AC 2012-3307: COMPUTATIONAL METHOD FOR IDENTIFYING INACCESSIBLE VOCABULARY IN ENGINEERING EDUCATIONAL MATERIALS

Mr. Chirag Variawa, University of Toronto

Chirag Variawa is a Ph.D. candidate in the Department of Mechanical and Industrial Engineering at the University of Toronto. He earned his B.A.Sc. in materials science engineering in 2009 from the same institution. His multi-disciplinary research uses principles from artificial intelligence, computational linguistics, higher-education, and aspects of neuroscience to investigate inclusive design of engineering learning environments.

Dr. Susan McCahan, University of Toronto

Susan McCahan is a professor in the Department of Mechanical and Industrial Engineering at the University of Toronto. In addition, she is currently the Vice Dean, Undergraduate for the Faculty of Applied Science and Engineering. She received her B.Sc. from Cornell University (1985), and M.S. (1989) and Ph.D. (1992) degrees from Rensselaer Polytechnic Institute in mechanical engineering.
Computational Method for Identifying Inaccessible Vocabulary in Engineering Educational Materials

Introduction

Instructors often face the challenge of making students feel more included in the classroom, especially in freshmen classes of engineering. In the freshman classroom, instructors are more often finding that their students are departing from the “traditional” homogenous demographics of engineering in the past. Engineering classrooms have broader representation from all cultural and socio-economic backgrounds and even greater variance in approaches to learning leading to greater diversity. This increased diversity among students may also lead to barriers that impede accessibility to learning and as a result, inclusivity.

Universal Instructional Design (UID) is a pedagogical philosophy that has emerged in the field of higher education research. It aims to increase accessibility to learning materials. The core concept of UID, universal design, is from civil engineering and calls for increasing accessibility to physical structures by incorporating accessibility as a priority in the design process. Applied to education, this design philosophy attempts to “make instruction accessible to the greatest extent for the largest number of people possible”.\(^1\) The literature on this subject suggests the use of seven principles that guide teachers to create accessible learning material by increasing clarity, transparency, flexibility and usability of instruction. However, the use of UID has not been rigorously examined within the context of engineering education as a tool to create more inclusive learning environments. The premise of our study is to use a UID-inspired approach to make engineering education more accessible to the greatest diversity of students possible to the greatest degree; we hope to maximize accessibility to engineering course material with the goal of making learning environments more inclusive.

One particular learning barrier that our students face is inaccessible language. In engineering, we generally encourage the development of a robust engineering vocabulary to help students develop as professionals. However, a critical look at the language we use in the classroom may raise questions about the accessibility to course material when we begin to use vocabulary that is external to the corpus of language our diverse students bring with them. So, while we claim to promote professional language development, we may inadvertently create a less than inclusive environment for our students. Specifically, when we use uncommon language that is neither discipline-specific nor explicitly taught, we are creating an environment that is biased towards learners that have the same corpus of vocabulary as the instructor – this is particularly evident when colloquial or cultural vocabulary is used in the classroom. In particular, our study attempts to investigate the communication barriers that exist between instructors and diverse student populations within the context of engineering education.
Final examinations are a standardized artifact of the engineering classroom whose purpose is to assess the student’s understanding of course material. As a summative evaluation technique, the exam probes the students mastery of what was taught in class and how well they can apply this material to answer the provided questions. In many cases, instructors attempt to provide realistic problems using course material that is contextualized within a particular setting under given conditions. The goal of providing such authentic, contextualized questions is to mimic a “real world” situation where engineering knowledge can be applied. In doing so, the instructors may inadvertently test the student’s cultural knowledge of the contextualized environment rather than testing course-specific instruction exclusively. Specifically, the vocabulary used in creating authentic engineering problems on such assessments may cause an inaccessible and non-inclusive environment for some students. For example when we contextualize an engineering problem by using societal references that we assume are “widely known”, some students may find the vocabulary to be unclear or foreign; the challenge of the question becomes trying to understand that vocabulary, rather than using engineering knowledge to answer the question. By extension, the use of such vocabulary may yield an invalid performance assessment because the final exam is no longer testing what it purports to test: a student’s understanding and mastery of course material. The use of accessible vocabulary while maintaining an authentic assessment environment may lead to final exams of higher quality that may promote robust vocabulary development as well. In our investigation, we aim to maximize accessible vocabulary while leaving the course and discipline-specific vocabulary as-is so that the integrity of the course material is unaffected by creating a more inclusive exam.

The goal of the study is to develop a computational approach to identify potentially inaccessible vocabulary and bring it to the attention of the instructor, while ignoring engineering-specific vocabulary explicitly taught in a course. In the process of doing so, we must examine the language currently used on engineering final exams to determine if there is a method to distinguish course-specific or discipline-specific language from the rest. Specifically, to find inaccessible vocabulary we are going to find words that are course-specific and then, in future work, take the complement as a possible source of inaccessible vocabulary. This is to ensure we are not inadvertently labeling course-specific words as being inaccessible. This paper focuses on this specific aspect of the larger study on inaccessible language. In particular we find that literature from the field of linguistics, computer science, computational linguistics, higher education and even some work from culture studies suggest tools for this type of work. While there are some limited corpora of vocabulary that are discipline specific, our intention is to establish a dataset using language relevant to engineering education at a typical North American engineering institution. We begin this particular component of the larger study by trying to find course-specific language.

Computational linguistics suggests the use of keyword-generation algorithms to establish a corpus of words that are characteristic of a particular piece of work. In our case, we can use this approach to develop a quantitatively-described hierarchy of potential keywords for final
exams used in engineering. The hypothesis is that we can then compare the keywords found in one exam to others, as a group or individually, to inform trends that describe the use of language in engineering education. The powerful use of keyword comparison can help us see how language changes across disciplines, how vocabulary of freshman classes differ from upper years, and so on. The aim is to compare keywords of different test cases to suggest a usable approach for determining course-specific language on final exams used in engineering. Although the use of keyword-generating algorithms is not the only approach to this issue, and neither is it exhaustive at that, the goal is to explore the intersection of computational linguistics and engineering education and see if we can use a tool from one discipline to help make the other more accessible.

Methodology

This paper focusses on classifying the vocabulary found on engineering final exams using a quantitative computational method. The work presented here builds on earlier studies that examined students’ self-assessment of vocabulary understanding, and word frequency analysis in engineering exams.\(^4,5\) Our methodology makes use of the results from these earlier works.

Specifically, this investigation compares keywords from different exams to one another with the goal of finding a method where course-specific terms are clustered and segregated from other vocabulary. Earlier studies found that word frequency analysis on-its-own was not sufficient for this purpose.\(^4\) The field of computational linguistics and computer science suggest several potential approaches that can be used to generate and compare keywords among groups of documents that are an improvement over word frequency analysis. Although the algorithms used in these approaches are often different, the goal remains to find words that effectively characterize the vocabulary used in a particular document.

One such approach is the Term-Frequency Inverse-Document Frequency (TFIDF) algorithm. This technique compares the frequency of words in a single document (TF) to the vocabulary used in a set of documents. The mathematical formula for TFIDF is:

\[
TFIDF = TF \times IDF
\]

where

\[
TF = \left( \frac{\text{# of occurrences}}{\text{total # of words}} \right)_{\text{in a single document}}
\]

and

\[
IDF = \log \left( \frac{\text{# of documents}}{\text{# of documents containing the word}} \right)_{\text{in a set of documents}}
\]
The IDF is a measure of how important a particular term is within a set of documents, and is calculated by dividing the total number of documents by the number of documents in the set which contain that term, and then takes the logarithm of the quotient. The TFIDF formula assigns a score to each word in the test document.

As an information retrieval tool, TFIDF is effective in finding characteristic words in a document when multiple documents are compared. The existing literature about TFIDF describes it as a technique used to classify documents based on keywords and modifiers. Specifically, TFIDF is used to describe documents using hierarchical subclasses, or other creative methods where the algorithm is used repeatedly per subclass. For example, a keyword for a computer hardware part might be described as “comp.sys.ibm.pc.hardware”, and this is an example of where the algorithm is used repeatedly in a loop within each subclass. From a computational perspective this puts a large load on the processor(s), and as such is quite intensive, but the results are generally accurate. Although we are not using a repeated looping method within subclasses for this study, we can still use the TFIDF to provide insight about words that are diagnostic for a document. Further, since we are using less processing loops, we are less concerned about computational overload and limitations on available computer memory.

Research shows that there are several other methods used to generate keywords used in computational linguistics and computer science. Research also shows tangential algorithms that can be used to classify documents and their strengths and weaknesses. In particular, the TFIDF method seems to be recognized as a validated approach to finding keywords in documents according to existing literature. The existing work also demonstrates approaches for accurately describing the magnitude of word-grouping behaviour using more sophisticated mathematical and statistical techniques including investigating the bottleneck method for disambiguation and greedy heuristic analysis approaches. In particular, the existing literature focuses on the algorithm itself and ways to describe documents using subclasses (for the sake of information retrieval) but not how to classify individual vocabulary in a study context similar to our work.

A higher TFIDF score means that the particular word being examined is diagnostic of that particular document and a low TFIDF score means that the word is not a keyword for the document. This algorithm is particularly effective because it tends to filter out common terms in a set of documents; the algorithm is able to filter out terms based on the documents being compared. As such, we can use the TFIDF method to potentially find words that are diagnostic of a particular exam and filter out words that are commonly used on engineering exams in general.

We use a repository of electronically available final exams in the Faculty of Applied Science and Engineering at the University of Toronto for this research. The total number of usable exams is 2254, with a final total of over 2,300,000 English words being examined in our work. The dataset spans the last ten years, covers all departments in the Faculty and is robust enough to
cover over 98% of all words on each exam used including captions on figures, etc. Words that contain numbers or foreign characters are discarded from the study: only words that contain ASCII characters from 65-122 (inclusive) are taken into account.

This phase of our study begins by converting the approximately 3000 electronically-available engineering exams from an image-PDF format to plain text using Optical Character Recognition (using Adobe Acrobat®). Then, all of the words from each exam are extracted and “scrubbed” to remove words that contain numbers or foreign characters. This word set is then automatically placed in its own new text-only file that is cross-referenced with the course designation, year, and discipline. Then, the user can select which of these files to compare against which group of exams to develop TFIDF values for that word set. The data is then exported and further scrubbed using a Microsoft Excel® macro to remove duplicates. The TFIDF processing code was created by the investigators. The investigators used four test cases for the TFIDF comparisons and these are detailed in the subsequent sections.

Results

The TFIDF computational method identifies keywords in a particular document by comparing it to a specified group of other documents. For this phase of the study, we compared one particular exam to four different groups of exams from the same institution to generate four case scenarios. The exam chosen as the “control” is from the Department of Materials Science and Engineering (MSE), for a third-year undergraduate course called ‘Mechanical Behavior of Materials’ held in 2009 (MSE316). This exam was chosen because the author is already familiar with the course-specific vocabulary and the course is very typical of a technical engineering course. We tried comparing this exam against 4 different document sets:

1. This exam is compared against all electronically available exams;
2. all exams created in the year 2009;
3. compared against all exams from the same department (MSE);
4. then compared against all exams from a different engineering department (Civil engineering).

In all cases, each word in the control exam is given a TFIDF score and ranked based on decreasing score. As mentioned, a higher TFIDF value means that the specific word has a higher probability of being a diagnostic word for that document.
Table 1 - Shows TFIDF values in decreasing order for exam words across four test cases; bolded words indicate that the word is potentially course or discipline-specific

<table>
<thead>
<tr>
<th>Rank</th>
<th>Case 1: MSE316-2009 vs. All Exams</th>
<th>Case 2: MSE316-2009 vs. All Exams in 2009</th>
<th>Case 3: MSE316-2009 vs. All MSE exams</th>
<th>Case 4: MSE316-2009 vs. All CIV exams</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Word</td>
<td>TFIDF score</td>
<td>Word</td>
<td>TFIDF score</td>
</tr>
<tr>
<td>1</td>
<td>dislocation</td>
<td>0.046887</td>
<td>dislocation</td>
<td>0.054035</td>
</tr>
<tr>
<td>2</td>
<td>stress</td>
<td>0.026217</td>
<td>stress</td>
<td>0.026728</td>
</tr>
<tr>
<td>3</td>
<td>ofll</td>
<td>0.023793</td>
<td>dislocations</td>
<td>0.022067</td>
</tr>
<tr>
<td>4</td>
<td>gb</td>
<td>0.020656</td>
<td>gb</td>
<td>0.021895</td>
</tr>
<tr>
<td>5</td>
<td>dislocations</td>
<td>0.018314</td>
<td>grain</td>
<td>0.018767</td>
</tr>
<tr>
<td>6</td>
<td>cry</td>
<td>0.017527</td>
<td>gb</td>
<td>0.01759</td>
</tr>
<tr>
<td>7</td>
<td>creep</td>
<td>0.017253</td>
<td>cry</td>
<td>0.01718</td>
</tr>
<tr>
<td>8</td>
<td>grain</td>
<td>0.017148</td>
<td>slip</td>
<td>0.01653</td>
</tr>
<tr>
<td>9</td>
<td>subgrain</td>
<td>0.016841</td>
<td>crystal</td>
<td>0.015391</td>
</tr>
<tr>
<td>10</td>
<td>partials</td>
<td>0.015869</td>
<td>deformation</td>
<td>0.014911</td>
</tr>
<tr>
<td></td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>20</td>
<td>material</td>
<td>0.011816</td>
<td>tensile</td>
<td>0.011252</td>
</tr>
<tr>
<td>30</td>
<td>formation</td>
<td>0.009302</td>
<td>strain</td>
<td>0.008904</td>
</tr>
<tr>
<td>40</td>
<td>continuation</td>
<td>0.008237</td>
<td>softening</td>
<td>0.007719</td>
</tr>
<tr>
<td>50</td>
<td>onset</td>
<td>0.007397</td>
<td>agb</td>
<td>0.007072</td>
</tr>
<tr>
<td>60</td>
<td>test</td>
<td>0.006167</td>
<td>theoretical</td>
<td>0.005505</td>
</tr>
<tr>
<td>70</td>
<td>crystalline</td>
<td>0.005691</td>
<td>yx</td>
<td>0.004857</td>
</tr>
<tr>
<td></td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>100</td>
<td>matrix</td>
<td>0.004542</td>
<td>lowering</td>
<td>0.004413</td>
</tr>
<tr>
<td>200</td>
<td>offset</td>
<td>0.002845</td>
<td>atoms</td>
<td>0.002752</td>
</tr>
<tr>
<td>300</td>
<td>fx</td>
<td>0.00157</td>
<td>lb</td>
<td>0.001529</td>
</tr>
<tr>
<td>400</td>
<td>so</td>
<td>0.000285</td>
<td>what</td>
<td>0.000403</td>
</tr>
</tbody>
</table>

The table above shows a small sample of the 457 words from the MSE316 exam, along with the corresponding TFIDF value for each word ranked in decreasing order. The table also shows course or discipline-specific words, and these are highlighted in bold for each case. These words are explicitly taught in class and are expected to be known for this exam. For instance, the instructor lectures on crystal lattice structures and mentions deformation behavior along slip planes: these words would then become part of the “course-specific” corpus of this class. It should be noted we were hoping that this method would put course-specific words near the top of the list, as this is indicative of a tool that can successfully isolate these terms computationally.

The data shows that Case 2 has the highest number of course-specific terms near the top of the TFIDF list. Specifically, eight out of ten highest-ranked TFIDF words are course-specific with the farthest outlier ranked at 20. This means that Case 2 emerges as potentially the best approach when compared to the other three cases. However, to confirm this conclusion further testing with other exams may be required. The data shows that Case 1 has seven of its top ten words as
however we begin to see some clustering behavior of course-specific words near the rank of 70. By “clustering” this means that some course-specific terms appear together in a group. This low-end clustering behavior is also seen with Case 4, but the clustering behavior is around both 20 and 70. This type of behavior indicates that the computational method is leaving some course-specific words scattered throughout the list when we begin to compare an exam to the entire repository of exams or to just one other discipline. From the data we also see that comparing MSE316 to all exams from MSE the course-specific terms are often clustered near the top of the list, but the top ten also contains other words – this shows that comparing an exam to the exams of the same discipline does not yield particularly accurate results either. In particular, with the data computed and the particular scope of this study, it appears that comparing one exam to all exams in the same particular year shows the highest number of course-specific terms among the top ten with minimal outliers elsewhere.

Discussion

The data shows that the TFIDF method appears to be gathering course-specific terms at the top of the word lists. In computer science, the purpose of the TFIDF method is to find words characteristic of a particular document when comparing to a group of other documents. In our case, TFIDF appears to have done that successfully for all test cases, with some limitations. However, we see that this trend does have its limitations, based on the test cases reported in this study. First, not all course-specific terms are flagged by this algorithm. We see that some course-specific terms still appear at other locations on the word lists, but the majority of them appear near the top. This tells us that the computational method can be improved to become more accurate.

Another important limitation of this study is that the data are not sufficient to demonstrate a reliable process. We have compared just one exam to different groups of exams to generate our sample test cases. As an exploratory test case, it shows that the TFIDF method does gather course-specific terms near the top of the word lists. However, it does not show if this is applicable for other exams when other courses are used instead of MSE316-2009. As such, the need for expanding this study to include exams other than MSE316-2009 is necessary to better evaluate the reliability of the TFIDF method.

Another limitation of this study is that the TFIDF method only analyzes single words and not phrases. Specifically, the program is blind to the fact that multiple words appearing in a particular sequence can be classified as a course-specific phrase. For example, the term “face centered cubic” is treated as three words instead of as one specific phrase by the TFIDF algorithm while “FCC” is treated as one word even though both are course-specific to MSE316-2009. This limitation needs to be addressed.

It should also be noted that the top ten words represent only the top 2% of words for this particular exam, and it does not provide particularly conclusive evidence about the applicability
of the computational process; we might need to examine more than just the top ten words. However, the purpose for this exploratory investigation is to compare cases to one another and determine which of these comparison types holds the most promise for being computationally effective – future work is needed to extend the scope of investigating these lists to include a more diverse set of exams.

Based on the literature in computer science, TFIDF is quite effective at making keyword lists that are characteristic of a particular document when compared to groups of other documents. Our study uses the TFIDF tool and tests it in the context of engineering education. Specifically, it extends the work found in the literature and shows that TFIDF is potentially able to find characteristic words on engineering exam documents as well. As instructors, we see from the data that “characteristic words” of exams are synonymous with course-specific terms. By extension, it appears that the algorithm originally developed in computer science can be used in engineering education for future work in the area of vocabulary analysis of assessment instruments.

The data shows that comparing one engineering exam to all engineering exams in the same year results in course-specific vocabulary having high TFIDF scores. In particular, the course-specific terms tend to demonstrate grouping behavior near the top of the list rather than being dispersed throughout the ranked vocabulary wordlist. The literature does not define why this behavior exists because such work has not been performed in this context before. Therefore, it is difficult to accurately describe the reasoning behind such behavior based on existing work. However, TFIDF is well known as a method to identify words that characterize a single document with respect to a body of documents. So from this perspective it makes sense that comparing to a wide range of other exams from the same year (i.e. use the same slang, or same current English colloquialisms) would allow for the identification of keywords that differentiate this exam from others.

While the data set presented here is limited, this study begins to offer insight into the development of more accessible course material for engineering. The vocabulary analysis process can potentially categorize words as course-specific, common, or into a third category; uncommon and not course-specific. It is this third category of terms that may pose accessibility challenges for students and can be brought to the attention of the instructor. Additional clarification or other assistance (such as a visual aid) can be used to improve the accessibility of the text. As instructors begin to create performance assessments for their students that contain more accessible language, they are also promoting a more valid assessment with the ability to contextualize material for more authentic questions. The premise of inclusive learning environments in engineering education is critical for the transition of students into post-secondary education as the goal is to develop a bias-free education system for the diverse nature of the learning population.
Future Work

The study attempts to investigate the language used in engineering education to promote inclusive learning environments. This particular investigation looks at the vocabulary used in engineering examinations and explores a computational method to distinguish course-specific vocabulary. By analyzing four cases, it is found that comparing a particular exam to all exams in that particular year is a promising approach for the computational approach to further investigate. Further work in this area will include a more diverse set of test cases and exams we compare to one another. In addition, we need to address the limitations of this study before we can conclusively state the effectiveness of using TFIDF as a method of finding course-specific terms. This exploratory study does however provide insight into the use of a computational tool to investigate the vocabulary used in engineering education. Future work in this area would help maintain the integrity of course-specific vocabulary on engineering examinations while we explore other ways to identify inaccessible language in engineering education.

References