Spam Detection Using Machine Learning

Authors:

Supriya Yerakaraju, [1, a)] P Gopala Krishna, [1, b)] N V Ganapathi Raju, [1, c)]
[1] Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad-500090, Telangana State, India
[a)] supriya.yerakaraju98@gmail.com

[b)] gopalakrishna@griet.ac.in

[c)] nvgraju@griet.ax.in

Abstract: The rapid development of technologies and the widespread use of mobile phones has resulted in various risks, such as spam and phishing attacks. Machine learning is one of the most widely utilized and well-known technology for detecting spam in online chats. With the help of machine learning techniques such as Naive Bayes, logistic regression, and other classifiers, the spam detection model has been developed in this project. The prediction and classification of spam information from the user data will be separated using various data analysis techniques. A final robust model will be developed for the enhancement of information categorization from spam, which will result in the storage of secure data in the device. The study investigates the various machine learning techniques and the benefits of the Transformer model in this sector. Finally, the research came to a close by advocating that certain fields adopt these machine learning approaches in order to obtain more accurate findings in the future.

Keywords: Spam, HAM, Tweets, NLP, KNN, Decision Trees, Naïve Bayes, Logistic Regression, etc.

Chapter 1 -Introduction

1.1 Problem Statement

SMS spam has been increasingly prevalent in recent years. SMS spam is defined as any fictitious text message that is distributed via a mobile network without the recipient's knowledge. They are a source of concern for users [2].

68 percent of mobile phone users [3] have been impacted by SMS spam, according to a recent survey. SMS spam can incorporate

malicious actions such as smishing, which is a type of phishing. Smishing is a cyber-security assault on mobile users that involves sending spam SMS messages that contain a link or malicious software, or both, in an attempt to deceive the recipient. It is made up of two words: SMS and Phishing [3] that are joined.

Short messaging service in mobile phones is used by humans for communication and business purposes. SMS has just surpassed all other data services as the most widely utilized data service in the world. SMS is vital for corporate communications since the world sent 8.3 trillion SMS messages in 2017, and the amount of SMS messages sent monthly is 690 billion, according to the International Telecommunications Union [1]. SMS spam has been increasingly prevalent in recent years.

SMS spam is defined as any fictitious text message that is distributed via a mobile network without the recipient's knowledge. They are a source of concern for users [2]. 68 percent of mobile phone users [3] have been impacted by SMS spam, according to a recent survey. SMS spam can incorporate malicious actions such as smishing, which is a type of phishing. Smishing is a cyber-security assault against mobile phone users that involves sending spam SMS messages that contain a link, malicious software, or both in order to

deceive the recipient. Phishing [3] and SMS [3] are words that are combined to form Smishing.

1.2 Ambition

In our project, we aim to detect Spam Messages using Machine Learning techniques of K Nearest Neighbors, Decision Trees, Naive Bayes, Logistic Regression, etc.

1.3 Objectives

Our objectives in this project are as follows:

- To discover the works done by other researchers.
- To identify the depth of the problem.
- To study all the algorithms of KNN, Decision Trees, K Nearest Neighbors, Logistic Regression algorithms, etc.

1.4 Significance of the study

Over the past few years, spam, also known as unsolicited commercial bulk emails, has risen in popularity and has become a big source of irritation on the internet, especially for those who are not familiar with the term. The spammer is the individual who is in charge of sending unsolicited emails to recipients. Individuals that behave in this manner obtain email addresses from a variety of sources, including websites, chat groups, and computer viruses. With spam accounting for more than 77 percent of all worldwide email traffic every year, spam is becoming a more serious problem on an annual basis. Those who receive spam emails that they did not request find them to be a very unappealing experience.

Chapter 2 – Literature Survey

2.1 Related Works

A model named "Smishing Detector" was proposed in [7] to recognize smishing messages with a lower bogus positive rate than the present status of the craftsmanship model. The model that has been proposed is partitioned into four modules. This module's goal is to analyze the substance of instant messages and recognize perilous data utilizing the Naive Bayes arrangement procedure, which will be utilized in the ensuing modules. In the subsequent case, the URL held inside the sends is analyzed. The third module is devoted to looking at the source code of the site that has been alluded to in the messages. An APK download identifier is remembered for the last module, and its motivation is to decide if a malevolent document is downloaded when the URL is called. On this model, the trial tests led by the creators uncovered a precision of 96.29 percent as indicated by the consequences of the tests.

Late examination [2] by Roy et al. proposed the utilization of profound figuring out how to sort SMS messages as Spam or Not-Spam because of their substance. The objective of their methodology is to join two profound learning procedures: CNN and LSTM, to accomplish the best outcomes. Basically, the objective is to order instant messages and recognize those that are spam and those that are not spam. It was important to analyze the proposed approach against other AI calculations, like the Naive Bayes and Random Forest calculations, as well as the Gradient Boosting, Logistic Regression, and Stochastic Gradient Descent calculations, to assess its presentation.

2.2 Insights from Other Researchers

When contrasted with other AI models, the CNN and LSTM models performed much better, as indicated by the obtained information. The "S-Detector" model was presented by Joo et al. [20] in one more paper that was planned to distinguish smishing interchanges and was named "S-Detector." There are four modules in this methodology, which are as per the following: a part for checking SMS action, an analyzer for examining the substance of the SMS, a determinant for ordering and obstructing Smishing instant messages, and an

information base for putting away the SMS information. The characterization approach utilized in this model is the Naive Bayes strategy. As per Jain and Gupta [24], a standard-based order system to distinguish phishing SMS was introduced. Their strategy has brought about the recognizable proof and sifting of nine models that might recognize phishing SMS from typical SMS. The creators' trial trying uncovered a genuine negative pace of almost 100% and a genuine positive pace of 92% in their actual negative and genuine positive tests.

To recognize smishing messages, Sonowal and partners [22] introduced an algorithmic methodology, named "SmiDCA," because of AI procedures for the distinguishing proof of smishing interchanges. The originators of the model chose to utilize relationship methods to remove the 39 most critical properties from smishing messages for incorporation in the model. Following that, they applied four AI classifiers to their model to assess its general presentation. Irregular Forest, Decision Tree, Support Vector Machine, and AdaBoost were the four classifiers utilized in this review. The exactness of this model utilizing Random Forest Classifier was 96.4 percent because of the consequences of the exploratory appraisal. A component-based procedure to recognizing smishing correspondences was introduced in [23], as per the creators.

When contrasted with other AI models, the CNN and LSTM models performed much better, as indicated by the gained information. The "S-Detector" model was presented by Joo et al. [20] in one more paper that was expected to distinguish smishing interchanges and was named "S-Detector." There are four modules in this methodology, which are as per the following: a part for checking SMS action, an analyzer for investigating the substance of the SMS, a determinant for characterizing and impeding Smishing instant messages, and a data set for putting away the SMS information.

The grouping approach utilized in this model is the Naive Bayes strategy. As indicated by Jain and Gupta [24], a standard-based order system to distinguish phishing SMS was introduced. Their procedure has brought about the ID and separating of nine standards that might recognize phishing SMS from ordinary SMS. The creators' trial trying uncovered a genuine negative pace of close to 100% and a genuine positive pace of 92% in their actual negative and genuine positive tests.

This procedure removes 10 attributes that, as per the creators, can be utilized to distinguish counterfeit signs from ham transmissions. To assess the presence of the proposed method, the elements were carried out on a benchmarked dataset and tried utilizing five grouping calculations to perceive how well they performed. The exploratory evaluation uncovered that the model can identify smishing messages with a genuine positive pace of 94.20 percent and a general precision of 98.74 percent in 94.20 percent of cases.

Chapter 3 Methodology and Requirements

3.1 Existing System

Nowadays, attackers have discovered that SMS is a convenient method of communicating with their victims [4]. The majority of those who fall prey to smishing and phishing assaults are smartphone users [5].

A link or direct contact with victims through SMS messages [6] is used by the attackers in an effort to acquire confidential information from users, such as credit card numbers, bank account data, and other personal information. Furthermore, when compared to email spam, which is supported by modern techniques of spam filtering [7], SMS spam filtration in cellphones is still not particularly robust [8]. Deep neural networks (DNNs) are one of the most recent technologies that have shown to be helpful in handling these types of challenges.

3.2 Proposed System

Specifically, the paper investigates the various machine learning methodologies accessible in this domain, as well as the advantages of employing the Transformer model in this context. Finally, but certainly not least, the research concluded by advising those specific regions to adopt these machine learning approaches in order to obtain more accurate findings in the future. In order to differentiate between ham emails and spam emails, as previously said, this project will create an efficient and sensitive classification model that has high accuracy while also having a low false-positive rate, which will then be tested.

With the purpose of detecting potentially important characteristics in the email collection, the Greedy Stepwise feature search technique has been specifically implemented. For the purpose of evaluating several machine learning classifiers (such as KNN, Decision Tree, Naive Bayes, and Logistic regression), the data was divided into three groups and the results were displayed in a table. There are a variety of variables (such as F-measure (accuracy), False Positive Rate, training length, and others) that may be used to study and evaluate the classifiers under consideration. It will be possible to identify the optimal model by taking into account all of these factors in their totality and merging them. Models with high accuracy but low false positive rates will be the most effective, as will models with low false positive rates.

3.3 Software and Hardware Requirements

A minimum of 8GB of RAM is necessary for the great majority of deep learning operations, with 16GB or more RAM being recommended for the vast majority of workloads. In terms of processor, it is advised that you use an Intel Core i7 processor from the 7th generation or above. A functional CPU, 1080 HD display Monitor, High-end Internet connection, etc.

Additionally, there is the issue of storage capacity, which is crucial given the increasing quantity of enormous Deep Learning data sets that require an increase in the amount of available storage space. Our research team carried out a comprehensive evolutionary examination of the most important aspects of email spam, including their origins, evolution, and development over time, as well as their evolution and development through time. As a consequence of this, we were able to discover several fascinating research gaps and potential study subjects.

As a consequence, we presented our results on several open research challenges connected to spam filtering in a comprehensible manner and advised proactive approaches for the development of machine learning techniques in order to prevent the formation of new kinds of spam that may find it easier to defeat filters in the future.

3.4 Algorithm Insights

LR algorithm: The most basic type of logistic regression, in its most basic form, is a supervised classification strategy that is used to predict outcomes. As long as the goal variable (or output) is a collection of characteristics (or inputs) in a classification issue, the target variable (or output) can only take discrete values for that set of characteristics (or inputs). Contrary to common assumption, logistic regression is a type of regression model, not a regression technique.

NB Technique: According to the NB classifier, each feature is assumed to be independent of the others and that they do not interact with one another so that each feature contributes independently and equally to the chance that a sample belongs to a certain class. For this reason, the NB classifier is successful even when dealing with extremely large datasets with high dimensionality, as it is simple to create and computationally efficient.

KNN Method: If you're looking for a non-parametric, slow-learning method for machine learning, the KNN algorithm is a good choice because it's widely used. It is important to examine a database in which the data points have been divided into a large number of distinct categories in order to anticipate the categorization of a new sample point. In other words, rather than the other way around, the data is used to determine the structure of the model's structure rather than vice versa.

Chapter 4 – Experiment Implementation and Outputs

4.1 Project Architecture

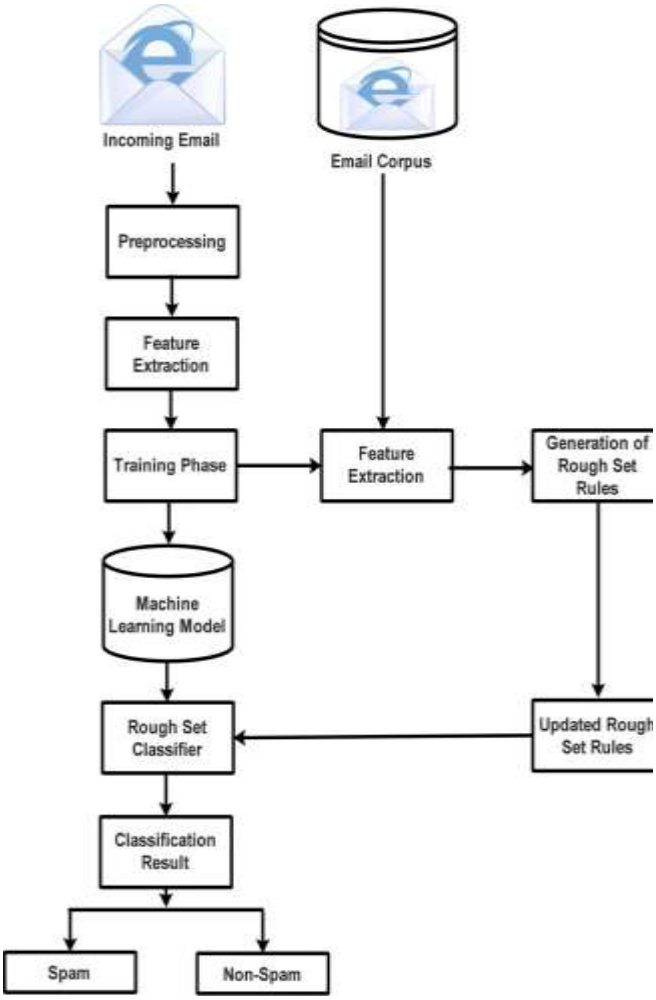
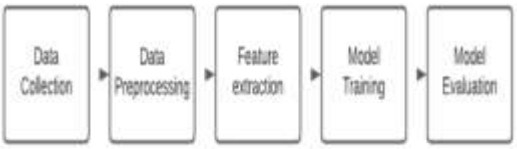


Fig 1: Project Flowchart

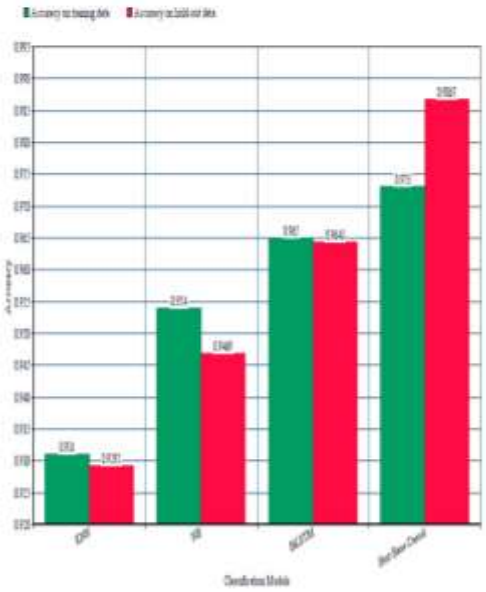
4.2 Project Flow

The overall approach, as well as the tools and techniques that were used to complete the spam email detection work, are all covered in great detail in the following section. Generally speaking, each NLP activity is separated into five primary phases: data collection, pre-processing of data, extraction of features, training, and evaluation of a model, and evaluation of a model. The flow diagram for each of the steps is depicted in the figure to the right. Following are some examples of how the extraction of features will be carried out automatically as part of the deep learning model training in this study:



4.3 Derived Output

A comparison study of the algorithms is intended to yield the desired outcome, as exemplified in the following example.



Output 1: Accuracy Table

Chapter 5 – Conclusion and Future Scope

In this review, a mixture model for sorting SMS spam is depicted that depends on CNN and LSTM, and that addresses SMS settings by utilizing CNN and LSTM, (for example, portable organization messages, Facebook courier messages, WhatsApp messages). To gain a genuine dataset for the appraisal dataset, an assortment of interchanges in both Arabic and English is accumulated and investigated. Because of the procured information, support vector machines (SVM), K-closest neighbors (KNN), Multinomial Naive Bayes (NB), Decision Trees (DT), Logistic Regression (LR), Random Forest (RF), AdaBoost (AB), Bagging classifier, and Extra Trees are totally used to distinguish SMS spam.

The exploratory evaluation of the proposed procedure has uncovered that the CNN-LSTM model beats different techniques as far as the arrangement of SMS spam, as per the outcomes. Following the investigations, our CNN-LSTM model showed an exactness of 98.37 percent, accuracy of 95.39 percent, review of 87%, an F1-Score of 91.48 percent, and an all-out region under the bend of 93.7 percent. This innovation can significantly work on the security of cell phones by screening spam messages and restricting the dangers that are related to smishing assaults in portable settings. As future work, we need to build a refined system fit for separating spam messages in cellphones with further developed accuracy. To grow the usefulness, further highlights, for example, the assessment of URLs or documents joined to messages and the assessment of phone numbers remembered for messages are being created.

References

1. Morreale, M. Daily SMS Mobile Usage Statistics. 2017. Available online: https://www.smseagle.eu/2017/ 03/06/daily-SMS-mobile-statistics/ (accessed on 15 June 2020).

2. Roy, P.K.; Singh, J.P.; Banerjee, S. Deep learning to filter SMS Spam. Future Gener. Comput. Syst. 2020, 102, 524–533. [CrossRef]

3. Tatango. Text Message Spam Infographic. 2011. Available online: https://www.tatango.com/blog/textmessage-spam-infographic/ (accessed on 15 June 2020).

4. Goel, D.; Jain, A. Smishing-Classifier: A Novel Framework for Detection of Smishing Attack in Mobile Environment. In Proceedings of the Smart and Innovative Trends in Next Generation Computing Technologies (NGCT 2017), Dehradun, India, 30–31 October 2017; pp. 502–512.

5. Goel, D.; Jain, A.K. Mobile phishing attacks and defense mechanisms: State of art and open research challenges. Comput. Secure. 2018, 73, 519–544. [CrossRef]

6. Jain, A.K.; Yadav, S.K.; Choudhary, N. A Novel Approach to Detect Spam and Smishing SMS using Machine Learning Techniques. IJESMA 2020, 12, 21–38. [CrossRef]

7. Mishra, S.; Soni, D. Smishing Detector: A security model to detect smishing through SMS content analysis and URL behavior analysis. Future Gener. Comput. Syst. 2020,

108, 803–815. [CrossRef]

8. Graves, A. Offline Arabic Handwriting Recognition with Multidimensional Recurrent Neural Networks. In Guide to OCR for Arabic Scripts; Springer: London, UK, 2012, pp. 297–313.

9. Chherawala, Y.; Roy, P.P.; Cheriet, M. Feature Set Evaluation for Offline Handwriting Recognition Systems: Application to the Recurrent Neural Network Model. IEEE Trans. Cybern. 2016, 46, 2825–2836. [CrossRef] [PubMed]

10. Elleuch, M.; Khairallah, M. An Improved Arabic Handwritten Recognition System Using Deep Support Vector Machines. Int. J. Multimed. Data Eng. Manag. 2016, 7, 1–20. [CrossRef]

11. Yousfi, S.; Berrani, S.A.; Garcia, C. Contribution of recurrent connectionist language models in improving LSTM-based Arabic text recognition in videos. Pattern Recognit. 2017, 64, 245–254. [CrossRef]

12. El-Desoky Mousa, A.; Kuo, H.J.; Mangu, L.; Soltau, H. Morpheme-based feature-rich language models using Deep Neural Networks for LVCSR of Egyptian Arabic. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 8435–8439.

13. Deselaers, T.; Hasan, S.; Bender, O.; Ney, H. A Deep Learning Approach to Machine Transliteration. In Proceedings of the Fourth Workshop on Statistical Machine Translation, Athens, Greece, 30–31 March 2009; StatMT '09; Association for Computational Linguistics: Stroudsburg, PA, USA, 2009; pp. 233–241.

14. Guzmán, F.; Bouamor, H.; Baly, R.; Habash, N. Machine Translation Evaluation for Arabic using Morphologically-enriched Embeddings. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; The COLING 2016 Organizing Committee: Osaka, Japan, 2016; pp. 1398–1408.

15. Jindal, V. A Personalized Markov Clustering and Deep Learning Approach for Arabic Text Categorization; In Proceedings of the ACL 2016 Student Research Workshop, Berlin, Germany, 7–12 August 2016.

16. Dahou, A.; Xiong, S.; Zhou, J.; Haddad, M.H.; Duan, P. Word Embeddings and Convolutional Neural Network for Arabic Sentiment Classification. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; The COLING 2016 Organizing Committee: Osaka, Japan, 2016; pp. 2418–2427.

17. Al-Sallab, A.; Baly, R.; Hajj, H.; Shaban, K.B.; El-Hajj, W.; Badaro, G. AROMA: A Recursive Deep Learning Model for Opinion Mining in Arabic as a Low Resource Language. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 2017, 16, 1–20. [CrossRef]

18. Al-Smadi, M.; Qawasmeh, O.; Al-Ayyoub, M.; Jararweh, Y.; Gupta, B. Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews. J. Comput. Sci. 2018, 27, 386–393. [CrossRef] 19. Zhou, C.; Sun, C.; Liu, Z.; Lau, F.C.M. A-C-LSTM Neural Network for Text Classification. arXiv 2015, arXiv:cs.CL/1511.08630.

20. Joo, J.W.; Moon, S.Y.; Singh, S.; Park, J.H. S-Detector: an enhanced security model for detecting Smishing attack for mobile computing. Telecommun. Syst. 2017, 66, 29–38. [CrossRef]

21. Delvia Arifin, D.; Shaufiah.; Bijaksana, M.A. Enhancing spam detection on mobile phone Short Message Service (SMS) performance using FP-growth and Naive Bayes Classifier. In Proceedings of the 2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), Bandung, Indonesia, 13–15 September 2016; pp. 80–84.

22. Sonowal, G.; Kuppusamy, K.S. SmiDCA: An Anti-Smishing Model with Machine Learning Approach. Comput. J. 2018, 61,

1143–1157. [CrossRef]

23. Jain, A.K.; Gupta, B.B. Feature-Based Approach for Detection of Smishing Messages in the Mobile Environment. J. Inf. Technol. Res. 2019, 12, 17–35. [CrossRef]

24. Jain, A.K.; Gupta, B. Rule-Based Framework for Detection of Smishing Messages in Mobile Environment. Procedia Comput. Sci. 2018, 125, 617–623. [CrossRef]

25. Almeida, T.A.; Silva, T.P.; Santos, I.; Hidalgo, J.M.G. Text normalization and semantic indexing to enhance Instant Messaging and SMS spam filtering. Knowl. Based Syst. 2016, 108, 25–32. [CrossRef]

26. Yadav, K.; Kumaraguru, P.; Goyal, A.; Gupta, A.; Naik, V. SMSAssassin: Crowdsourcing Driven Mobile-Based System for SMS Spam Filtering. In Proceedings of the 12th Workshop on Mobile Computing Systems and Applications, Phoenix, AZ, USA, 1–2 March 2011; HotMobile '11; Association for Computing Machinery: New York, NY, USA, 2011; pp. 1–6. [CrossRef]

27. Agarwal, S.; Kaur, S.; Garhwal, S. SMS spam detection for Indian messages. In Proceedings of the 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, 4–5 September 2015; pp. 634–638.

28. Almeida, T.A.; Hidalgo, J.M.G.; Yamakami, A. Contributions to the Study of SMS Spam Filtering: New Collection and Results. In Proceedings of the 11th ACM Symposium on Document Engineering, Mountain View, CA, USA, 19–22 September 2011; DocEng '11; Association for Computing Machinery: New York, NY, USA, 2011; pp. 259–262. [CrossRef]

29. Chen, T.; Kan, M.Y. Creating a live, public short message service corpus: The NUS SMS corpus. Lang. Resour. Eval. 2013, 47, 299–355. [CrossRef]

*30. Zhang, W.; Yoshida, T.; Tang, X. A comparative study of TF*IDF, LSI and multi-words for text classification Expert Syst. Appl. 2011, 38, 2758–2765. [CrossRef]*

31. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, AZ, USA, 2–4 May 2013.

32. Pennington, J.; Socher, R.; Manning, C. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1532–1543. [CrossRef]