Final Report

CSE-6242 Fall 2018

Team 20: Legendata

Srinath Abburi, Swati Gupta, Dakota Hauesler,
Ali Lakdawala, Elizabeth Stanford, James Wiggins

Motivation: Adding images to text boosts understanding and retention

Research shows that seeing pictures along with reading text greatly increases both retention and understanding (Glenberger, 1992) (Hibbing, 2003). Motivated by these studies, we developed a program to increase readers comprehension, retention, and enjoyment by populating a body of text with relevant images. We accomplished this through leveraging and consolidating tools in natural language processing and image tagging.

The problem: Lack of visuals in classic texts and decreasing rates of retention.

There is an overabundance of books that are difficult to read due to a lack of images and visuals that make them less accessible to young readers. Other readers face problems retaining and comprehending text in our increasingly distracted world. (Levine, 2007)

Our objectives with this project were three-fold; create supplemental illustration to further aid text comprehension, increase information retention by providing a visual framework, and boost reader enjoyment and engagement.

Survey: What we've read on the subject

Text Summarization/Selection

Four of the most popular methodologies and algorithms for text summarization:
1) Graph-Based Rank (Mihalcea, 2004) ; Similar to Page rank but for text.
2) Sentence-Scoring (Ferreira, 2013) ; Highlight the most important sentences.
3) RAKE (Rose, 2010) ; Rapid Automatic Keyword Extraction.
4) TF - IDF (Leskovac, 2016) ; Text Frequency, Inverse Document Frequency.

We wrote and tested of each of these methods and also developed our own combinations of these approaches that we describe in detail later in the approach section.

The rest of the survey section is incorporated into the approach justification sections. We have cited literature references in that section.

Intuition: What currently exists and what we are innovating.

There are a few tools that attempt to derive connections between text and images, but most pair in the opposite direction of our approach - generating text based on an image. There are several algorithms that are developed for accurate mapping of images. (Barnard, 2003) We will use image tags mainly for validation in our project, providing a way to check image quality. (Leong, 2010)

There are limited ways to find images based on text. Most search engines have a robust image search; however, our method involves finding many images for a large amount of text, e.g. a novel (big dataset), which is a more unique task. One study had the end user select the most relevant image from a set of suggested images (Taj, 2014) and another converted single sentences into sentences made out of images (Mihalcea, 2008). Our methods will be more user-friendly and larger-scale than these. The most similar project is Image Illustration of Text using Natural Language Processing (Zainab, et al, 2015). Zainab's project aimed to improve comprehension of complex topics. They used NLP models to create specific queries to find one image per sentence. Our process differs by tackling a much larger body of text in a more abstract manner.

Approach:

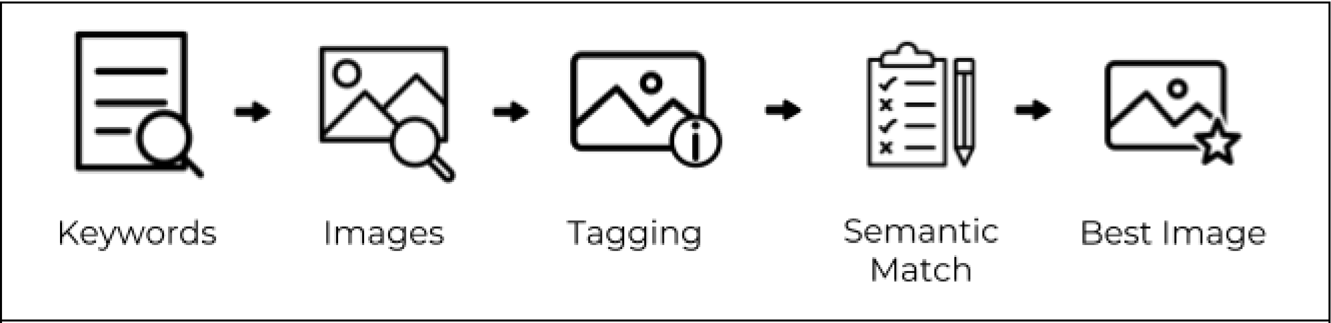


Fig 1: Approach general flow.

PDF Conversion: A messy task

PDF files are notoriously difficult to work with as there are no particular conventions on formatting. There are several open source packages designed to extract text from a pdf, including PyPDF2 and PDFMiner. However, these packages struggle to understand the context of pdf files. We explored several options to make this a reliable, automatic process using python-libreoffice conversion tools. Finally, after significant cleaning we had seven classic texts as datasets.

Our final cleaned books:

1. Sherlock Holmes - 104,420 words
2. A Tale of Two Cities - 135, 630 words
3. Tom Sawyer - 69,978 words
4. Grimm's Fairytales - 100, 843 words
5. Stories from Tagore - 47,208 words
6. The Hollowland - 14,765 words
7. The Giver - 13,478 words

Choosing an NLP Library

To find an NLP library to implement we looked into Stanford's CoreNLP, Google Cloud NLP, NLTK, and SpaCy. We opted to use both NLTK and SpaCy capitalizing on the strengths of each. We used NLTK for keyword extraction because the library is the most robust, widely trusted, easy to understand and implement, written for Python, and able to work with outside add-ins, e.g. a rake algorithm extension. SpaCy was built for speed, so we used it for Semantic match in our image validation. CoreNLP had Python extensions, but is a java program, which could make it less accessible if issues arise. Google Cloud NLP was easy to use, but there is very little documentation, or user acclaim for the tool, and it was difficult to tell what algorithms they were implementing.

Extracting keywords: Comparing different approaches

Text Frequency Inverse Document Frequency (TF-IDF)

TF-IDF scores the words based on their frequency in the current text vs. other texts. It has two components: Text Frequency and Inverse Document Frequency.

Text Frequency is the number of times the word appears in the current document.

Inverse Document Frequency is the inverse of ratio of number of documents that the given word appears in and number of documents analysed.
The TF-IDF is a product of TF and IDF.

Implementation of TF-IDF

First, we tokenized and lemmatized the text and removed stopwords. Stopwords are a set of commonly used words in a language. They are considered to be uninformative or meaningless as they appear often in all text. Some examples of stopwords are: ‘a’, ‘the’, ‘for’, ‘but’, etc. We used NLTK’s list of stopwords. We modified TF-IDF to include only nouns as potential keywords. NLTK’s part-of-speech tagging was used to identify nouns from the text. After that we calculated the Text Frequency of each word. To calculate the Inverse Document Frequency, we used the Brown Corpus within NLTK. Based on the calculated TF-IDF score of each word, we chose the top 10 words as keywords.

Drawbacks

1. Calculating IDF for each word is costly. Processing time increases significantly with larger texts.
2. The keyword results significantly depend on the kind of documents used to calculate the IDF. For example, using fictional literature to evaluate score on a news article might not give the best results.

Rapid Automatic Keyword Extraction (RAKE)

The principle behind RAKE is that it is rare to find stopwords as a part of a keyword for text. The algorithms breaks down the text into candidate keywords by partitioning them based on punctuation and stopwords. Keywords often contain multiple words.

The degree of each word is defined as the number of time that word appears in the text plus the number of other words in all the keywords it appears in. The score of each word in the keyword is the ratio of its degree and its frequency in the text. Each keyword’s score is then the sum of individual scores of the words that appear in the keyword.

The RAKE algorithm is very fast compared to the TF-IDF algorithm and can be easily applied to new domains as it does not take into account any other text.

Drawbacks

1. The RAKE algorithm gives better results for a larger text. If the sample text is small, we might end up getting similar scores for all keywords
2. It is a new algorithm. Compared to TD-IDF, it does not have many variations that can be used tailored to a specific problem

Text Rank - Graph based Ranking

This method converts a given text into a graph of individual words. The words that appear close to each other in the text are linked in the graph. Then, the score of each node is a linear function of sum of ratio of each neighbour's score and degree. Thus high degree and high neighbor's score result in high score for a keyword. Based on the score, the top 5 words were chosen (Mihalcea, 2004).

Sentence-Scoring

Sentence scoring is essentially graph based ranking based on sentences instead of words. However, it would not be efficient to use this for smaller texts. Since we plan to generate images for a small batches of text, this would not be an optimal choice

Combined Approaches

RAKE-Noun

This was a modification to our implementation of RAKE which focused on phrases with at least one noun. Our hypothesis was that these phrases are more likely to be the most important keywords within the paragraph and would provide better performance than the original theory RAKE was based on. We used NLTK's parts-of-speech tagging to identify nouns from the text. Any phrase without a noun was scored zero. For all other phrases, the scoring was kept similar to RAKE

RAKE-Text Rank

Similarly to RAKE-Noun, we modified the RAKE algorithm to consider the importance of individual words within the RAKE phrase. We created a composite score where the score of each word was multiplied with the Text Rank score of the same word, in an attempt to ensure that the highest priority words were included in the top phrases. The keyword's score was then the sum of the composite score of each individual word.

We will discuss our choice of algorithms in the experimental section.

Matching Text to Image

We translated keywords to easily locate a relevant image. We could have assigned attribute-value pairs from each description to provide search parameters. This would have required predefined attributes. (Ghani, Rayid, et al., 2006). We also could have analyzed sentiment to search for positive or negative images, but the methods presented were less effective for ambiguous texts. Thus, we chose a simple google image search using extracted keywords. These searches had approximately 70% accuracy which was sufficient for the purposes of this project. If we had more resources we would make our own collection of stock images to search from rather than relying on Google Image.

Image tagging for validation

There were two approaches we considered for image tagging. The first and most desirable was to classify with TensorFlow and Keras open source tools. We trained a neural network model on 82,000 pre-tagged images from COCO datasets on Amazon cloud computing services. Even after training on a significantly large dataset for an extensive amount of time our image-tagger was not up to par.

The second option was to use an online image tagging service. This service would essentially be free, easy to implement, and fast but less fun. We compared several services including fastphototagger, Flickr's Image tagging API and Google Cloud Vision API. Google provided us with the best results in terms of speed and accuracy to tag the image with appropriate labels. (Leong, 2010). Google charges $1.5 for every thousand images. Since we had a $300 credit, the free trial credit was sufficient for our purpose.

Hosting the project

Our main goal with the project was to make a user friendly web interface that makes our project interactive and fulfills the goals we set out to accomplish. We designed a simple web interface and ran python through Flask. We initially started with the text box which a user playground to enter their own text for processing by our algorithm. We also hosted all of our cleaned books on our website for anyone to read and enjoy with the accompanying images.

We do not store or pre-run our algorithm. Whenever a user selects a book or inputs text we do the entire approach from finding keywords -> validating -> providing the best image every single time.

We initially hosted our website on heroku, the URL is:
https://cse6242-project20.herokuapp.com/
We used Heroku's free tier server which provided us 1 dyno, and only has 512MB of RAM. This is slow and sometimes unreliable, causing our website to timeout. As such, we also attempted to deploy on AWS for scalability. It was more successful, but we did not have sufficient free credits left to keep our deployment live. We have kept our Heroku website live since it is free of cost.

Experiments/ Evaluation

Overview: Questions we sought to answer

We tested whether our project would increase user retention and comprehension based on the psychology papers (Glenberger, 1992) (Hibbing, 2003). We got survey responses from 30-60 respondents on three topics. The first survey was designed to test the claim that images increased retention / comprehension, especially using images derived using our model. The second survey aimed to capture whether users felt that the images our model produced were actually relevant to the texts displayed. The last survey paired the text with the keywords obtained from each of our models to test whether users felt that certain keywords were more representative of a paragraph.

We also developed a relevancy score to give a clear metric for how frequently our algorithms returned a relevant image, specifically on the large book datasets. While it was helpful to understand how relevant the images were for each text, we also needed to understand how frequently we were returning relevant images. Lastly, we did semantic match testing for each of our 5 algorithms to have a data metric to assist in choosing the best algorithm to use in our final product.

Selecting the best algorithm :

We tested each of the algorithm to determine which provides us the best semantic match score with the images returned. We found that objectively TF-IDF returned the best images as it returned single important keywords within each paragraph which could be separated with "OR" conditions to retrieve relevant images for each paragraph. Although RAKE was fastest and provided phrases which appeared to make more sense as summaries to paragraphs, using those phrases to retrieve images was challenging since phrases are more likely to be open to interpretation by a search engine compared to keywords that provide less variance and room for interpretation.

Thus TF-IDF provided the highest score during testing of each of the algorithms and the lowest variance which is why we chose it for final production.

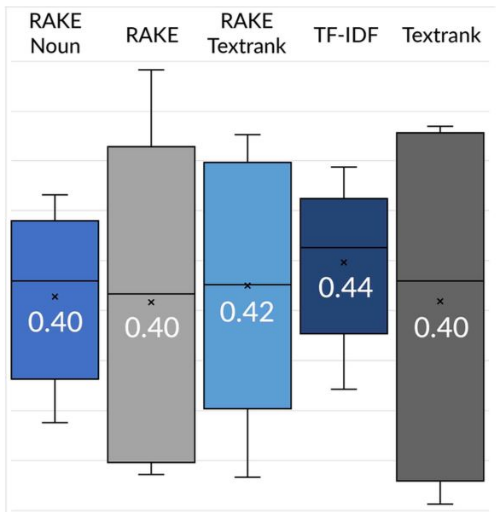


Fig 2. Semantic match scores for each of the algorithms.

Survey Results:

Our text comprehension survey used four paragraphs, of roughly equal size and reading level, and the corresponding images our algorithm returned for each paragraph. Qualtrics randomly showed users either the text alone or the test paired with the image for a total of 30 seconds. The users were then asked to answer a multiple choice setting related question and a multiple choice question about the action occurring in the text. We found that on average respondents did 6% better on the setting question when shown an image, and 21% better on the action question when shown an image.

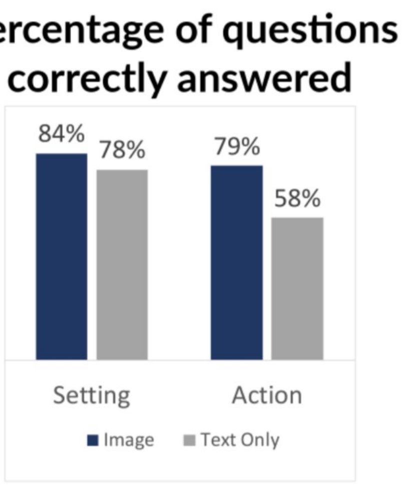


Fig 3. Survey Results for our Comprehension & Retention Survey

The second survey showed that most respondents felt that images were at least somewhat relevant to the text, with only 12% of responses rating the average image as irrelevant / very irrelevant. These results were an average of the relevancy ratings across the 5 text and image pairs in the survey.

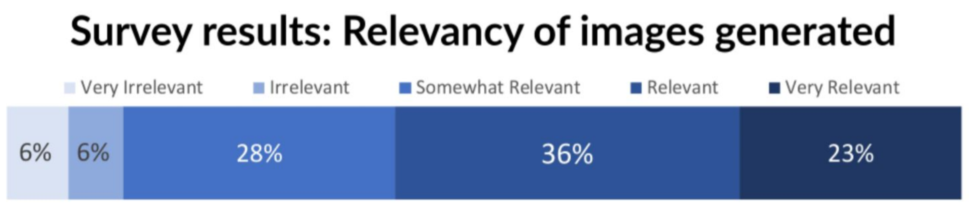
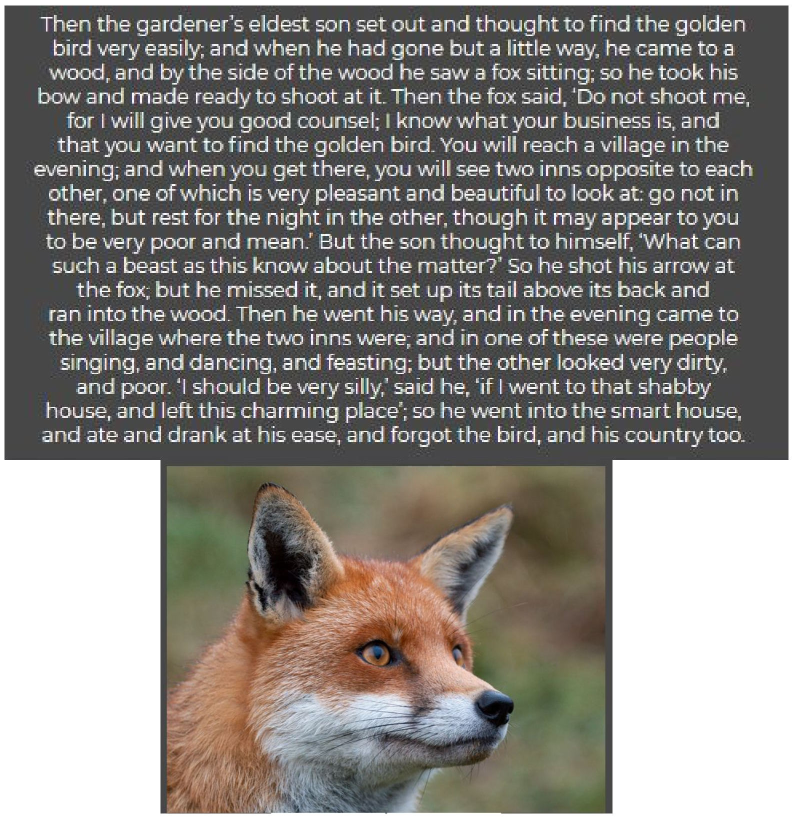


Fig 4. Survey Results for the Relevancy of Images Survey

For the third survey, we tallied all of the keywords that users marked as relevant, and then looked at what algorithm(s) produced the top 5 keywords for each paragraph. Across the five texts, we found that users gravitated the most to keywords from the noun-rake algorithm, the rake algorithm, and the rake-text rank algorithm. Text rank came in 4th, and the TF-IDF algorithm had the least selected keywords. Given that we found the TF-IDF algorithm the most effective at selecting relevant images, we were initially surprised by this finding. However, it makes sense that users are drawn to phrases (which rake is likely to produce) as being the most relevant. For instance, it is reasonable that users would rate “wind blows hard” higher than “wind”.

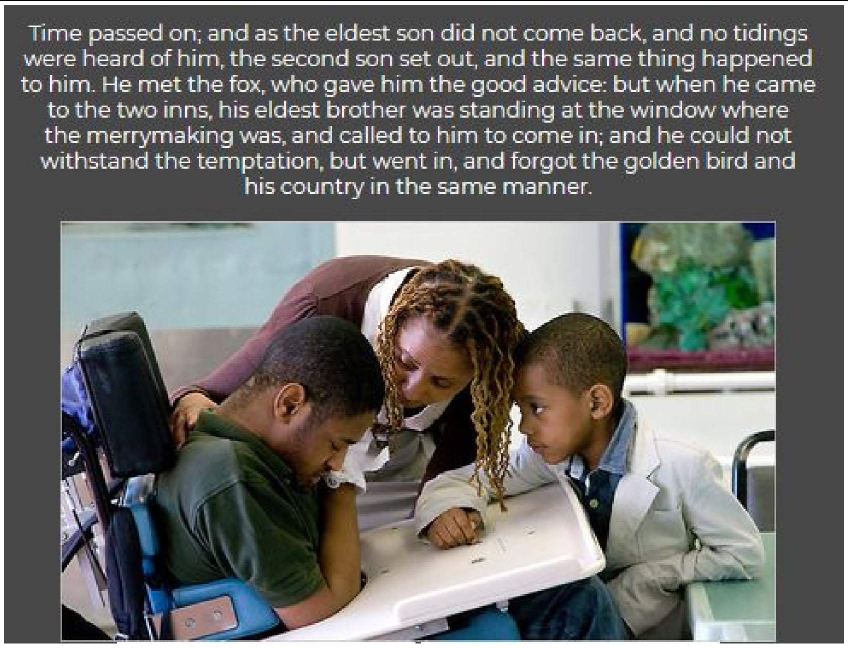
Relevancy Score

In addition to user surveys, we calculated a relevancy metric to capture the frequency with which our algorithm returned relevant scores for paragraphs in a book. We went through 10 pages of each book that we loaded and marked whether the image was relevant or not. We then took the total number of relevant images divided by the total number of paragraphs checked. We calculated a relevancy score of 56% across our preloaded books. We found that the images tended to be more accurate in certain settings though. The first few paragraphs of a book were more likely to be relevant, likely due to the descriptive nature of introductions. Likewise, texts that had a one consistent concept and more descriptive text returned better results on average.



Then the gardener's eldest son set out and thought to find the golden bird very easily; and when he had gone but a little way, he came to a wood, and by the side of the wood he saw a fox sitting; so he took his bow and made ready to shoot at it. Then the fox said, 'Do not shoot me, for I will give you good counsel; I know what your business is, and that you want to find the golden bird. You will reach a village in the evening; and when you get there, you will see two inns opposite to each other, one of which is very pleasant and beautiful to look at: go not in there, but rest for the night in the other, though it may appear to you to be very poor and mean.' But the son thought to himself, 'What can such a beast as this know about the matter?' So he shot his arrow at the fox; but he missed it, and it set up its tail above its back and ran into the wood. Then he went his way, and in the evening came to the village where the two inns were; and in one of these were people singing, and dancing, and feasting; but the other looked very dirty, and poor. 'I should be very silly,' said he, 'if I went to that shabby house, and left this charming place'; so he went into the smart house, and ate and drank at his ease, and forgot the bird, and his country too.

Fig 5. Example of a Relevant Image



Time passed on; and as the eldest son did not come back, and no tidings were heard of him, the second son set out, and the same thing happened to him. He met the fox, who gave him the good advice: but when he came to the two inns, his eldest brother was standing at the window where the merrymaking was, and called to him to come in; and he could not withstand the temptation, but went in, and forgot the golden bird and his country in the same manner.

Fig 6. Example of an Irrelevant Image

Conclusions and Discussion

We produced an end to end solution that is capable of taking an entire body of text, running it through several intermediate processes, and outputting images to accompany each paragraph in an easy-to-use website accessible to anyone. Originally we hoped to allow users to input an entire PDF, but given the complexity of cleaning PDFs, we stuck to the set of options that were pre-cleaned. We were able to successfully implement the algorithm on an entire book, as well as user inputted text, in an intuitive and streamlined manner. We found that the images returned were often more accurate for single user-input paragraphs than for entire books.

Given more time, we would continue testing keyword extraction algorithm improvements to improve our relevancy score for the books. One approach we would like to take is to take a larger segment of text for the book, and run the algorithms on those texts. Then when we ran the algorithm on individual paragraphs, to only return the paragraph image when the validation score was high enough, otherwise we would return a general image for the larger body of text. Alternately, we thought about only returning an image when the validation was above a certain threshold. Ultimately though, it is difficult to have machines truly interpret the relevant keywords that are important in attributing a relevant image to a text. We had the best results when downloading 5-10 images and having a person rate which image was the most relevant to the text. In this situation, it would be interesting to implement a training model to improve the process. All improvements aside, we were pleased that we were able to return an image relevant to the text an average of one out of every two paragraphs, and that we were able to provide a user interface that made it easy (albeit a little slow) to interact with our product.

Distribution of team member effort

All team members contributed equally to the project.

References

1. Barnard, Kobus, et al. "Matching words and pictures." Journal of machine learning research 3.Feb (2003): 1107-1135.
2. Clark, James M., and Allan Paivio. "Dual coding theory and education." Educational psychology review 3.3 (1991): 149-210.
3. Cohen, Jacob. "A coefficient of agreement for nominal scales." Educational and psychological measurement 20.1 (1960): 37-46.
4. Fan, Weiguo, et al. "Tapping the power of text mining." Communications of the ACM 49.9 (2006): 76-82.
5. Ferreira, Rafael, et al. "Assessing sentence scoring techniques for extractive text summarization." Expert systems with applications 40.14 (2013): 5755-5764.
6. Ghani, Rayid, et al. "Text mining for product attribute extraction." ACM SIGKDD Explorations Newsletter 8.1 (2006): 41-48.
7. Glenberg, Arthur M, and William E Langston. “Comprehension of Illustrated Text: Pictures Help to Build Mental Models.” Journal of Memory and Language., vol. 31, no. 2, (1992), pp. 129–151.
8. Graber, Doris A. "Say it with pictures." The annals of the American academy of political and social science 546.1 (1996): 85-96.
9. Hibbing, Anne Nielsen, and Joan L. Rankin-Erickson. "A picture is worth a thousand words: Using visual images to improve comprehension for middle school struggling readers." The reading teacher 56.8 (2003): 758-770.
10. Jurafsky, Daniel and James H Martin. “8.” *Speech and Language Processing*, Pearson, 2014.
11. Jurafsky, Daniel and James H Martin. “11.” *Speech and Language Processing*. Pearson, 2014.
12. Lee, Sangno, Jaeki Song, and Yongjin Kim. "An empirical comparison of four text mining methods." Journal of Computer Information Systems 51.1 (2010): 1-10.
13. Leong, Chee Wee, Rada Mihalcea, and Samer Hassan. "Text mining for automatic image tagging." Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics, 2010.
14. Levine, Laura E., Bradley M. Waite, and Laura L. Bowman. "Electronic media use, reading, and academic distractibility in college youth." CyberPsychology & Behavior 10.4 (2007): 560-566.
15. Mihalcea, Rada, and Chee Wee Leong. "Toward communicating simple sentences using pictorial representations." Machine translation 22.3 (2008): 153-173.
16. Mihalcea, Rada. "Graph-based ranking algorithms for sentence extraction, applied to text summarization." Proceedings of the ACL 2004 on Interactive poster and demonstration sessions. Association for Computational Linguistics, 2004.
17. Mihalcea, Rada, and Paul Tarau. "Textrank: Bringing order into text." Proceedings of the 2004 conference on empirical methods in natural language processing. 2004.

18. Miner, Gary, John Elder IV, and Thomas Hill. Practical text mining and statistical analysis for non-structured text data applications. Academic Press, 2012.
19. *Mining of Massive Datasets*, by Jurij Leskovec et al., Cambridge University Press, 2016, pp. 1–19.
20. Nahm, Un Yong, Mikhail Bilenko, and Raymond J. Mooney. "Two approaches to handling noisy variation in text mining." Proceedings of the ICML-2002 workshop on text learning (TextML'2002). 2002.
21. Nasukawa, Tetsuya, and Jeonghee Yi. "Sentiment analysis: Capturing favorability using natural language processing." Proceedings of the 2nd international conference on Knowledge capture. ACM, 2003.
22. Paivio, Allan, and Kalman Csapo. "Picture superiority in free recall: Imagery or dual coding?." Cognitive psychology 5.2 (1973): 176-206.
23. Rose, Stuart & Engel, Dave & Cramer, Nick & Cowley, Wendy. (2010). Automatic Keyword Extraction from Individual Documents. Text Mining: Applications and Theory. 1 - 20. 10.1002/9780470689646.ch1.
24. Senthil, Karthik, Abhi Arun, and Kamath S. Sowmya. "A Content-Based Visual Information Retrieval Approach for Automated Image Annotation." Progress in Intelligent Computing Techniques: Theory, Practice, and Applications. Springer, Singapore, (2018). 69-80.
25. Shi, Lei, and Rada Mihalcea. "An algorithm for open text semantic parsing." Proceedings of the 3rd Workshop on RObust Methods in Analysis of Natural Language Data. Association for Computational Linguistics, 2004.
26. Tariq, Amara, and Hassan Foroosh. "Designing a symmetric classifier for image annotation using multi-layer sparse coding." Image and Vision Computing 69 (2018): 33-43.
27. Taj, Kausar, Pooja Dutta, and Sona Mary Francis. "A Text to Image Story Teller Specially Challenged Children-Natural Language Processing Approach." 2014.
28. Uricchio, Tiberio, et al. "Automatic image annotation via label transfer in the semantic space." *Pattern Recognition* 71 (2017): 144-157.
29. Zainab, Pirani, et al. "Image illustration of text using natural language processing." International Journal of Computer Applications 115.21 (2015).

Appendix

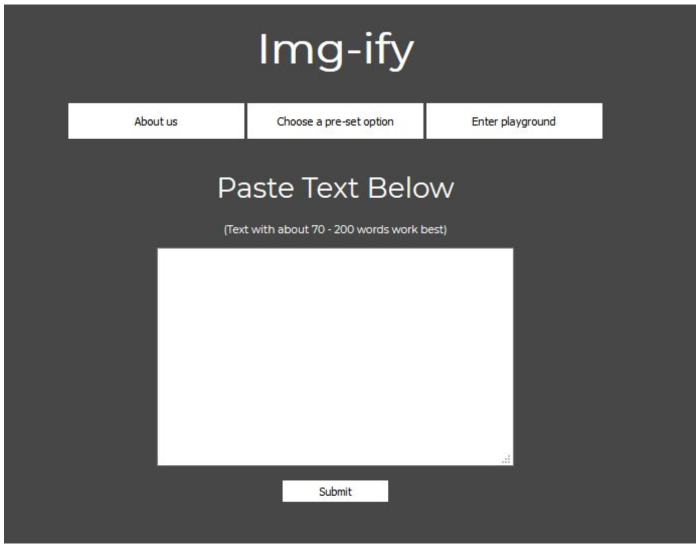
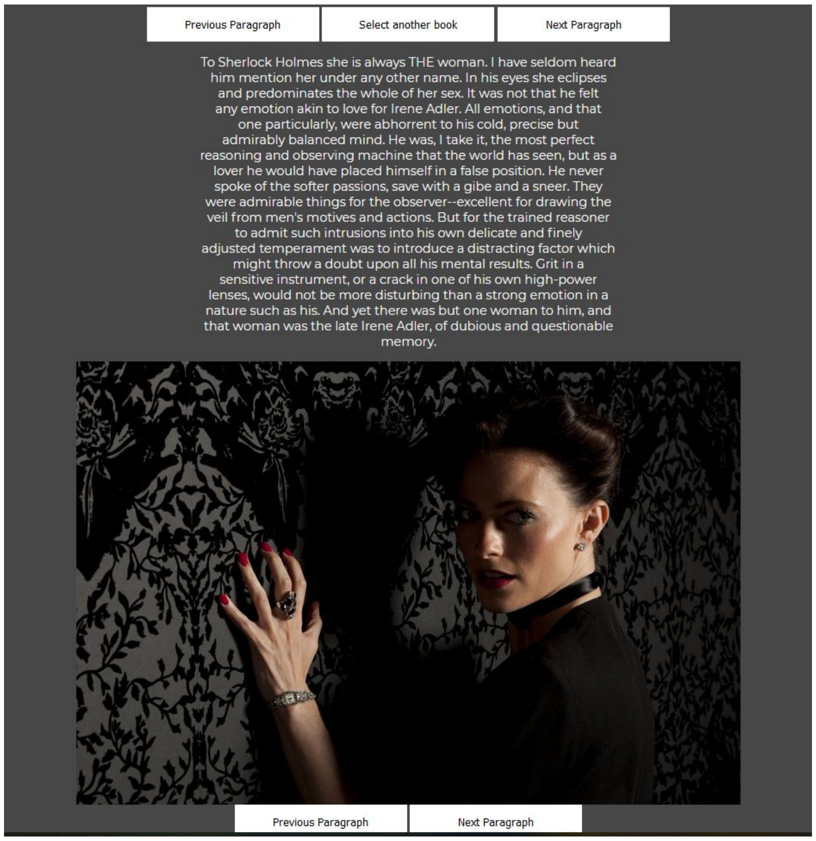


Fig 7. Screenshot of the “Free Text” portion of our website



To Sherlock Holmes she is always THE woman. I have seldom heard him mention her under any other name. In his eyes she eclipses and predominates the whole of her sex. It was not that he felt any emotion akin to love for Irene Adler. All emotions, and that one particularly, were abhorrent to his cold, precise but admirably balanced mind. He was, I take it, the most perfect reasoning and observing machine that the world has seen, but as a lover he would have placed himself in a false position. He never spoke of the softer passions, save with a gibe and a sneer. They were admirable things for the observer--excellent for drawing the veil from men's motives and actions. But for the trained reasoner to admit such intrusions into his own delicate and finely adjusted temperament was to introduce a distracting factor which might throw a doubt upon all his mental results. Grit in a sensitive instrument, or a crack in one of his own high-power lenses, would not be more disturbing than a strong emotion in a nature such as his. And yet there was but one woman to him, and that woman was the late Irene Adler, of dubious and questionable memory.

Fig 8. Screenshot of a paragraph in the Sherlock Holmes book within the preloaded book portion of our website