An Automated Approach for Finding Course-specific Vocabulary

Mr. Chirag Variawa, University of Toronto

Chirag Variawa is a Ph.D. candidate in Industrial Engineering at the University of Toronto. His research is in using artificial intelligence to maximize the accessibility of language used in engineering education instructional materials. His work on the Board of Governors at the University of Toronto further serves to improve accessibility for all members of the university community.

Dr. Susan McCahan, University of Toronto

Dr. Susan McCahan is vice-dean, Undergraduate, and is a professor in the Department of Mechanical and Industrial Engineering in the Faculty of Applied Science and Engineering at the University of Toronto.

Prof. Mark Chignell, University of Toronto

Mark Chignell is a professor of Mechanical and Industrial Engineering at the University of Toronto where he has been on the faculty since 1990. Prior to that he was an assistant professor in Industrial and Systems Engineering at the University of Southern California from 1984 to 1990. He earned the Ph.D. in Psychology from the University of Canterbury in New Zealand in 1981, and an M.S. in Industrial and Systems Engineering from Ohio State in 1984. Mark is currently president of Vocalage Inc., a University of Toronto spinoff company, director of the Interactive Media Lab, and a visiting scientist at both the IBM Centre for Advanced Studies and Keio University in Japan.

An Automated Approach for Finding Course-specific Vocabulary

Introduction

This study introduces methods to increase the transparency of specific learning outcomes expected in an engineering course. Freshman engineering students face the challenge of absorbing a new set of terminology associated with their discipline, while also adjusting to the university environment. As they learn, students may inaccurately grasp course concepts due to lack of understanding of domain vocabulary. One strategy for addressing this problem is to make design of vocabulary part of overall course design. This requires explicitly identifying the vocabulary that students need to learn in the course of their studies. Proper specification of vocabulary is likely to be particularly important in introductory courses that form the foundation of engineering disciplines.

Identifying discipline-specific words helps instructors establish clear expectations of required vocabulary knowledge, while building robust technical communication skills. If students have a clear understanding of required vocabulary, then instructors will be able to develop higher quality teaching and assessment material. As a result, instructors will likely be confident in the knowledge that students will not be handicapped by language usages that are neither part of their cultural background nor inherent to the course or domain. At the freshman level, vocabulary lists might be developed that highlight terms pertinent to the field. However, language has a fluidity that cannot be accurately captured by static wordlists that do not accommodate context. However, manual updating of word lists each year is an additional (and probably unwelcome) burden on instructors.

In this study, the authors investigate an efficient and semi-automated approach for developing up-to-date course-specific vocabulary lists while requiring minimal contextual input from the instructor. The focus of this research is on engineering course material with the ultimate goal being to help freshman students adjust to new terminology in their field of study, without increasing the workload of teaching faculty. The goal is to find a computational method that can be used to create a software tool which automatically compiles a unique list of course-specific vocabulary for the instructor.

Literature

There are several approaches that can characterize language in document text. The fields of research that contain literature in this area include education, linguistics, computational linguistics, industrial engineering, as well as several others. Specifically, literature in the field of education pertinent to the study ranges from the Plain Language Movement to language acquisition and English as a Second Language research.[1-3] These approaches aim to simplify

language structure and vocabulary to maximize accessibility.[2,4] Further, research in this area focuses on the relationship of words to generate meaning and on how language development is affected by choice of vocabulary.[1,2,4] The research informs an understanding of the importance of language development and the motivation to use accessible, yet immersive, language in learning environments.[4-6] While this is important in public documents (i.e. tax forms) overly simplifying language does not suit the purposes of the engineering classroom. Engineering students need to develop robust vocabulary ability that is authentic to their field and will stand them in good stead when they take up their careers.

The fields of linguistics and computational linguistics are particularly broad, and they study language from several perspectives. Some approaches examine the development of language, symbolic meaning, and the structure of words.[7, 8] Other approaches look at differences between languages and their evolution over time.[9, 10] Computational approaches tend to convert the complexities of language into bits of information that can be quantified, classified and analyzed as packets of data. More specifically, this field investigates algorithms and tools that can measure and quantify vocabulary.[11-14] Some algorithms and methods are broadly applicable across a range of linguistic fields. Classification algorithms use various corpora to organize words into hierarchical structures. Word hierarchies can also be elaborated with syntactic and semantic information to create a comprehensive representation of knowledge about the English lexicon. The most extensive tool of this type is WordNet, a database that contains words and their synonyms classified by relevance and similarities with each other, referred to as synsets.[14] This approach forms lexical repositories of words that can be used to analyze the relationship of sets of words with one another.[12-14] An advantage of using this approach is to develop a common lexical database of words pertinent to a field, but a disadvantage is that this repository continues to grow in size without a structured ability to prune words over time.[12-14] Further tools like wordnet are designed to deal with the vocabulary of language in general and are less useful in organizing and explaining domain specific vocabularies. Thus an approach is needed that generates manageable, and domain specific, vocabulary lists.

Another area of research in computational linguistics is keyword-generation (automated indexing), and the development and application of algorithms to statistically determine the characteristics of words based on frequency. Some of the more frequently used approaches in this area include frequency analysis of words, keyword generation algorithms and artificial intelligence methods. Frequency analysis of words is an approach that attempts to correlate the frequency of use of a word in a target document to a corpus of English, or specific discipline.[14, 16, 17] Prior work in this area shows that this method is useful to understand natural language, and can be used with algorithms that are supplemented by statistical theory, like Zipf's Law[16-18] Another approach is Latent Semantic Indexing, for example, which uses singular value decomposition (analogous to principal components analysis on large, sparse matrices) to identify associations between words based on their context, and which can also be used to generate data about the meaning of words when used in similar contexts.[11, 13, 19] Multiword Expressions are

another set of approaches that investigate the meaning of words based on lexeme analysis,[11, 12, 20] Specifically, multiword helps us understand that words can change meaning based on how they are used in a sentence, and this can inform a keyword generation procedure.[11, 20] This general field of language analysis using computational approaches falls under a category of computer science and engineering defined as artificial intelligence (AI) because words are being translated from human vocabulary to computer-based computations, and then to a form that allows us to better understand its characteristics.

TF-IDF Approach

A preliminary analysis shows that a simplistic approach such as frequency-analysis on its own is inadequate to determine characteristic terms to a piece of text.[21- 23] Frequency-analysis alone only generates information about how often certain words are used. This information is not particularly useful to this study because characteristic words on engineering documents are not necessarily those that appear most frequently. Specifically, literature shows that commonly occurring words are indicative of natural language, and not a measure of diagnostic vocabulary to an input document.[13, 15, 18] As such, a more advanced approach is required: one that can characterize diagnostic words in documents, while requiring minimal contextual data other than the documents themselves, and one that can handle large sets of words and documents.

Term Frequency Inverse-Document Frequency (TF-IDF) analysis is a well known index method in information retrieval, and it is used to characterize vocabulary across sets of documents.[11, 13, 15, 18, 23-25]

The TF-IDF 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:

$$\mathrm{TFIDF} = \mathrm{TF} \times \mathrm{IDF}$$

where

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

and

$$\mathrm{IDF} = \log\left(\frac{\#\ \text{of documents}}{\#\ \text{of documents containing the word}}\right)_{\text{in a set of comparitor documents}}$$

There are two main parts to the TF-IDF algorithm, and they work together to assign a score for each word in the target document. The TF counts the number of occurrences of a particular word, and divides that number by the total number of words in the target document, which is a simple measure of frequency. 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 its logarithm. The TFIDF formula multiplies these together and attaches the resulting score to each unique word in the target document. 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 approach allows us to differentiate common vocabulary from words that are characteristic to the target document, like course-specific language in this case. This approach works reasonably well for one target document (e.g. the final exam in a course), but does not do a good job at differentiating course-specific or discipline-specific vocabulary from words that appear infrequently in natural language. For example, it might identify both "enthalpy" and "circulation" as characteristic words on a thermodynamics exam because these both are likely to occur rarely in a comparator document set. But a thermodynamics instructor would easily recognize "enthalpy" as being key disciplinary jargon, and "circulation" as not specific to the discipline.

Interpreting the mechanics of the approach

The authors propose a method that should improve the effectiveness of the TF-IDF algorithm for the purposes of investigating the language used by engineering documents. Specifically, we suggest developing two TF-IDF scores for each word in a document and then calculating their difference to maximize accuracy in finding course-specific vocabulary. The approach would be to use two different contexts for the same document to calculate two TF-IDF scores:

1. Compare a target document to all documents in engineering, minus those that are in the same discipline. This should highlight terms that are characteristic of the discipline.
2. Compare a target document to all documents within the same discipline as that input document. This should highlight terms that are characteristic to that course.

This method generates two wordlists – one from each context listed above. These lists can then be sorted alphabetically while subtracting the TF-IDF scores for context #2 from context #1. This produces a list where words that are both course-specific and discipline-specific are given a high score, whereas all other types of words are given a lower score. This modified use of the TF-IDF algorithm can be expressed as:

$$\text{TFIDF} = \text{TF} \times \text{IDF}$$

$$TFIDF_{mod} = TFIDF_1 - TFIDF_2$$
$$TFIDF_{mod} = TF(IDF_1 - IDF_2)$$

Where, subscripts 1 and 2 would represent context #1 and context #2 respectively, and TF would be identical for both because input exam is the same.

And where,

D_E = # documents in engineering, minus discipline
$D_{E,W}$ = # documents in engineering, minus discipline, containing the same word
D_D = # documents in discipline
$D_{D,W}$ = # documents in discipline containing the same word

Condensing and simplifying:

$$IDF_{mod} = \log\left(\frac{D_E \bullet D_{D,W}}{D_{E,W} \bullet D_D}\right)$$

Using this approach, words can be characterized based on how prevalent they are in engineering and in their respective discipline.

- if $D_E \bullet D_{D,W}$ is large, because there are lots of documents in the discipline containing the same word, then it causes the numerator to increase, resulting in the IDF_{mod} becoming larger, which then amplifies the $TFIDF_{mod}$ value; this means that the word frequently occurs in the discipline but not necessarily all of engineering, which implies it is likely discipline-specific
- conversely, if $D_{E,W} \bullet D_D$ is large, as a result of many documents in engineering containing the same word, then IDF_{mod} will get smaller, which will reduce the $TFIDF_{mod}$ value; this means that the word occurs frequently in engineering but is not necessarily unique to the discipline, which implies it may not be discipline-specific.

As a result, when there is a word that has a high term frequency in a document, but occurs frequently in the discipline but not in all of engineering, then the modified approach would boost the score of that word. However, if that word does not occur frequently in the discipline but is common to engineering, then this algorithm would shrink its score. Therefore, the boosting effect only significantly affects words that are characteristic of that document, meaning it appears preferentially in the discipline but not necessarily in all of engineering

Methodology

This study develops a method for characterizing wording in engineering documents. In particular, we are interested in developing an approach to automatically identify course-specific language so that instructors can help first year students adjust to the terminology used in their chosen field of study. This is relevant to the field of accessible language in general, because it identifies vocabulary that students need to be familiar with in a professional context. The approach is outlined in Figure 1 below. Words are prepared for analysis by converting all input

documents to text-only format, then the TFIDF algorithm is used to develop word lists based on a target document (e.g. the final exam for a course) and sets of comparator documents. These word lists are then used to differentiate and highlight course-specific vocabulary that characterizes the target document.

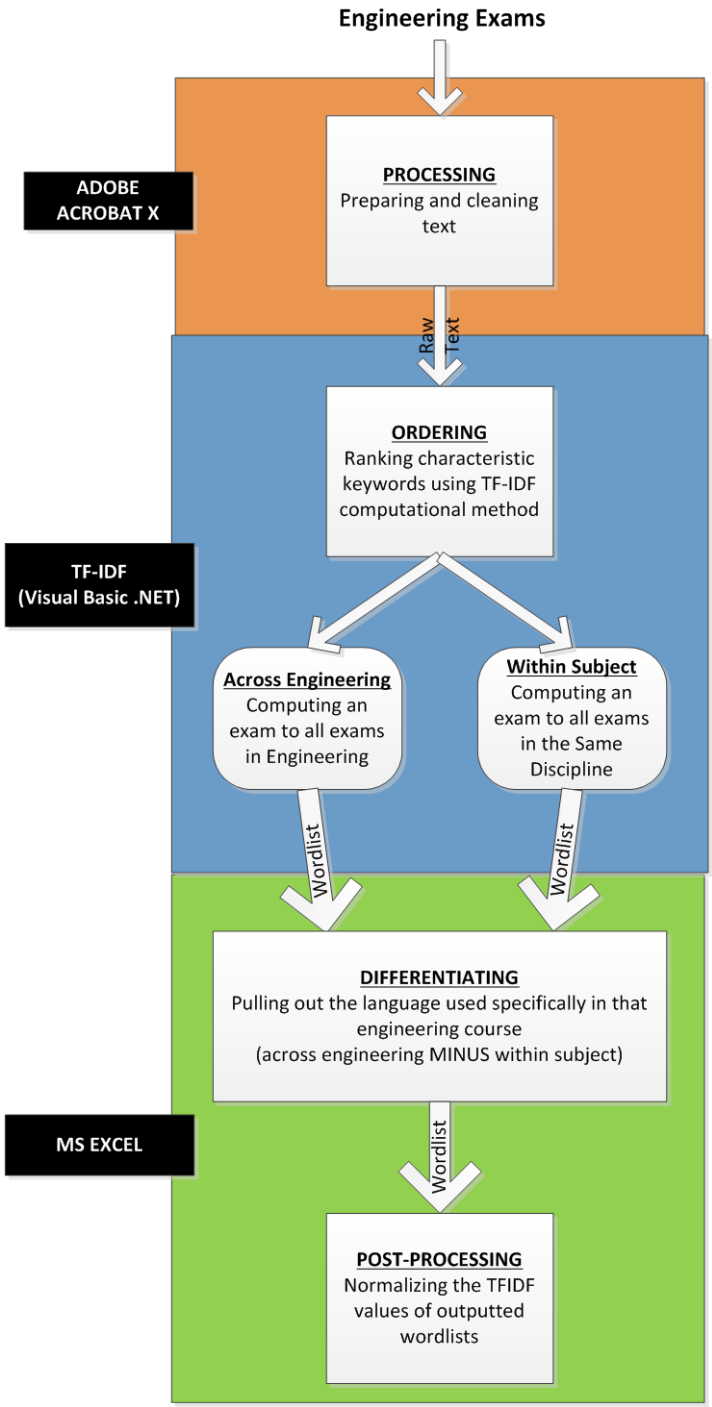

Figure 1- Shows graphically the methodology used in this study from top to bottom

The type of engineering document chosen for this study is engineering final exams. These documents are standardized artifacts of the engineering learning environment and are publicly-available for research and study purposes at the University of Toronto. The large dataset of exams spans several years, creating a substantial amount of vocabulary that can be examined. For this study, the authors begin by acquiring all electronically-available engineering exams at the University of Toronto. In total 2254 exams were used in the Faculty of Applied Science and Engineering between the years 1999 to 2009. These exams are in a variety of graphics and document formats, but they were converted to PDF-format using Adobe Acrobat X Professional to simplify subsequent coding and processing.

Clean Data and OCR exams

The text for each exam was subjected to an optimization process, as outlined in the top-most box of Figure 1. This process removes the majority of non-word artifacts that occur because of the original hardcopy-to-electronic conversion. Some of these artifacts included specks, misshapen words, improperly-oriented pages, equations, and foreign non-ASCII characters. Text conversion failed for roughly 20 of the exams, which were excluded from the remainder of the analysis.

TF-IDF Algorithm and equations

Once the text files for each of the exams in the study are created and optimized, the authors developed an applet in Visual Basic.NET that would compute the TFIDF score for words in target documents. Specifically, the program prompts for an input document and a folder where comparator documents are located. It computes the TFIDF score for each word in the target document based on the words found within text files contained the folder specified earlier. It then generates a list of words and their associated TFIDF scores and outputs that as another text file. Each sample exam is run through this program twice. One pass compares the exam against a comparator set of exams within the same discipline, while the other compares the exam to all exams in the repository. This procedure results in the creation of two word lists.

For each of the input exams, the $TFIDF_{mod}$ score is developed by subtracting the two word lists for each of the target documents, as outlined figure 1. This step is critical to the process because it helps to distinguish between vocabulary used in a discipline from vocabulary used across engineering. Specifically, this approach is used to highlight and further differentiate course-specific words from other vocabulary on the sample exam by increasing the spread of TFIDF values and outputting them as a scored wordlist.

Post-processing the TFIDF scores

The wordlist generated from the previous step is plotted graphically. This step graphically depicts the quantity and range of TFIDF values across an exam.

Results

Sample Case –Materials Engineering Exam

The results below track an exam from a course called “Fracture and Failure of Engineering Materials”, which is part of the Materials Engineering curriculum at the University of Toronto where we did this research. The data shows the TFIDF scores for a sample exam from the repository. Table 1 shows a ranked list of words in the target exam, in order of decreasing TFIDF scores. Figure 2 shows the rank of all of the words from the same exam plotted against their corresponding TFIDF score.

Table 1 - Shows the TFIDF scores (top 25 and selected others) for a sample exam from the course "Fracture and Failure of Engineering Materials"

Rank	Word	ModifiedTFIDF Score
1	dislocation	0.046749
2	dislocations	0.016992
3	cry	0.016379
4	grain	0.015939
5	crystal	0.014845
6	stress	0.013639
7	material	0.011965
8	strength	0.010907
9	deformation	0.008955
10	creep	0.008446
11	partials	0.008165
12	ofll	0.007426
13	intermetallic	0.007198
14	subgrain	0.007193
15	tensile	0.007181
16	metallic	0.006853
17	gb	0.006749
18	hardening	0.006659
19	boundaries	0.006414
20	hallpetch	0.006259
21	crss	0.00569
22	composite	0.005598
23	strengthening	0.005518
24	elastic	0.005376
25	lattice	0.005137
…		
200	fact	0.000435
…		
350	able	-0.000104
…		
450	equals	-0.001426

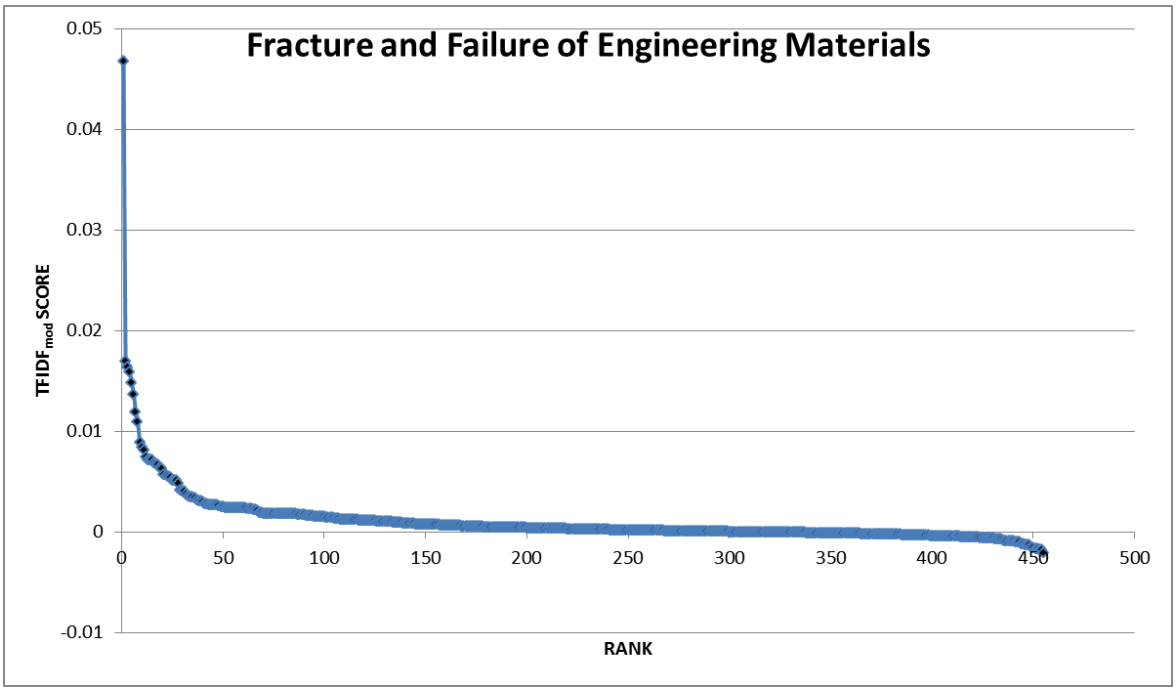

Figure 2 - Shows all of the TFIDF-scored words from a course called "Fracture and Failure of Engineering Materials"

The wordlist in Table 1 contains a high number of course-specific vocabulary, especially near the top of the list. This is the expected result as words that are characteristic of the sample document are assigned a higher TFIDF score than words that are commonly found on all engineering exams. As the rank gets larger, the number of non-course-specific words increases significantly. Though there are too many words to list individually here, a number of sample words at various points along the TFIDF scale are included. For example, looking at word 350 "able", shows that it is assigned a negative value, and this is a direct result of the $TFIDF_{mod}$ shrinking the value because it occurs frequently in all of engineering, the discipline, and the exam.

It is also worth noting that there are some "non-sensical" terms that are prevalent on this exam. Though only a small portion are seen in Table 1, like "ofll", "gb" and "crss", most of them exist in the ranks greater than what is shown. Further, it is important to note that "gb" is shorthand for "grain boundary", and "crss" is shorthand for "critical resolved shear stress", both of which are words characteristic to the course and might be interpreted otherwise. Other terms such as "ae", "gc", "ndx", "derisity", and many others pollute the dataset even though the exams have been carefully processed. Unfortunately, these words continue to exist on all of the datasets and affect the computation of accurate TF-IDF values. This shows that though the approach shows promise to distinguish course-specific words from "everyday" language, there remain many artifacts that compromise the accuracy of using this method as currently defined.

Figure 2 graphically depicts the words and their corresponding TFIDF score, ranked in decreasing order going from left to right. The data show that there is a small subset of vocabulary – seen here as being ranked from 1 to roughly 50 – that have a much higher TFIDF score than the majority of other words on the list. That is, a few words have a high TFIDF score while the majority have a consistently lowscore. It is also important to note that the tail of the data in Fig. 2 shows a downward (negative) trend as it approaches the lowest TFIDF scores. In the wordlist, these words are typically nonsensical artifacts that pollute the dataset and are not course-specific.

Discussion

Critique of the Approach as situated in the Literature

The TD-IDF method is one in a spectrum of approaches that range in utility and feasibility when applied to investigating the discipline specific terminology in a course. Ideally, a method can be found that is easy to employ with minimum effort by the instructor (i.e. highly feasible), and produces a list that is of high value to the freshman student (i.e. of high utility). On one end of the spectrum there are approaches that examine just the frequency of words (e.g. Zipf's Law[12, 13, 18]) and these approaches are highly feasible but low in utility. Frequency information is useful as it explains how 'conversational-sounding' the text is[12], and which words are used more or less frequently than others, etc. but it does not provide much utility towards the purpose identified here. The ease of implementation is high though, because documents can be submitted to a software program that tallies the occurrences of each word and graphs this information. The shape of the graph can then easily be used to characterize the language.[13, 18]

Conversely, there exist approaches that use synsets, or the relationship of words to one another using language corpora, that can be used to characterize language on documents.[13, 14] These approaches rely on comparing the meaning of words to one another, and these meanings are identified using tools such as WordNet, etc. These synset-based approaches produce a large quantity of rich data about the vocabulary. This information would include the meaning of words in sentences and how they evolve with the context in which they appear. Though very informative and thus high in utility, the feasibility of using such approaches is low because the amount of information required about the vocabulary being explored a priori is high. For example, the corpora used in identifying meaning needs to be continuously updated by an expert (or the instructor) to take into account the ever-changing vocabulary. As such, synset-based approaches require a large amount of support to produce and use corpora that include not only a list of words, but also information about how they associate. This may be preferable, but also necessitates the creation of a large back-end support system versus an overly-simplistic frequency analysis approach that does not provide much utility.

The method we have identified, a modified application of the TFIDF algorithm, works toward creating a dataset that has higher utility than pure frequency analysis yet is more feasible to

implement than synset-based approaches. This is because the approach does not require multiple corpora and systems to understand the specific meaning and relationships between words, but instead uses contextual information provided by the comparator document set. Specifically, the user provides comparator sets in the form of groups of exams or other teaching materials. Users are not required to know specific details about the comparator sets, other than the course and disciplines, but they need to provide the document sets in a machine-readable format. This approach is a tradeoff between utility and feasibility because although it requires some contextual information, which comes in the form of other documents, it does not require a continuously-updating support system to update the meanings and relationships between words. As such, the TF-IDF method has a higher utility than purely a frequency analysis approach, while also being more feasible to implement than approaches based on synsets, yet still provide information that can assist us in separating discipline-specific language from others.

One can imagine a system that automatically files final exams into a database based on course information and a few key words that identify the field, e.g. materials science fundamentals or materials and metallurgy, etc. The instructor could simply identify the target document or documents (such as last year's midterm exam, or final exam for her course), identify courses in the same discipline by course number or keyword, and then hit "run". The program would automatically produce a word list for her to distribute to her students at the start of the new term. For a freshman student a word list like this lessens anxiety about what they need to learn and creates a starting vocabulary of terms relevant to the field.

Critique of the Methodology as it exists right now

The preliminary results suggest that the modified TFIDF approach is able to distinguish discipline-specific vocabulary from other words. The methodology is soundly grounded in existing methods of automated indexingand the TFIDF lists that we have produced appear to be largely discipline-specific for the first 50 or so words.

This method needs further improvement to eliminate artifacts before progressing further. The data currently shows a high number of nonsensical terms that do not appear to be in the English language. Specifically, the data contains terms like 'ofll' and 'gb' Ideally, artifacts would be eliminated before TFIDF calculations are made. However, this is not a straightforward task because of the use of acronyms and other anomalies in engineering jargon. Suggested approaches to this problem are as follows. First, it may be possible to remove words that do not include vowels a priori to being scanned. In doing so, we remove a significant portion of terms that might not exist in the English language[6, 7] but risk removing important engineering acronyms. Another strategy is to incorporate a 'spell-checker' application that can scan text to highlight these terms using a combination of English-language as well as existing Corpus-based tools such as WordNet. This approach uses a larger corpus of language comparison tools than most word processors because it can draw on language from engineering corpora as well as standard English corpora. However more involved, this second strategy does not remove words

automatically and thus ensures that the process is not removing vocabulary that may be pertinent to the student, like technical jargon. However, as a result, each word on the outputted wordlists would need to be extracted manually and this could be a lengthy process that reduces the feasibility of a computational approach for an instructor. Such a method might also be used instead to highlight terms on the final outputted list. This way, an instructor can visually see words that appear high in the TF-IDF lists and appear in relevant corpora as well. Ideally, a combination of the vowel-removal and corpora-scan methods could be employed to remove as many non-English words as possible to maximize the effectiveness of a computational approach characterizing the language in engineering courses. As an added benefit an instructor could input a draft of an exam or other piece of course material and explicitly identify the vocabulary they are testing.

Conclusions

This study uses a modified approach from the field of computational linguistics to characterize vocabulary on engineering exams. The objective is to increase the transparency of learning outcomes expected in an engineering classroom, specifically the development of a professional vocabulary. By using a repository of 2229 exams, a modified term-frequency inverse-document frequency (TFIDF) algorithm assigns a weight to each word in an input exam by comparing it to the occurrences of those words across all exams; the weight represents the degree to which that word is characteristic to that document. The data show that this method does appear to preferentially give course-specific words higher ranking. However, we also found that these wordlists are polluted with non-English words and that further work in cleaning the input text files a priori is required. The next step is to refine the algorithm and test it with users.

References

1. Mazur, Beth. "Revisiting Plain Language." *Technical Communication: Journal of the Society for Technical Communication* 47.2 (2000): 205-11.
2. Robinson, Peter, and Nick C. Ellis, eds. *Handbook of cognitive linguistics and second language acquisition*. London and New York: Routledge, 2008.
3. Ahearn, Laura M. "Language Acquisition and Socialization." *Living Language: An Introduction to Linguistic Anthropology* (2011): 50-64.
4. Bunch, George C., Percy L. Abram, Rachel A. Lotan, and Guadalupe Valdés. "Beyond sheltered instruction: Rethinking conditions for academic language development." *TESOL Journal* 10, no. 2-3 (2001): 28-33.
5. Braun, Sabine. "From pedagogically relevant corpora to authentic language learning contents." *ReCALL* 17.1 (2005): 47-64.
6. Krashen, Stephen D. *Explorations in language acquisition and use*. Portsmouth, NH: Heinemann, 2003.
7. De Saussure, Ferdinand. *Course in general linguistics*. Columbia University Press, 2011.
8. Jackendoff, Ray. *Foundations of language: Brain, meaning, grammar, evolution*. Oxford University Press, USA, 2002.
9. Dąbrowska, Ewa, and James Street. "Individual differences in language attainment: Comprehension of passive sentences by native and non-native English speakers." *Language Sciences* 28.6 (2006): 604-615.
10. Aitchison, Jean. *Language change: progress or decay?*. Cambridge University Press, 2000.

11. Church, Kenneth W., and Robert L. Mercer. "Introduction to the special issue on computational linguistics using large corpora." *Computational Linguistics*19.1 (1993): 1-24.
12. Mitkov, Ruslan, ed. *The Oxford handbook of computational linguistics*. Oxford: Oxford University Press, 2003.
13. McEnery, Tony, Andrew Wilson, and Geoff Barnbrook. "Corpus linguistics." *Computational Linguistics* 24.2 (2003).
14. Budanitsky, Alexander, and Graeme Hirst. "Evaluating wordnet-based measures of lexical semantic relatedness." *Computational Linguistics* 32.1 (2006): 13-47.
15. Bybee, Joan L., and Paul Hopper, eds. *Frequency and the emergence of linguistic structure*. Vol. 45. John Benjamins Publishing Company, 2001.
16. Phillips, Betty S. "Lexical diffusion, lexical frequency, and lexical analysis." *Typological Studies in Language.* 45 (2001): 123-136.
17. Roland, Douglas, Frederic Dick, and Jeffrey L. Elman. "Frequency of basic English grammatical structures: A corpus analysis." *Journal of Memory and Language* 57.3 (2007): 348-379.
18. Montemurro, Marcelo A. "Beyond the Zipf–Mandelbrot law in quantitative linguistics." *Physica A: Statistical Mechanics and its Applications* 300.3 (2001): 567-578.
19. Bellegarda, Jerome R. "Exploiting latent semantic information in statistical language modeling." *Proceedings of the IEEE* 88.8 (2000): 1279-1296.
20. Maynard, Diana, and Sophia Ananiadou. "Trucks: a model for automatic multiword term recognition." *Journal of Natural Language Processing* 8.1 (2000): 101-126.
21. Variawa, Chirag, and Susan McCahan. “Identifying Language as a Learning Barrier in Engineering.” *International Journal of Engineering Education* 28.1 (2012): 183-191.
22. SHI, Congying, Chaojun XU, and X. Yang. "Study of TFIDF algorithm." *Journal of Computer Applications* 29 (2009): 167-170.
23. Robertson, Stephen. "Understanding inverse document frequency: on theoretical arguments for IDF." *Journal of Documentation* 60.5 (2004): 503-520.
24. Singhal, Amit. "Modern information retrieval: A brief overview." *IEEE Data Engineering Bulletin* 24.4 (2001): 35-43.
25. Han, Eui-Hong, and George Karypis. "Centroid-based document classification: Analysis and experimental results." *Principles of Data Mining and Knowledge Discovery* (2000): 116-123.