Столыпинский вестник

Научная статья

Original article

УДК 002.304

АВТОМАТИЧЕСКОЕ ОБНАРУЖЕНИЕ И АНАЛИЗ САРКАЗМА В ТЕКСТАХ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

AUTOMATIC DETECTION AND ANALYSIS OF SARCASM IN NATURAL LANGUAGE TEXTS

Сламкул Данияр, Магистрант 2 курс, факультет «Программная инженерия», Казахстанско-Британский технический университет Казахстан, г. Алматы

Slamkul Daniyar, 2nd year undergraduate, Faculty of Software Engineering, Kazakh-British Technical University of Kazakhstan, Almaty, da_slamkul@kbtu.kz

Аннотация. Это исследование решает сложную задачу автоматического обнаружения сарказма в текстах Твиттера, что имеет решающее значение, поскольку социальные сети становятся центральным средством обмена мнениями. В исследовании оцениваются как традиционные методы, основанные на правилах, так и передовые методы машинного обучения.

Мы применили классические алгоритмы, такие как случайный лес, гауссов наивный байесовский метод, машина опорных векторов (SVM) и k-ближайшие соседи (KNN), для выявления сарказма, причем SVM показал заметную точность. Кроме того, мы усовершенствовали модель TinyBERT, упрощенную версию BERT,

предназначенную для обнаружения сарказма, которая превзошла традиционные модели с показателем F1 0,87.

Наш набор данных был хорошо сбалансирован, что способствовало надежности наших моделей. Исследование посвящено конкретным проблемам Twitter, включая изменение языка и сокращений. Результаты закладывают основу для эффективного обнаружения сарказма в Твиттере с использованием сочетания старых и новых методов. Будущие исследования могут включать более крупные модели трансформаторов и более обширные наборы данных для повышения точности.

Abstract. This research tackles the complex task of automatic sarcasm detection in Twitter texts, crucial as social media becomes central to sharing opinions. The study evaluates both traditional rule-based methods and advanced machine learning techniques.

We applied classic algorithms like Random Forest, Gaussian Naive Bayes, Support Vector Machine (SVM), and k-Nearest Neighbors (KNN) to identify sarcasm, with SVM showing notable accuracy. Additionally, we enhanced the TinyBERT model, a simplified version of BERT designed for sarcasm detection, which outperformed traditional models with an F1 score of 0.87.

Our dataset was well-balanced, aiding the robustness of our models. The research addresses the specific challenges of Twitter, including its changing language and abbreviations. The findings establish a groundwork for effective sarcasm detection on Twitter using a mix of old and new methods. Future research might include larger transformer models and more extensive datasets to improve accuracy.

Ключевые слова: НЛП, машинное обучение, обнаружение сарказма, нейронная сеть.

Keywords: NLP, Machine Learning, Sarcasm Detection, Neural Network.

1. INTRODUCTION

In recent years, social media platforms like Twitter have gained immense popularity and significance. These platforms have evolved into extensive ecosystems

where users freely express their opinions and ideas. Many companies leverage their presence on social media for marketing, customer assistance, and post-sales service, engaging with public sentiment about their products or services.

As the volume of data on social media grows rapidly, companies increasingly rely on various tools for data analysis and providing customer services. Tasks such as sentiment analysis, content management, and extracting relevant messages for customer support review become crucial. However, these tools often lack the sophistication to decipher more subtle forms of language, such as sarcasm or humor, where the message's meaning is not always evident and explicit.

This adds additional challenges for social media teams, already inundated with customer messages, to identify and appropriately respond to sarcastic content. This underscores the importance of sarcasm detection to eliminate intentional ambiguity.

The goal of our research is to address the intricate problem of sarcasm detection on Twitter. Although sarcasm detection inherently poses a complex task, the nature and style of content on Twitter further complicate the detection process. Twitter is informal by nature, with a constantly expanding lexicon of abbreviations and slang, making sarcasm detection particularly challenging.

Moreover, the dynamic and real-time nature of Twitter content necessitates adaptive and efficient sarcasm detection algorithms to keep pace with the evolving language trends and user behaviors on the platform. Our research aims to contribute to the development of robust and effective sarcasm detection methods tailored to the unique characteristics of Twitter's communication landscape.

2. LITERATURE REVIEW

In recent times, there has been a growing research interest in the field of sarcasm detection in text. Numerous studies have delved into the analysis of sarcasm using data collected from various sources, including tweets on Twitter, Facebook posts, Amazon product reviews, website comments, and more.

The identification of emotions expressed on social media has garnered significant attention in recent years. Researchers have developed techniques for analyzing emotions

within text, categorizing them into six basic emotions: anger, fear, disgust, joy, surprise, and sadness [5]. To aid in opinion mining, SenticNet, which employs common sense reasoning techniques and emotion categorization models, is commonly used. Researchers have also utilized a combination of SentiWordNet and WordNetAffect to discern emotions in web-based content [4]. WordNet is employed to gauge the similarity of text and identify emotion synsets, although the challenge of word context variation remains. Additionally, emoticons have found application in sarcasm detection. As one study suggests, emojis do not always directly convey emotional content. For example, a positive emoji may serve to disambiguate a vague sentence or complement an otherwise relatively negative text [7].

Latent Semantic Analysis (LSA) provides a vector space model that offers a uniform representation for words, word sets, sentences, and texts. Within the LSA space, emotions can be represented in various ways, such as the vector of a specific word denoting the emotion (e.g., 'love'), the vector representing the synset of the emotion (e.g., 'choler' and 'ire'), and the vector of all words in the synsets associated with the emotion [5].

For sarcasm detection in dialogue, sequence labeling serves as a learning mechanism. New features are introduced based on the available dataset information. A comparison is made between two sequence labelers (SEARN and SVMHMM) and three classifiers (SVM with oversampled and under-sampled data, and Naïve Bayes) [3]. Another approach involves the utilization of a Novel Bootstrapping Algorithm, which autonomously learns lists of positive sentiment phrases and negative situation phrases from sarcastic tweets. SVM classifiers are then employed for classifying tweets [5].

The majority of research focused on detecting sarcasm relies on sentiment or emotional data. Maynard and Greenwood (2014) [6] explored the presence of sarcasm in tweets and its impact on sentiment analysis. Their findings indicate that accurate identification of sarcasm can enhance sentiment analysis by nearly 50\%. Despite correctly categorizing tweets as sarcastic or non-sarcastic, the accuracy of their program

remained low. Over the years, researchers have devised various methods within the field of sarcasm detection.

For sarcasm detection in dialogue, sequence labeling serves as a learning mechanism. New features are introduced based on the available dataset information. A comparison is made between two sequence labelers (SEARN and SVMHMM) and three classifiers (SVM with oversampled and under-sampled data, and Naïve Bayes) [3]. Another approach involves the utilization of a Novel Bootstrapping Algorithm, which autonomously learns lists of positive sentiment phrases and negative situation phrases from sarcastic tweets. SVM classifiers are then employed for classifying tweets [5].

The majority of research focused on detecting sarcasm relies on sentiment or emotional data. Maynard and Greenwood (2014) [6] explored the presence of sarcasm in tweets and its impact on sentiment analysis. Their findings indicate that accurate identification of sarcasm can enhance sentiment analysis by nearly 50\%. Despite correctly categorizing tweets as sarcastic or non-sarcastic, the accuracy of their program remained low. Over the years, researchers have devised various methods within the field of sarcasm detection.

3. DATASET

4.1 Data Source. This investigation examines and identifies instances of sarcasm within the Kaggle data set [4]. The data set comprises approximately 27,000 comments, with 12,000 classified as sarcastic and the remaining 15,000 as non-sarcastic. The average length of each headline is 12.5 words, with a standard deviation of 4.3 words. The data set consists of three columns: ‘headline’ (the comment), ‘is sarcastic’ (1 if the headline is sarcastic, 0 otherwise), and ‘article_link’ (the source link).

Here are a few examples of sarcastic comments from the data set:

- “new history textbook makes hatred of history come alive for students”
- “thirtysomething scientists unveil doomsday clock of hair loss”
- “area man does most of his traveling by gurney”

And here are examples of non-sarcastic comments:

- “Does the Internet really exacerbate bullying?”

- “How tired I am of these boring posts”

These comments exhibit ambiguity, prompting our effort to develop a model and analyze its ability to distinguish between the two classes.

4.2 Preliminary Data Analysis

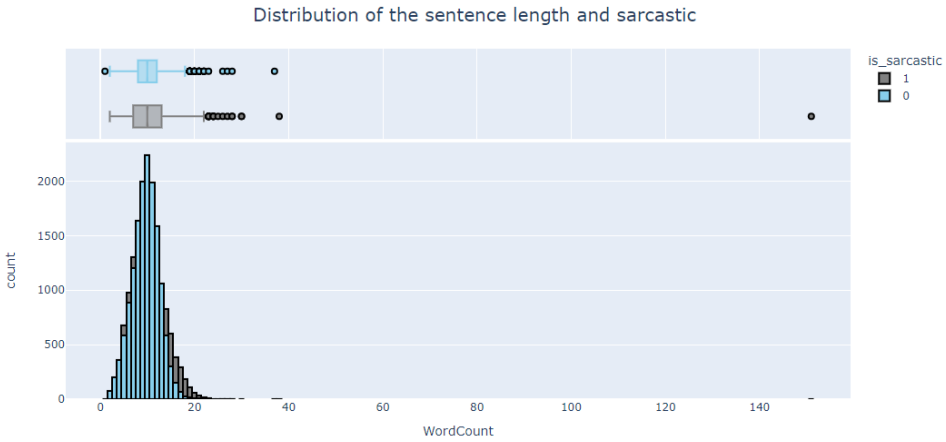

Fig. 1. Distribution of the sentence length and sarcastic

From the depicted figure, it is evident that:

- The label distribution within the sarcasm dataset is notably balanced
- The distribution of sarcastic and non-sarcastic sentences exhibits a significant similarity

The dataset demonstrates a considerable balance, obviating the necessity to employ techniques for handling imbalanced classes, such as undersampling, oversampling, focal loss, etc. This balance in class distribution contributes to a more robust and unbiased training process. Consequently, the application of standard procedures for model training is deemed sufficient, ensuring a reliable and representative learning experience.

4. METRICS

We employ performance metrics, namely precision, recall, and F1-Score, derived from the principles of True Positive, True Negative, False Positive, and False Negative.

Precision quantifies [1] the accuracy of positive predictions, indicating the proportion of correctly identified positive instances among all instances predicted as positive.

Recall assesses the classifier's [4] ability to correctly identify positive instances in relation to all actual positive instances in the dataset. It is alternatively known as Sensitivity.

F1-Score [7], a composite metric, integrates both precision and recall. Commonly defined as the harmonic mean of the two, the harmonic mean is a method for calculating an "average" of values, particularly suited for ratios (such as precision and recall) compared to the conventional arithmetic mean.

F1 is calculated as follows:

$$F1 = 2 \times \left(\frac{precision \times recall}{precision + recall}\right)$$

where:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

In "macro" F1, a separate F1 score is calculated for each species value and then averaged.

5. MODELING

6.1 Classic Machine Learning

In this chapter, we employ various machine learning models to address the classification task of sarcasm detection. The initial attempt involves the application of classic machine learning algorithms, including Random Forest, Gaussian Naive Bayes, Support Vector Machine (SVM), and k-Nearest Neighbors (KNN). The outcomes of these techniques are presented in the later sections of this paper.

1. Random Forest.

Random Forest [1] is a widely used ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes (classification) of the individual trees. In our implementation, we utilize the RandomForestClassifier from scikit-learn, which offers a versatile and powerful framework for classification tasks.

$X = \frac{1}{N}\sum_{i=1}^{N} fi(X)$

Where X denotes the mean (or mode) of individual tree predictions, N is the number of trees in the forest, and $fi(X)$ represents the individual tree prediction (i). In a classification problem, the result can be interpreted as the probability of belonging to class 1.

2. Gaussian Naive Bayes

Gaussian Naive Bayes [6] is a probabilistic classification algorithm based on Bayes' theorem with the assumption of independence between features. The scikit-learn library provides the GaussianNB class, allowing us to implement this algorithm for sarcasm detection.

3. Support Vector Machine (SVM)

Support Vector Machine [7] is a powerful algorithm for both classification and regression tasks. It works by finding the hyperplane that best separates the data into different classes. For sarcasm detection, we leverage the SVC class from scikit-learn, providing an efficient implementation of SVM.

4. K-Nearest Neighbors (KNN)

k-Nearest Neighbors [6] is a simple and effective algorithm for classification tasks. It classifies a data point based on the majority class among its k-nearest neighbors. The scikit-learn library offers the KNeighborsClassifier class for implementing the KNN algorithm in our sarcasm detection model.

Hereafter, the outcomes obtained from classical models are presented for examination and analysis.

Model	Precision	Recall	Accuracy	F1 Score	AUC
Random Forest	0.810	0.810	0.810	0.810	0.805
Gaussian Naive Bayes	0.690	0.660	0.670	0.650	0.659
SVM	0.820	0.820	0.820	0.820	0.821
KNN	0.700	0.700	0.700	0.690	0.695

Table 1. Performance metrics of classic models.

6.2 Performance metrics of classic models.

BERT [3] represents a state-of-the-art model architecture based on transformers, initially developed by Google. Its versatility allows for pre-training followed by fine-tuning to cater to specific tasks, such as classification. The pre-training of BERT involves two simultaneous tasks: Masked Language Modeling (MLM), where 15\% of tokens are masked and the model predicts the masked words, and Next Sentence Prediction (NSP), which involves predicting whether sentence B follows sentence A.

The Hugging Face framework [3] facilitates the straightforward acquisition and training of state-of-the-art pre-trained models, offering advantages in terms of computational efficiency, environmental impact, and time savings compared to training models from scratch.

For the present investigation, the choice is made to fine-tune TinyBERT [1] due to constraints imposed by limited GPU resources, specifically for a classification task using the Sarcasm Dataset.

A. Procedural Steps

The procedural steps commence with installing the transformers package and importing requisite libraries. The utilization of the pre-trained TinyBERT model, sourced from *prajjwal1/bert-tiny*, is notable for its compact size (approximately 16MB), facilitating rapid training on standard PCs. Following this, the creation of a Python class named Dataset is undertaken. This class encapsulates labels and texts from the dataset and inherits properties from *torch.utils.data.Dataset*, enabling the utilization of functionalities like multiprocessing. It contains essential information such as labels and text at each pass, ensuring efficient retrieval through methods like get item.

B. Network Initialization

The network initialization stage involves defining the layers, comprising the Bert layer for the pre-trained model (obtained from the Hugging Face hub), a linear layer featuring dropout and ReLU activation.

C. Model Training

Subsequent to the network initialization, the model is trained using the Adam optimizer to optimize the CrossEntropyLoss. The training loop involves feeding inputs through the model, computing losses, and optimizing weights using *backward()* and *step()*. PyTorch automates these computations. Additionally, accuracy scores are calculated in each batch, culminating in an average at the end of the training loop.

D. Hyperparameter Tuning for Bert

The fine-tuning process in the sarcasm dataset necessitates only a few epochs to achieve a high level of accuracy on the training set. In this context, the following hyperparameters are configured:

- Number of epochs *Num_epochs*: 5, 7, 9, 10
- Learning rate *lr*: 0.0001, 0.001, 0.005
- Batch size: 8, 16, 32

7. RESULTS

Our research initiation aimed to automate the identification of sarcastic sentences through the application of machine learning/deep learning (ML/DL) techniques. Following the loading and cleansing of our dataset, we explored its key characteristics. Subsequently, we subjected these features to preprocessing and employed various machine learning algorithms, including Random Forest, Gaussian Naive Bayes, Support Vector Machine (SVM), k-Nearest Neighbors (KNN), and BERT. The outcomes of our experimentation revealed that TinBert yielded the most favorable results Table 2, achieving an F1 score of 0.87.

Model	Precision	Recall	Accuracy	F1 Score	AUC
Random Forest	0.810	0.810	0.810	0.810	0.805
Gaussian Naive Bayes	0.690	0.660	0.670	0.650	0.659
SVM	0.820	0.820	0.820	0.820	0.821
KNN	0.700	0.700	0.700	0.690	0.695

Table 2. Performance metrics of different models.

8. CONCLUSION

In this research proposal, we tackled the challenging task of automatically detecting sarcasm in natural language texts, particularly from Twitter. This is crucial given sarcasm's prevalence in online communication and the need for sophisticated detection methods. We examined a range of techniques from traditional rule-based to modern machine learning models.

Classic machine learning algorithms such as Random Forest, Gaussian Naive Bayes, Support Vector Machine (SVM), and k-Nearest Neighbors (KNN) were utilized for sarcasm classification. Among these, SVM was particularly accurate.

We also advanced into deep learning, specifically by fine-tuning the TinyBERT model—a streamlined version of BERT optimized for sarcasm detection. TinyBERT notably surpassed the conventional models, achieving an F1 score of 0.87.

Our analysis covered not only technical aspects but also the peculiar challenges of Twitter's informal, dynamic nature and evolving language, which complicates sarcasm detection. The dataset used was balanced in terms of sarcasm presence, enhancing model robustness and providing unbiased training results.

In conclusion, our study provides a solid base for developing effective sarcasm detection methods on Twitter, combining both classic and advanced models. Moving forward, refining these methods with larger datasets and more complex models will be crucial for improving accuracy and applicability.

Список литературы

1. Сильвио Амир, Байрон К. Уоллес, Хао Лю и Паула Карвалью Марио Дж. Сильва. Моделирование контекста с использованием пользовательских вложений для обнаружения сарказма в социальных сетях. Препринт arXiv arXiv: 1607.00976, 2016.
2. Раджеш Басак, Шамик Сурал, Нилой Гангули и Сумья К. Гхош. Публичный онлайн-позор в Twitter: выявление, анализ и смягчение последствий. Транзакции IEEE в вычислительных социальных системах, 6(2):208-220, 2019.

3. Дарио Бертеро и Паскаль Фунг. Система долговременной кратковременной памяти для прогнозирования юмора в диалогах. В материалах конференции Североамериканского отделения Ассоциации компьютерной лингвистики 2016 года "Технологии человеческого языка", стр. 130-135, 2016.
4. Сантош Кумар Бхарти, Корра Сатья Бабу и Санджай Кумар Джена. Распознавание сарказма в данных Twitter на основе анализа. В материалах Международной конференции IEEE/ACM 2015 года по достижениям в области анализа и разработки данных в социальных сетях, 2015, страницы 1373-1380, 2015.
5. Сантош Кумар Бхарти, Рамкрушна Прадхан, Корра Сатья Бабу и Сан-Джей Кумар Джена. Анализ сарказма в данных Twitter с использованием методов машинного обучения. Тенденции в анализе социальных сетей: распространение информации, моделирование поведения пользователей, прогнозирование и оценка уязвимостей, страницы 51-76, 2017 г.
6. Кристофер М. Бишоп и Нассер М. Насрабади. Распознавание образов и машинное обучение, том 4. Springer, 2006.
7. Дэвид Чикко и Джузеппе Юрман. Преимущества коэффициента корреляции Мэтьюса (mcc) по сравнению с показателем f1 и точность при оценке бинарной классификации. Геномика BMC, 21(1):1-13, 2020.

References

1. Silvio Amir, Byron C Wallace, Hao Lyu, and Paula Carvalho Mario J Silva. Modelling context with user embeddings for sarcasm detection in social media. arXiv preprint arXiv:1607.00976, 2016.
2. Rajesh Basak, Shamik Sural, Niloy Ganguly, and Soumya K Ghosh. Online public shaming on twitter: Detection, analysis, and mitigation. IEEE Transactions on computational social systems, 6(2):208–220, 2019.
3. Dario Bertero and Pascale Fung. A long short-term memory framework for predicting humor in dialogues. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human

Language Technologies, pages 130–135, 2016.

4. Santosh Kumar Bharti, Korra Sathya Babu, and Sanjay Kumar Jena. Parsing- based sarcasm sentiment recognition in twitter data. In Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, pages 1373–1380, 2015.
5. Santosh Kumar Bharti, Ramkrushna Pradhan, Korra Sathya Babu, and San- jay Kumar Jena. Sarcasm analysis on twitter data using machine learning approaches. Trends in Social Network Analysis: Information Propagation, User Behavior Modeling, Forecasting, and Vulnerability Assessment, pages 51–76, 2017.
6. Christopher M Bishop and Nasser M Nasrabadi. Pattern recognition and ma- chine learning, volume 4. Springer, 2006.
7. Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. BMC genomics, 21(1):1–13, 2020.

Для цитирования: Сламкул Д. АВТОМАТИЧЕСКОЕ ОБНАРУЖЕНИЕ И АНАЛИЗ САРКАЗМА В ТЕКСТАХ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ Научный сетевой журнал «Столыпинский вестник" №4/2024.