Project Number: DM3 IQP AAGV

Understanding Video Lectures in a Flipped Classroom Setting

A Major Qualifying Project Report

Submitted to the Faculty

Of

Worcester Polytechnic Institute

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

By

Phillip Simon

Monday, April 28th, 2014

Approved by:

Professor Joseph Beck, Project Advisor

Abstract:

Online courses are a more recent development in teaching. This project is designed to analyze how flipping a classroom from a traditional setting where the professor and students are in a room together to a setting where the students watch previously filmed lectures. I parsed words from a file that contains the transcription of the closed captioning. There were two sets of files: one for Coursera videos and one for YouTube videos. The Porter Stemming Algorithm was then used to create a language data model of the parsed words. The purpose of parsing these files was to count the number of occurrences of certain technical words and see how often they are missed. Once the files were parsed and the results were analyzed, they showed that the Coursera transcription is a better tool for online courses. It is clear that the Coursera videos have a higher word count for technical words than the YouTube videos. The analysis of the results of the project will help to better understand if online courses are an effective method of teaching for technical material.

Acknowledgments

I would like to thank Professor Joseph Beck for advising the project and helping throughout my senior as I worked to complete the project. I would also like to thank Gregor Kiczales for allowing the WPI Computer Science department to use his video lecture for their class. Lastly, I would especially like to thank the WPI Computer Science department for providing me with the knowledge over my four years at WPI to complete my project successfully.

Table of Contents

1. Introduction

1.1 Problem Statement

Online courses or some form of online courses, such as WPI's class capture, are used in universities around the world. Some problems with these courses can be poor video quality, poor examples used in lectures, or confusing terminology used. For this project, we will focus more on the language aspect of the videos. The data model provided by the project will highlight which terms are used in explaining technical material during an online lecture as well how often these terms are used. The results of this project will serve as introduction into the analysis of online lectures.

1.2 YouTube vs Coursera

The purposes of using YouTube in this project are: ease of access, being free, and the fact that the online course being used by WPI is uploaded on to this site. YouTube also provides closed captioning for the certain videos which makes it easier to analyze the language as long we have some method to extract the closed captioning.

Coursera has similar aspects to YouTube but is designed for a different purpose. Coursera is "an education platform that partners with top universities and organizations worldwide, to offer courses online for anyone to take, for free."[5] It requires an account for students to be able to access the video lectures. The closed captioning for Coursera is human transcribed so of course it is of better quality than the closed captioning algorithm used by YouTube.

1.3 Goals

The goals of this project are to develop some type of analysis for the online courses being used so the process can be improved upon moving forward. We are also providing an analysis for the closed captioning algorithm used by YouTube. By comparing it against the human transcribed Coursera closed captioning, we can get estimate of the accuracy rating of the algorithm. Alongside the estimated accuracy, we will use a language model that consists of word types and word count to justify any conclusions.

2. Background

2.1 WPI’s Intro to Programming Course

The intro to programming course introduces principles of computation and programming with an emphasis on program design. Topics include design and implementation of programs that use a variety of data structures (such as records, lists, and trees), functions, conditionals, and recursion. Students will be expected to design, implement, and debug programs in a functional programming language[2]. The language used for coding is a functional language called Racket which is a descendent of a language called Scheme. The interactive development environment used is known as DrRacket. All computer science majors are required to take one of two versions of the course: the original intro to programming (CS 1101) or the accelerated intro to programming (CS 1102) which provides an accelerated introduction to design and implementation of functional programs. CS 1102 presents the material from CS 1101 at a fast pace (so students can migrate their programming experience to functional languages), then covers several advanced topics in functional programming (potential topics include macros, lazy programming with streams, and programming with higher-order functions). Students will be expected to design, implement, and debug programs in a functional programming language[2].

2.2 MOOC

MOOC is an acronym for massive open online course. It is designed to be a free course that is easily accessible via the web and allows unlimited participation[3]. MOOC’s are used worldwide. Some features of MOOCs include peer grading and automatically graded assignments. The MOOC used in this project comes from the University of British Columbia. Something similar to a MOOC would be WPI’s class capture. A professor is recorded teaching a class and then the class lecture is posted via WPI’s blackboard portal. Students can then watch the lecture an unlimited amount of times. WPI’s class capture is slightly different because it is a live stream at first. One key distinction is that class capture is not intended to be primarily online while MOOCs are fully online.

2.3 Closed Captioning

Closed captioning (CC) is a process that displays text on a screen in order to provide additional support information. Closed captioning can be used for the hearing impaired as a way to watch shows, videos, movies, etc. Closed captioning provides free text to be analyzed which is key because we are going to repurpose the closed captioning in order to analyze the contents of the videos.

2.4 Parsing

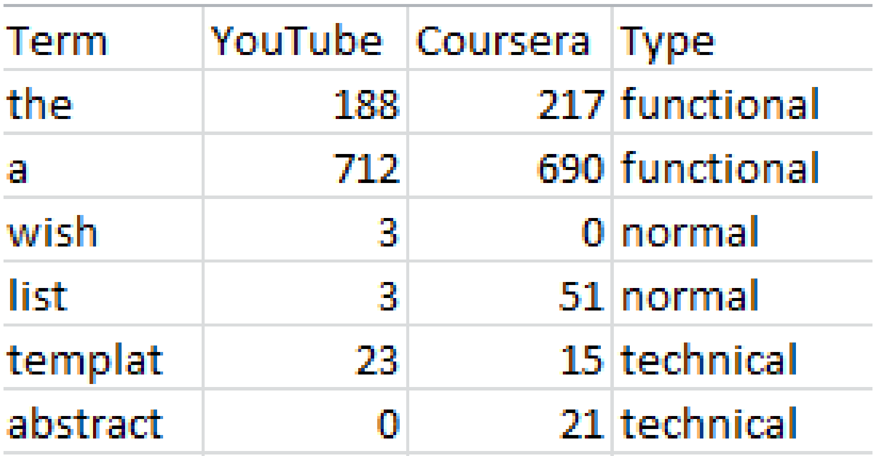
Parsing is a process used to analyze a string of symbols, both in natural language and computer languages. Parsing is typically used to interpret sentences. It can also be used to emphasize “the importance of grammatical divisions such as subject and predicate” (citation).

2.5 Stemming

Stemming is a process used to reduce a word to its root or base. The stem may not “necessarily be the same as the morphological root of the word” (citation). The stem also need not be a valid word. Algorithms that encompassing stemming have been incorporated in computer science since the 1960s. Stemming programs may be referred to as stemming algorithms.

2.6 Porter Stemming Algorithm

The porter stemming algorithm is a process for removing the commoner morphological and inflexional endings from words in English. Its main use is as part of a term normalization process that is usually done when setting up Information Retrieval systems[1]. Some examples of the porter stemming algorithm are below:



Term	YouTube	Coursera	Type
the	188	217	functional
a	712	690	functional
wish	3	0	normal
list	3	51	normal
templat	23	15	technical
abstract	0	21	technical

2.7 Unigram Data Model

A unigram data model is a subcategory of a statistical language model. Statistical language models “assign a probability to a sequence of words by means of a probability distribution” (citation). Language models are used in many applications such as parsing, speech recognition, and information retrieval. Unigram models are the more commonly used models in information retrieval[4].

3. Tools Used

Tools Used	**Purpose**
YouTube	YouTube[6] videos were used in this project because they are easily accessible and have closed captioning available.
Coursera	Coursera[5] is an education platform that partners with top universities and organizations worldwide, to offer courses online for anyone to take, for free[6].
Eclipse	Eclipse was used for this project because it is a free source interactive development environment that has Java capabilities.
Java	Java was used because it is an object oriented language.
Microsoft Excel	Excel sheets were an easy way to keep count of the words and to stem them
Google2SRT	A tool that can download “not embedded” subtitles (Closed Captions – CC) from YouTube/Google Video videos (if those are present) and convert them to a standard format (SubRip – SRT) supported by most video players[5].
Porter Stemming Algorithm	The Porter stemming algorithm (or ‘Porter stemmer’) is a process for removing the commoner morphological and inflexional endings from words in English. Its main use is as part of a term normalisation process that is usually done when setting up Information Retrieval systems[1].

4. Methodology

countLines:

This method counts the number of lines in a file.

stemWords:

This method takes a file, counts the number of lines in that file, and reduces each word to its base then puts those words into an array. The Porter Stemming Algorithm code is adapted from a source listed[1].

parseFiles

This method takes words from the parsed closed captioned files, removes any characters attached to them, and prints them into an excel document. Depending on the set of files (Coursera and YouTube), the word gets printed to a different column in the excel sheet.

The method works by calling the getCol method to obtain an array of strings of the names of the videos which are contained in an excel sheet. Next, the method takes a file from a either the YouTube or Coursera directory that contains the closed captioning files of the YouTube closed captioning and strips them of any undesired characters and prints them to an excel sheet. The method repeats this process for files in a Coursera directory as well. This process is repeated for every video listed in the excel file that contains the video names for both the Coursera and YouTube videos.

getCol

This method takes all the words in a column of an excel sheet and puts them in an array.

unigram:

This method counts the number of occurrences of a word in a sheet in an excel document and builds a unigram model. The method runs for every video listed in the excel document containing the list of Coursera and YouTube videos.

There are three text files containing different types of words. The first of the three types are normal words which are simply: nouns, adjectives, adverbs, etc. The second of the three types are function words. Function words are those that have ambiguous meanings such as: the, from, all, his, etc. The last of the three types are technical words. Technical words are those that relate directly and specifically to the material being covered in the lectures. The stemWords function is then run using each of the three files in order to produce three separate arrays of stemmed words for each category of words. Words that do not appear in the file are unaccounted for.

For each text file, the method compares a word in that file to a word in a file generated by parseFiles that contains the words in the closed captioning files. This is done for both YouTube and Coursera files. The count is kept for the number of times of the word in the text file appears in the file generated by parseFiles. The word, the count, the type of the word, and the accuracy ratings (one for each word and one the aggregates the accuracies) are all printed in an excel file in a unigram data model. To see results, refer to sheet within the zip file labeled unigram_results.

main

The main method calls parseFiles and then calls unigram. A line stating “Program is done” is printed when the main method has finished running.

5. Results

The results of the project are contained in an excel sheet marked unigram_results. The focus of the results was on the technical types of words. The function and normal words provide good data as well but were counted simply because they appear in the language.

After analyzing the results, it is clear that the Coursera video has more occurrences of the technical words than the YouTube. For instance, the word “abstract” appears 21 times in the in one of the Coursera closed captioning but 0 times in one of the YouTube closed captioning and the word “function” appears in in one of the Coursera closed captioning 65 times but only 21 times in one of the YouTube closed captioning. There are some results that where words appear more often in the YouTube closed captioning than the Coursera closed captioning but that is not the majority of the data.

The accuracies were calculated by comparing the number of occurrences of a word in a YouTube file with that of a Coursera file. The Coursera file is used as a base because it is the human transcribed closed captioning which leaves less room for error. I divided the number of occurrences in a YouTube file by the number of occurrences in a Coursera file. These results are also in the file marked unigram_results.

When the accuracies are compared, 88.3% of the parsed YouTube files have accuracy ratings over 100%. This means that parsed word occurred more times in the YouTube video than in the Coursera video. Although the data seems to not point towards Coursera it actually does. These accuracies were not separate by types of words so the focus of the project, the technical words, is lost in the larger sample size. The higher counts originate in the normal and function words more than the technical words. This agrees with the idea that the YouTube algorithm has errors and changes words of a technical nature (i.e. abstraction) to something such as “a bat some”. The three words that the technical word could possibly be changed to are categorized under either the normal or technical. This means the word counts for these types increase while the technical count remains stagnant thus causing the large discrepancies in the accuracy ratings.

Words that are longer and have more syllables were the ones that were incorrectly translated and misspelled especially, the technical words. In some cases, words such “now” or even some prepositions were also incorrectly translated. The reason for this is most likely the speech patterns of the speaker. People may slur or the algorithm used by YouTube may not have picked up on an accent and this causes random errors.

6. Conclusion

It appears that Coursera closed captioning is better for technical videos than YouTube's algorithm. This conclusion was the expected outcome. Coursera's hand transcribed closed captioning is an almost perfect match to the video lectures. The reason for the discrepancies noticed in the YouTube video can be from a number of factors. Some of those factors are: the lecturer is talking to fast, the lecturer has an accent, or the lecturer is using words that YouTube's algorithm is not designed to catch. For this project, the last of these factors is probably the most likely.

Words such as abstract and function are probably not typical words outside of academic fields. Although I am unsure of how YouTube's algorithm works, I know that some speech recognition algorithms use common speech patterns to guess the next word such as the android texting algorithm that tries to preselect the next word in text messages based on previous conversations held by the user.

In summary, Coursera's closed captioning model is a better one for technical videos. YouTube's algorithm is close but should not be considered close enough for a college course. Many of the words are mixed and confused for other words. Watching only 1 minute of the YouTube's version of the lectures with the closed captioning turned on will highlight this fact. Moving forward, human transcribed closed captioning is a better model for massive open online courses.

It is important to note that using closed captioning to infer the content of the videos (for doing searching, for example), would fail for programming courses, or any other courses that consists of a technical vocabulary.

7. Future Works

There are many additions that can be made to the project. One of the original ideas was to analyze the comments, likes, and thumbs up and thumbs down count from the YouTube videos and cross examine those results with the grades of the students in the courses. This idea never came to fruition due to some features being disabled by the owner of the YouTube videos. One idea for future work would be to take the results of this project and cross examine them with the grades from the students in the course. The videos where the number of technical words is noticeably greater in the Coursers closed captioning would be the key videos. If in the case where students did poorly for this section of the course, it can be analyzed to see if there is any correlation with the number of misses in the YouTube video.

Another course of action would be to have one lecture watch only the YouTube videos and another lecture watch only the Coursera videos. The students grades can then be analyzed throughout the lectured and compared to see which class performed better: the YouTube class or the Coursera class. This may be slightly tricky as there are many factors that go into college courses such as what time in the year are they taken, who is the professor, the sample size, etc.

The results of the project can also be improved. Currently, the three types of word files are words input by a user. To improve the project, the types of words should be taken from the videos. This would produce truer and better results.

8. Sources

1. http://tartarus.org/martin/PorterStemmer/
2. http://www.wpi.edu/Images/CMS/Pubs-Catalogs-Ugrad/UGCat14-15FinalWEB.pdf
3. http://www.washingtonpost.com/blogs/campus-overload/post/what-in-the-world-is-a-mooc/2012/09/24/50751600-0662-11e2-858a-5311df86ab04_blog.html
4. http://www.phontron.com/slides/nlp-programming-en-01-unigramlm.pdf
   http://google2srt.sourceforge.net/en/
5. https://www.coursera.org/
6. http://www.youtube.com/