Information Extraction from Textbooks for Physics Knowledge

Arman Bolat
ECE MS '15
abolat@andrew.cmu.edu

Bryan Tan
ECE MS '15
bstan@andrew.cmu.edu

Abstract

The goal of this project is to build a classifier that is able to extract physics knowledge from textbooks. Information extraction of textbooks is a useful problem to tackle because of the many applications that the extracted information could be used for. In the course of this project, textbooks in the PDF will be converted to a text format, and classifiers of different complexity will be tested for accuracy. Training data sets will be created by extracting laws, theorems, definitions, and other knowledge in the text format textbook, and attaching any relevant material, such as context or metadata.

1 Introduction

As more and more information is becoming digitalized, the importance of information extraction through machine learning methods has increased. The goal of this project is to extend machine learning methods to extract physics knowledge from textbooks. In particular, this project will focus on extracting theorems, laws, and definitions relevant to newtonian physics within a textbook. The resulting knowledge will be presented as a glossary, corresponding words and phrases with their respective definitions.

Information extraction from textbooks has a number of practical applications. For one, a problem from a textbook can be analyzed, and the relevant laws, theorems, and equations can be determined by using the knowledge extracted from textbooks, and potentially even a step by step solution for the problem could be provided. The extracted physics knowledge from a given textbook can also show a quick summary of the material covered in the textbook and allow for a classification of the textbook based on depth and breadth of material covered. Even more advanced would to be build a customized textbook through extracted information based on personalized needs.

The goal of this project is to extract physics knowledge from a set of textbooks. The rest of this report is organized as follows: related works to this project will be covered in section 2, the proposed method of creating and training the classifier will be covered in section 3, and preliminary results will be covered in section 4.

2 Related Works

The employment of information extraction and machine learning techniques for textbooks has not seen much work. However, there has been research into the application of knowledge extraction, such as the implementation of some of the example applications mentioned in the introduction. It is possible that the ease of access to resources that provide the information and data necessary (without the use of textbooks specifically) for such research is an explanation for the lack of research into information extraction for textbooks.

[4] explores a method of automatically solving geometry questions. To accomplish this, both the diagram and the accompanying text must be understood. A visual definition of key concepts and terms must be learned, instead of a textual definition as in this project.

[5] attempts to create a tool that is able to assess the comprehension burden, or the difficulty in understanding the textbook material. A property of well-written textbooks is that concepts are presented in sequential order, such that each concept is explained before it occurs in examples or other concepts. A textbook with high comprehension burden thus does not present concepts in sequential order. In order for this to be determined, the occurrences and definition of concepts must be found.

[6] enriches the learning experience by automatically finding relevant images for each chapter. In order to do so, key nouns, phrases, and ideas must be found, as well as their corresponding relevant images. This is similar to this project in that key ideas must be determined. [7] also applies similar techniques to find key concept phrases.

[1] aims to create a concept hierarchy of a textbook. Concepts are identified and extracted by using Wikipedia as a resource. This is very similar to this project - the intermediate goal of extracting concepts is shared, albeit without the use of an external source as a direct reference. The end goal is more complex in the hierarchy extraction paper, as the relationship between concepts must be determined. [2], a system designed at Penn State, extends upon research done in concept hierarchy extraction to automatically create open versions of textbooks by updating the textbook content through online resources such as Wikipedia.

3 Proposed Method

3.1 Training Data Extraction

It is important to generate the training data with the appropriate set of features that would be analyzed by the classifier. The training data used for this project will be extracted from the physics textbooks from [3], and specifically the chapters on the Newtonian physics. For each sentence in the textbooks, a single training data point will be created, which will be represented as a sample object. The sample object will include all the features that could be used by the classifier.

Specifically, each sample object will include:

- the sentence itself
- the sentence before the current sentence
- the sentence following the current sentence
- the type or label of the sample object (theory/law, definition, or none)
- any other relevant information, such as if the sentence is bold or if the sentence contains a mathematical formula or expression

The classifiers will be using not only the content of the sentence but also the context of where the sentence is located as well as the metadata of the sentence. By including a greater number of features, a more general training data set for different classifiers can be created. Since different classifiers might be considering different set of features, it is better to include all possible features which decide if a particular sentence is a piece of physics knowledge or just a regular sentence.

Determining the start and end of each sentence is a difficult problem, and often can be a machine learning problem on its own. In order to avoid creating a tokenizer that is overly specific to the training data, many of the training samples contain multiple sentences, or have some formatting issue. This is an area of improvement that will not be handled very thoroughly and accurately for the sake of the scope of this project.

3.2 Naive Bayes Implementations

This project will begin with implementing a simple Naive Bayes classifier and compare the accuracy by gradually increasing the number of features the classifier considers. Specifically, a classifier implementing a bag-of-words model which only looks at the words used in the sentence will be created. After the completion of the simple classifier, the implementation will be modified to account for the sentence metadata as well. The final Naive Bayes implementation will take into account all features of a sentence in order to classify it. Multiple iterations will be made to identify important features for use in future implementations.

3.3 AdaBoost Implementation

After creating multiple Naive Bayes implementations and identifying important features, an AdaBoost implementation will be created. The implementation will incorporate important features and integrate more weak classifiers beyond Naive Bayes. Once a reasonable set of weak classifiers has been determined, the weights of each classifier can then be optimized and determined.

3.4 ??? Implementation

4 Preliminary Results

For this midpoint report, multiple Naive Bayes implementations were created. The implementations were created through an iterative process, each new implementation improving on the previous iteration by removing unnecessary features and adding relevant features. The resulting classification accuracies can be seen in tables 1 and 2 below.

4.1 Simple Bag-of-words Approach

The initial approach of using a bag-of-words model seemed fairly successful with a training data accuracy of 81.70% and a testing data accuracy of 71.72%. However, upon inspection, it was clear that this initial approach was failing: for the testing data, the accuracy for general (non-knowledge) sentences was 95.58%, but the accuracy for knowledge sentences was only 3.17%. This indicated that the bag-of-words model had a feature set that was too large to effectively classify the sentences, which all had a relatively small number of features, or words, compared to other problems solved by a bag-of-words approach.

The poor results highlighted areas of interest worth tackling to improve the classification accuracy: adding more weight to important and distinguishing phrases or words, and taking into account the other features of the text (namely, the existence of bolded words). The script used to split textbook chapters up into sentences was also further improved to handle edge cases more effectively.

4.2 Bag-of-words Approach with Weighting

After implementing the changes to fix problems pointed out in the initial approach, the new classifier improved in training data knowledge sentence classification accuracy, going from 17.83% to 21.99%, but regressed in testing data knowledge sentence classification accuracy, going from 3.17% to 1.35%. The weighting did not improve the classification accuracy, as the size of the vector of indicator features was still too large in comparison to the size of the sample sentences.

4.3 Naive Bayes with Key Phrases

The results of the second iteration showed that a bag-of-words approach would not work, even after adding weights and accounting for different features. Thus, instead of using a bag-of-words approach, a few key phrases such as ”is defined” and ”is called” were selected and used as the indicator features, along with the presence of bolded words. The result of the simpler classifiers were substantially better: 35.29% classification accuracy for testing data knowledge sentences. After looking through the misclassified examples, it became clear that the training data itself actually had many misclassified samples. After going through the data set to look for any missed definitions, the classification accuracy rose to 76.47%. The features used in this implementation and their respective frequencies can be seen in table 3 below.

Table 1: Comparison of classification accuracies for training set data

Training Data Set Classification Accuracies			
Classifier	Knowledge Sentence	General Sentence	Overall
Simple bag-of-words	17.83%	98.40%	81.70%
Weighted bag-of-words	21.99%	98.01%	86.69%
Naive Bayes with Key Phrases	43.06%	96.68%	95.66%
Naive Bayes, fixed data set	86.11%	96.55%	96.45%

Table 2: Comparison of classification accuracies for testing set data

Testing Data Set Classification Accuracies			
Classifier	Knowledge Sentence	General Sentence	Overall
Simple bag-of-words	3.17%	95.58%	71.72%
Weighted bag-of-words	1.35%	95.41%	81.15%
Naive Bayes with Key Phrases	35.29%	97.80%	96.18%
Naive Bayes, fixed data set	76.47%	97.80%	97.25%

Table 3: Probability of features occuring in knowledge and general sentences

Feature	Probability in Knowledge Sentence	Probability in General Sentence
Contains bold word	51.87%	10.45%
Contains ”is called”	20.32%	0.08%
Contains ”is defined”	8.56%	0.08%
Contains ”are defined”	0.00%	0.05%

References

[1] K. Bowen, B. Brautigam, C. L. Giles, C. Liang, B. Pursel, S. Saul, S. Wang, H. Williams, K. Williams, Z. Wu: BBookX: An Automatic Book Creation Framework. DocEng 2015.

[2] K. Bowen, B. Brautigam, C. L. Giles, C. Liang, B. Pursel, S. Saul, S. Wang, H. Williams, K. Williams, Z. Wu: Concept Hierarchy Extraction from Textbooks. DocEng 2015.

[3] Online Textbooks. *National Council of Educational Research and Training.* Web. 5 Nov. 2015. (`http://www.ncert.nic.in/ncerts/textbook/textbook.htm`)

[4] O. Etzioni, A. Farhadi, H. Hajishirzi, M. J. Seo: Diagram Understanding in Geometry Questions. AAAI 2014.

[5] R. Agrawal, S. Chakraborty, S. Gollapudi, A. Kannan, K. Kenthapadi: Empowering Authors to Diagnose Comprehension Burden in Textbooks. KDD 2012.

[6] R. Agrawal, S. Gollapudi, A. Kannan, K. Kenthapadi: Enriching Textbooks with Images. CIKM 2011.

[7] R. Agrawal, S. Gollapudi, A. Kannan, K. Kenthapadi: Identifying Enrichment Candidates in Textbooks. WWW 2011.