

Data Mining User Data to Determine SAT Word Complexity and Fairness

August 9, 2012

Dmitry Simakov (dmitrysmkv@gmail.com)

Abstract

A+ ITestYou Vocabulary, an online educational application with word banks for many standardized tests, was selected for the study. The data mining of the user data was conducted on 500,000 user responses. The statistical analysis algorithm was designed to extract a list of the hardest and the easiest SAT vocabulary words from this data. Three different software tools were used to conduct the analysis: Excel 2010, RapidMiner 5 and R. Several recommendations were made to improve the learning process, to increase accuracy, and to prevent cheating.

Introduction

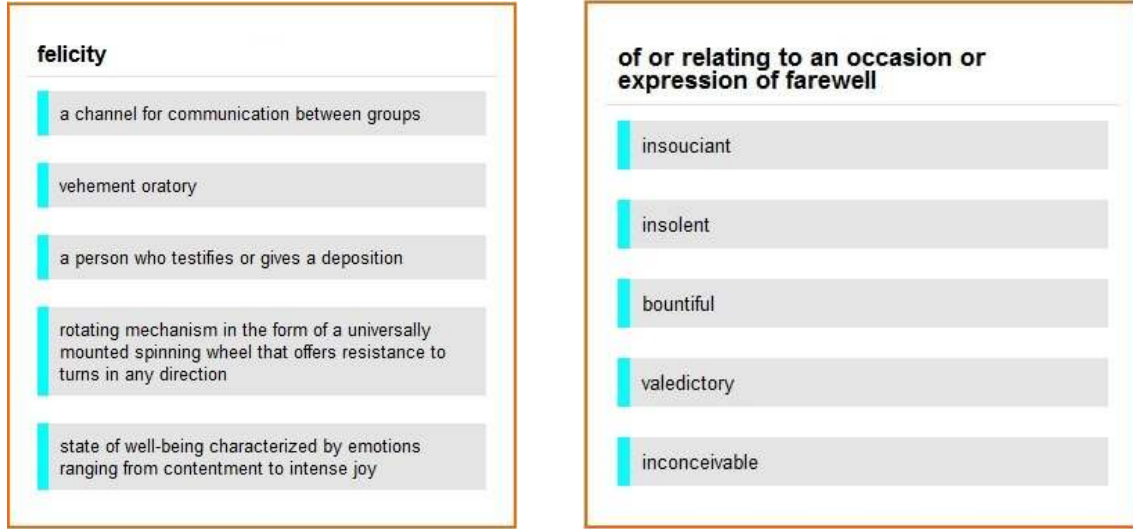
Every year approximately 1.5 million [1] young adults take the Scholastic Aptitude Test (SAT)[2]. This test is taken so that students can prove to the Colleges and Universities that they are ready academically for the workload. The test score also shows what level of complexity the future student can handle. The math and writing sections are skill based that can be mastered simply by studying the rules of the subject, but the critical reading section is not something that can easily be taught because it involves a lot of vocabulary.

What makes the SAT vocabulary so hard is the sheer volume of words that one needs to know. Vocabulary, a lot of the times, can only be mastered through rigorous memorization. Even after spending countless hours studying, remembering the intricate difference between *callow* and *candor* can be difficult. Over the years many software tools and games were created to simplify the studying process and empower the student. In this study I focused on one of these tools, ITestYou [3], which offers A+ ITestYou Vocabulary - a mobile and web application for learning vocabulary.

A+ ITestYou Vocabulary has word banks for many standardized tests, including SAT, Graduate Management Admissions Test (GMAT) [4], Test of English and Foreign Language Test (TOEFL) [5] and Graduate Record Examination (GRE) [6]. The format of the questions asked in the application closely resembles the format of the questions given on the standardized tests. This is done so that the user can have realistic experience. The format for the SAT words has two variations. For the first variation of the question, the user has to find the appropriate definition for the mystery word asked. For the second variation of the question, the user has to find the appropriate word for the mystery definition.

ITestYou did not just want to create an application that presents words to users randomly. It implements a variation of the *spaced repetition* algorithm [7]. Words are presented to a user to verify if user knows the word. Unknown words are re-introduced frequently so the users can remember them better, while known words are shown less often, so that the users can preserve the knowledge of the word. The ITestYou team is interested in improving the algorithms and the efficiency of learning and allowed me to conduct this study.

Picture: A+ ITestYou Vocabulary user interface: variation 1 (left) and variation 2 (right).



Goals

As a high school student myself, I understand first hand how difficult it can be to master vocabulary and to achieve a competitive score on the Critical Reading Section [8] on the SAT. As with any online service for general public, ITestYou has many users and lots of data. There are many aspects of the data that one could have measured and analyzed, but in order to benefit students the most I decided to pursue 3 key goals.

The first goal is to extract a list of the hardest and the easiest SAT vocabulary words from user responses. This is one of the top three goals because I want to give students a basic outline of what words they should spend most of their time studying. Giving students such a list allows for optimal usage of their time.

The second goal is to choose the most convenient data mining and analysis software suitable for even non-programmers. There are many different statistical and data mining software titles available, but I focused on the three most popular: Microsoft Excel [9], R [10], and RapidMiner [11]. In the end of this study I want to recommend one software tool that I know can provide the best data mining service for the type of analysis performed here.

The third and final goal is to understand the functionality of A+ ITestYou Vocabulary application. I am interested to better understand software algorithms behind the product. With that, I want to provide suggestions on how ITestYou team can improve the user interface, increase learning efficiency, and reduce the number of opportunities for cheating.

Source Data

The complete set of words that could potentially be asked on the SAT test is unfortunately not disclosed to the public. Additionally, new words are added to the SAT word bank everyday. Therefore, the ITestYou team can only gather words that are reported by past test takers to be on the official SAT vocabulary list. The ITestYou team used a web crawler to collect the words from various sources claiming to have collected them from the past test takers. Over time a word bank of 5202 SAT vocabulary words was assembled and is now available on the A+ ITestYou SAT Vocabulary [12].

Before I start to analyze the data it is very important to discuss its origin and features first. The data under study was collected by ITestYou during different time periods in the first and second quarter of

2012. The data was given to me as a CSV [13] file 168 MB in size. The data was anonymized; all users were assigned random identifiers so that we can conduct analysis, but not identify a specific person. The data also contained user responses for all three ITestYou applications: A+ ITestYou Languages, A+ ITestYou Math, and A+ ITestYou Vocabulary. Each row contains these columns:

Table: Source data column names, description and type.

Column Name	Description	Possible Values
REQ_ID	identification number for the user response	integer value sequentially increasing over time
REQ_USR_ID	identification number of user	integer value, unique value for each user
REQ_UNI_ID	identification number of different problem type	integer value, unique value for each type of problem (115 is for SAT vocabulary problem)
REQ_VAR_ID	identification number of vocabulary word (for vocabulary problems) or variation (for math problems)	integer value ranging between 0 - 5201, unique value for each word in the SAT vocabulary
REQ_PASSED	value that signifies if problem was correctly answered	0 – if user answered question incorrectly 1 – if user answered question correctly

There were other data columns, but they had no relevancy to this study.

Data Mining Algorithm

I have used different tools to perform the analysis, but in all cases this general procedure was followed:

1. Import data into an analytical tool (Excel, R or RapidMiner).
2. Filter data so that only user responses made for SAT vocabulary remain.
3. Run data through a formula filter to extract only first encounters: users first interaction with a word. We are interested only in first encounters because it shows us the users true knowledge of the word before using vocabulary application.
4. Count total number of correct and incorrect answers for each word first encounters across all user.
5. Calculate the total number of times a word has been presented by adding correct and incorrect answers.
6. Calculate binomial probability [14] of getting so many correct and incorrect answers purely by chance assuming that the the word is fair and has 50/50 probability to be known by a user. I am using binomial probability because there can only be two outcomes, either the user knows the definition of the word or he does not.
7. Order the words from the smallest binomial probability to the greatest binomial probability.
8. Extract lists of top ten hardest words and top ten easiest words.

Now that the general algorithm has been outlined, I proceed with the implementation in Excel, Rapid Miner and R.

Implementation in Excel

In order to perform this study I used an Acer Aspire 1830T laptop with the Intel i3 dual core U330 CPU running at 1.2 GHz with 3 GB of RAM running 64-bit Windows 7 operating system. I used Excel version 14.0.6112.5000, which is part of Microsoft Office Professional Plus 2010. I imported the CSV data file into Excel. The process of import took approximately 4.5 seconds. There we total of 514,009 data rows in the file.

After importing the data, I created a header row to name the columns as shown in the table above. I used an Excel filter function to select all the answers corresponding only to SAT problem types (REQ_UNI_ID = 115). After filtering, there were 84,615 rows left.

Next, one needs to find the first encounters of every word with every user. It is very important for the study that we only select the first encounters, because they show us the basic knowledge of user vocabulary before using ITestYou’s educational application. Extracting first encounters requires several steps. The first step is to sort the data by three columns: REQ_USR_ID, REQ_VAR_ID, and REQ_ID, respectively. This step properly orders the data for the formula later on. Once sorting is done, it is critical to copy the filtered data into new a worksheet. This is important because all formulas implemented on data will not skip filtered out and invisible rows and we need these rows removed.

Once the data was copied into a new worksheet as “values only”, I created a formula to compare the selected row and the row above it. If the selected row is the same as the row above it, the value of “True” is returned and the selected row is not a first encounter. If the selected row is not the same as the row above it, the value “False” is returned and it is a first encounter. Next, I filtered out all the rows that are “True”, and only first encounters remain, 51,031 rows in total.

Picture: Excel spreadsheet that shows first encounters for SAT vocabulary words.

	A	B	C	D	E	F	G	H
1	REQ_ID	REQ_USR_ID	REQ_UNI_ID	REQ_VAR_ID	REQ_ISSUED_ON	REQ_RECEIVED	REQ_PASSED	rec sam
30458	116887	0	115	8	2/3/2012 2:21	2/3/2012 2:21	1	FALSE
30459	381931	0	115	8	4/16/2012 13:26	4/16/2012 13:27	1	TRUE
30460	292359	0	115	16	3/15/2012 16:11	3/15/2012 16:11	1	FALSE
30461	207628	0	115	37	2/22/2012 16:59	2/22/2012 17:00	1	FALSE
30462	460820	0	115	43	5/15/2012 19:22	5/15/2012 19:22	1	FALSE
30463	146441	0	115	58	2/9/2012 22:02	2/9/2012 22:02	1	FALSE
30464	209754	0	115	61	2/23/2012 7:55	2/23/2012 7:55	1	FALSE
30465	116878	0	115	70	2/3/2012 2:20	2/3/2012 2:20	1	FALSE

Having first encounters, I created a pivot table to compute the sum of correctly and incorrectly answered words and the total amount of times a word has been shown to all users. Now that we have the sums of passed and failed we can calculate the binomial probability of each and every word and its fairness. All 5202 unique SAT words were presented to the user at least once and some as many as 22 times.

Picture: Pivot Table shows the pass, fail, and total shown count, also the binomial probability of pass rates.

	A	B	C	D	E
1	Not First Encounter	FALSE			
2					
3	Row Labels	Sum of REQ_PASSED	Sum of REQ_FAILED	Total Played	Binomial Probability
4	630	1	15	16	0.000244141
5	3563	4	14	18	0.011672974
6	1154	5	14	19	0.02217865
7	4519	3	13	16	0.008544922
8	1871	7	13	20	0.073928833
9	2199	3	13	16	0.008544922
10	1756	6	13	19	0.051750183
11	1024	9	13	22	0.11859417

Using the binomial coefficient and the probability mass function formulas below, I was able to determine the fairness of each word in the SAT vocabulary. I used the binomial distribution formula $\text{BINOMDIST}(n, k, p, \text{cumulative})$ [14] provided in Excel, which can also be manually calculated for n - number of total trials, k - number of successes, and p - probability of success using [15]:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

$$f(k; n, p) = \Pr(K = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Since I am using a binomial distribution, I need to take note of the fact that it is a two sided distribution. This means that when you have two words with inverted pass and fail rate, the binomial probability value will be the same. Therefore, I need to differentiate between the two cases. When the pass rate \geq fail rate, the word is considered to be "EASY". When the pass rate $<$ fail rate, the word is considered to be "HARD". Now that I have a way to determine easy versus hard, I can use a simple Excel filter to select one or the other. This makes it easy for us to then extract the list of top ten hardest and easiest words.

Overall, Excel was an excellent analytical tool to do this kind of data mining. It gave me a quick responses and was not complex to use. There were only two complications with Excel. First, I had to manually copy all filtered data into new worksheet to create a pivot table. Second, the entire process described above had to be redone manually each time the source data changed. But as a whole, Excel turned out to be a great analysis tool.

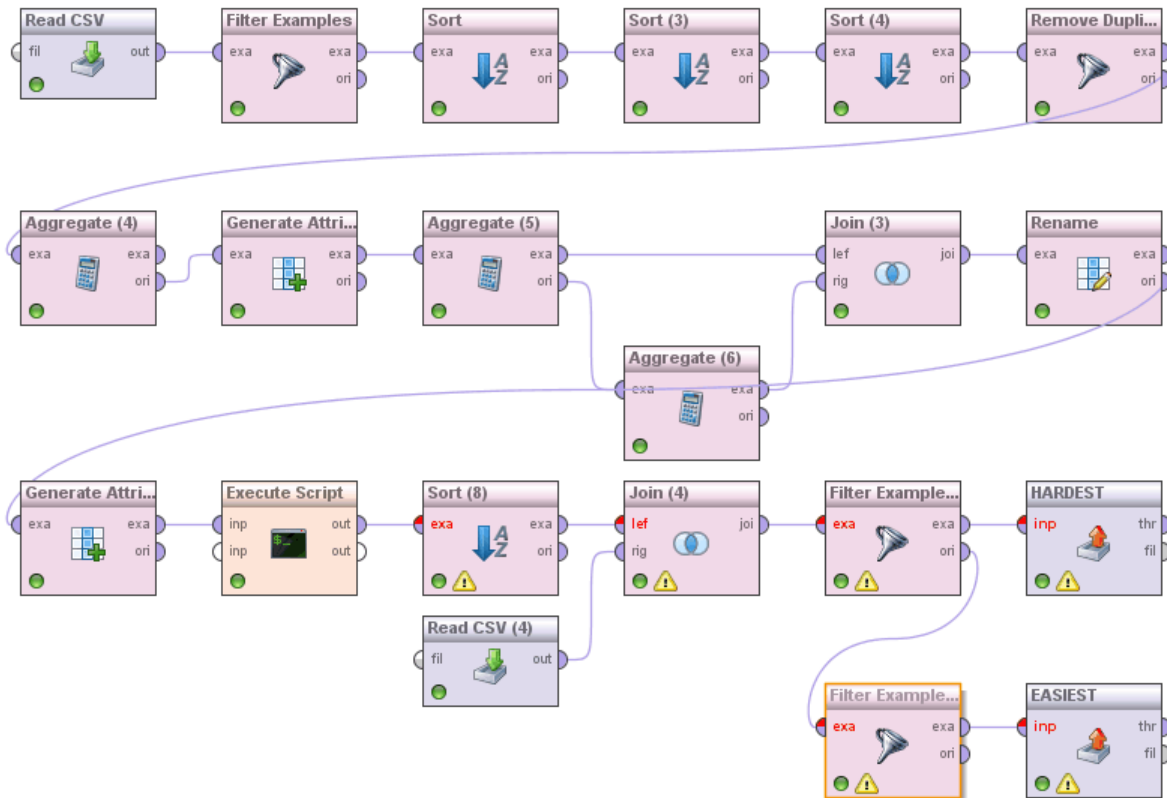
Implementation in RapidMiner 5

Another tool that I used to conduct this study was RapidMiner 5 [11]. It is an environment for data mining, predictive analytics and business analytics. The unique feature of RapidMiner 5 is that almost all possible functions are preprogrammed into the system as "operators". All the user has to do is create a processing pipeline that shows how he wants the data to be manipulated by the operators. A pipeline can

be built by connecting two different “operators” using “connectors” and setting different properties of each operator.

I imported the source data file into RapidMiner and filtered, sorted and aggregated the data. However, I found that the online manual had too little information and did not describe well complex processing. I experimented with different operators and tried different settings. Ultimately, with significant help from the ITestYou engineer, I was able to complete the processing pipeline shown below.

Picture: Data mining process in RapidMiner 5



Here are all the operators used in the pipeline with their brief descriptions:

- Read CSV file - This operator reads csv files, where all values are separated by a comma.
- Sort - This operator sorts a given data set by a single attribute.
- Aggregate - This operator aggregates specific column values across all rows and creates a new value with the their sum or average.
- Filter - This operator takes a dataset and returns a new dataset with rows that fulfill a specific condition (for example: REQ_UNI_ID = 115).
- Generate - This operator adds a new column to a dataset and sets attribute value for each row.
- Rename - This operator renames any attribute specified.
- Execute Script - This operator allows the user to built his own program in the programming language Groovy [16].
- Join - This operator combines two different dataset, based on identical value of one of the attributes.
- Write CSV - This operator outputs data into a comma separated file.

It took about 45 seconds to run the pipeline. It produced the output identical to the one produced in Excel above.

Overall, I found RapidMiner 5 to be a very convenient analytical tool. There were many positives I benefitted from. First, the program was easy to use and the visual model for data processing is very intuitive. I was able to create simple processing pipeline in minutes. Second, the program was able to run very fast even with the large dataset I had. Third, all the operators, even very complex ones, were preprogrammed into the software allowing the user to create very complex models with little programming.

I also had difficulties. First, the RapidMiner 5 online manual had gaps and did not give examples showing how to use for each operator and how to combine them. Second, RapidMiner 5 binomial probability function `binom()` returned strange results that did not match those of Excel. So we found an article [17] that described the process of making a custom binomial probability function for RapidMiner 5. The article helped me complete the study.

Implementation in R (Programming Language)

As planned, I also used R [10]. Unfortunately, the experience did not turn out as I hoped. There were several issues that did not allow me to complete the study. First, my computer did not have the power to load the entire file with 514,009 rows. The computer would hang indefinitely. I had to partially filter data in Excel for the `REQ_UNI_ID = 115` down to 84,615 records before the file was able to load into R workspace.

Picture: Using R to import user data and construct an interquartile range for all categories

```
> data = read.table ("C:\\Dima-Documents\\2012\\dima_ity_data_final(csv1).csv", header = T, ",")
>
> summary (data)
```

REQ_ID	REQ_USR_ID	REQ_UNI_ID	REQ_VAR_ID	REQ_ISSUED_ON
Min. : 30458	Min. : 1000	Min. : 115	Min. : 1	1/17/2012 2:45 : 7
1st Qu.: 86147	1st Qu.: 3252	1st Qu.: 115	1st Qu.: 1268	1/17/2012 4:39 : 7
Median : 247176	Median : 5718	Median : 115	Median : 2553	5/28/2012 11:09 : 7
Mean : 251672	Mean : 5685	Mean : 115	Mean : 2597	1/17/2012 6:39 : 6
3rd Qu.: 398720	3rd Qu.: 8029	3rd Qu.: 115	3rd Qu.: 3970	1/17/2012 7:21 : 6
Max. : 513985	Max. : 10094	Max. : 115	Max. : 5199	1/18/2012 9:48 : 6
				(Other) : 15791

REQ_RECEIVED_ON	REQ_PASSED	First
1/17/2012 4:39 : 7	Min. : 0	Mode : logical
1/18/2012 9:48 : 7	1st Qu.: 0	FALSE: 15830
5/28/2012 11:09 : 7	Median : 0	NA's : 0
1/17/2012 10:10 : 6	Mean : 0	
1/17/2012 2:41 : 6	3rd Qu.: 0	
1/17/2012 2:45 : 6	Max. : 0	
(Other) : 15791		

Second, learning R programming language was much more difficult than I expected. I had watched many video tutorials on R and read the different sections of the manual, but could not get confident working with data. R did not have the multiple level filtering and grouping capabilities that was required for this study or I was not able to understand how to use these features properly in the time allowed. Ultimately, I was not able to properly filter and group the records and complete other calculation steps of the data mining process.

As a whole R is a great analytical tool, but requires significant programming skills and a background in computer science. R did not suit our needs, so I cut short and did not pursue further.

Results

I was able to successfully extract the top ten hardest and easiest words in the SAT vocabulary. The tables below shows the results of this study. The first table shows the top ten hardest words. The second table shows the top ten easiest words.

Table: The top ten hardest SAT vocabulary words.

REQ_VAR_ID	Number of Correct Answers	Number of Incorrect Answers	Total Number of Answers	Binomial Probability	Word
630	1	15	16	0.000244	bowdlerize
4574	1	12	13	0.0015869	specious
1677	0	9	9	0.0019531	encyclical
4800	0	9	9	0.0019531	tenet
833	1	11	12	0.0029296	commissariat
1600	1	11	12	0.0029296	effete
1728	1	11	12	0.0029296	ephemeral
2515	1	11	12	0.0029296	indigence
5046	1	11	12	0.0029296	vapid
5151	0	8	8	0.0039065	wane

Table: The top ten easiest SAT vocabulary words

REQ_VAR_ID	Number of Correct Answers	Number of Incorrect Answers	Total Number of Answers	Binomial Probability	Word
4885	19	0	19	0.0000019	transition
322	18	0	18	0.0000038	anthropology
863	18	0	18	0.0000038	collapsible
4992	17	0	17	0.0000076	unicellular
331	16	0	16	0.0000152	antique
1452	16	0	16	0.0000152	disobedience
1604	16	0	16	0.0000152	efficient
4981	16	0	16	0.0000152	undervalue
1412	15	0	15	0.0000038	discolor
1531	15	0	15	0.0000038	dominate

As seen above the word **bowdlerize** is the hardest word, having been answered incorrectly by 15 out of 16 users seeing it for the first time. The easiest word is **transition**., having been answered correctly 19 out of 19 times. I have just selected the top ten words, but I have a complete ranking for all 5202 words in the SAT vocabulary.

Discussion

My first goal was to determine SAT vocabulary word difficulty. I was able to identify in this study the most difficult and the easiest SAT vocabulary words. I developed a statistical analysis approach to comparing words difficulty for different users based on binomial probability and were able to apply it to analysis of A+ITestYou SAT vocabulary testing.

I have found several related reports, but was surprised how little information was given about the procedures or data used to produce them. For example, Education.com has an article “50 Words the SAT Loves” cites Princeton Review providing them the “most frequently tested” words [18]. But specifies no details on how these word were selected. None of the words in my most difficult word list are mentioned in this list.

Sparknotes.com publishes separate reports for “The Top 250 Most Difficult SAT Words” [19] and “1000 Most Common SAT Words” [20], but does not describe where the data comes from of how word difficulty was determined. I have compared the lists and found two of top ten words (*ephemeral* and *vapid*) to match Sparknotes.com most difficult list and word *wane* to be present in the most common list.

Mygretutor.com has the closes approach to the one I took [21]. They keep track the number of people get the words right in their MyGRETutor interactive section of the website. Instead of binomial probability the user population is measured. Mygretutor.com defines a word as easy if it is answered correctly by at least 80% of users. Medium words are those words whose definition is, on average, correctly selected between 50 and 80 percent of the time, and difficult words are those words whose definition is correctly selected less than 50% of the time. Mygretutor.com does not publish the entire list and I was not able to crossreference it with my results. I think the best approach is to combine the binomial distribution with the population based metrics like it is done at Mygretutor.com.

My second goal for this study was to choose an analytical tool that works most effectively for data analysis. I used three different tools to conduct this study: Excel 2011, RapidMiner 5, and R. Every tool has its own positives and negatives. Excel was able to handle the dataset size and complete the calculation, but I had to manipulate data manually. R has many video tutorials because it is a popular analytical tool, but it was difficult to use and required programming skills I did not have. RapidMiner 5 has a very intuitive interface so that the user can build complex logical pipelines, but did not have a binomial probability operator in the program. Ultimately, given my skill set and experience an Excel turned out to be the best data mining tool for me.

My third goal was to understand ITestYou software algorithms and provide useful insights on how ITestYou can improve them. There were two big issues: opportunities for user cheating and vocabulary test user interface. Students using ITestYou vocabulary application earn points for giving correct answers to the problems. The competition is based on a point system where every right answer gets you a positive points and every wrong answer gets you negative points. At the end of every week the winner, user that has the most points, is selected.

This system works very well in driving the engagement up, but it also creates opportunities for cheating. Many users are very dedicated at getting perfect scores and many of them end up cheating to get the score they desire. When the user sees a word they do not know they can simply quit the program and later restart without having to face the unknown word. Having users cheat like this makes the data skewed and seem more perfect than it actually is. Cheating also might have changed the ranks of how difficult the words actually are, because responses for the skipped words are not recorded now. The most difficult words might have been the ones for which we have no answers at all! ITestYou team needs to record and consider all problems presented to the user, even if user does not provide answers.

Another set of improvements can be made in the user interface. The vocabulary problems are presented as multiple-choice. Since the definitions given are random the user can have a really easy time deducing the correct answer. This is true because some words like *anthropology* everybody knows that it is the study of something, but they don't know the study of what. Since the definitions, that are supposed to trick the user, are randomly selected, guessing is often easy.

I TestYou can improve this issue by applying two different tactics. First, they can give the users questions that are structured a certain way. The questions have to be inverted, meaning that we give the user a definition and they have to select the correct word. Second, to make the questions more complex (so that the word tests the users full knowledge of the word) the definitions have to be similar so it tests the intricate differences between the words. One can do this by assembling all the words and splitting them into key parts - root, suffix, and prefix. Even though there are three different key parts, only the root can help us create a more challenging problem for the user. One can do this by grouping all the words that have the same root together. Then when a word is selected to be asked a user the definitions that are meant to be the wrong answers can only be selected from the group of the same root words. Similar ideas were explored in [22], where it was proposed not to use multiple-choice questions at all, or to supplement them with in-depth follow up questions.

Conclusion

Online educational software generates unprecedented amount of useful data. To supplement, the excellent quality statistical analysis tools are readily available to anyone with almost none or just minimal software programming experience. It is now possible to optimize the learning process and provide personalization of education to each individual user. It is also possible to compare performance of a community of users and to tune the test complexity to a desired level.

References

1. College Confidential Blog. <http://talk.collegeconfidential.com/sat-act-tests-test-preparation/301818-how-many-people-take-sat.html>
2. Official SAT Reasoning Test. <http://sat.collegeboard.org/register/>
3. I TestYou. <http://www.itestyou.com/>
4. Official Graduate Management Admissions Test. <http://www.mba.com/the-gmat.aspx>
5. Official Test of English and Foreign Language Test. <http://www.ets.org/toefl/>
6. Official Graduate Record Examination. <http://www.ets.org/gre/>
7. Spaced repetition algorithm. <http://en.wikipedia.org/wiki/SuperMemo>
8. Official College Board. <http://sat.collegeboard.org/about-tests>
9. Official Microsoft Excel. <http://office.microsoft.com/en-us/excel/>
10. R programming language. <http://www.r-project.org/>
11. RapidMiner 5. <http://rapid-i.com/content/view/181/196/>
12. A+ I TestYou SAT Vocabulary. <http://www.itestyou.com/cms/unit-115>
13. Comma-separated values (CSV). http://en.wikipedia.org/wiki/Comma-separated_values
14. Official Excel website for Binomial Distribution formula. <http://support.microsoft.com/kb/827459>
15. Binomial probability and binomial distribution. <http://www.stat.yale.edu/Courses/1997-98/101/binom.htm>
16. Groovy Programming Language. <http://groovy.codehaus.org/>
17. Binomial Probability in RapidMiner 5. <http://www.softwaresecretweapons.com/jspwiki/user-defined-java-functions-in-rapidminer-5>
18. 50 Words the SAT Loves. http://www.education.com/magazine/article/Ed_50_Words_SAT_Loves_2/
19. The Top 250 Most Difficult SAT Words. <http://www.sparknotes.com/testprep/books/newsat/chapter15section4.rhtml>
20. The 1000 Most Common SAT Words. <http://www.sparknotes.com/testprep/books/newsat/chapter15section3.rhtml>

21. GRE Difficult Vocabulary Words, Randomly Selected. <http://www.mygretutor.com/vocabularyGREDifficult.aspx>
22. A Multiple-Level Vocabulary Assessment Tool: Measuring Word Knowledge Based on Grade-Level Materials. http://vaseproject.ucsc.edu/publications/57thYrbook_Scott.pdf

Acknowledgments

This study was performed during 2012 summer internship. Author is thankful to the ITestYou.com team for the opportunity to conduct this study, for the access to user data, for help with the statistical data analysis and discussion of results.

Author Profile

Dmitry Simakov is a rising senior at Henry M. Gunn High School of Palo Alto, California. Alongside doing statistical analysis for ITestYou, he was the organizer for 2012 Tedx and currently serves as tutor and developer at Club Academia. Dmitry led financial analysis, projections and presented winning arguments for his team, winning the “Best Company of the Year” Junior Achievement of Northern California award. Outside of business, he is a two year Varsity Water polo player and a third degree black belt in Tae Kwon Do. You can reach Dmitry by email at dmitrysmkv@gmail.com.



Product Profile

ITestYou.com is an adaptive education platform for Math, English Vocabulary, Foreign Languages and other subjects. The learning units are designed by the *publishers* and are delivered to the *students* in the game-like mobile and web applications. ITestYou.com platform provides: a) design tools and hosting for interactive content, b) mobile and web applications for content delivery, c) analytics for measuring user's engagement and subject comprehension, d) algorithmic personalization of learning process.

Privately funded, first working prototype was built in 2005; first public web site was launched in 2007. In 2011, mobile applications for Google Play and Apple AppStore were released. ITestYou.com currently offers: a) 500+ interactive math problems covering grades 1-12 of California Math Standard, all available in different languages (EN, RU, ZH, ES, NL, ...), b) advanced English Vocabulary lessons for SAT, GRE, GMAT and TOEFL standardized tests, c) basic Foreign Languages Vocabulary available in 36 different languages.