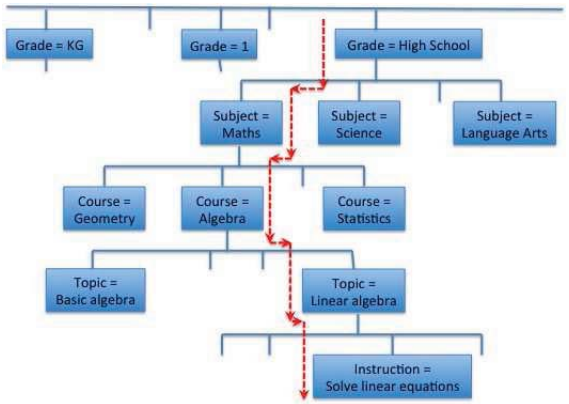

Figure 1: Hierarchical structure of learning standard

Mathematics

ALGEBRA I CC

A - Algebra

- interpret expressions that represent a quantity in terms of its context (Emphasis on linear expressions and exponential expressions with integer exponents.) (CCGPS) (MAL1_A2012-1/MCC9-12.SSE.1)
- interpret parts of an expression such as terms, factors, and coefficients (Emphasis on linear expressions and exponential expressions with integer exponents.) (CCGPS) (MAL1_A2012-2/MCC9-12.SSE.1_a)

C - Geometry

- identify and use special right triangles (GPS) (MAA2_C2009-20)
- determine the lengths of sides of 30° - 60° - 90° triangles (GPS) (MAA2_C2009-21)
- determine the lengths of sides of 45° - 45° - 90° triangles (GPS) (MAA2_C2009-22)

Figure 2: Example Learning Standard: High School

erarchy as illustrated in Figure 1 - a Grade has multiple Subjects[4] being taught, a Subject can have different Courses[5], a Course covers multiple topics and finally there are multiple instructions under a given topic. An *instruction* specifies the specific skill a student is expected to learn when pursuing a Subject/Course in a given Grade. Instructions are the most granular objects in any Learning Standard and are associated with a short textual description that details a specific skill. For example, "*Use rational approximations of irrational numbers to compare the size of irrational numbers, locate them approximately on a number line diagram, and estimate the value of expressions* " is an instruction under *Grade:*8/*Subject:*Math in the Common Core learning standard. Figure 2 shows a few more examples of such instructions for an Algebra course in High School Mathematics.

Learning material annotated with such instructions, ingested into an LCMS would allow teachers and students to efficiently browse through the labeled and cataloged collection of the learning materials. However, there are no automated systems that can label learning content with learning standards. Relying on subject and curriculum experts to tag every piece of learning material is not only expensive but also cannot scale to the rapidly increasing amounts of content being generated. Moreover, learning standards can have tens of thousands of instructions (for example, the Academic Knowledge and Skill (AKS)[6] learning standard has over 30,000 instructions) and each learning content could be related to one or more of these instructions. Further, learning standards are revised periodically. All the above reasons necessitate the need for an automated system that labels learning content with the learning standards.

The problem of automatically labeling content with the learning standards is challenging given the fact that instructions are often ambiguous because of their shorter lengths and the choice of words used in their descriptions. Hence use of information only from the instructions may not be enough to achieve accurate labeling. For example the instruction "*relate temperature, pressure and volume of gases to the behavior of gases*" refers to the concepts of Charles' and Pascal's law - laws in physics and chemistry, which relate pressure, volume and temperature of gases. While the instruction description makes no explicit reference to these concepts, any system that aims to label content with learning instructions would benefit from being able to establish those references.

The labeling task is effectively a multi-label classification problem, where the set of all instructions defines the set of classes to be assigned to the learning materials. In the presence of training data this could be solved by training classifiers in a supervised manner. However, given the fact that learning standards typically contain a large number of instructions (*e.g.*, 30,000 AKS instructions), it is impractical to assume an availability of training data to train classifiers.

In this paper, we describe an unsupervised method to automatically label documents with the appropriate learning instructions. We model each instruction as a collection of terms (features) that are relevant to that instruction and use external knowledge sources to overcome the lexical gap between learning standard instructions and learning content. To the best of our knowledge the problem of automatically labeling content with learning instructions has not been addressed earlier despite its practical applications in the education domain[7]. Our paper makes the following contributions:

- We define the novel task of automatically labeling learning content with the learning standard instructions.
- We describe a first implemented and deployed system that automatically labels learning materials with the learning standard instructions without requiring supervision.

[4]Math, Science, Language Arts *etc.*

[5]For instance courses under Math subject : Algebra, Trigonometry *etc.*

[6]`http://publish.gwinnett.k12.ga.us`

[7]This problem was motivated from requirements gathered during discussions with two of the largest school districts in the United States.

- We evaluate our system by conducting experiments using a real world learning content data set obtained from a US public school district made of two subjects - High School Mathematics and High School Science. The documents have been labeled using a publicly available learning standard by curriculum experts from same US public school district. We found that our system had a labeling accuracy of 71% on Science content and about 81 % on Mathematics content.

In the next section, we formally describe our problem and the rest of the paper is organized as follows: In section 3, we present some previous work relevant to our work. In section 4, we describe our solution along with detailed descriptions of the components and the design choices, section 5 presents detailed experiments demonstrating the effectiveness of our approach, section 6 describes how our system has been deployed in production. Lastly, section 7 concludes the paper summarizing our contributions and scope for future work.

2 Problem Statement

Instructions in a learning standard are grouped by *topic*, *course*, *subject* and *grade*. This grouping exhibits a natural and explicit hierarchy as illustrated in Figure 1- a *grade* has multiple subjects being taught, a *subject* can have different courses, each *course* has a set of topics that are covered in that course and finally a *topic* has a set of instruction associated with it[8]. For example the following example, which is taken from the AKS (Academic Knowledge and Skills) learning standard, shows this hierarchy – *Grade*: 9 $\rightarrow$ *Subject*: Mathematics $\rightarrow$ *Course*: Algebra I CC $\rightarrow$ *Topic*: Algebra $\rightarrow$ *Instruction*: "*interpret expressions that represent a quantity in terms of its context (Emphasis on linear expressions and exponential expressions with integer exponents.)*". With this background, the task is then to label[9] the content with the most appropriate instructions from the learning standard.

3 Prior and related work

The last few years have seen a considerable interest in the area of educational content mining. This includes work on quantitatively assessing the comprehension burden [1] that a textbook imposes on the reader, automatically enriching text books with additional content *etc*[3][2]. In [1], authors present a formal definition of comprehension burden and propose an algorithmic approach for computing it. This tool, which has been applied to a corpus of high school text books from India, has been shown to be effective in identifying sections of text books that can benefit from reorganizing the presented material. Considering that a large number of text books lack clear and adequate coverage of important concepts authors in [3] propose a technological solution for algorithmically identifying those sections of a book that are not well written and could benefit from better exposition. The authors provide a decision model based on the syntactic complexity of writing and the dispersion of key concepts, which helps identify sections of text books that can benefit from content enrichment. In a similar spirit, authors in [2] seek to augment text books using digital visual material to enhance the learning experience. Our work contributes to this area of educational content mining by proposing and solving a novel and practical problem of aligning educational content with learning standard.

The use of external knowledge sources, in particular Wikipedia, for machine learning and NLP tasks is not new. There is a considerable amount of literature where authors have used Wikipedia for semantic relatedness [16] [15],word sense disambiguation [10] [12], co-reference resolution [18], multilingual alignment [7] *etc*. On similar lines, in this work, we use external knowledge sources to build lexicons for each instruction in a learning standard. Authors have also used Wikipedia for information retrieval task using the concept of explicit semantic analysis (ESA). ESA is a vectorial representation of text (individual words or entire documents) that uses a document corpus as a knowledge base. Specifically, in ESA, a word is represented as a column vector in the *tf-idf* matrix of the text corpus and a document (string of words) is represented as the centroid of the vectors representing its words. Authors [8] have used the Wikipedia corpus for ESA and shown good results for retrieval problems. More specifically, in the method proposed by [8] machine learning techniques are used to explicitly represent the meaning of any text as a weighted vector of Wikipedia-based concepts. Assessing the relatedness of texts in this space amounts to comparing the corresponding vectors using conventional metrics. As evident one could use a similar technique for finding instruction and content similarity, *i.e.*, map both instruction and content into the ESA space as defined by wikipedia category and calculate the similarity of the corresponding vectors using say cosine similarity. However, due to the very specific/focused nature of our instructions this approach is not feasible for our setting.

4 Our Approach

This section provides an architectural overview of our system (refer Figure 3) and provides details of the different components involved. The end goal, as mentioned earlier, is to be able to automatically link or annotate content with instructions. To achieve this we first build semantic representations

[8] One should note that there might be some variation in this hierarchy structure - based on grade and learning standard - for instance in elementary and middle school there is no *course* and *topic* artifact, all instructions are organized directly under *subject*

[9] A valid Grade $\rightarrow$ Subject $\rightarrow$ Course $\rightarrow$ Topic $\rightarrow$ Instruction combination from the learning standard hierarchy.

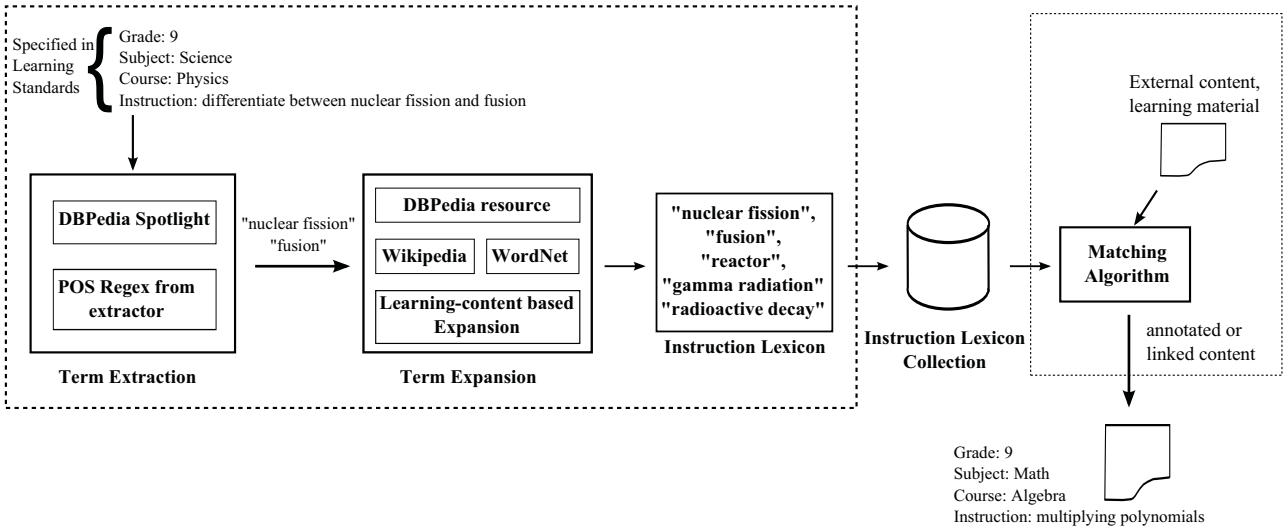

Figure 3: System Architecture

(later referred as lexicon) of the learning standard instructions using external knowledge sources such as Wikipedia and domain text books. We then use these semantic representations to build vector space models of the learning standard instructions and the learning material to compute their similarity scores. These scores are then used in a framework that utilizes structural constraints imposed by the hierarchy of the learning standards to assign labels to the learning materials. The next section describes how we build the semantic representations of the learning standard instructions.

4.1 Building the Instruction Lexicon: In order to build an instruction specific lexicon, we first extract important terms from the instruction and then build *instruction lexicons* using *term* based and *instruction* based expansion methods. The *term/instruction based expansion* methods are aimed at discovering a set of words/phrases that are closely associated with one or many of the extracted terms or the entire learning instruction. These expansions are then filtered based on a relevancy criteria as described in section 4.1.4.

4.1.1 Term Extraction: We extract key terms from learning standard instructions using parts of speech (POS) tags and annotations from the DBPedia spotlight service [9].

Typically nouns and noun phrases constitute an important component of the concept expression and hence it is important to not miss them. We use Stanford tagger[10] to generate POS tags and spot noun phrases in the instructions and mark them as key terms.

In addition, we use the DBPedia Spotlight service [9] to identify additional key terms. DBpedia Spotlight uses data from Wikipedia information boxes to annotate text. Apart from identifying key terms DBPedia also tries to associate a Wikipedia article page with each spotted term. This allows one to retrieve additional information about the spotted term, for instance, the associated Wikipedia Category *etc*.

A union of all the key terms obtained from each instruction using the steps above are used as the set of key terms extracted for that instruction. The next section describes the process of generating *expansions* for each of these *key* terms.

4.1.2 Term Expansions: An "expansion" in our context refers to related words/phrases for a given term. We classify the expansions we generate as *term* based expansions and *instruction* based expansions.

Term based expansions are the set of related words/phrases identified by looking up sources such as WordNet, word embeddings (generated using a recursive neural network (SENNA) [5] *etc*.

Instruction based expansions, on the other hand, use the complete set of terms extracted from an instruction to identify related words and phrases.

- **Term based expansion using Wikipedia:** For each extracted key term that contains a Wikipedia reference[11], we use the corresponding Wikipedia article to identify the corresponding expansion terms using TextRank[11] algorithm. For key terms that do not contain a Wikipedia reference, we use Wikipedia articles whose title matches exactly with the key term, with an additional constraint that the page category of the Wikipedia article should also match with the topic (in the instruction tree) of the instruction from which the key term is extracted. The TextRank algorithm extracts key phrases from a given document by building a graph where the vertices are the words in the document and

[10] http://nlp.stanford.edu/downloads/tagger.shtml

[11] As provided by DBpedia spotlight

edges between them are assigned with weights equal to the measure of similarity between them. In our implementation, the edges between words occur only if constituent words occur within a window of 20 words in the document. The algorithm uses a variation of PageRank [14] adapted to work on sentences, to extract important words and phrases. We run the TextRank algorithm only on the introductory passages of the Wikipedia article and not the complete article to avoid noisy expansions. This is done because a Wikipedia article may contain very specialized sections which may be beyond the scope of high school curriculum and selecting these can lead to conceptually distant terms getting added to the lexicon. For instance, the Wikipedia article on "*Gravitation*" contains references to advanced concepts such as "*Lyman-Alpha Forests*" which may not be relevant for high school Physics. The introduction section too sometimes contains references to such advanced concepts and we describe how we remove them in section 4.1.4

- **Term based expansion using Word Vector embeddings:** We make use of word embeddings generated using a recursive neural networks (SENNA) [5] to estimate additional term expansions. This has 50 dimensional feature representations for about 130000 words generated by training network on Wikipedia and Reuters news data sets. We use cosine similarity between vectors corresponding to two terms as a measure of semantic similarity. Cosine similarity between two vectors v_1 and v_2 is given by

$$sim(v_1, v_2) = \frac{v_1.v_2}{|v_1||v_2|} \tag{4.1}$$

Let WV_t be the word embeddings vector associated with a term t and let the word embeddings vector of a candidate expansion term t_{exp} be $WV_{t_{exp}}$. We then go over all the words (V) and select a set terms (T_{exp}) as an expansion of term t. Where T_{exp} is defined as:

$$\{t_{exp} : t_{exp} \in V \ \& \ sim(WV_t, WV_{t_{exp}}) > \delta\} \tag{4.2}$$

4.1.3 Instruction based expansion: In order to generate *instruction* based expansions, we use the complete set of *key* terms extracted from an instruction and build a disjunctive query which is then issued on a collection of Wikipedia articles as well as educational content (prescribed text books, content *etc.*) available. Accordingly instruction based expansions are generated using two sources:

- **Educational Content**: These include documents such as digitized reference books, textbook *etc*. For each instruction we used the set of key terms T extracted from the instruction to build a disjunctive query and look up an Apache Lucene[12] index containing the learning content. The TextRank algorithm[11] is then used on the top k documents[13] returned from each query to extract terms and phrases. These sets of terms and phrases are included in the set of "expansions" for the instruction.
- **Wikipedia Articles**: In this step, we utilized the same disjunctive text query as described previously to retrieve Wikipedia articles[14]. The top *k* Wikipedia articles that matched the query are identified. Related Wikipedia articles (found by applying Green's method [6] on Wikipedia's hyperlink network structure) are also added to this set. After having identified the set of articles relevant to the instruction, key terms are extracted from this set of articles using TextRank. Similar to term extraction method described in section 2.2.1, these terms are also extracted only from the introductory passages of the Wikipedia article.

4.1.4 Pruning the lexicon: Once the terms are expanded, these list of expansions need to be cleaned to remove references to named entities that are not *relevant* for learning concepts in Mathematics and Science and also references to uncommon word senses.

We determine the relevancy of an expansion by calculating its relevance score with respect to all the terms T. The *relevance score* of an expansion (t_{exp}) is given by :

$$RelScore(t_{exp}, T) = \frac{\sum_{\forall t \in T} w_{t,t_{exp}}}{|T|} \tag{4.3}$$

where $w_{t,t_{exp}}$ is a similarity score between an expansion $t_{exp} \in T_{exp}$ and term $t \in T$. In other words, the relevancy score associated with each expansion is the average of its similarity values with all the terms present in T.

We compute the similarity score using the **Normalized Wikipedia distance (NWD)**. We define the Normalized Wikipedia Distance similar to the Normalized Google Distance (NGD). The Normalized Google Distance [4] is a semantic distance measure that uses the number of document hits returned for a given set of keywords using the Google[15] search engine.

Thus, for our work, we adapted the NGD to work on the Wikipedia Encyclopedia corpus. We indexed the Wikipedia corpus [16] in Apache Lucene and we define the NWD (Normalized Wikipedia distance) the same way as the NGD but in this case

$$NWD(t_1, t_2) = \frac{max(log(N_{t_1}), log(N_{t_2})) - log(N_{t_1,t_2})}{logN - min(log(N_{t_1}), log(N_{t_2}))} \tag{4.4}$$

where N_{t_1} is the number of articles returned from the Wikipedia dataset when term t_1 is used as the keyword for

[12] http://lucene.apache.org/core

[13] from the Apache Lucene index

[14] We locally indexed English Wikipedia articles using Apache Lucene

[15] http://www.google.com

[16] http://download.wikipedia.org

querying , N_{t_2} is the number of articles returned from the Wikipedia dataset when term t_2 is used as the keyword for querying and N_{t_1,t_2} is the number of documents when both terms t_1 and t_2 are used to query the Wikipedia data set and N is the total number of Wikipedia articles indexed.

In order to prune the list of expansions using the NWD measure we define the edge weights in Equation 4.3 as

$$w_{t,t_{exp}} = 1 - NWD(t, t_{exp}) \tag{4.5}$$

4.2 Labeling Content: Our aim is to estimate a set of most relevant learning standard instructions to be associated/linked with each learning content. To identify the set of such relevant instruction, the document is compared with each "*instruction lexicon*" to compute matching scores. Then the instructions are ranked according to their matching scores to estimate top matching instructions as the most relevant. In our implementation instead of computing matching scores independently for each instruction, we actually utilize the hierarchical structure of the learning standards to achieve a robust search of the top ranking match list.

An important step in the matching task is to build suitable representation of the documents and the instruction lexicons to compute matching scores between them. For this purpose, we use a tf-idf[17] based vector space model as described next.

4.2.1 Vector Space Model: To compute *tf-idf* vector for a given content document we use a large collection of generic documents, denoted by set G. Each document in G is annotated with the generic domain it belongs to, such as '*Science*' or '*Mathematics*'[18] . Given a content document d, we first select a subset of documents G_s i.e. $G_s \subset G$ whose generic domain is same as that of d, and use document set G_s to compute *tf-idf* vector for d.

To compute tf-idf vector for instructions, "*instruction lexicon*" (computed as described in section 4.1.2) are treated as documents. As shown in Figure 1, learning standards are hierarchically organized in a tree structure, with leaf nodes corresponding to instructions. To compute the tf-idf vectors for each of these leaf nodes, the generic documents are used as the reference corpus. As explained in next section, for labeling we use not only the matching scores of the leaf nodes but also the matching scores of the parent nodes in the tree corresponding to topics, courses, subjects and grades. To compute their matching scores we need to first build tf-idf vectors for those nodes too.

Lexicons for the parent nodes are built in a bottom up fashion. For each parent node, its "*instruction lexicon*" is built as a union of "*instruction lexicon*" of its child nodes. Once the "*instruction lexicon*" for all the nodes are computed, their tf-idf vectors are computed similar to that for the instruction nodes, *i.e.*, the generic documents are used as a reference corpus to compute the tf-idf vectors. The next section describes how we label documents using vector space representations of topic, course, subject and grade information as described above.

4.2.2 Matching: Instead of linking tree nodes independent of each other, we impose constraints based on hierarchical structure of the learning standards. For example, linking of a topic is allowed only if its course and subject also gets reasonably good matching scores. Such constraints help in eliminating spurious matches. We aim to annotate with complete root to leaf paths not just the leaf nodes corresponding to the instructions. If in the learning standard tree, the set of nodes corresponding to grades are denoted by g_i, nodes corresponding to subjects are denoted by s_i, nodes corresponding to courses are denoted by c_i, nodes corresponding to topics are denoted by t_i, and nodes corresponding to instructions are denoted by a_i, then a path p_i in the tree is constituted by the set of nodes $p_i = (g_i, s_i, c_i, t_i, a_i)$. An example of such path is illustrated in Figure 1 marked by a dotted line with arrows from root to a leaf node. Annotating/linking with such a path would effectively link the corresponding instruction, its topic, its course, its subject and its grade together to a content document.

In order to find paths best matching with the content, the set of all such paths p_i are scored against the content to compute their matching score. To compute a rank list of the paths: 1) Each path is assigned a matching score equal to the matching score of its leaf node (instruction level matching), 2) Set of all paths are ranked according to their matching scores, and 3) Validity of each path in the rank list is determined and invalid paths are discarded from the list. The criteria used to determine the validity of a path is based on the scores of its constituent nodes. Specifically, we consider a path as valid only if matching score for every node in the path (except its leaf node) is at least a fraction of matching score of its child node. In our system we assume the fraction of parent-to-child matching score should be at least 0.1. Since the topics in the learning standards represent multiple instructions, their matching scores are expected to be lower than the instruction level matching score, on the other hand because of the broader level topic match the score is not expected to be too low. Any path that does not satisfy this constraint is discarded from the rank list. Matching scores of the nodes are computed using cosine similarity metric defined in Equation (4.1).

[17] http://nlp.stanford.edu/IR-book/html/htmledition/the-vector-space-model-for-scoring-1.html

[18] This corpus is built by crawling publicly available sources of mathematics and science documents/articles as described in the Experiment section

Subject	#Courses	#Topics	#Instructions	#Words per Instruction
HM	29	43	1621	16.39
HS	13	22	1759	12.22

Table 1: Details of the learning standards. HM: High School Mathematics, HS:High School Science

Configuration	With Pruning		No Pruning	
	HM	HS	HM	HS
WordNet	31.96	30.84	69.96	55.48
Wikipedia	26.23	23.98	37.47	27.99
Word Embedding	97.4	87.16	275.38	198.25
Wikipedia+WordNet	46.33	41.73	91.67	71.40
Wikipedia+Word Embedding	69.93	140.47	296.21	214.33

Table 2: Average size of lexicon per instruction. HM: High School Mathematics, HS: High School Science

5 Experimental Setup & Results

In this section, we describe experiments demonstrating the performance of our system and highlight the benefits of the different components used.

5.1 Data: We used two datasets for our experiments : The first dataset (text dataset) consisted of 179 educational documents for two subjects from High School - Mathematics and Science, provided to us with gold labels, by a school district in the United States. The documents were in different formats. We used Apache POI[19] to convert them into plain text files for ease of processing. These documents were labeled with grade, subject, course and topic labels based on the Academic Knowledge and Skills (AKS) learning standard by curriculum experts in the school district. The details of the learning standards for High School Mathematics and High School Science are given in Table 1.

The second dataset (video dataset) was a collection of 30 High School Mathematics and 30 High School Science videos from Khan Academy. The duration of the videos ranged between 10-15 minutes. The audio from these videos was extracted and transcribed using the speech transcription engine[20]. These transcriptions were used as documents for labeling.

5.2 Building the Instruction Lexicon: For each of the learning standard instructions we build a lexicon of terms as described in sections 4.1.1 and 4.1.2. In case of term level expansions using Word Embeddings, the value of δ in Equation 4.2 was set as 0.6. Table 2 provides the average size of the lexicon per instruction, generated using these configurations.

As described in section 4.1.3 we make use of educational content and Wikipedia articles to generate expansions at the instruction level. The educational content collection comprises of 8595 documents related to Mathematics and Science sourced from freely available online resources such as Wikibooks[21] and Project Gutenburg[22]. These are also used as the *generic documents* mentioned in section 4.2.1.

5.3 Evaluation Measures: We evaluate the performance of our system using the following measures:

- **Minimal Match Accuracy**: To compute the minimal match accuracy we evaluate whether any of the top m labels assigned by our system are present in the gold standard label. Let the set of gold standard labels associated with a document i be G_i and let the set of top m labels assigned by the system for document i be S_i. Then the minimal match accuracy at m (minimal match accuracy @m) is defined as:
$$\frac{\sum_i^N \mathbb{1}\{|G_i \cap S_i| \geq 1\}}{N} \tag{5.6}$$
where $\mathbb{1}\{\}$ is the indicator function and N is the number of labeled documents.

- **Full Match Accuracy** : The full match accuracy is more strict than the minimal match accuracy and evaluates if all of the gold standard labels are present in the top m labels assigned by our system. Thus, if G_i is the set of gold standard labels and S_i is the set of top m labels assigned by our system, the full match accuracy at m (full match accuracy @m) is given by:
$$\frac{\sum_i^N \mathbb{1}\{|G_i \cap S_i| = |G_i|\}}{N} \tag{5.7}$$
where $\mathbb{1}\{\}$ is the indicator function and N is the number of labeled documents.

- **Mean Reciprocal Rank (MRR)** : In a ranked list of labels S assigned by our system for a given document, if the label at rank r is the first correct label assigned, the reciprocal rank for the labeling would be defined as $\frac{1}{r}$. The mean reciprocal rank for N documents is given by:
$$\frac{1}{N}\sum_i^N \frac{1}{r_i} \tag{5.8}$$

- **Recall** : The recall of the system is given by:
$$\frac{1}{N}\sum_i^N \frac{|G_i \cap S_i|}{|G_i|} \tag{5.9}$$
where G_i and S_i are the gold standard labels and system generated labels respectively for document i.

[19] https://poi.apache.org

[20] Details about the data set, transcription process and experiments performed on this dataset are available in the supplementary notes posted on the first author's webpage (DOI: 10.13140/2.1.3728.4166).

[21] http://www.wikibooks.org

[22] http://www.gutenberg.org

Configuration	High School Mathematics							High School Science						
	MMA (%)		FMA (%)		Recall		MRR	MMA (%)		FMA (%)		Recall		MRR
	@5	@10	@5	@10	@5	@10		@5	@10	@5	@10	@5	@10	
NoExpansion	75.00	83.33	66.67	66.67	0.68	0.73	**0.82**	79.85	88.48	76.97	86.33	0.77	0.86	0.72
WordNet	58.33	83.33	41.67	75.00	0.52	**0.84**	0.72	76.97	84.17	73.38	81.29	0.74	0.82	0.68
Wikipedia	**83.33**	**83.33**	**75.00**	**75.00**	**0.79**	0.79	0.70	**87.77**	**90.64**	**84.89**	**88.48**	**0.85**	**0.88**	**0.74**
Word Embedding	83.33	83.33	66.67	66.67	0.79	0.79	0.74	79.13	85.61	75.53	82.73	0.76	0.83	0.69
Wikipedia+WordNet	58.33	75.00	41.66	66.67	0.47	0.73	0.66	80.57	87.76	77.69	84.89	0.78	0.85	0.73
Wikipedia+Word Embedding	75.00	83.33	66.67	75.00	0.73	0.84	0.68	81.29	85.61	77.69	82.01	0.78	0.83	0.72

Table 3: Topic level labeling performance for text dataset. **MMA: Minimal Match Accuracy; FMA: Full Match Accuracy**

Configuration	High School Mathematics		High School Science	
	P@1	P@2	P@1	P@2
Wikipedia	0.81	0.79	0.71	0.65

Table 4: Precision scores at instruction level for the test dataset

Configuration	High School Mathematics		High School Science	
	P@2	P@5	P@2	P@5
NoExpansion	0.422	0.406	0.641	0.363
Wikipedia	0.597	0.522	0.656	0.556

Table 5: A comparison of the labeling precision achieved on video dataset (transcripts)

5.4 Evaluation: Table 3 shows our system's performance on text dataset. The experiments were performed using different methods for generating instruction lexicons. In all configurations listed in Table 3 educational content based expansion (as explained in section 4.1.3) was used. These lexicons were then used by the content matching method as described in section 4.2 to generate the top scoring labels for each document.

Five different sources were used to generate the lexicons, namely, WordNet, Wikipedia, Word vector embeddings and combinations of Wikipedia with WordNet and Word embeddings. As can be seen the Wikipedia configuration out performs all other configurations in both Mathematics and Science. We also observe that combining word embedding and Wikipedia expansions do not improve the results. One of the primary reasons for this is that the word embeddings were trained on a general corpus and not Mathematics/Science documents.

The topic level full match accuracy @5 is 84.89% for Science and 75.00% for Mathematics and the full match accuracy @10 is 88.48% for Science and 75.00% for Mathematics. Note that MRR is higher in *NoExpansion* configuration. However, the low recall and FMA suggest that there are less number of accurate instructions in top N.

As mentioned previously the data provided to us was labeled at grade, subject, course and topic level. In order to evaluate the precision of our system at the instruction level we manually evaluated a subset of the documents[23] and the instruction labels assigned by our system. Four evaluators were requested to browse the content and the system assigned labels and determine if the instructions labels were valid. We compute the precision scores and report these results in Table 4. We observed that the instruction level precision for Science was lower than that of Mathematics. One of the reasons for the lower precision scores in Science is due to the higher *confusion per topic* in Science as compared to Mathematics, *i.e.*, the number of learning standard instructions per topic for Science is twice that for Mathematics (See Table 1).

Further analysis of the results revealed that a large number of Science documents were *"activity based documents"*, *i.e.*, these documents described general real-world phenomenon so as to teach students how to apply concepts they may have studied in the classroom in the practical world. Thus, the references to science "concepts" were found to be much lower in such documents and therefore our system was less confident while asserting matches. Activity documents from Mathematics, on the other hand, were found to be less generic and contained direct references to mathematical concepts and methods.

For the second dataset of videos, we report the accuracy[24] for two different configurations using the transcribed video lectures: 1) using lexicon that contains only the terms extracted from instructions (without expansion; "*NoExpansion*" in Table 5) and 2) using expanded lexicon ("*Wikipedia*" in Table 5). An improved annotation accuracy obtained both for High School Mathematics and Science transcripts demonstrate the usefulness of the such semantically related words/phrases in the lexicons. However the accuracy on the video dataset is lower due to errors from automatic speech transcription and the spontaneous nature of human speech.

6 Deployment

The system described in this paper has been deployed for a pilot group of teachers at the Gwinnett County Public

[23] 37 Mathematics documents and 78 Science documents were used

[24] Since we did not have gold labels for this dataset, the results were manually inspected. Detailed analysis available in supplementary notes.

Schools, a school district in Gwinnett County, Georgia, USA. Gwinnett County Public Schools (GCPS) has been a three-time finalist for *The Broad Prize for Urban Education* and 2010 winner of The Broad Prize, designating it as one of the nations top urban school districts. This deployment is part of Gwinnett's eCLASS initiative which aims at providing an integrated learning management system to enhance the learning process.

In the current deployment teachers can upload learning content into a content repository. A crawler periodically crawls this repository for newly added content, which is then passed to our algorithm. We package our algorithm as a UIMA annotator and execute it within the IBM Watson Content Analytics (WCA) pipeline. This design has several advantages. WCA provides multiple options for scalability and robustness, thus freeing the annotator developer from such issues. Moreover, our annotator can easily utilize existing WCA services such as *text extraction* – extracting text from different file formats, *e.g.,* MS Word, MS PowerPoint, PDF, Lotus Word and many more, *character set detection*, *UTF8 encoding etc*. After the content has been labeled, if required, the labeled content can be subjected to a *review* process. In this step curriculum experts can look at the labels assigned by the system and provide feedback - incorrect labels can be deleted, or partially correct labels corrected. The automatically labeled content is used by teachers when developing personalized academic intervention for students. Teachers use the academic risk profile of students to assess which AKS instructions or topics student needs help in. Based on this input, teachers assign AKS instruction specific learning content to the students. Currently, the pilot group of teachers include 8 Middle School and 2 High School teachers from different schools in the district.

7 Discussion and Conclusion

In this paper, we described a system that can automatically label educational content with the appropriate learning standard instructions. This was a challenging problem due to the lack of training data, the large number of instructions as well as the ambiguity and terseness of these instructions. We approached this problem by building a lexicon for each learning standard instruction by using available educational content as well as external sources such as Wikipedia, WordNet and word embeddings. Using the lexicon and the hierarchical nature of learning standards we determined the best matching instructions for each document. One key reason that affects performance of our system is that learning standard instructions frequently contain subtle but important differences which our system fails to account for. For instance, our system is unable to distinguish between the following two instructions related to mathematical functions "*determine inverses of linear, quadratic, and power functions*" and "*analyze the graphs of functions and their inverses*"'. To be able to distinguish between these instructions would require deeper semantic analysis of both the learning content as well as instructions which we plan to address in our future work.

References

[1] R. Agrawal, S. Chakraborty, S. Gollapudi, A. Kannan, and K. Kenthapadi. Empowering authors to diagnose comprehension burden in textbooks. In *KDD*, pages 967–975, 2012.

[2] R. Agrawal, S. Gollapudi, A. Kannan, and K. Kenthapadi. Enriching textbooks with images. In *CIKM*, pages 1847–1856, 2011.

[3] R. Agrawal, S. Gollapudi, A. Kannan, and K. Kenthapadi. Identifying enrichment candidates in textbooks. In *WWW (Companion Volume)*, pages 483–492, 2011.

[4] R. L. Cilibrasi and P. M. B. Vitanyi. The google similarity distance. *IEEE TKDE*, 19(3):370–383, Mar. 2007.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[6] D. G. Duffy. *Green's functions with applications*. CRC Press, Abingdon, 2001.

[7] S. Ferrández, A. Toral, Óscar Ferrández, A. Ferrández, and R. M. noz. Applying wikipedia's multilingual knowledge to cross-lingual question answering. pages 352–363. 2007.

[8] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. IJCAI'07, pages 1606–1611, 2007.

[9] P. N. Mendes, M. Jakob, A. Garcia-Silva, and C. Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *(I-Semantics)*, 2011.

[10] R. Mihalcea. Using wikipedia for automatic word sense disambiguation. In C. L. Sidner, T. Schultz, M. Stone, and C. Zhai, editors, *HLT-NAACL*, pages 196–203, 2007.

[11] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *EMNLP 2004*, July 2004.

[12] D. N. Milne, I. H. Witten, and D. M. Nichols. A knowledge-based search engine powered by wikipedia. CIKM '07, pages 445–454, New York, NY, USA, 2007.

[13] D. of Education and G. o. S. A. Children's Services. Choosing and using teaching and learning materials: Guidelines for preschools and schools. 2004.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[15] S. P. Ponzetto. Creating a knowledge base from a collaboratively generated encyclopedia. In *NAACL-HLT 2007 Doctoral Consortium*, 2007.

[16] M. Strube and S. P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. AAAI'06, pages 1419–1424. AAAI Press, 2006.

[17] J. C. Whitmer. Logging on to improve achievement: Evaluating the relationship between use of the learning management system, student characteristics, and academic achievement in a hybrid large enrollment undergraduate course. Phd, 2012.

[18] X. Yang and J. Su. Coreference resolution using semantic relatedness information from automatically discovered patterns. In *ACL*, 2007.