Topic: Amazon product review spam detection

Literature review and Project Plan

ELEC-E5550: Statistical Natural Language Processing

Student name:

- Khoa Nguyen (khoa.nguyen@aalto.fi)
- Huyen Pham (huyen.pham@aalto.fi)

1. Literature review

Nowadays, customers have the ability to express their opinions about products and services easily in the form of reviews on purchasing platforms. E-commerce platforms like Amazon have suffered from fake review problems as some shoppers accept gifts and other incentives in exchange for positive reviews, despite a ban on these activities. Hence, having a model that assists companies in detecting such issues would help customers shop with confidence knowing the reviews they read are authentic and relevant. Engaging in spam detection might also create a competitive advantage for firms regarding customer experience.

Due to the significance of detecting spam reviews, various research studies on this problem were established, including numerous methods. The original spam review detection study concentrates on e-mail spam that was considered to be similar to spam reviews. Sahami et al. utilized probabilistic learning methods in collaboration with a concept of misclassification cost to develop filters that are particularly well suited to the complexity of this task. Carreras et al. established a way to illustrate that AdaBoost is more effective than Naive Bayes and decision trees by considering domain-specific elements of this problem and the raw text of e-mail messages accurate filters may be constructed. Spam became a severe problem for the World Wide Web in the late 1990s. Web-spam filtering based on content analysis was further investigated to detect spam pages. Davison et al. used the decision trees method to detect link-based web-spam. He used heuristics to identify and remove "internal" links between sites for reasons other than quality. SVM was used by Drost et al. to categorize web spam using content-based characteristics. They defined the challenge of link spam detection and offered a method for generating training data by showing the efficiency of intrinsic and relational attribute classes. Machine learning techniques and probability distributions over strings were applied by Mishne et al. to detect spam comments. They compare the language models used in the blog post, comments, and pages linked from the comments. Finally, there were a variety of potential approaches for detecting spam text. In the following part, we will discuss the strengths and weaknesses of several methods in greater detail in the next section.

In this project, we aim to apply NLP methods to classify spam reviews from the Amazon Reviews dataset. The dataset consists of text and its corresponding label (spam or non-spam). Supervised models are applicable for this dataset. There are some possible methods we can use to classify spam from non-spam reviews. Deep learning models can provide some excellent classifying in this case. The paper Spam review detection by Shahariar et al. compared the results of multiple Machine Learning methods and Deep Learning methods. According to the paper, neural networks such as Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), and Multilayer Perceptron (MLP) consistently achieve accuracy of higher than 90% compared to traditional machine learning methods such as Support Vector Machine (SVM). A very promising approach proposed in this paper is the use of a word embedding method such as Word2Vec and uses the output as a numerical representation of the dataset as input of LSTM and CNN. CNN gives an accuracy of around 95.5%, and LSTM has about 96.75% accuracy. These are some potent methods that we can try on our dataset. Deep Learning methods like Word2Vec are better at providing vector representations of words, improving the accuracy of the classifiers. However, the drawback of these Deep Learning models is that they require more intensive training, requiring more significant amounts of data and hardware compared to traditional Machine Learning methods.

2. Reference:

1. M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz (1998) *A Bayesian approach to filtering junk e-mail.* Workshop on Learning for Text Categorization, Madison, Wisconsin, pp. 98-105.
2. Davison B. D. (2000) *Recognizing Nepotistic Links on the Web*. Workshop on Artificial Intelligence for Web Search, 2000, pp.23- 28.
3. Carreras, X. and Marquez, L. (2001) *Boosting trees for anti-spam e-mail filtering*. 4th International Conference on Recent Advances in Natural Language Processing, pp. 58-64.
4. I. Drost and T. Scheffer. (2005) *Thwarting the nigritude ultramarine: Learning to identify link spam*. 16th European Conference on Machine Learning, Berlin, Germany, pp.96-107.
5. Mishne, G., Carmel, D. and Lempel, R. (2005) *Blocking blog spam with language model disagreement*. Adversarial Information Web (AIRWeb), Chiba, Japan, pp. 1-6.
6. Radulescu, C., Dinsoreanu, M., & Potolea, R. (2014). *Identification of spam comments using natural language processing techniques. 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP).* doi:10.1109/iccp.2014.6936976.
7. Shahariar, M., Biswas, S., Omar, F., Shah, M., and Hassan, S. (2019) *Spam Review Detection Using Deep Learning*. *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (ICON)*, pp. 0027-0033, DOI: 10.1109/IEMCON.2019.8936148.

3. Project plan

- Dataset:
  - Link: https://www.kaggle.com/naveedhn/amazon-product-review-spam-and-non-spam
- Data preprocessing:
  - Change the data format from JSON to the data frame
  - Remove empty rows, retain important columns (text, labels, category of product, review title)
  - Combine review title and review text as the data
  - Apply tokenization and lemmatization, remove stop words, lowercasing
  - Balance class distribution
  - Save the data
- Data visualization (EDA)
  - Apply vectorization methods: TF-IDF, Word2Vec
  - Embed the matrices and visualize them on 2D scatter plots to see if there are any clusters or structures in the data
- Modeling
  - Apply NN models: LSTM, CNN, or other models if any interesting ones arise
  - Hyperparameter tuning for each model
- Model validation and comparison
  - Monitor classification accuracy of train and test sets for each model
  - Compare accuracy between the models