Smart Systems, Security, and Data Science (SSSDS) **2025, 1(1): 1-7**
https://www.longmanpublishers.com/

Testing Topological NLP Transformers on Text Message Data to Determine the Best Online Phishing Detection Technique

Harrison N., Theodore D.
Monash University, Melbourne, Australia

Abstract: This study uses topological sentence transformer techniques to create an ideal classification model for online SMS spam detection. Malicious actors' increasingly complex and disruptive actions are the reason behind the study. We offer a workable and lightweight way to combine sklearn capability with pre-trained NLP repository models. The study design presents a user-extensible spam SMS solution and replicates the spaCy pipeline component architecture in a downstream sklearn pipeline implementation. We use linguistic NLP transformer approaches to short-sentence NLP datasets and use HuggingFace (RoBERTa-base) large-text data models via spaCy. Using a normal sklearn pipeline architecture, we iteratively retest models and compare their F1-scores. An optimal F1-score of 0.938 is obtained by applying spaCy transformer modeling; this result is comparable to research output from modern BERT/SBERT/'black box' prediction models. Using semantically similar paraphrase/sentence transformer techniques, this study presents a lightweight, user-interpretable, standardized, predictive SMS spam detection model that produces the best F1-scores for an SMS dataset. For a Twitter assessment set, significant F1-scores are also produced, suggesting possible real-world applicability.
Keywords: Dependency parsing, Phishing, Topological transformer processing, Transfer learning.

1. Introduction

All Since 2016, text-based dataset modeling has seen a significant transformation due to NLP language machine learning frameworks (Brown et al., 2020). The work presented here shows methodological relevance for short-text spam detection problems with limited resources and builds upon fundamental modern approaches to predictive linguistic modeling.

Threat vectors have become increasingly automated, and SMS spam detection is still a required effort (Haynes et al., 2021). Predictions from short-text message models are weak since they are usually supported by outdated technological tools. Tasks involving the classification of SMS spam data utilizing novel transformer techniques don't seem to be thoroughly studied or methodologically compatible with transformer structures (Roy et al., 2020). This article includes exploratory research from The University of Adelaide that looks at how big vector approaches and complex topological transformers affect the classification of short-text SMS spam.

These designs are well suited for short-text data since the current transformer iteration may embed dense tensors into topological frameworks from sentence-based inputs. For modeling NLP tasks, high-rank embeddings and pre-trained libraries—like RoBERTa (Liu et al., 2019)—have proven essential (Yang et al., 2017). The statistical model for the spaCy transformer pipeline is integrated with the self-supervised, sentence-based processing of RoBERTa-base. Roberta-base is a good option for a task-based transparent solution since it produces encoded weights that may be used in downstream standard classifiers. In order to listen to the transformer component (output), SpaCy offers a tagger tokenizer, a (dependency) parser, and an entity recognizer. Our work uses a Sklearn pipeline to classify spam text data using the resulting output.

The relevance of spam messaging content is usually transient, and the goal of this study is to determine the best state-of-the-art design using common categorization modeling criteria. The selection of language sample sizes and the particular vocabulary contained in datasets are significant problems for all models (Conneau et al., 2019). Haynes et al. (2021) stress the importance of avoiding risky websites and recommend that phishing detection research use publically accessible datasets. The sample size of available data is instantly limited by this methodological limitation, which also poses a recurring issue for SMS researchers. Growing illegal SMS message generating methods might produce redundant training datasets that are mainly unrepresentative of current trends, which is a problem for spam detection systems. With spaCy component-modeling pipelines, we find that this problem is negligible.

NLP-based classification techniques are used in this study under time constraints. The study addresses the transient nature of SMS message and creates an appropriate solution for settings with limited resources and time. Resolving limitations promotes an agile, iterative design implementation process.

1.1. Our methodology

To duplicate previous base-level research, a simple NLTK Regex model is first developed and evaluated. Using pattern-matching approaches, this model generates extremely accurate predictions; nevertheless, further testing yields overfitted results that are deemed insufficient for benchmarking. In the current literature, pattern-matching spam filters are redundant technological tools. When developing spam identification solutions, these approaches are not seen as appropriate or forward-looking (Shirazi et al., 2023).

Our design offers an appropriate model template for creating improved, forward-looking models. The work uses open-source component pipeline designs to repeatedly fit a classic SMS dataset (Kaggle, 2017) to a predictive classification model. Two lightweight statistical NLP models that use pre-trained neural network (NN) embeddings and finish in less than five minutes were developed thanks to constant verification of the model outputs. A transformer pipeline insertion was utilized in the second comparison model, while a vast language collection of unique vectors from the web was used in the first. When modeling, package defaults are used. This study indicates that spaCy transformer statistical modeling produces better F1 scores when compared to sentence-based transformer spam classification techniques and word-vector similarity modeling.

Because it effortlessly integrates pre-trained CommonCrawl data from the RoBERTa-base model (Liu et al., 2019), hosted on the Hugging Face (n.d.) repository, the en_core_web_trf spaCy transformer model is selected for transformer modeling. Using entity identification technologies and dependency parsers, the transformer pipeline design creates embedded weights (Gormley et al., 2015). The pipeline architecture generates outputs for downstream categorization tasks using standard components (sentencizer vs. senter, for example). The downstream implementation optimizes the spaCy statistical models on a CPU and builds a bespoke predictor using the sklearn pipeline feature. To balance the SMS dataset and avoid overfitting, the pipeline uses a unique SMOTE oversampling technique (Abid et al., 2022).
We evaluate the F1-score that is produced for every modeling cycle, and an iterative implementation process gives us assurance that the final product is optimal and reproducible. Cutting edge In order to input into a pipeline and produce predictions, short-text binary transformer modeling finds inferences and generates contextual topological embeddings. In accordance with earlier studies, we validate the implementation of open-source semantic similarity detection approaches on phrases against a Twitter dataset (Liu et al., 2021). Compared to earlier studies that use transformer topologies from scratch, this work uses spaCy transformer pipeline modeling to produce better classification results (accuracy and F1-scores). When pre-trained, default spaCy modeling pipelines were used, our lightweight CPU-based solution obtained accuracies comparable to GPU processing (0.9845 with an SVC classifier), and open-source, extensible architectures offered better options to new-build NN research.
The successful creation of a user-extensible, highly accurate spam detection system for SMS data based on topological transformers is a noteworthy contribution of our study. The study shows that cutting-edge transformer solutions offer a distinct path for SMS classifier development in the future.

2. Literature Review

The In the field of security research, SMS spam detection has received little attention (Roy et al., 2020), and the majority of spam detection studies have given priority on fraudulent email modeling (Chiew et al., 2019; Tan et al., 2020).

Complex, new-build NN implementations have been used to predict spam from the SMS dataset (Liu et al., 2021; Roy et al., 2020). According to Haynes et al. (2021), new NN builds require a lot of memory and fail to properly separate the required classification techniques from user-extensible components (such as pipelines).
Sentence transformers and solution time complexity are not typically taken into account in older studies. Prior work on the SMS dataset employing Hidden Markov Models (HMM) revealed favorable temporal complexity processing overheads (Xia & Chen, 2020). Nevertheless, HMM models are word-centric and use techniques that are inherently connected to forward-fed deep learning, taking into account just one previous state. Word embedding modeling has mostly been abandoned in favor of more complex sentence/paraphrasing techniques in post-2019 research.

The vast range of outcomes that can be obtained when analyzing linguistic datasets using word-based modeling techniques is depicted in Figure 1.

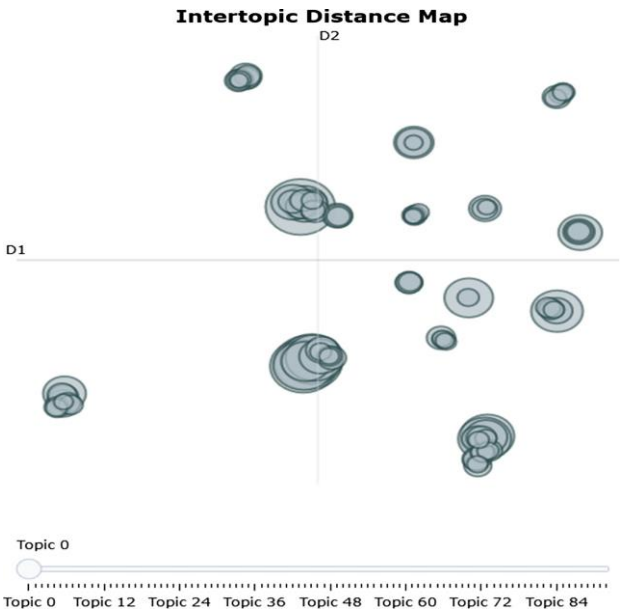


Figure 1: Inter-topic modelling using BERT topic methods – SMS data

The linguistic content inherent in paraphrases cannot be adequately captured by naively used vectorizers, such as GloVe and word2Vec word embedding methods, which fail bias assessment tests (May et al., 2019). This flaw with RoBERTa models is specifically highlighted on the Hugging Face website. Detecting and addressing bias in models is a real issue as, according to Hugging Face (n.d.), bias is an intrinsic drawback of pre-trained models. The use of transformer-based solutions instead of word embedding techniques is becoming more and more popular as a way to address algorithmic bias (Islam et al., 2020).

Low-rank topological vectorization has been utilized to allow task-specific approaches to classification (Brown et al., 2020; Shwartz-Ziv & Armon, 2021), while tensor topologies are capable of capturing spatial/relational information (Tumas et al., 2022). Gormley et al. (2015) studied how to train effective dependency parsers for text by analyzing lexical feature embeddings using low-rank tensors. Using comparably huge training data, Reimers and Gurevych (2019) investigated topological semantic similarities for sentence pairings using dependency parsing techniques in their model SBERT (SentenceBERT). To compare brief text passages, these writers looked into sentence paraphrasing mining techniques, which is a crucial tool for categorizing SMS spam (Reimers & Gurevych, 2019).

Backpropagation transformer modeling is necessary to concisely capture underlying latent complexity, and spam SMS identification in the wild is acknowledged as a non-trivial task (Shirazi et al., 2023) (Xu et al., 2023). Applying pre-trained transformer models to a specific topology has produced great accuracies on unlabeled data and unsupervised modeling. Transformer models use backpropagation to predict output from a huge pre-trained corpus (Jain, 2022).

The significance of topological field analysis for differentiating between German paraphrases was illustrated by de Kok and Hinrichs (2016). For SMS spam identification, it is very useful to examine a topological interpretation of sentence sentiment. If latent sentence associations are not properly recorded, SMS data cannot be analyzed effectively (Gormley et al., 2015). SpaCy implements NN components that have been identified as crucial for short-textual modeling, such as dependency parsers and sentencers (Hu et al., 2022). An expansion of SBERT ideas, a RoBERTa-derived model is modified for the development of spaCy transformers. The dependency parser pipeline component of the spaCy en_core_web_trf model (Honnibal & Johnson, 2014) incorporates backpropagation for NN sentence assessment processing and sentence encoding. In order to support high-rank processing, this modifies the vanilla RoBERTa transformer model and adds a SoftMax modification (Yang et al., 2017). When evaluating vector cosine similarities, this innovative adaptation provides a nuanced representation of raw data by producing lexical embeddings to facilitate paraphrase mining and accommodate sentences. SpaCy makes use of RoBERTa's masking approaches (Liu et al., 2019) and offers superior inputs for downstream modeling, allowing for high output accuracies. SpaCy pipeline tools make it simple to see outcomes and improve user comprehension of internal modeling. Pipelines are essential components of the spaCy platform and are given priority as tools for creating user-extensible models. These modifications and advancements guarantee the feasibility of the task-specific procedure for choosing the best model.

Through approximation-aware techniques, contextualized sentence processing and dependency parsing have made it

possible to achieve restricted runtimes (Gormley et al., 2015). To prevent going above O(n3T) runtime, inherent edge approximations must be used when generating topological inferences. Exponential processing overheads can be efficiently limited by implementations that compare particular topological sentence components (Hu et al., 2022). According to Honnibal and Johnson (2014), selecting suitable within-model parameters allows for the deployment of lightweight transformers. They show how joint incremental dependency parsing, a crucial component of the spaCy transformer model, can be optimized to satisfy low-resource and low-latency requirements. It should be mentioned that researchers don't always reveal their time complexity evaluations. The persistent underutilization of spaCy software, particularly improper transformer use, was a key topic found in this literature assessment. Invalid results and subpar models might be produced by experimental approaches that are not implemented effectively.

Utilizing topological advancements and lightweight processing techniques to find and generate an ideal, implementable model has been a primary goal of our study. This paper's research focuses on finding the best SMS spam detection model based on important assessment criteria, such as strong F1 scores, lightweight implementation capabilities, and features that are easy for users to understand and expand.

2.1. Conceptual structure

Large text blocks can be classified using topic identification models that have already been trained. Basic word counts can be shown using exploratory data analysis (EDA) approaches, such as simple word cloud production. Initial EDA could help an analyst orient the dataset, as seen by inter-topic visualizations. Although topic clustering was unable to improve spam/2-dimensional dataset classification tasks, the inter-topic visualization produced for the SMS Kaggle dataset was instructive (Kaggle, 2017). The dearth of nuance produced by discrete word analysis suggested that techniques like sentencizer processing have a higher chance of producing significant NLP modeling outcomes.

By developing architectural frameworks based on theoretical language modeling and applying the full range of linguistic theories, modern improvements in NLP processing have been made. For the creation of NLP models, the linguistic domain has taken precedence over topic collation (Sartran et al., 2022). Taking a comprehensive approach to NLP modeling offers a theoretical rationale for our study as well as a sound basis for linguistic-based transfer learning analysis (Sasikala et al., 2022).

Whole-of-linguistic techniques in machine learning have been shown to enhance modeling capabilities in previous studies (Güngör et al., 2020). A study of semantics, or meaning derivation, morphology, or structure, and sentence structure mechanics, or syntax, are necessary to integrate the main branches of language theory into a model (Harvard, 2023). Transformer model-based predictive linguistic classifiers are widely acknowledged as essential structures for large-scale sequence-based learning. Modern modeling techniques, such as the application of approximation-aware algorithms, leverage concepts from the linguistic domain to improve machine learning capabilities (Gormley et al., 2015).

Predictive modeling and language phonology (tonal inflections) are not as extensively discussed in the machine learning literature as morphological modeling is. Establishing the applicability of morphological or dependency parsing transformer techniques to spam detection prediction models is the sole focus of this paper.

A linguistic taxonomy or representation of linguistic relationships for all language groups is shown in Figure 2 (Booij & Audring, 2017).

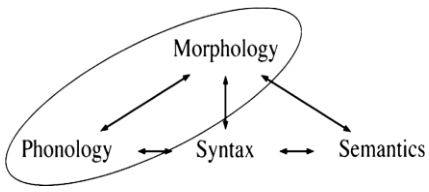


Figure 2: Connections between theories of linguistics

3. Research Methodology

3.1 Research design

Complex Sentence transformers were used in the study to determine the best processing model for SMS spam. For transformer and non-transformer modelling, we employed the topological, open-source NLP techniques built into the spaCy models.

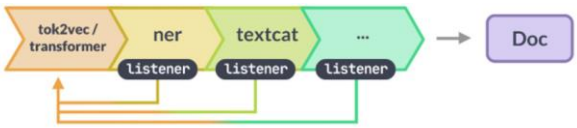
Two publicly accessible datasets, provided by Kaggle and utilized in previous studies, have been used to test three distinct approaches. It is deemed appropriate and suitable to utilize these Kaggle datasets for this research (Kaggle, 2017, 2019). Following the removal of superfluous features, the Twitter and SMS databases are both two-dimensional. The English language sentences in the text-based (domain) dataframe column vary in length. These sentences are categorized as either "ham" or "spam." Before duplicates are removed, the SMS dataset has a length of 5572; after preliminary cleaning, it has a length of 5169. The Twitter dataset has 11,968 and 11,787 metrics, respectively.
The research design that was used was as follows:

1. Execute normal predictive modeling without spaCy objects. This modeling made use of rudimentary processing and NLTK classification tools from Sklearn. We created "Word-Cloud" visualizations using the most frequently occurring terms. Word frequency histograms were created as part of further EDA to illustrate the intrinsic characteristics of the spam dataset. Since the overfitting ruled out any baseline use, we finally threw out the model's results. Additionally, we believed that the processing overheads were too high for CPU implementations.
2. Modeling the SpaCy pipeline with core SpaCy models: The findings from en_core_web_md (_md medium) and en_core_web_lg (_lg large) were compared. SpaCy accesses pre-trained models from language sources. A training pipeline contained the _lg model. By including dependency parsing natively, these models made advantage of topological modeling and pre-trained weights.
3. The model of spaCy en_core_web_trf (_trf transformer): The full-transform model makes use of pre-trained weights and a deep-learning architecture based on RoBERTa. It is designed to make use of a CPU or GPU. Tensor transformations obtained from topological sentence/paraphrase embeddings are optimized using the transformer model. It is intended for usage on downstream jobs with fine-tuning. Transformers hosted on the Hugging Face website are wrapped by SpaCy models.

The Twitter and SMS spam datasets were imported into separate Jupyter notebooks as CSV files. The generated dataframes had one column for the binary classification type (spam or ham) and a column of text "values" or entries encoded in Latin-1. Given that it included significantly more records (11,968) than the SMS data (5572), the validation data (Twitter) was judged appropriate for use as an evaluation dataset. These records, which included slang, short phrases, and a lot of punctuation, were comparable across the two datasets. Information on the binary class numbers was provided by the first EDA, which also showed that while the SMS dataset was extremely unbalanced, particularly after duplicates were eliminated, the Twitter dataset was balanced. Using imported, built-in spaCy routines to eliminate stop words was part of the data cleaning process for the text. To make tokenization and processing within spaCy simple, lemmatizations, stemming, "split," and "lower" techniques were applied to the datasets (Dataquest, 2019). Implementing a Bag-of-Words (BoW) vectorizer and contrasting it with a pipeline that had a TF-IDF vectorizer was one of the first processing strategies. When a BoW vectorizer failed to converge on the Twitter dataset in _trf model tests, the design decision was made to solely process with TF-IDF vectorizers. BoW was rejected as a vectorizer technique after being confirmed to be suitable exclusively for topic modeling. To comprehend topic clustering, topic modeling visuals were created. The unpredictability of word-based approaches was discovered during this pre-processing step, which led to the design decision to use superior sentencizer methods to handle the dataset. The identical SMS message was rendered using the spaCy "DisplaCy" tool, which showed that topological sentence processing outperformed word-based techniques. The efficiency of cosine similarity comparisons used to generate sentence similarity metrics (L2 norm dot products) was demonstrated using sentence similarity pre-processing techniques. After importing pre-trained transformers from the Hugging Face (n.d.) repository, _trf models for dependency parsing and transfer learning were produced. To enable spaCy sentencizing (transfer learning between discrete paraphrases), each message was entered as an input sentence string into the statistical model. In order to preserve training settings, spaCy transferred "remembered" sentence embeddings between pipeline components. The sklearn imbal SMOTE method, tokenized spaCy embeddings, cleaning procedures, and a classifier prediction component were then developed using the sklearn pipeline components.

The spaCy English model pipelines' default configuration parameters are shown in Figures 3 and 4. An Accuracy Evaluation for Sentence Segmentation (F-score) of 0.91 and 0.9, respectively, are possessed by en_core_web_lg and en_core_web_trf. The transformer model outperforms the _lg model for part of speech (pos), ner, and unlabelled dependencies, although it has a marginally poorer sentencizer evaluation accuracy. According to spaCy, the assessed tokenization accuracy for both models is 1.0, as seen in Figure 4 (spaCy, 2023).

All spaCy pipeline components were instantiated, and final modeling was carried out without hyperparameter adjustment of the sklearn downstream models (Peters et al., 2019). To train and test the SMS NLP data, we employed the conventional train_test_split approach with test size 0.2 and seed = 42. To take advantage of tensor processing and inherent topological functionality and steer clear of extra dependencies on deep learning products (like Keras), a sklearn transformer class was created. Because it yielded comparable outcomes to the _trf model (e.g., entity recognition), the _lg pre-trained model was selected to develop comparative measures. Both models may run on a CPU, and the _lg model is made to calculate similarities using tensors that are shared with the pipeline (spaCy, 2023).



PIPELINE	PARSER	TAGGER	NER
en_core_web_trf (spaCy v3)	95.1	97.8	89.8
en_core_web_lg (spaCy v3)	92.0	97.4	85.5

Figure 3: Components of spaCy architecture

The modeling constructions listed below were employed:
1. _lg model using SMOTE oversampling on the SMS dataset
2. The Twitter dataset without SMOTE oversampling; 3. The SMS dataset with SMOTE oversampling; 4. The Twitter dataset without SMOTE oversampling; and 5. The _lg model.
Predictions for the Logistic Regression (LR), Naïve Bayes (NB), Random Forest (RF), and Support Vector Machine (SVC) classifiers were produced by running the modeling four times for each architecture. When the model was ran on the Twitter dataset, this pipeline architecture made it simple to remove the SMOTE oversampling pre-processing phase. The results would have been invalid and meaningless if the model had been used mindlessly without a pipeline adjustment. Accuracy, precision, recall, and F1 were the metrics produced for every model. In all statistical tests, models were compared using F1-scores. For binary classification, this industry standard is a good metric option.
The general model creation process, which adheres to accepted NLP methods, is explained below:
1. Load data and check for irregularities, etc. With the exception of SMS/Tweet text rendering and classification categorization (spam vs. not spam), remove every column. A CPU is used in this investigation to process models.
2. Bring in Python, SpaCy, and Sklearn (load _lg/_trf models). To process text and clean data using generalized methods—that is, to remove stop words and convert words in the ingested sentence to lowercase—create a custom base class.
3. Develop a function that uses built-in spaCy pipeline components (such lemmatization) and incorporates spaCy embeddings. Make sure the outputs from models (_lg or _trf) can be utilized in the chosen sklearn classification model after ingesting SMS text as a sentence. SpaCy learns textural representations by passing "contexts" between built-in components.
4. Build a sklearn pipeline using the relevant classification model, a spaCy tokenizer that is called from a TF/IDF vectorizer, and the custom transformer. There is no more hyperparameter adjustment made to the sklearn classifiers. LR, RF, SVC, and NB classifiers for the _lg and _trf spaCy models, along with a train_test_split of 80:20, are used to process the sklearn pipeline.
5. Use the principal F1-score statistic to analyze and compare the results. Other metrics that are presented include accuracy, precision, and recall. To verify results, create visualizations and confusion matrices for the classification of discrete sentences.

3.2. Statement of Ethics

Both of the research's datasets are widely utilized and openly accessible. The SMS dataset, which was made public with users' cooperation, includes SMS messages from around 2012. The data set was

initially utilized in a doctoral dissertation (Tagg, 2012), and this study is cited in the Kaggle ethics statement (2017). Although it has been utilized in previous studies (Liu et al., 2021) and a Kaggle competition held by the University of Tennessee (Kaggle, 2019), the Twitter dataset is not created for a particular study. On Kaggle, the mitigation technique

employed by the Twitter dataset's data owners is not made explicit. By utilizing just two columns from the dataset—the Type identifier and the Tweet content column—we reduced the possibility of personal identification. This tactic follows the procedures followed by other researchers in this field and duplicates the strategy utilized in the Liu et al. (2021) publication.
The study cannot be generalized to make claims about other languages because the datasets are a collection of sentences spoken in English.

4. New Results

4.1 Original baseline

According Although the basic NLTK analysis produced a word cloud and identified significant terms with ease, modeling using this method was found to be insufficient. The goal of creating a sustainable and reusable model was defeated when training on the Twitter dataset was terminated after 24 hours of processing and modifications to the NLTK implementation were not feasible. When building customized NLTK models for the SMS dataset, processing problems were also noted. The NLTK results had little value because of the extent of overfitting caused by modeling a short dataset. Training on a tiny dataset is inappropriate due to the ease with which exceptionally high accuracies can be reached, according to Gormley et al. (2015). According to Clark et al. (2019) and Vaswani et al. (2017), this offered a compelling argument for including data augmentation (SMOTE) and investigating transformer-based models that do not depend on pattern-matching methods.

4.2. Modeling topologically

It was observed that the default parameters produced remarkably good results on this dataset and that NN Learning Rates for the _trf model did not require adjustment. Fine-tuning classifier hyperparameters resulted in minimal accuracy, while the system's processing time-load costs increased tremendously. As a result, it was decided to employ untuned Sklearn classifiers and train without hypermeters (Peters et al., 2019).
When testing entity recognition methods, this study found model variance. The smaller (_md) statistical spaCy model was unable to effectively classify or describe the SMS data by utilizing built-in topological capability. Only the _lg and _trf implementations were compared because, in the majority of cases, they yielded comparable outcomes. The _lg and _trf models were designed as lightweight systems and were created on a CPU with outstanding processing times (all less than five minutes). These calculated implementation choices guaranteed correct and effective model deployment and output. In this work, the evaluation of topological embeddings was supported by an iterative acceptance testing approach.

Using a transparent pipeline technology, SpaCy was utilized in an evaluation capacity to produce the best lexical findings (Spring & Johnson, 2022). Compared to conventional "black box" models, SpaCy pipeline processing allowed for a deeper comprehension of the transformer output (Honnibal & Montani, 2019).
Other studies typically showed that RF was a better modeling option, however we discovered that F1-scores were the highest for the models of SVC. It seems unusual that 1.0 accuracy was attained on RF (see Table 3). RF_lg model results show a great deal of variability.

Table 1: SMOTE oversampling on _lg sms model

_lg	Accuracy	Precision	Recall	F1
LR	0.9749	0.8846	0.9127	0.8901
NB	0.9768	0.8751	0.9444	0.9084
RF	0.9768	0.9811	0.8254	0.8966
SVC	0.9816	0.9422	0.9048	0.9231

Table 2: Excluding SMOTE oversampling on _lg twitter model

_lg	Accuracy	Precision	Recall	F1
LR	0.8545	0.8380	0.8693	0.8534
NB	0.8613	0.8943	0.8198	0.8506
RF	0.8584	0.8818	0.8188	0.8491
SVC	0.8630	0.8559	0.8641	0.8599

Table 3: SMOTE oversampling on _trf sms model

_trf	Accuracy	Precision	Recall	F1
LR	0.9807	0.9380	0.9098	0.9237
NB	0.9749	0.8741	0.9399	0.9058
RF	0.9758	1.0	0.8120	0.8963
SVC	0.9845	0.9606	0.9173	0.9385

Table 4: Excluding SMOTE oversampling on _trf twitter model

_trf	Accuracy	Precision	Recall	F1
LR	0.8660	0.8471	0.8841	0.8652
NB	0.8774	0.9141	0.8256	0.8676
RF	0.8715	0.8983	0.8240	0.8595
SVC	0.8711	0.8656	0.8678	0.8678

This study deviated from the currently accepted methodology for SMS data analysis by not using a topic/word-modelling approach. The tf_IDF vectorizers were only able to produce accurate ner recognition and dependency parsing on the _trf and _lg models. Additionally, SpaCy visualizations validated that the _trf sentencizer model outperformed the _lg model in terms of processing power. Predictive accuracy increased as a result of this base architecture selection (Table 3).

The results produced by using non-optimal configuration designs are shown in Tables 1, 2, and 4. The best outcomes were obtained by applying the SMOTE augmentation approach to the SVC model. Accuracy, precision, recall, and F1 metrics were used to compare the best model to predictions made by conventional Sklearn classifiers. Models were produced effectively on a CPU because the transformer (tensor) topological pre-processing techniques worked well with the input. Because of the class imbalance and high cost of misclassification while predicting spam, F1 was employed as a comparative statistic rather than accuracy (Statology, 2021).

Best design option for SMS data analysis. High accuracies are achieved via topological transformers using approximation-aware rapid dependency parsing. Edge processing can resolve as approximation by using spaCy transformers (Gormley et al., 2015). This study confirmed that, even with CPU usage, runtimes can continuously appear as O(n3) (Gormley et al., 2015; Honnibal & Johnson, 2014). Our study confirmed that in order to take advantage of the best runtime complexities on a CPU, transformer application extensions need to be properly implemented. Runtime performance is severely impacted when these techniques are used improperly. When compared to _lg non-transformer approaches, untuned statistical spaCy transformer models obtained a good F1-score. The SVC classifier was optimized by fitting SMS data on embedded topological clustering, which also took advantage of built-in topologies using dependency parsing techniques (de Kok & Hinrichs, 2016). The introduction of SoftMax extension applications and high-rank data renderings made it possible to access hitherto unexplored topological data expressions (Yang et al., 2017). Both short-text and large-text NLP tasks seem to benefit from prior studies that looked at how tuning affects sequential inductive transfer learning (Peters et al., 2019). At both the transformer and downstream classifier levels, we have evaluated our model with little user involvement. To the best of our knowledge, the use of untuned pipelines for classification modeling is a novel technique, and the SMS dataset is made up of short-sentence components.

In order to mine comparable paraphrases within sentences, dependency parsing is integrated as a core component of the spaCy pipelines. A basic method for applying spaCy models is pipeline processing. A pipeline architecture is used in the study to guarantee expandable production functionality. Sklearn pipeline modeling allowed for the smooth generation of F1-scores by mirroring the spaCy architecture. POS tagging and entity recognition visualizations were used to demonstrate the intrinsic sentence transformer structure. The pre-trained model began contextual "learning" through the classification of semantically related sentences, which made it possible to uncover hidden links (Gormley et al., 2015). Nuanced learning of short-message data was made possible by SpaCy's open-source technologies, which guaranteed access to the best processing techniques. By using succinct sentence dependency parsing techniques, tensors could be rendered topologically effectively (Moliner et al., 2020). Semantically related latent expressions embedded in SMS messages were successfully captured by appropriate system design decisions. Because RoBERTa uses BERT pre-trained models that were obtained from internet scraping data, it is biased (Hugging Face, n.d.; Jain, 2022). Through transparent modeling, bias and iterative reprocessing modifications can be informed by precision and recall (Bartička et al., 2022). Minority class predictions are supported by giving accuracy results on unbalanced SMS data priority (Brownlee, 2020). Given the significant variability of SMS data, algorithmic fairness may be enhanced by

examining topological transformer techniques through adversarial sampling, accuracy, and F1-scoring (Zhang et al., 2018). Transparent, user-centric evaluation techniques are given priority in current work on unbiased transformer modeling (Modarressi et al., 2022).

This work develops a lightweight, user-extensible, cutting-edge solution to the SMS spam detection problem using open-source techniques. Due to its ease of processing and compatibility with sklearn algorithms, SpaCy was selected for implementation tasks. One of the inherent implementation hazards of relying on open-source technology is that a system may not be able to meet certain SMS data needs without requiring extensive model deconstruction. Work on Bernoulli latent variables has to be monitored, because SoftMax advancements for discrete data have not yet reached their full potential (Yang et al., 2017). For advanced forms of SMS text spam to be processed efficiently, system changes could be necessary.

5. Conclusion

The study's findings show that SMS data can be processed using contemporary NLP techniques. Based on a combination of classifiers and sampling strategies, the tested models yielded a range of outcomes for the Twitter and SMS datasets. The study found that the best use of topological data was achieved via sklearn pipeline implementations and spaCy sentence transformers, which produced a maximum F1-score of 0.938. Because a lightweight, transparent, and user-extensible architecture was used to provide superior F1-scores, the modeling can be regarded as optimal. These characteristics were thought to be suitable assessment tools for impartially determining manufacturing suitability. The study showed that short-sentence datasets derived from SMS texts might be handled as documents and categorized as best as possible. Since SpaCy is a product that is always changing, this study offers a design strategy that calls for little involvement from end users.

References

Abid, M. A., Ullah, S., Siddique, M. A., Mushtaq, M. F., Aljedaani, W., & Rustam, F. (2022). Spam SMS filtering on text features and supervised machine learning techniques. *Multimedia Tools and Applications*, *81*(28), 39853–39871. https://doi.org/10. 1007/s11042-022-12991-0

Booij, G., & Audring, J. (2017). Construction morphology and the parallel architecture of grammar. *Cognitive Science*, *41*(S2), 277–302. https://doi.org/10.1111/cogs.12323

Bartička, V., Pražák, O., Konopík, M., & Sido, J. (2022). Evaluating attribution methods for explainable NLP transformers. In *2022 International Conference on Text, Speech, and Dialogue*, 3–15. https://link.springer.com/content/pdf/10.1007/978-3-030-58323-1.pdf

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., .. . , & Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcon & H. Lin (Eds.), *Advances in neural information processing systems, 33*, (pp. 1877–1901). Curran Associates, Inc. https://arxiv.org/pdf/2005.14165.pdf

Brownlee, J. (2020). *How to calculate precision, recall, and F1- measure for imbalanced classification*. Retrieved from: https://machinelearningmastery.com/precision-recall-and-f-measure-for- imbalanced-classification/

Chiew, K. L., Tan, C. L., Wong, K. S., Yong, K. S. C., & Tiong, W. K. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection systems. *Information Sciences*, *484*, 153–166. https://doi.org/ 10.1016/j.ins.2019.01.064

Clark, K., Khandelwal, U., Levy, O., & Manning, C. D. (2019). *What does BERT look at? An analysis of BERT's attention*. *arXiv Preprint: 1906.04341*. https://doi.org/10.48550/arXiv.1906.04341 Conneau, A., Khandelwal, K. M., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., . . . , & Stoyanov, V. (2019). *Unsupervised cross- lingual representation learning at scale*. *arXiv Preprint*: https://doi. org/10.48550/arXiv.1911.02116

Dataquest. (2019). *Tutorial: Text classification in Python using spaCy*. Retrieved from: https://www.dataquest.io/blog/tutorial-text-classification-in-python-using-spacy

de Kok, D., & Hinrichs, E. (2016). Transition-based dependency parsing with topological fields. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2, 1–7.

Gormley, M. R., Dredze, M., & Eisner, J. (2015). Approximation- aware dependency parsing by belief propagation. *Transactions of the Association for Computational Linguistics*, *3*, 489–501. https://doi.org/10.1162/tacl_a_00153

Güngör, O., Güngör, T., & Uskudarli, S. (2020). EXSEQREG: Explaining sequence-based NLP tasks with regions with a case study using morphological features for named entitity recognition. *PLoS ONE*, *15*(12), e0244179. https://doi.org/ 10.1371/journal.pone.0244179

Harvard. (2023). *Linguistic theory*. *Harvard Kenneth C. Griffin Graduate School of Arts and Sciences*. Retrieved from: https://gsas.harvard.edu/policy/linguistic-theory.

Haynes, K., Shirazi, H., & Ray, I. (2021). Lightweight URL-based phishing detection using natural language processing transformers for mobile devices. *Procedia Computer Science*, *191*, 127–134.

https://doi.org/10.1016/j.procs.2021.07.040

Honnibal, M., & Johnson, M. (2014). Joint incremental disfluency detection and dependency parsing. *Transactions of the Association for Computational Linguistics*, *2*, 131–142. https://aclanthology.org/Q14-1011.pdf

Honnibal, M., & Montani, I. (2019). *SpaCy meets transformers: Fine-tune BERT, XLNet and GPT-2*. https://explosion.ai/blog/spacy-transformers

Hu, C., Gong, H., & He, Y. (2022). Data driven identification of international cutting-edge science and technologies using SpaCy. *PLoS ONE*, *17*(10), e0275872. https://doi.org/10.1371/ journal.pone.0275872

Hugging Face (n.d.). *RoBERTa-base*. Retrieved from: https:// huggingface.co/RoBERTa-base

Islam, M. R., Liu, S., Wang, X., & Xu, G. (2020). Deep learning for misinformation detection on online social networks: A survey and new perspectives. *Social Network Analysis and Mining*, *10*(1), 82. https://doi.org/10.1007/s13278-020-00696-x

Jain, S. M. (2022). *Introduction to transformers for NLP*. USA: Apress. https://doi.org/10.1007/978-1-4842-8844-3_4

Kaggle. (2017). *SMS spam collection dataset*. Retrieved from: https:// www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

Kaggle. (2019). *Utkml's Twitter spam detection competition*. Retrieved from: https://www.kaggle.com/competitions/utkmls-twitter-spam-detection-competition/overview

Liu, X., Lu, H., & Nayak, A. (2021). A transformer model for SMS spam detection. *IEEE Access*, *9*, 80253–80263. https://doi.org/ 10.1109/ACCESS.2021.3081479

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., …, & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv Preprint*. https://doi.org/ 10.48550/arxiv.1907.11692

May, C., Wang, A., Bordia, S., Bowman, S. R., & Rudinger, R. (2019). On measuring social biases in sentence encoders. *arXiv Preprint*. https://doi.org/10.48550/arXiv.1903.10561

Modarressi, A., Fayyaz, M., Yaghoobzadeh, Y., & Pilehvar, M. T. (2022). GlobEnc: Quantifying token attribution by incorporating the whole encoder layer in transformers. *arXiv Preprint*. https:// doi.org/10.48550/arXiv.2205.03286

Moliner, P. E., Heunen, C., & Tull, S. (2020). Tensor topology. *Journal of Pure and Applied Algebra*, *224*(10), 106378. https://doi.org/10.1016/j.jpaa.2020.106378

Nguyen, N. T. H., Ha, P. P. D., Nguyen, L. T., Nguyen, K. V., & Nguyen, N. L. T. (2022). SPBERTQA: A two-stage question answering system based on sentence transformers for medical texts. *arXiv Preprint*. https://doi.org/10.48550/ arXiv.2206.09600

Peters, M., Ruder, S., & Smith, N. A. (2019). To tune or not to tune? Adapting pretrained representations to diverse tasks. *arXiv Preprint*. https://doi.org/10.48550/arXiv.1903.05987

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT networks. *arXiv Preprint*. https://doi.org/10.48550/arXiv.1908.10084

Roy, P. K., Singh, J. P., & Banajee, S. (2020). Deep learning to filter SMS spam. *Future Generation Computer Systems*, *102*, 524–533. https://doi.org/10.1016/j.future.2019.09.001

Sartran, L., Barrett, S., Kuncoro, A., Stanojevic´, M., Blunsom, P., & Dyer, C. (2022). Transformer grammars: Augmenting transformer language models with syntactic inductive biases at scale. *Transactions of the Association for Computational Linguistics*, *10*, 1423–1439. https://doi.org/ 10.1162/tacl_a_00526

Sasikala, S., Ramesh, S., Gomathi, S., Balambigai, S., & Anbumani, V. (2022). Transfer learning based recurrent neural network algorithm for linguistic analysis. *Concurrency and Computation: Practice and Experience*, *34*(5), e6708. https://doi.org/10.1002/cpe.6708

Shirazi, H., Muramudalige, S. R., Ray, I., Jayasumana, A. P., & Wang, H. (2023). Adversarial autoencoder data synthesis for enhancing machine learning-based phishing detection algorithms. *IEEE Transactions on Services Computing*, *16*(4), 2411–2422. https://doi.org/10.1109/TSC2023.3234806.

Shwartz-Ziv, R., & Armon, A. (2021). Tabular data: Deep learning is not all you need. *arXiv Preprint*. https://doi.org/10.48550/ arXiv.2106.03253

SpaCy. (2023). *What's new in v3.0*. Retrieved from: https://spacy.io/ usage/v3#features-transformers

Spring, R., & Johnson, M. (2022). The possibility of improving calculation of measures of lexical richness for EFL writing: A comparison of the LCA, NLTK and SpaCy tools. *System*, *106*, 102770. https://doi.org/10.1016/j.system.2022.102770

Statology. (2021). *F1 score vs. accuracy: Which should you use?* Retrieved from: https://www.statology.org/f1-score-vs-accuracy/.

Tagg, C. (2012). *Discourse of text messaging: Analysis of SMS communication*. UK: Continuum International Publishing Group.

Tan, C. L., Chiew, K. L., Yong, K. S. C., Sze, S. N., Abdullah, J., & Sebastian, Y. (2020). A graph-theoretic approach for the detection of phishing webpages. *Computers & Security*, *95*, 101793. https://doi.org/10.1016/j.cose.2020.101793

Tumas, V., Rivera, S., Magoni, D., & State, R. (2022). Topology analysis of the XRP ledger. *arXiv Preprint*: https://doi.org/ 10.48550/arXiv.2205.00869

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett (Eds.), *Advances in neural information processing systems, 30,* (pp. 5998–6008). Curran Associates, Inc.

Xia, T., & Chen, X. (2020). A discrete hidden Markov model for SMS spam detection. *Applied Sciences*, *10*(14), 5011. https://doi.org/10.3390/app10145011

Xu, L., Yan, X., Ding, W., & Lu, Z. (2023). Attribution rollout: A new way to interpret visual transformer. *Journal of Ambient Intelligence and Humanized Computing*, *14*(1), 163–173. https://doi.org/10.1007/s12652-022-04354-2

Yang, Z., Dai, Z., Salakhutdinov, R., & Cohen, W. W. (2017). Breaking the softmax bottleneck: A high-rank RNN language model. *arXiv Preprint*. https://doi.org/10.48550/ arXiv.1711.03953

Zhang, B. H., Lemoine, B., & Mitchell M. (2018). Mitigating unwanted biases with adversarial learning. *arXiv Preprint*. https://doi.org/10.48550/arXiv.1801.07593