Received: 27 May 2023 | Revised: 27 June 2023 | Accepted: 12 July 2023 | Published online: 12 July 2023

RESEARCH ARTICLE

Journal of Data Science and Intelligent Systems
2024, Vol. 2(1) 173–181
DOI: 10.47852/bonviewJDSIS32021131

BON VIEW PUBLISHING

Establishing an Optimal Online Phishing Detection Method: Evaluating Topological NLP Transformers on Text Message Data

Helen Milner[1] **and Michael Baron**[2,*]

[1]*University of Adelaide, Australia*

[2]*CME Department, Charles Sturt University, Australia*

Abstract: This research establishes an optimal classification model for online SMS spam detection by utilizing topological sentence transformer methodologies. The study is a response to the increasing sophisticated and disruptive activities of malicious actors. We present a viable lightweight integration of pre-trained NLP repository models with sklearn functionality. The study design mirrors the spaCy pipeline component architecture in a downstream sklearn pipeline implementation and introduces a user-extensible spam SMS solution. We leverage large-text data models from HuggingFace (RoBERTa-base) via spaCy and apply linguistic NLP transformer methods to short-sentence NLP datasets. We compare the F1-scores of models and iteratively retest models using a standard sklearn pipeline architecture. Applying spaCy transformer modelling achieves an optimal F1-score of 0.938, a result comparable to existing research output from contemporary BERT/SBERT/'black box' predictive models. This research introduces a lightweight, user-interpretable, standardized, predictive SMS spam detection model that utilizes semantically similar paraphrase/sentence transformer methodologies and generates optimal F1-scores for an SMS dataset. Significant F1-scores are also generated for a Twitter evaluation set, indicating potential real-world suitability.

Keywords: dependency parsing, phishing, topological transformer processing, transfer learning

1. Introduction

NLP linguistic machine learning frameworks have fundamentally altered text-based dataset modelling since 2016 (Brown et al., 2020). The research presented here extends foundational contemporary approaches to predictive linguistic modelling and demonstrates methodological relevance for resource-constrained, short-text spam detection tasks.

SMS spam detection remains a necessary task (Haynes et al., 2021), and threat vectors have become highly automated. Short-text message model predictions are typically serviced by older technological instruments, making them vulnerable. SMS spam data classification tasks using new transformer methods do not appear to be well-researched (Roy et al., 2020) or methodologically aligned to transformer architectures. This study presents exploratory research from The University of Adelaide examining the effect of both sophisticated topological transformer and large vector methods on short-text SMS spam classification.

The current iteration of transformers can imbed dense tensors into topological frameworks from sentence-based inputs, making these architectures a good fit for short-text data. High-rank embeddings and pre-trained libraries have proved crucial for modelling NLP tasks (Yang et al., 2017), for example RoBERTa (Liu et al., 2019). The self-supervised, sentence-based processing of RoBERTa-base is embedded as the statistical model for the spaCy transformer pipeline. Roberta-base generates encoded weights for use in downstream standard classifiers, making this model a suitable choice for a task-based transparent solution. SpaCy provides a tagger tokenizer, a (dependency) parser and an ner (named entity recognizer) to listen to the transformer component (output). The generated output is used by our study to classify spam text data within an sklearn pipeline.

Spam messaging content relevance is typically short-lived, and this research focuses on identifying an optimal state-of-the-art design, based on typical classification modelling metrics. A major issue for all models is the choice of language sample sizes and the specific vocabulary included in datasets (Conneau et al., 2019). Haynes et al. (2021) highlight the need to avoid visiting dangerous sites and advocate using publicly available datasets in phishing detection research. This methodological constraint immediately limits the sample size of available data and presents a persistent problem for SMS researchers. The issue for spam detection systems is that evolving illicit SMS message generation techniques can result in redundant training datasets, largely unrepresentative of current trends. We find that this issue is not significant when using spaCy component-modelling pipelines.

This research applies NLP-based classification methods in a time-constrained environment. The study responds to the short-lived nature

*Corresponding author: Michael Baron, University of Adelaide and CME Department, Charles Sturt University, Australia. Email: mbaron@csu.edu.au

of SMS messaging and develops a suitable solution for time-critical and resource-constrained environments. Addressing constraints encourages an iterative, agile design implementation.

1.1. Our approach

A basic NLTK Regex model is initially implemented and tested to replicate prior base-level research. This model produces highly accurate predictions using pattern-matching techniques; however, subsequent testing reveals overfitted results considered inadequate for benchmarking purposes. Pattern-matching spam filters are redundant technological instruments in the contemporary literature. These methods are not considered future-facing or fit for purpose when designing spam identification solutions (Shirazi et al., 2023).

Our design provides a suitable template for developing enhanced, future-facing models. The study iteratively fits a classic SMS dataset (Kaggle, 2017) to a predictive classification model, using open-source component pipeline architectures. Constant checking of the model outputs enables the development of two lightweight statistical NLP models, both leveraging pre-trained neural network (NN) embeddings and completing in <5 mins. The first comparison model uses a large language collection of unique vectors from the web; the second used a transformer pipeline insertion. Modelling is conducted using package defaults. This work compares word-vector similarity modelling with sentence-based transformer spam classification methods and finds that spaCy transformer statistical modelling generates superior F1 scores.

The en_core_web_trf spaCy transformer model is chosen for transformer modelling as it seamlessly incorporates pre-trained CommonCrawl data from the RoBERTa-base model (Liu et al., 2019), hosted on the Hugging Face (n.d.) repository. The transformer pipeline architecture generates embedded weights using dependency parsers and entity detection technologies (Gormley et al., 2015). The pipeline architecture uses standard components (e.g., sentencizer vs senter) to generate outputs for downstream classification tasks. Downstream implementation uses sklearn pipeline functionality to create a custom predictor and fine-tunes the spaCy statistical models on a CPU. The pipeline incorporates a custom SMOTE oversampling method to balance the SMS dataset and prevent overfitting (Abid et al., 2022).

We assess the generated F1-score for each modelling cycle, and an iterative implementation approach provides confidence that the ultimate result is reproduceable and optimal. State-of-the-art short-text binary transformer modelling identifies inferences and creates contextual topological embeddings to feed into a pipeline and generate predictions. Open-source semantic similarity detection techniques on sentences are implemented, and we validate this work against a Twitter dataset, as per previous research (Liu et al., 2021).

This work achieves superior classification results (accuracy and F1-scores) using spaCy transformer pipeline modelling, compared to previous research that implements transformer architectures from scratch. Open-source, extensible architectures provided superior alternatives to new-build NN research, and our lightweight CPU-based implementation achieved comparable accuracies to GPU processing (0.9845 with an SVC classifier) when pre-trained, default spaCy modelling pipelines were utilized.

A significant contribution of our work is the successful generation of a user-extensible, highly accurate, topological transformer-based spam detection method for SMS data. The study demonstrates that state-of-the-art transformer solutions provide a clear direction for the future of SMS classifiers.

2. Literature Review

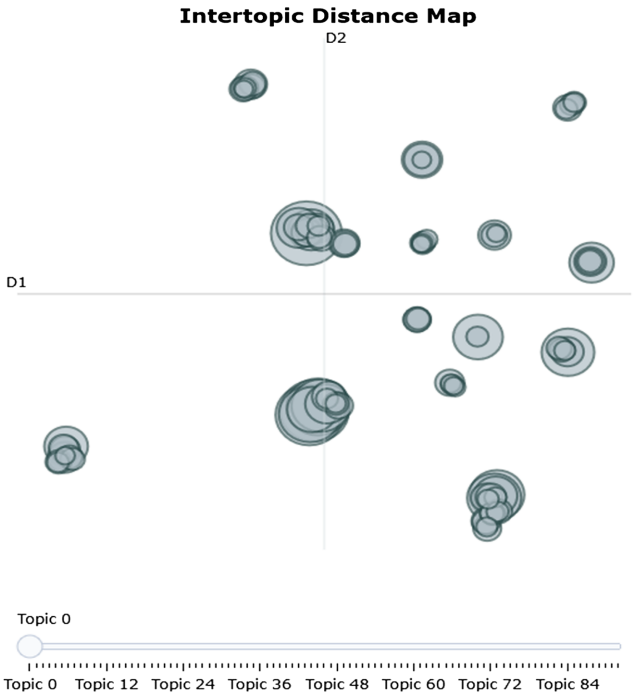
SMS spam detection has been relatively neglected within the security research domain (Roy et al., 2020), and prevalent spam detection studies have prioritized fraudulent email modelling (Chiew et al., 2019; Tan et al., 2020).

The SMS dataset has been employed to predict spam by utilizing complex, new-build NN implementations (Liu et al., 2021; Roy et al., 2020). New NN builds are memory intensive (Haynes et al., 2021) and do not appropriately compartmentalize the necessary classification methods with user-extensible components (e.g., pipelines).

Older research does not employ sentence transformers or generally consider solution time complexities. Previous research using Hidden Markov Models (HMM) on the SMS dataset reported favourable time complexity processing overheads (Xia & Chen, 2020); however, HMM models are word-centric and utilize methods intrinsically linked with forward-fed deep-learning, considering only one prior state. Contemporary post-2019 research has largely discarded word embedding modelling for sophisticated sentence/paraphrasing methods.

Figure 1 illustrates the wide variance of results when word-based modelling methods are used for linguistic dataset processing.

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



Naively employed vectorizers, for example GloVe and word2Vec word embedding methods, fail bias assessment tests and cannot appropriately capture linguistic meaning embedded within paraphrases (May et al., 2019). The Hugging Face site explicitly highlights this weakness of RoBERTa models. Hugging Face states that bias is an inherent limitation of pre-trained models (Hugging Face, n.d.); therefore, identifying and actioning bias in models is a live problem. Implementing transformer-based solutions over word embedding methods is gaining acceptance as a solution to the problem of algorithmic bias (Islam et al., 2020).

Tensor topologies capture spatial/relational information (Tumas et al., 2022), and low-rank topological vectorization has been used to enable task-specific approaches to classification (Brown et al., 2020; Shwartz-Ziv & Armon, 2021). Gormley et al. (2015) examined lexical feature embeddings via low-rank tensors and focused on training efficient dependency parsers for text. Reimers and Gurevych (2019) utilized dependency parsing techniques in their model SBERT (SentenceBERT) to investigate topological semantic similarities for sentence pairs, utilizing similarly large training data. These authors examined sentence paraphrasing mining techniques to compare short blocks of text, an important option for classifying SMS spam (Reimers & Gurevych, 2019).

Spam SMS identification in the wild is recognized as a non-trivial task (Shirazi et al., 2023), and backpropagation transformer modelling is required to succinctly capture embedded latent complexities (Xu et al., 2023). Transformer models use backpropagation to predict output from a large pre-trained corpus and applying pre-trained transformer models to a targeted topology has achieved high accuracies on unlabelled data/unsupervized modelling (Jain, 2022).

de Kok and Hinrichs (2016) demonstrated the importance of topological field analysis for discriminating between German paraphrases. Interrogating a topological rendering of sentence sentiment is particularly applicable to SMS spam detection. SMS data cannot be adequately analysed if latent sentence relationships are not fully captured (Gormley et al., 2015). Components of NNs identified as important for short-textual modelling, for example sentencizers and dependency parsers, are implemented within spaCy (Hu et al., 2022). A RoBERTa-derived model is adapted for spaCy transformer implementation and is an extension of SBERT concepts. Backpropagation for NN sentence assessment processing and sentence encoding is implemented in the spaCy en_core_web_trf model (Honnibal & Johnson, 2014) and embedded in the dependency parser pipeline component. This adapts the vanilla RoBERTa transformer model and introduces a SoftMax change to enable high-rank processing (Yang et al., 2017). This novel adaption generates lexical embeddings to accommodate sentences and support paraphrase mining, providing a nuanced representation of raw data when assessing vector cosine similarities. SpaCy utilizes the masking techniques of RoBERTa (Liu et al., 2019) and provides excellent inputs to the downstream modelling, enabling high output accuracies. SpaCy pipeline tools enable easy visualization of results and enhance user understanding of the internal modelling. Pipelines are prioritized as tools for developing user-extensible models and are intrinsic to the spaCy platform. These adaptations and developments ensure that the task-specific process of selecting an optimal model is viable.

Contextualized sentence processing/dependency parsing has enabled the realization of constrained runtimes, via approximation aware methods (Gormley et al., 2015). Inherent edge approximations during topological inference generation must be implemented to avoid exceeding $O(n^3T)$ runtime. Implementations based on comparing specific topological sentence components (Hu et al., 2022) effectively limit exponential processing overheads. Honnibal and Johnson (2014) argue that lightweight transformer implementation is achieved by choosing appropriate within-model parameters. They demonstrate that low-resource and low-latency requirements can be met by optimizing joint incremental dependency parsing, a key construct of the spaCy transformer model. It should be noted that time complexity assessments are not regularly disclosed by researchers. A major theme identified from this literature review was the consistent underutilization of spaCy software, including inappropriate transformer use. Ineffective implementations of experimental methods can generate sub-standard models and invalid results.

A major objective of our research has been to implement lightweight processing methodologies and use best-practice topological developments to identify and produce an optimal, implementable model. The research presented in this paper focuses on identifying an optimal SMS spam detection model from key evaluation criteria, including high F1 scores, lightweight implementation capabilities and user interpretability/extensibility features.

2.1. Theoretical framework

Topic identification models are suitable for classifying large text blocks via pre-trained models. Exploratory data analysis (EDA) techniques, for example rudimentary word cloud generations, can provide a visual understanding of basic word counts. Inter-topic visualizations demonstrated that initial EDA could orient the dataset for an analyst. The inter-topic visualization generated for the SMS Kaggle dataset was informative (Kaggle, 2017), but topic clustering proved incapable of enhancing spam/2-dimensional dataset classification tasks. The lack of available nuance generated from discrete word analysis implied that methods such as sentencizer processing are more likely to yield important NLP modelling results.

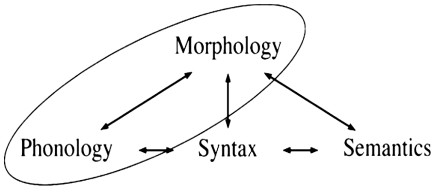
Modern advances in NLP processing have been achieved by creating architectural frameworks based on theoretical linguistic modelling and utilization of the entire suite of linguistic theories. The linguistic domain has become more important for NLP model construction than topic collation (Sartran et al., 2022). Approaching NLP modelling holistically provides a valid foundational approach to linguistic-based transfer learning analysis (Sasikala et al., 2022) and presents a theoretical justification for our research.

Prior research has demonstrated that whole-of-linguistic approaches in machine learning contribute to improved modelling capability (Güngör et al., 2020). Incorporating the major divisions of linguistic theory into a model involves an examination of sentence structure mechanics (syntax), morphology (structure) and semantics, or meaning derivation (Harvard, 2023). Predictive linguistic classifiers built with transformer models are now recognized as necessary architectures for sequence-based learning at scale. Contemporary modelling approaches use ideas inherent within the linguistic domain to enhance machine learning capabilities, for example implementation of approximation aware methods (Gormley et al., 2015).

The machine learning literature does not address relationships between linguistic phonology (tonal inflections) and predictive modelling to the same extent as morphological modelling. This paper is limited to establishing the relevance of morphological or dependency parsing transformer techniques to spam detection prediction models.

Figure 2 (Booij & Audring, 2017 illustrates a linguistic inter-relationship representation or taxonomy for all language groups, derived by Booij and Audring (2017).

Figure 2
Connections between theories of linguistics



3. Research Methodology

3.1. Research design

The study involved identifying an optimal processing model for SMS spam using sentence transformers. We used the open-source, topological NLP methods inherent within the spaCy models for transformer and non-transformer modelling.

Three discrete methods have been tested on two publicly available datasets, as hosted by Kaggle and used in prior research. These Kaggle datasets are considered valid and legitimate to use for this study (Kaggle, 2017, 2019). The SMS and Twitter datasets are both two dimensional after dropping irrelevant features. The text-based (domain) dataframe column contains English language sentences of varying length. These sentences are classified as 'spam' or 'ham'. The SMS dataset is of length 5572 before deleting duplicates and 5169 once initial cleaning has been undertaken. The metrics of the Twitter dataset are 11,968 and 11,787, respectively.

The following research design was implemented:

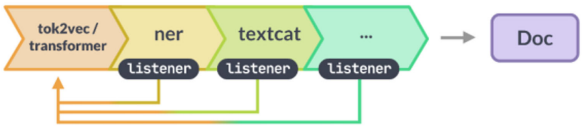
1. Run standard predictive modelling without spaCy objects: This modelling used NLTK classification tools from sklearn and primitive processing. We generated visualizations from the highest occurring words as a 'Word-Cloud'. Other EDA included production of word frequency histograms to demonstrate inherent properties of the spam dataset. We ultimately discarded the results from this model as the overfitting excluded any baseline use. We also considered the processing overheads to be excessive for CPU implementations.
2. SpaCy pipeline modelling using core spaCy models: Pre-trained models from linguistic repositories are accessed by spaCy, and we compared results from en_core_web_md (_md medium) and en_core_web_lg (_lg large). The _lg model was included in a training pipeline. These models leveraged pre-trained weights and used topological modelling, by natively embedding dependency parsing.
3. spaCy en_core_web_trf (_trf transformer) model: The full-transformer model utilizes a RoBERTa-base deep-learning architecture and pre-trained weights. It is built to use a GPU or CPU infrastructure. The transformer model optimizes tensor transformations derived from topological sentence/paraphrase embeddings. It is designed to be used with fine-tuning on downstream tasks. SpaCy models provide a wrapper for transformers hosted on the Hugging Face site.

The SMS spam and Twitter datasets were imported as.csv files into individual Jupyter notebooks. The resulting dataframes contained a latin-1 encoded column of text 'values'/entries and a single column for the binary classification category (spam or ham). The validation data (Twitter) were deemed relevant to use as an evaluation dataset as it contained substantially more records (11,968) than the SMS data (5572). These records were similar in both datasets, consisting of short sentences, slang and excessive punctuation. The initial EDA provided information on the binary class numbers and revealed that the Twitter dataset was balanced but the SMS dataset was highly imbalanced, especially when duplicates were removed. Data cleaning of the text involved applying imported, inbuilt spaCy functions to remove stop words. Lemmatizations, stemming, 'split' and 'lower' methods were applied to the datasets to ensure easy tokenization and processing within spaCy (Dataquest, 2019).

An initial processing tactic was to implement a Bag-of-Words (BoW) vectorizer and compare it to a pipeline containing a TF-IDF vectorizer. A design decision was made to only process with TF-IDF vectorizers when a BoW vectorizer could not converge on the Twitter dataset in _trf model tests. BoW was verified as only appropriate for topic modelling and discarded as a vectorizer method.

Topic modelling visualizations were generated to understand topic clustering. This pre-processing step identified the variability of word-based methods and resulted in a design decision to process the dataset using superior sentencizer methods. The spaCy 'DisplaCy' tool was used to visualize renderings of the same SMS message and demonstrated that topological sentence processes were superior to word-based methods. Sentence similarity pre-processing methods were used to demonstrate the effectiveness of cosine similarity comparisons utilized for generating sentence similarity metrics (L2 norm dot products). Pre-trained transformers from the Hugging Face (n.d.) repository were imported, and _trf models for transfer learning/dependency parsing were generated. Each message was fed into the statistical model as an input sentence string, to enable spaCy sentencizing (transfer learning between discrete paraphrases). spaCy passed 'remembered' sentence embeddings between pipeline components to retain training contexts. Sklearn pipeline components were subsequently implemented to action the sklearn imbal SMOTE method, call tokenized spaCy embeddings, call cleaning operations and initiate a classifier prediction component.

Figure 3
Components of spaCy architecture



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

In Figure 3 (spaCy, 2023), the tok2vec pipeline component is used by the _lg model and the transformer component is used by the _trf (RoBERTa-base) model. We generated higher F1 scores for the SMS dataset using the _trf model.

Figure 4

Model	Pipeline	Vectors	Type
en_core_web_lg	tok2vec, tagger, parser, attribute_ruler, lemmatizer, ner	514k keys, 514k unique vectors, 300 dim	Vocab., syntax, entities, vectors
en_core_web_trf	transformer, tagger, parser, attribute_ruler, lemmatizer, ner	0	Vocab, syntax, entities

Figures 3 and 4 reference default configuration parameters embedded in spaCy English model pipelines.

en_core_web_lg has an Accuracy Evaluation for Sentence Segmentation (F-score) of 0.91, en_core_web_trf has an Accuracy Evaluation for Sentence Segmentation (F-score) of 0.9. The

transformer model has a slightly lower sentencizer evaluation accuracy but surpasses the _lg model for part of speech (pos), ner and unlabelled dependencies. As Figure 4 shows, spaCy reports that the evaluated accuracy of tokenization for both models is 1.0 (spaCy, 2023).

Final modelling was conducted without hyperparameter tuning of the sklearn downstream models (Peters et al., 2019), and all spaCy pipeline components were instantiated. We used a standard train_test_split method, test size 0.2 and seed = 42 to train and test the SMS NLP data. An sklearn transformer class was implemented to utilize tensor processing/inherent topological functionality and avoid additional deep-learning product (e.g., Keras) dependencies. The _lg pre-trained model was chosen to generate comparison metrics because it produced similar results to the _trf model (e.g., entity recognition). The _lg model is designed to compute similarities via tensors shared with the pipeline (spaCy, 2023), and both models can operate using a CPU.

The following modelling constructions were used:

1. _lg model on the SMS dataset with SMOTE oversampling
2. _lg model on the Twitter dataset without SMOTE oversampling
3. _trf model on the SMS dataset with SMOTE oversampling
4. _trf model on the Twitter dataset without SMOTE oversampling.

Modelling was run four times per construction to generate predictions for Logistic Regression (LR), Naïve Bayes (NB), Random Forest (RF) and Support Vector Machine (SVC) classifiers. This pipeline architecture ensured easy removal of the SMOTE oversampling pre-processing step when running the model on the Twitter dataset. Using the model blindly without a pipeline adjustment would have produced irrelevant and invalid results. The metrics generated for each model consisted of accuracy, precision, recall and F1.

All statistical testing used F1-scores to compare models. This industry standard is a suitable metric choice for binary classification.

The general construction of models is described below and follows standard NLP practices:

1. Load data and examine for anomalies etc. Delete all columns except SMS/Tweet text rendering and classification categorization (spam vs. not spam). This study utilizes a CPU to process models.
2. Import sklearn, python, spaCy (load _lg/_trf models). Create a custom base class to process text and clean data using generalized methods, that is, convert words in the ingested sentence to lowercase and remove stop words.
3. Create a function to incorporate spaCy embeddings and use inbuilt spaCy pipeline components (e.g., lemmatization). Ingest SMS text as a sentence and ensure outputs from models (_lg or _trf) can be used in the selected sklearn classification model. spaCy passes 'contexts' between inbuilt components, to learn textural representations.
4. Construct an sklearn pipeline from the custom transformer, a spaCy tokenizer called from a TF/IDF vectorizer and relevant classification model. No additional hyperparameter tuning is applied to the sklearn classifiers. The sklearn pipeline is processed using LR, RF, SVC and NB classifiers for the _lg and _trf spaCy models and a train_test_split of 80:20.
5. Analyse and compare results using the primary F1-score metric. Accuracy, precision and recall are also reported as additional metrics. Generate confusion matrices and visualizations of discrete sentence classification to validate results.

3.2. Ethics statement

The two datasets used for the research are publicly available and widely used. The SMS dataset contains SMS messages from approx. 2012 and released with the consent of the participants. The dataset was originally used in a PhD thesis (Tagg, 2012) and the ethics statement on Kaggle (2017) references this study. The Twitter dataset is not generated for a specific paper, although it has been used in prior research (Liu et al., 2021) and a Kaggle competition, hosted by the University of Tennessee (Kaggle, 2019). The mitigation strategy used by the data owners of the Twitter dataset and is not clearly specified on Kaggle. We mitigated personal identification risks by only using two columns from the dataset, namely the Tweet content column and the Type identifier. This strategy replicates the approach taken in the Liu et al. (2021) paper and adheres to methodologies used by other researchers in this field.

The datasets are a collection of English-speaking sentences, and therefore, the study cannot be extrapolated to make assertions on other languages.

4. Results

4.1. Original baseline

The basic NLTK analysis easily identified key words and generated a word cloud but modelling with this method proved to be inadequate. Training on the Twitter dataset was cancelled after 24 h of processing, and adjustments on the NLTK implementation were infeasible, defeating the object of developing a sustainable and reusable model. Processing issues were also observed when constructing modified NLTK models for the SMS dataset. The degree of overfitting from modelling a small dataset rendered the NLTK results meaningless. Gormley et al. (2015) commented that training on a small dataset is not appropriate because of the ease with which extremely high accuracies can be achieved. This provided a good justification for including data augmentation (SMOTE) and exploring transformer-based models that do not rely on pattern-matching techniques (Clark et al., 2019; Vaswani et al., 2017).

4.2. Topological modelling

It was noted that NN Learning Rates for the _trf model did not need to be adjusted and that the default settings achieved extremely good results on this dataset. Minimal accuracies gained from fine-tuning classifier hyperparameters generated exponentially large processing time-load costs to the system. This resulted in a design decision to train without hypermeters (Peters et al., 2019) and use untuned sklearn classifiers.

This research identified model variance when testing entity recognition techniques. The smaller (_md) statistical spaCy model could not leverage inbuilt topological functionality to successfully categorize or characterize the SMS data. Comparisons were only made between _lg and _trf implementations, as they produced similar results in most scenarios. The _lg and _trf models were generated on a CPU with excellent processing times (all <5 mins) and implemented as lightweight systems. These strategic implementation decisions ensured that model deployment and output was both accurate and efficient.

An iterative approach to acceptance testing was used throughout this work to support the evaluation of topological embeddings.

SpaCy was used in an evaluation capacity to generate optimal lexical results with a transparent pipeline technology (Spring & Johnson, 2022). SpaCy pipeline processing enabled a better understanding of the transformer output than typical 'black box' models (Honnibal & Montani, 2019).

RF was generally identified by other research as a superior modelling choice; however, we found that F1-scores were highest

for the SVC models. The 1.0 precision achieved on RF (see Table 3) appears anomalous. Results from RF _lg models demonstrate extreme variability.

A topic/word-modelling approach was not implemented in this research, diverging from the current consensus approach for SMS data analysis. Correct ner recognition and dependency parsing could only be generated on the _trf and _lg models using tf_IDF vectorizers.

SpaCy visualizations also confirmed the superior processing capabilities of the _trf sentencizer model over the _lg model. This choice of base architecture resulted in improved predictive accuracy (Table 3).

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

Tables 1, 2 and 4 are illustrative of results generated by employing non-optimal configuration designs.

The SVC model with SMOTE augmentation method application produced optimal results. The optimal model was evaluated using the accuracy, precision, recall and F1 metrics with predictions generated by standard sklearn classifiers. Due to effectiveness of the transformer (tensor) topological pre-processing methods on the data, models were generated efficiently on a CPU. F1 was used as a comparison metric over accuracy due to the class imbalance and high cost of misclassification when predicting spam (Statology, 2021).

5. Discussion of Key Findings

This work has identified an optimal classification model for short-text SMS data. The model achieves an SVC F1-score of 0.938 and consistently low processing times. This solution is user-extensible and interpretable, due to transparent implementation methods. Optimal topological rendering is achieved with sentence encoders inherent within the spaCy models. spaCy leverages pipeline methods to imbed dense vectors/tensors into a topological architecture and successfully fits the tested SMS modelling data. A spaCy pipeline with oversampling achieved the reported F1-scores on the SMS data and the evaluation dataset.

Short-text modelling methods do not appear to have been investigated or tested to a sufficient extent. This study demonstrates that the application of topological sentence transformer methods is an optimal design choice for analysing SMS data. Approximation-aware fast dependency parsing enables topological transformers to achieve high accuracies. Applying spaCy transformers enables edge processing to resolve as approximate (Gormley et al., 2015). This research has verified that runtimes can consistently present as $O(n^3)$ (Gormley et al., 2015; Honnibal & Johnson, 2014), even with CPU use. Our research verified that transformer application extensions must be correctly implemented to leverage optimal runtime complexities on a CPU. The incorrect use of these methods has a devastating effect on runtime performance.

Untuned statistical spaCy transformer models achieved an excellent F1-score compared to _lg non-transformer methods. Fitting SMS data on embedded topological clustering optimized the SVC classifier and leveraged inherent topologies from dependency parsing methods (de Kok & Hinrichs, 2016). High-rank data renderings and the implementation of SoftMax extension applications (Yang et al., 2017) enabled access to previously unutilized topological data expressions. Previous research examining the effects of tuning on sequential inductive transfer learning (Peters et al., 2019) appears to be relevant for short-text and large-text NLP tasks. We have tested our model from a perspective of minimal user intervention, at both the transformer level and the downstream classifier level. The SMS dataset is comprised of short-sentence components and the application of untuned pipelines to classification modelling is, to the best of our knowledge, a new approach.

Dependency parsing is incorporated as a native constituent of the spaCy pipelines and used to mine similar paraphrases within sentences. Pipeline processing provides a fundamental approach for utilizing spaCy models. The study implements a pipeline design to ensure extensible production functionality. Sklearn pipeline modelling mirrored the spaCy architecture and enabled seamless generation of F1-scores. The inherent sentence transformer topology was illustrated via POS tagging and entity recognition visualizations. Classification of semantically similar sentences initiated contextual 'learning' within the pre-trained model and enabled the identification of latent relationships (Gormley et al., 2015). SpaCy open-source technologies ensured access to optimal processing methods and enabled nuanced learning of short-message data.

Effective topological renderings of tensors (Moliner et al., 2020) were achieved by implementing concise sentence dependency parsing methods. Correct system design choices effectively captured semantically similar latent expressions embedded within SMS messages. RoBERTa is subject to bias as it uses BERT pre-trained models, sourced from internet-scraped data (Hugging Face, n.d.; Jain, 2022). Precision and recall can be used to inform bias and influence iterative reprocessing adjustments, via transparent modelling (Bartička et al., 2022). Prioritizing precision outcomes on imbalanced SMS data

supports minority class predictions (Brownlee, 2020). SMS data are extremely variable and interrogating topological transformer methodologies via adversarial sampling, precision and F1-scoring could improve algorithmic fairness (Zhang et al., 2018). Contemporary work on unbiased transformer modelling prioritizes transparent, user-centric evaluation methods (Modarressi et al., 2022).

This work uses open-source methods to generate a lightweight, user-extensible, state-of-the-art solution to the SMS spam detection problem. SpaCy was chosen for implementation tasks because it proved straightforward to process and could be integrated with sklearn methods. Inherent implementation risks of open-source technological reliance include the inability of a system to respond to specific SMS data requirements, without significant model deconstruction. SoftMax developments have not been fully realized for discrete data and work on Bernoulli latent variables requires monitoring (Yang et al., 2017). System modifications may be required to effectively process evolved versions of SMS text spam.

6. Conclusion

The results from this study demonstrate that modern NLP processing methods are suitable for use with SMS data. The tested models produced a variety of results for the SMS and Twitter datasets, based on a combination of classifiers and sampling techniques. The work identified that spaCy sentence transformers and sklearn pipeline implementations generated a maximum F1-score of 0.938, optimally utilizing topological data. The modelling can be considered optimal because a lightweight, transparent, user-extensible architecture was leveraged to produce excellent F1-scores. These attributes were considered appropriate evaluation mechanisms to objectively assess production suitability. The research demonstrated that SMS text-based datasets of short sentences could be treated as documents and optimally classified. SpaCy is a constantly evolving product and this study presents a design approach requiring minimal end-user intervention.

6.1. Implications for further research

There are various avenues to use this work as a baseline for future research, including adversarial data augmentation strategies (Shirazi et al., 2023). Adversarial sampling incorporates synthetic data generation and could benefit projects working with dangerous/hard-to-retrieve data.

The opaque/'black box' nature of NNs does not generally afford users the opportunity to investigate statistical models and assess efficacy or bias. Textual meanings are interpretable and a proven ability to demonstrate generated inferences is of paramount importance to a system (Spring & Johnson, 2022). Speech is notoriously difficult to categorize, and assessment of bias must be an ongoing maintenance task in a production environment. These mitigations do not guarantee a model free from bias but do allow user transparency (May et al., 2019; Sartran et al., 2022).

Dense embedding is used by RoBERTa, but Multiple Negative Ranking (MNR) loss sentence embedding has surpassed recorded accuracies since 2019. MNR manipulates the cosine similarity metric by comparing opposing vectors (Nguyen et al., 2022). Spam datasets could be tested using this technique once the open-source implementation version becomes available.

Additional hyperparameter tuning was not enacted on classifiers, and no complementary sampling strategies were applied to the Twitter dataset, excluding the inbuilt _trf generation sampling. Application of additional sampling could improve F1-scores (Peters et al., 2019) for Twitter data. SpaCy is RoBERTa-based and provides scope to adjust parameters of pre-trained weights in the pre-processing stage, circumventing sklearn hyperparameter tuning. Intra-model hyperparameter tuning degrades the performance of BERT, requiring careful manipulation, but could be achieved with new attribution techniques (Modarressi et. al., 2022; Xu et al., 2023).

Acknowledgement

This work has been supported by the Computer Science Department of the University of Adelaide.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in [Kaggle] at https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset and https://www.kaggle.com/competitions/utkmls-twitter-spam-detection-competition/data

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., Stanojević, 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

How to Cite: Milner, H. & Baron, M. (2024). Establishing an Optimal Online Phishing Detection Method: Evaluating Topological NLP Transformers on Text Message Data. *Journal of Data Science and Intelligent Systems* 2(1), 173–181, https://doi.org/10.47852/bonviewJDSIS32021131