

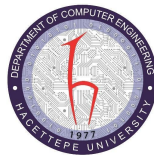
Content Rating System for Documents' Proposal



PREPARED FOR

BBM419 / DEL#1

Hacettepe University



Signed as accepted by client:

Hacettepe University,
C.E Department

PREPARED BY

Hazem Alabiad "Architect",
Abdaljalil Jarjnaze "PM",
Malek Baba "Analyzer",
Abdullah Agha "Tester"

© Code_it Team

A. PROJECT VISION:

PROJECT SUMMARY:

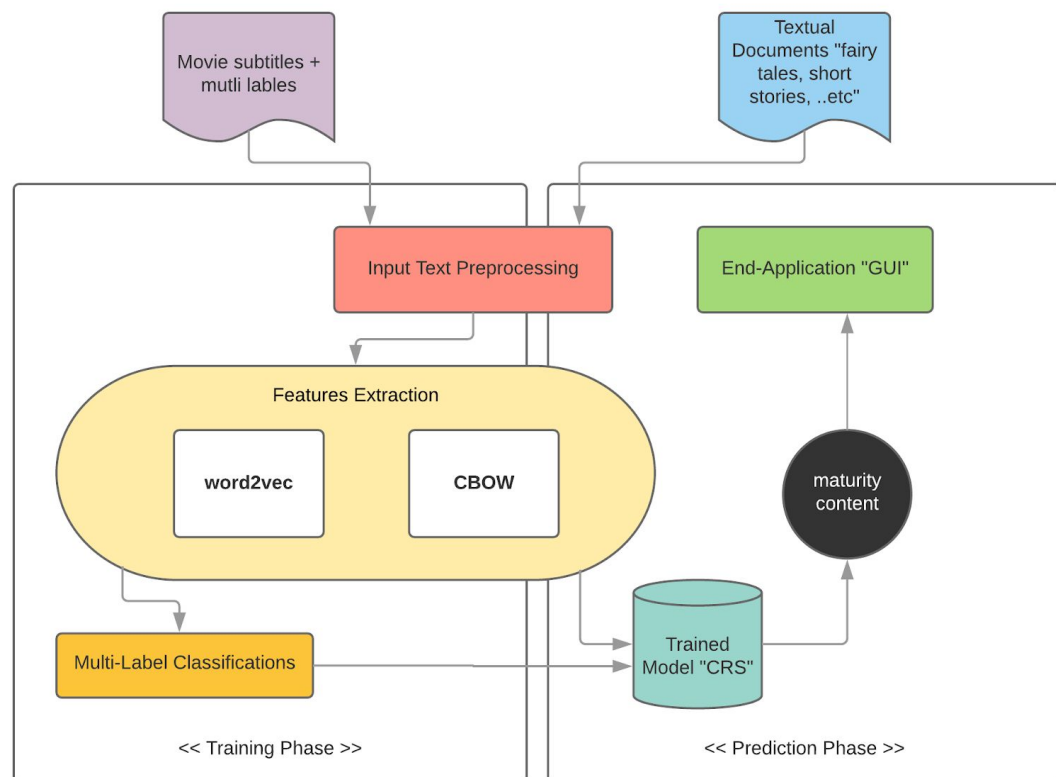
Movies and video games have been being rated for years, and for a very good reason. Different people have vastly different comfort levels when it comes to entertainment.

Content Rating is more important now than ever because society's view of what's acceptable has changed, but not all content is acceptable to everyone especially, if they are kids. Until now, there has been no way for the community to easily, effectively, and automatically rate large numbers of books for content.

Our solution is aiming to provide an easy to use, accurate, time-saving, and near cost-free solution to this problem; by building an automated system that everyone can use to rate contents of a given textual document as appropriate to a certain age of readers.

PRODUCT FEATURES:

- Our End-System is designed to be able to classify the content of the textual documents according to their maturity level based on specific criteria and rating policies to decide whether a book is suitable for children.
- The user can feed the textual documents in multiple file extensions such as txt, epub, pdf, ...etc.
- Our model applies *Transfer Learning* which trains the model on a domain then, applies the trained model and learned techniques on a varied domain.
- Basically, our Content Rating System *CRS* with the aid of *Deep Learning* learns how to judge and rate contents from the movie's subtitles, which are already classified, then decide the maturity level of textual documents that do not have any systematic or reliable rating policy.
- Context Diagram of the system:



SUMMARY OF STATE-OF-THE-ART:

As a result of our research, we found the following current methods for the content rating problem, which as we can see are rare:

- 1) **A manual solution for the problem:** where the raters are humans who are either volunteer or paid to rate the content. Like Book cave and Novelbookratings services for rating books.

Moreover, the process of rating movies and games is similar to that followed in books, where employees read the whole script of the movie or watch a long gameplay of the game to make decision about their contents. Like MPAA(Motion Picture

Association of America) for movies rating and
ESRB(Entertainment Software rating board) for games rating.

This solution has a lot of weak points; for example the high price of the rating in case of the paid method, also the latency, that may be occurred due to the difficulty to keep up the newly published contents

2) **An automated solution for the problem:** where an automatic rating system is created, using machine learning and NLP or computer vision techniques. A very astonishing solution was proposed by Bing Hu et al [1]. Where they proposed a system for rating mobile apps depending on their descriptions and generate the maturity level of the apps and the mature contents they contain. A similar solution was proposed by Chen et al [2]. However they just generated the maturity level for the apps.

In Bing Hu et al [1] solution, First, they extract novel features from App descriptions using deep learning techniques by considering the semantics of words. Second, they adapt SVM to capture label correlations in a multi-label setting. Their experiments on real-world datasets demonstrated that their approach can achieve a very high accuracy and substantially outperforms baseline methods.

In Chen et al [2] solution, they proposed the ALM algorithm. ALM is a semi-supervised learning algorithm, and it processes apps' descriptions and user reviews to determine maturity ratings. It follows a number of steps, first Building seed-lexicons for objectionable content detection. Then, Assigning initial weights to seed-terms. After that comes the Classification part, followed by Expanding seed-lexicons and adjusting weights. This research has several contributions. First, they practically examine the maturity rating policies on both Android and iOS platforms and discover the inconsistencies and ambiguities from both policies. Second, based on app descriptions and user reviews, the algorithm ALM is developed to automatically verify Android apps' maturity ratings that were based on developers' self-disclosure.

- **Chen et al [2]** solution is more robust to bias since they depend on not just the apps descriptions, but also on the user's reviews which is not the case for **Bing Hu et al[1]**.
- However, **Bing Hu et al[1]** solution was better in term of overcoming the problems that can be generated due to language synonymous and ambiguity, through leveraging feature augmentation for sensitive words. But in the case of **Chen et al [2]**, they depend on a keyword-matching approach while extracting the features.

As we notice that the automated solution is better than the manual one due to automation of the rating process which leads a much gain in the performance and cost.

INNOVATIVE ASPECTS:

We hope to develop a better solution for the textual document rating problem, than the currently available ones. By building an automated system, that not just depend on the textual documents to make a prediction(which is similar to Bing Hu et al [1] and Chen et al [2] where they depend on mobile apps descriptions), but to consider also the reviews of the readers from various websites like Goodreads and Amazon which add a robustness to our learned model.And to generate different rating policies depending on the geographic location of the reader.

POTENTIAL CONTRIBUTION(S) TO INDUSTRY AND ECONOMY:

- CRS End-Application will have many positive impacts as it eases the rating process and automates it rather than relying on the manual rating that costs a lot in terms of money, effort, time, and efficiency.
- CRS will be proposed as a supplementary service for the writers, publishers, and readers to choose convenient content.

- Making the rating process kind of free compared to the manual thing which could cost up to [\$75 per book of 250 pages, MBR rating, 2018,[Link](#)].

TECHNOLOGIES TO DEVELOP/USE AND UNIQUE ACHIEVEMENTS:

In order to develop our project, we will use the following technologies:

- Nltk v3.3, Natural Language Toolkit: for preprocessing and analyzing the linguistic structure of texts.
- Gensim library v3.6: to implement different Nlp algorithms like Word2Vec and Bag-of-words algorithms.
- Pytorch v0.4.1, deep learning framework for classification.
- Python v3.7.0 programming language.
- Android v9 (API level 28), for designing and developing the end Android application.
- Bootstrap framework v4.1.3, for front-end of the end web application.
- Node Js framework v8.0.12, for the back-end of the end web application and Android application.
- Postgres v10.5, for the database management.
- Trello tool, for work management.
- Slack messaging platform.

We hope to get a fast, highly accurate and easy to use solution to the book rating problem, that outperforms the available solutions.

METHOD TO FOLLOW:

- **Agile-Development** method is adopted while developing our product due to its flexibility and ability to accept multiple changes during the development phase.
- **The software process:**
 - 1) Specifications: Understanding the problem, and collecting information regarding the requirements in collaboration.
Output → *The Proposal*

- 2) Development: Based on the approved and discussed approaches, develop the core product. Output → **The initial version of CRS**
 - 3) Validation: Checking whether we are building “the product right” and “the right product”. Output → **The final version of CRS**
 - 4) Evolution: Releasing and maintaining the final End-product. Output → **Working and supported End-Application**
- **Project Core:** the basic steps of the project briefly consists of the followings:
- 1) Collecting Test & Train Data or Corpuses.
 - 2) Project Plan.
 - 3) Processing the Data.
 - 4) End-Product.

REFERENCES:

- [1] B. Liu, N. Z. Gong, D. Kong, H. Jin. Protecting Your Children from Inappropriate Content in Mobile Apps: An Automatic Maturity Rating Framework, in ACM, 2015.
- [2] Y. Chen, H. Xu, Y. Zhou, and S. Zhu. Is this app safe for children?: A comparison study of maturity ratings on android and ios applications. WWW '13, pages 201-212, 2013

B. PROJECT PLAN:

PROJECT GOALS:

The project's main goal is to help classifying the contents of a given document as appropriate to a certain age of readers or not. It also provides the user with live examples from the document backing up its decision about the document. By doing so, we avoid the need for manual reviewing of the documents by employing special people to read and rate the content, thus automating the process and saving effort and resources.

The project also aims to help documents' writers figuring out the level content, maturity wise, presented in their work. Moreover, it avoids them any unwanted criticism by their readers, and guide them to restore their work back on track.

PROJECT MILESTONES AND OBJECTIVES:

#	Milestone	Primary Objective	Due Date	Project Deliverable (if any)
1	propose a set of possible solutions.	Obtain the widest range of view about potential approaches.	7/Nov/2018	
2	Getting the solutions' steps and approaches discussed by the advisor	Using discussion feedback for future decisions.	13/Nov/2018	
3	Starting off the models training phase	Getting the candidate models trained and ready for testing.	28/Nov/2018	
4	Starting off the models testing phase	Obtaining the test result for each model.	5/Dec/2018	

5	The announcement of the testing results, and adoption of the single, final approach.	Assembling different test results in a single view.	10/Dec/2018	
6	Building up the web app's UI	Facilitating the usage of the product for the user.	20/Dec/2018	The product's user interface represented as a web application.
7	Release initial version of the final product.	Getting users' reviews.	22/Dec/2018	The initial version of the web application
8	Release final version of the final product.	Putting the product into actual use.	6/Jan/2019	The final version of the web application

PROJECT PRACTICES AND MEASURES:

#	Task	Task Description	Responsible Team Member	Start Date	Finish Date	Success Criteria
1	Researching Models.	Involves researching solutions and tools, proposing new ones and discussing them with the advisor for further insight.	All members (working in a parallel manner).	30-Oct-2018	19-Nov-2018	Picking a set of suitable models for the problem
2	Data collection and cleaning.	Collecting training and testing data. Performing cleaning operations on them to remove any possible fuzziness.	Tester, analyzer.	12-Nov-2018	18-Nov-2018	Gathering a sufficient amount of useful data.
3	Training models and selection of the most suitable one.	Building the models and training them with the obtained data. Decide on the best fit model based on testing results.	Project manager, architect, analyzer.	20-Nov-2018	1-Dec-2018	Obtaining a clear view of every model's performance and the ability to pick the best one.
4	Deployment of the model.	Programming an API for the model.	Project manager, analyzer.	2-Dec-2018	15-Dec-2018	Having a working API that fully delivers the result of a given request

5	GUI & the functionality of the app “front and back end”	Design the app’s UI. Establish the connection between the UI and the API of our trained model.	Tester, architect, analyzer.	3-Dec-2018	3-Jan-2019	The ability to submit a book evaluation request and receiving a correct result through the application.
---	---	--	------------------------------	------------	------------	---

PROJECT ORGANIZATION:

Hazem Alabiad (architect)	Task
#1	Research and discuss solutions, tools.
#2	Defining the post-research algorithms and the roadmap to follow while building the model.
#3	Building, training and testing models.
#4	Program the web application’s front end and designing its UI.

Abdaljalil Jarjnaze (project manager)	Task
#1	Research and discuss solutions, tools.
#2	Setting up the Project Plan, measuring the progress of the project to the set Project Plan, and handling the potential risks.
#3	Building, training and testing models.
#4	Programming an API for the trained model.

Malek Baba (analyzer)	Task
#1	Research solutions, tools.
#2	Collect training, testing data.
#3	Building, training and testing models.
#4	Programming the API and the application's front and backend

Abdullah Agha (tester)	Task
#1	Research solutions, tools.
#2	Collect testing, training data.
#3	Reporting and comparing the test results of the models “if more than 1 is available” to decide the most accurate and efficient one to adopt.
#4	Programming the back end of the web application.

PROJECT BUDGET:

Currently, the project has no clear budget, as it is not a crucial matter because the used technologies and tools are mainly free to use. However, we might ask for a grant from an institute if we were forced to use a paid service.

PROJECT RISKS:

#	Risk	Risk Description	Probability	Effect	How to handle its occurrence? (Plan-B)
1	Software Requirements risk	Ambiguity or misinterpreting in some requirements.	moderate	Producing undesired deliverables.	Urgent meetings with stakeholders to correctly understand the requirements and working on applying them
2	Project Plan risk	possible wrong estimations in terms of required effort for a task in Project Plan.	high	Delivering the product late and not being on schedule.	Revising the project plan and adjusting it, based on new estimates, trying to keep the deadline as is.
3	Software Design document & Coding Standard risks.	incompatibility issues between the used frameworks, languages, or technologies.	moderate	Having problems running the application on different platforms	Exploring and checking the official developer website in order to know which technologies can work well with theirs without problems.
4	Software Test result failures risk	Failure in software final test and having unexpected results.	moderate	Latency in delivering the final product.	Applying iterative component testing and detecting possible errors to fix them before integrating. Also, having a test expert to perform a supervised test.
5	Data size and quality risk	The inability to obtain useful and sufficient amounts of data	moderate	Models not trained properly, thus not getting optimal results.	Changing the search strategy. Employing more members to the task.

Tools and Technologies used to do and facilitate the mission:

- Nltk v3.3, Natural Language Toolkit. <https://www.nltk.org/>
- Gensim library v3.6. <https://radimrehurek.com/gensim/>
- Pytorch v0.4.1, deep learning framework. <https://pytorch.org/>
- Python v3.7.0 programming language. <https://www.python.org/>
- Android v9 (API level 28). <https://developer.android.com/>
- BootStrap framework v4.1.3. <https://getbootstrap.com/>
- Node Js framework v8.0.12. <https://nodejs.org/en/>
- Postgres v10.5, for the database management.
<https://www.postgresql.org/>
- Slack platform. <https://slack.com/>
- Trello platform. <https://trello.com/>
- Google docs: docs.google.com