**Detecting Abrasive Online User Content in Question and Answer Forums**

Team 3 – Deepak Rajan, Indranil Pal, Yasser Parambathkandy

**Dr. Lindi Liao**

**George Mason University**

**AIT614 Big Data Analytics**

**03/12/2023**

**Introduction**

**The amount of content produced online every day is growing at a rapid pace as more of our world is digitized. Instead of paying attention to the content itself, social media platforms have been emphasizing users to contribute to content creation. Any large website faces the challenge of locating and removing "toxic" information. By the time a user detects any hazardous content and the website administrators takes any action, the content may already have caused a great deal of harm. Question and answer forums like Quora that focus on providing credible information should take steps to moderate these types of posts. Members who regularly produce insincere content or who do not act in good faith are informally known as “trolls”. Insincere questions from trolls typically have political, non-neutral tone, or convey an extreme view of a group. The major objective of this project is to predict whether a question is sincere or insincere using machine learning techniques. There are some existing literature work on this topic that have attempted to solve this issue using word embeddings. Word embeddings provide a representation of a word in a numeric format which is better suited for machine learning. Recent work has demonstrated strong transfer task performance using pre-trained sentence level embeddings compared to word embeddings. This research will improve upon existing strategies by performing a comparative study between word and sentence embeddings in a deep learning construct.**

**Related Work**

**The various works done in the area of identifying insincere questions using similar methodologies are as follows: As pointed out by Nima, Prateek et al. (2019), in their research regarding insincere question classification, one of the most important parts in text classification and mining is preprocessing using techniques like lemmatization. In Yoon Kim’s (2014) paper, he has described the series of experiments performed with CNN on pre-trained word-vector for sentence classification. This CNN model improves upon the state of the art on four out of seven tasks which include sentiment analysis and question classification. In the work by S. T. Indra et al., (2016), the authors have used Logistic Regression models to classify tweets according to the topic. The tweets are transformed into vectors, which is one of the pre-procesing steps used in this project (Fan, H. and Qin, Y, 2018). A wide variety of classification techniques have been used to document classification. The models developed include Naive Bayes, Logistic Regression, Support Vector Machine (SVM), an ensemble of Naive Bayes and Logistic Regression and Random Forest. While all of the models provided high accuracy rates, the F1 score and ROC provided more meaningful model performance metrics due the data being imbalanced (Aggarwal, C.C., Zhai, C., 2012). Text classification was used for identifying tweets related to suicides (F. CHIROMA, H. LIU and M. COCEA, 2018). This was done with the motive of reducing the negative impact of tweets. Amongst all models, Decision Tree performed the best. Abdalraouf Hassana and Ausif Mahmood (2017) at the University of Bridgeport have performed research on Deep Learning for Sentence Classification. The paper observed that most of the machine learning algorithms require input to be denoted as a fixed-length feature like “bag of words”. They ignore the semantics of word and loss ordering of words. According to Prudhvi Raj, Dachapally and Srikanth Ramanam, typically CNN is used for image classification. CNN for NLP is not used often and is completely intuitive. They used two-tier CNN that classifies questions into their main and subcategories (Dachapally Prudhvi Raj, 2018).**

**This project extends related work: The common aspect in most of the existing research work is the use of word embeddings to represent text before feeding to machine learning techniques. The models implemented in this project use *sentence embedding* with the goal of obtaining better performance compared to word-based embedded modeling approach. Recent work has demonstrated strong transfer task performance using pre-trained sentence level embeddings compared to word embeddings. Pre-built sentence encoders developed recently have been used to vectorize questions. A comparative study between the two embeddings when subjected to supervised learning and neural network-based learnings is presented in this paper.**

**Objectives**

**The primary objectives of this project are:**

1. **Using machine learning techniques, develop model to detect insincere questions posted on question and answer forums**
2. **Compare the performance of the models developed using pre-trained word and sentence embeddings**
3. **Provide an automated way to flag incoming insincere questions using trained models which can then be used by website administrator for further review and action**
4. **Develop an end-to-end pipeline that is not limited to the Quora dataset, in a a cloud based environment**

**Proposed selected dataset**

**The Quora provided training dataset contains 1,303,122 questions. The dataset contains three labels: qid, question\_text, and target. The third feature, target, is a binary value and is labeled as 1 for an insincere question or 0 otherwise. Similarly, Quora has also provided a test dataset that contains 375,806 questions which only have two labels: qid, question\_text. This dataset is posted on Kaggle as a competition that doesn’t allow external files to receive score on test data. The project’s primary objective is to study sentence embedding and that is considered an external source and, as such, cannot use test data. For this reason, the training data has been split into train and test data and all model development and evaluation will be performed on this generated test data. The training data is further split for validation purposes. Dataset analysis shows that there are 1,225,312 questions that are sincere, labeled as 0, and 80,810 questions are insincere, labeled as 1. The data clearly exhibits class imbalance. The imbalance nature has been taken into consideration during model parameter selection and evaluation. The imbalance has been retained as it is because it reflects the daily Quora website activity where toxic questions appear much less frequently than genuine questions.**

**Description of proposed system**

**The training and prediction of detecting abrasive troll questions is achieved in the architecture utilizing various tools like Spark for data processing, Global Vectors for Word Representation (GloVe) for word embeddings, Amazon EMR (Elastic MapReduce) for cloud-based computation, MongoDB for database storage, and Amazon S3 for file storage. By combining various machine learning techniques and tools, this system can effectively classify questions and make accurate predictions. The detailed end-to-end architecture is shown in figure 1.**

***Training Data and Pre-processing:***

**The training data from Quora is pre-downloaded and pre-processed before training. Preprocessing of data should yield better quality of word embeddings on the dataset. Data preprocessing is a task that includes preparation and transformation of data into a suitable form. Data preprocessing aims to reduce the data size, find the relation between the data, normalize data, remove outliers and extract features for data. The steps necessary to carry out in preprocessing includes - removing punctuations, numbers, stop words like “is”, “are”, and lemmatization. The goal of lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form (Christopher D. Manning, 2008). If a question contains “caring” as a word, lemmatization will convert it to “care” and if another question has a word “cared”, that is also converted to “care”. This might mean the two questions may have similar context. This pre-processing step is required only for word embeddings. Sentence embeddings will be generated without any pre-processing to preserve the context as much as possible.**

A picture containing timeline

Description automatically generated

Figure : System Architecture

*Word Embeddings - GloVe:*

Word embedding is the language modeling technique in natural language processing where individual words or phrases are represented as a real-valued vector that can capture the context of the word in a document, semantic and syntactic similarities, and relation with other words (Brownlee Jason, 2019). There are various word embeddings available like the classic TF-IDF, word2vec, and GloVe. GloVe was developed as an open-source project at. GloVe is an unsupervised learning algorithm for obtaining vector representations for words. This model was created at Stanford university where training was performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. While other word embedding models specifically Word2Vec ignores the fact that some context words occurs more often than others and also, they only take into consideration the local context and hence failing to capture the global context.

*Sentence Embeddings – Universal Sentence Encoder:*

An illustration of sentence embedding being better for context awareness for question classification is provided as follows. The question “Is the duck swimming?” and “Should I duck when a cow attacks?” should be treated as contextually two different questions. The word “duck” has different meanings based on the context. Word embedding may treat the two equally, but sentence embedding does not. Google research department created models for encoding sentences into embedding vectors (Daniel Cer, 2018). “Universal Sentence Encoder” model has been used for getting embeddings of all questions in Quora’s dataset. Classification will be performed by finding semantically similar sentences. The encoder generates a 512-dimensional vector as output sentence embedding for each question. The pre-trained Universal Sentence Encoder is publicly available in Tensorflow-hub. It comes with two variations – 1) Trained with Transformer encoder and 2) Deep Averaging Network (DAN). The two have a trade-off of accuracy and computational resource requirement. While the one with Transformer encoder has higher accuracy, it is computationally more intensive. The one with DAN encoding is computationally less expensive and with little lower accuracy. This system has used the transformer encoder version of Universal sentence encoder.

*Spark, Spark MLlib, and Machine Learning Models:*

Spark is primariy chosen for its capability of processing large volumes of data both for training and handling real-time prediction for submissions from thousands of concurrent users. Spark MLlib can help simplify the machine learning process by providing pre-built algorithms and functions for common machine learning tasks. Spark and Spark MLlib have been chosen for this application because they offer scalability, distributed computing, integrated libraries, versatility, and are open-source. These features make them well-suited for handling the large volumes of incoming questions, complex computations, and machine learning tasks required in this application. This application uses Logistic regression as one of the supervised modelling technique. Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take values such as true/false. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 for sincere question or 1 otherwise)

*Amazon EMR, S3, API Gateway:*

EMR is a cloud-based service provided by Amazon Web Services (AWS) that allows for easy deployment and scaling of Spark clusters. EMR can help this application in the following ways – 1) uses a pay-as-you-go pricing model, and this makes it a cost-effective solution for running Spark clusters in the cloud; 2) makes it easy to deploy and manage Spark clusters in the cloud: 3) Integration with other AWS Services such as S3 and MongoDB, which are also being used in this application. This makes it easy to move data between services and to use EMR in conjunction with other AWS services; 4) provides several security features, such as encryption and secure access control, which are important for applications dealing with sensitive data. The other AWS components S3 is used to store model files. The API gateway is the interface between the end user and this application. User question submission is received via the gateway and sent for weeding out troll questions.

*MongoDB:*

MongoDB's flexible data model allows for easy storage and retrieval of unstructured data, such as question and answers. This makes it an ideal choice for this application dealing with text data that may have varying structures and formats. MongoDB is designed to scale horizontally, which means that it can handle large amounts of data and increased workload which is expected out of a question and answer forum application. Another reason to choose MongoDB for this application is that it has a connector for Spark, which allows for easy integration between the two platforms. This means that data can be easily moved between Spark and MongoDB, making it easier to process, train, and predict questions sincerity.

**Machine learning Pipeline**

**The machine learning workflow for of processing the input dataset, processing the questions, training the models with sentence and word embeddings is shown in figure 2.** The machine learning process will perform the following steps:

1. The training data is pre-downloaded as a manual step, and loaded into the EMR cluster.
2. The question text is cleaned and pre-processed after loading it as a Spark dataframe
3. The clean-up process involves removing stop-words and lemmatization.
4. For each question, GloVe and USE are used to create word and sentence embeddings respectively.
5. Use MLlib and perform logistic regression for each of the word and sentence embeddings
6. Compare and save the best model to S3 for real-time incoming question prediction

**Diagram

Description automatically generated**

Figure 2: Modeling with Machine learning pipeline

**Worflow to flag abrasive troll questions**

Figure 3 is the workflow to predict incoming questions in the forum and flag inappropriate ones.

Graphical user interface

Description automatically generated with low confidence

Figure 3: Real-time prediction

This workflow involves the following steps:

1. User (sincere and trolls) submit questions in the forum seeking answers or provoking reactions
2. AWS API Gateway receives the question and is routed to EMR cluster
3. The spark based application uses the best saved model from S3 and performs prediction
4. The prediction is then stored in MongoDB for administrator to review
5. Administrator can further review all of the flagged questions or on a certain threshold

**Proposed development platforms**

**The solution will be developed using PySpark and Spark MLlib and deployed in an AWS environment composed of AWS EMR, S3, API gateway, Mongo DB component. The details of the development tools and AWS components were described in the architecture section. In terms of component sizing, the starting specification of EMR cluster will be at least 4 vCPUs and 16 GB of memory. The EMR provides pre-configured Amazon Machine Images (AMIs) with various software packages installed, including Apache Spark and others. AWS SDK will be used within pySpark to transfer models from EMR to S3. Github will serve as the source code repository for collobarative development between different team members. The core development will be performed in a Windows machine and code pushed to github. As a stretch goal, github actions pipeline may be setup for automated deployments from github to AWS.**

**Project tasks and timeline**

**The project tasks and timelines are divided into multiple phases**

|  |  |  |  |
| --- | --- | --- | --- |
| ****S.no**** | ****Task**** | ****Assigned To**** | ****Timeline**** |
| **1** | **Finalize project scope and objectives** | **Yasser Parambathkandy**  **Deepak Rajan**  **Indranil Pal** | **3/6/2023 - 3/11/2023** |
| **2** | **Research GloVe and USE** | **Yasser Parambathkandy** | **3/12/2013 - 3/18/2023** |
| **3** | **Research AWS components – EMR, S3, API, cloudformation** | **Deepak Rajan**  **Indranil Pal** | **3/12/2023 - 3/18/2023** |
| **4** | **Data analysis and preprocessing using Spark** | **Deepak Rajan**  **Yasser Parambathkandy**  **Indranil Pal** | **3/19/2023 - 3/25/2023** |
| **5** | **Generate word embeddings and train using logistic regression** | **Yasser Parambathkandy** | **3/26/2023 – 4/8/2023** |
| **6** | **Generate sentence embeddings and train using logistic regression** | **Indranil Pal** | **3/26/2023 – 4/8/2023** |
| **7** | **Develop code to receive real-time question and perform prediction using saved model** | **Deepak Rajan** | **3/26/2023 – 4/8/2023** |
| **8** | **Deploy application in EMR cluster** | **Yasser Parambathkandy** | **4/9/2023 - 4/15/2023** |
| **9** | **Configure API gateway and S3 and other AWS components using Cloudformation template** | **Indranil Pal** | **4/9/2023 - 4/15/2023** |
| **10** | **Develop minimal UI for question submission and display predictions** | **Deepak Rajan** | **4/9/2023 - 4/15/2023** |
| **11** | **Testing end-to-end and review** | **Deepak Rajan**  **Yasser Parambathkandy**  **Indranil Pal** | **4/16/2023 – 4/24/2023** |
| **11** | **Project documentation and presentation** | **Deepak Rajan**  **Yasser Parambathkandy**  **Indranil Pal** | **4/16/2023 – 4/24/2023** |

**References**

**Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008, “Stemming and lemmatization”, pp. 86-91.**

**Brownlee, Jason (2019, August 07). What are word embeddings for text? Retrieved March 11, 2023, from** <https://machinelearningmastery.com/what-are-word-embeddings/>

**Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.**

**Nima, Prateek, Nikita Parab, Akshay Mungekar, Sanchit Pereira. (2019). Quora Insincere Questions Classification.**

**Y. Kim, ―Convolutional neural networks for sentence classification,‖ arXiv preprint arXiv:1408.5882, 2014**

**S. T. Indra, L. Wikarsa, and R. Turang, “Using logistic regression method to classify tweets into the selected topics,” in 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Oct 2016, pp. 385–390**

**Fan, H. and Qin, Y., “Research on Text Classification Based on Improved TF-IDF Algorithm” in Proceedings of the 2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018), May 2018, pp. 501–506**

**Aggarwal, C.C., Zhai, C. (2012). A Survey of Text Classification Algorithms. In: Aggarwal, C., Zhai, C. (eds) Mining Text Data. Springer, Boston, MA. https://doi.org/10.1007/978-1-4614-3223-4\_6**

**F. CHIROMA, H. LIU and M. COCEA, "Text Classification For Suicide Related Tweets," 2018 International Conference on Machine Learning and Cybernetics (ICMLC), 2018, pp. 587-592, doi: 10.1109/ICMLC.2018.8527039.**

**A. Hassan and A. Mahmood, ―Deep learning for sentence classification, 2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT). 2017**

**Dachapally Prudhvi Raj, ―In-depth Question classification using Convolutional Neural Networks, arXiv preprint arXiv:1804.00968, 2018**

**EMR. (n.d.). Retrieved March 16, 2023, from** <https://aws.amazon.com/emr/features/?nc=sn&loc=2&dn=1>

**Quora insincere questions classification. (n.d.). Retrieved March 16, 2023, from** <https://www.kaggle.com/competitions/quora-insincere-questions-classification/overview>

**Pennington, J. (n.d.). Retrieved March 16, 2023, from** <https://nlp.stanford.edu/projects/glove/>

**Appendix**

**A sample of the dataset from** <https://www.kaggle.com/competitions/quora-insincere-questions-classification/data>

|  |  |
| --- | --- |
| Question\_text | target |
| How did Quebec nationalists see their province as a nation in the 1960s? | 0 - Sincere |
| What is the dumbest, yet possibly true explanation for Trump being elected? | 0 - Sincere |
| How were the Calgary Flames founded? | 0 - Sincere |
| What is the best email auto-reply software bot? | 0 - Sincere |
| What is the process of regeneration of brain cells? | 0 - Sincere |
| How do I marry an American woman for a Green Card? How much do they charge? | 1 - Insincere |
| Why do these idiots keep listening to Steve Harvey for relationship advice? | 1 - Insincere |
| How do I lobotomize myself? | 1 - Insincere |
| Why so many questions? Do you people not have Google? | 1 - Insincere |