The Application of Pre-trained Transformers for Deceptive Review Detection

Martijn Laurent Daemen
STUDENT NUMBER: u1275106

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:
Assist. Prof. Dr. Yash Satsangi
Assoc. Prof. Dr. Emmanuel Keuleers

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
Jan 2021

Preface

This thesis 'The Application of Pre-trained Transformers for Deceptive Review Detection' concludes my master program Data Science and Society with specialization Business at Tilburg University. The thesis was written during September 2020 and January 2021.

This will not conclude my academic career yet. Next semester I will do an additional thesis for my second master Information Management. I hope to extend my research on transformers there as transformers generously sparked my enthusiasm.

By doing this research I realized that the introduction of transformers to the field of natural language processing has been revolutionary. The amount of context transformers can grasp and the performances these models can achieve is astonishing. As these models are fairly new, and better transformers are already in development, one can only wonder what the future might bring.

I would like to express my gratitude to a few people that helped me during the process of writing this thesis. First and foremost, I would like to thank my supervisor Yash Satsangi for his feedback while writing this thesis. His comments and constructive meetings helped me improve this thesis greatly. Additionally, I would like to thank my partner and parents who supported me whenever I was stuck in the writing process.

Abstract

Due to the rise of e-commerce, consumers increasingly consult online reviews before making a purchase. Driven by profit, opinion spammers write deceptive reviews to boost or demote products or services. Therefore, both researchers and businesses have been eager to detect these deceptive reviews and protect consumers. Existing studies mainly focus on using context-independent language representation models like Word2Vec and GloVe in combination with neural networks. However, these methods only partially address the overarching problem of scarcity of labelled data and the disability to grasp the contextual information of the review. In this study, we empirically explore the possibility of using pre-trained transformers for language representation for the detection of online opinion spam. Various transformer-based language representation models, precisely XLNet, RoBERTa and BERT, are used for sequence representation. These language representation models are empirically tested in combination with Logistic Regression, BiLSTM and BiLSTM with attention mechanism as classification models. Further, this study proposes a new approach for the detection of online review spam called RXB-Ensemble. RXB-Ensemble (RoBERTA, XlNet, BERT, Ensemble) is an ensemble model of the three best performing models (BERT-BILSTM-ATT, XLNET-BiLSTM-ATT, RoBERTa-BiLSTM-ATT) in this study. The results of RXB-Ensemble on the Ott Corpus and the deception dataset demonstrated that the proposed model, RXB-Ensemble, generates state-of-the-art performance for the detection of opinion spam.

Keywords: Opinion spam, Transformer, BERT, XLNet, RoBERTa, Combination model, Ensemble learning, Online review

1. Introduction

As e-commerce, review sites, and social media platforms continue to expand (Buettner, 2016), along with it comes an increasing amount of online user reviews on services and products (Mudambi & Schuff, 2010; Valent, 2015). According to a public survey, 95% of online customers read reviews before purchasing a product and 92% trust online reviews as much as peer recommendations (McCabe, 2020). Sadly, strict inspection mechanisms are usually not present, resulting in many fake and deceptive reviews being read by consumers.

Opinion spammers use deceptive reviews to alter a product or service's image for financial gains (Jindal & Liu, 2008). Deceptive reviews, also known as opinion spam, is defined as the use of illegitimate means to boost the target's rank positions (Jindal & Liu, 2008). The publisher of the fake review, defined as an opinion spammer, tries to influence the consumer's buying behaviour for financial gains. The opinion spammer can either publish negative reviews on competing products, websites, or services that can negatively impact the image and, thus, put competitors lower in rank (Jindal & Liu, 2008). Or it can boost its image and rank by manufacturing misleading positive reviews of its product or service as was recently the case for amazon third-party sellers (Jindal & Liu, 2008; Daen, 2020). Because humans can only distinguish poorly between deceptive and genuine reviews (Ott et al., 2011), is it possible for the opinion spammer to influence the consumer for financial gains (Jindal & Liu, 2008). This makes opinion spam detection an important and challenging problem.

Existing research for deceptive review detection has tried a variety of solutions for this problem. Deceptive review detection is a binary classification problem in which the model tries to classify spam and non-spam reviews. Traditional approaches select predefined features in combination with SVM, Naive Bayes, or Logistic regression (Ott, et al., 2011; Jindal & Liu, 2008). Others address the classification problem with neural networks to overcome the selection of pre-defined features and use the full review as input (Ren & Ji, 2017; Zhang et al., & Wang, 2018).

The main shortcoming of existing research is that it does not take the surrounding contextual information of the review into account (Zhang et al., 2018). Ott et al. (2011) state that a person can successfully write a misleading review that cannot merely be differentiated on word occurrences. An example of contextual information is that the meaning of the word 'rock' differs in 'rock concert' and 'throw a rock'. Additionally, ground truth labelled data is scarce which further complicates the problem. This all creates the need for more advanced techniques which can detect contextual differences and mitigate the limited data availability.

Pre-trained transformers, e.g. BERT (Bidirectional Encoder Representations from Transformers), reduce this problem and generate a significant increment in performance for text classification tasks (Devlin et al., 2019). A transformer is an encoder-decoder model architecture that uses self-attention for the

representation of a sentence (Vaswani et al., 2017). It can map the dependencies between all words in a sentence. BERT is a language model that uses the encoder of the transformer in its architecture for sequence representations. (Devlin et al., 2019). It is pre-trained with immense amounts of English language data. This results in BERT being able to detect subtle differences in language because it identifies the dependency between words in a sentence based on the context (Devlin & Chang, 2018). For example, BERT can represent the contextual dependencies of the word 'bank' in the sentences 'Let's get some cash at the bank.' and 'Look at that beautiful bank of the river.'. This makes BERT perform well on tasks that are contextually dependent such as the classification of opinion spam (Barsever et al, 2020). Other transformers, such as XLNET and RoBERTa, all work on a somewhat similar derived architecture or are pre-trained with different datasets but vary in their performance (Yang et al., 2019; Lui et al., 2019).

The main objective of this research is to study the effectiveness of applying pre-trained transformers to identify online opinion spam. Prior research often uses Word2Vec or GloVE for sequence representations (Wang, 2017; Lozano & Fernquist, 2018). Word2Vec and GloVe differ from transformer-based language models because they are context-independent sequence representation models (Huang et al, 2020). Transformer-based language models, compared to Word2Vec and GloVe, can represent richer contextual information. (Huang et al, 2020). Barsever et al.(2020) has introduced BERT embedding for opinion spam detection and has managed to improve accuracy on a benchmark dataset. This study proceeds by empirically researching various combinations of transformer-based language models and classification models. First, language models BERT, RoBERTa and XLNet are combined with Logistic Regression, BiLSTM (Bidirectional Long Short-Term Memory), and BiLSTM with attention mechanism in order to identify the most ideal set-up for opinion spam classification. Second, no studies have attempted to ensemble multiple transformers for opinion spam detection. Recently, studies have concluded that ensembling transformer-based classifiers enhance performance for various text classification tasks (Chang et al., 2020, Lee et al., 2019). By ensembling our three best classification models we are able to contribute to this underdeveloped research field. Combined, this study explores the possibility of transformers for opinion spam detection by applying pre-trained language models BERT, XLNet and RoBERTa and pairing them with a variety of classification models. Additionally, we explore the performance impact of ensembling our best models. Therefore, the study addresses the following research question:

"*How can we use pre-trained transformers for opinion spam detection?*"

To answer the main research question the following sub-research questions are to be answered:

"How can BERT, RoBERTa or XLNet be applied for opinion spam detection?"

"To what extent does ensembling transformer-based classifiers enhance the performance for opinion spam detection?"

The contribution of this study can be summarized as follows. First, we describe a method for combining pre-trained language models with Logistic Regression, BiLSTM and BiLSTM with attention mechanism to detect fake reviews. Secondly, extensive experiments are conducted to demonstrate the performance of the variety of models on the Ott corpus and the best models are further examined on the deception dataset. Third, this study proposes a new method, RXB-Ensemble *(RoBERTA, XLNet, BERT - Ensemble)*, for opinion spam detection in which three classifiers (BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT, XLNet-BiLSTM-ATT) are ensembled using a majority voting strategy.

The remainder of this study is structured as follows. Section 2 discusses related work on online opinion spam detection, transformers and ensemble learning. Section 3 describes the experimental set-up and the architecture of the proposed RXB-Ensemble model. Thereafter, section 4 will present the results. Finally, Section 5 discusses the results and section 6 concludes by way of providing a final summary of this study.

2. Related work

2.1 Online opinion spam detection

Opinion spam detection is a long-standing topic of interest in email and web-page domains. Due to the rise of e-commerce, research on opinion spam detection shifted to the customer review domain. Various approaches have been researched in order to detect online opinion spam. Research has been done on exploiting features outside the review content. Mukherjee et al. (2013) analyzed customer-specific reviewing behaviour features, e.g. extreme rating, number of reviews and duplication of the review, with unsupervised Bayesian inference frameworks to detect opinion spam. Lim et al. (2010) suggest rule-based pattern mining for discovering unusual data patterns related to a user's brand and rating distribution to detect opinion spammers. Li et al. (2014) propose PU-learning, a graph-based approach, with user ID, IP address and review as features to spot fraudulent reviews. This study focuses on the exploitation of the content of the review.

Researchers have proposed a wide variety of strategies to detect review opinion spam based on the content of the review. Ott et al. (2011) propose n-gram based categorization of text in combination with

Naive Bayes and Support Vector Machine (SVM) classifiers. Feng et al. (2012) has implemented context-free grammar parse trees and part-of-speech (POS) tags for the detection of fake reviews. Mukherjee et al. (2013) have trained SVM classifiers on linguistic features (n-grams, POS) in combination with behavioural features.

These methods stumbled upon the problem of limited availability of labelled data and the disability of spotting context for classification of review spam. Only limited labelled data is available for online opinion spam detection because manually labelling data is time-consuming, costly, and can be very subjective (Li et al., 2018). This makes leveraging vast amounts of data, not an option. Besides limited labelled data, there is a lacking ability to detect the context of the review (Zhang et al., 2018). For the classification of opinion spam the contextual information of the review is more decisive than word occurrences (Ott et al., 2011).

More recently, neural networks are used to better grasp the contextual information in the review and deal with the scarcity of labelled data. Ren and Ji (2017) propose the application of gated recurrent neural networks that managed to identify the contextual information of the sentences in the review. Further, they managed to partially mitigate the scarcity of labelled data by training a continuous bag-of-words model with a large Amazon review corpus. Zhang et al. (2018) leverage a Skip-gram model in combination with a recurrent convolutional neural network to learn the word-level contextual information of the words in the deceptive reviews. Generative Adversarial Networks with recurrent neural networks as classifier and discriminator are proposed to overcome the problem of limited available training data by generating new reviews with reasonable perplexity (Stanton & Irissappane, 2019).

Due to limited ground-truth labelled online opinion spam datasets, it is challenging to fully exploit the potential of neural networks for opinion spam detection. Recently, a major step forward is set by introducing a pre-trained transformer language representation model, such as BERT, to opinion spam detection. Barsever et al. (2020) has managed to train a classifier using BERT (Bidirectional Encoder Representations from Transformers) for word embedding that generates state-of-the-art performance for online opinion spam detection. The proposed model RXB-Ensemble in this study builds forth on this model by introducing better pre-trained language models and leveraging multiple models in an ensemble architecture.

2.2 Pre-trained language models for language representation

Generally speaking text classification with neural networks can be split up into an upstream and downstream part of the model. The upstream part is defined as the way the sequence of words will be represented to the downstream part of the model (Huan et al., 2020). The downstream part is the model that classifies the represented sequence of text. This terminology will be further used in this study.

2.2.1 Language representation models

Natural language processing (NLP) tasks all start by converting a sentence of words in a way that it is compatible to process with a machine learning model. In general, it is the transformation of a sequence of words and punctuation into digits which a machine learning model can use to learn. Traditionally language representation models often implied the transformation of a sequence of words into a digital vector (Pennington et al., 2014). For creating these digital vector representations often one-hot encoding, tokenization techniques using vocabulary or embedded as neural words in which they are matched with inserts of fixed length are used. Most popular techniques include GloVe and Word2Vec (Pennington et al., 2014). The main disadvantage of these models, such as Word2Vec, is that the representation of a word is independent of the context it appears in (Huang et al, 2020).

Recently, state-of-the-art pre-trained transformer-based language representation models demonstrated the ability to represent the context of a sequence of words. (Devlin et al., 2019). A transformer is a model architecture, consisting of an encoder and decoder, that uses self-attention for the representation of a sequence (Vaswani et al., 2017). The function of the self-attention mechanism is to focus on one part of the input in order to evaluate the effects on the output sequence (Vaswani et al., 2017). This self-attention mechanism allows the transformer model to compute dependency relationships between all words in a sequence (Vaswani et al., 2017). The transformer's encoder is the base for most state-of-the-art transformer-based language representation models such as BERT (Devlin et al., 2019).

Bidirectional Encoder Representations from Transformers (**BERT**) is a language representation model that superimposes multiple layers of transformer encoders (Devlin et al., 2019). BERT is trained on word prediction and next sentence prediction (NSP),. BERT is learned by randomly masking 15% of the tokens in a sequence and predicts these masked tokens with a softmax layer over the last encoder. For the second task, NSP, BERT predicts whether or not sentence B follows sentence A.

The input sequence is converted into tokens and a positional embedding (Devlin et al., 2019). Two special tokens, [CLS] and [SEP], are respectively added to the start and end of a sequence. The [CLS] token is used for text classification tasks, including NSP (Devlin et al., 2019). The [SEP] token is a separator token used for separating, for example, questions from answers or two sentences for NSP tasks.

BERT comes in two model sizes: a base size (L=12, H=768, A=12, Total Parameters = 110M) and large size (L=24, H=1024, A=16, Total Parameters=340M)), where L indicates the number of encoder layers (e.g. Transformer encoder blocks), H the hidden units and A the amount of multi-head attention heads (Devlin et al., 2019). Both models have been trained on English Wikipedia and BookCorpus (Zhu et al., 2015) datasets, which have a combined size of 16GB.

RoBERTa (Robustly optimized BERT approach) has a similar architecture to BERT but is trained on more language data and with a dynamic masking technique (Lui et al, 2019). Additionally, it is trained

on longer sequences. RoBERTa beats BERT with performances up to 2-20% on the GLUE benchmark (Lui et al, 2019). The GLUE benchmark consists of 8 text classification tasks, such as whether or not a sentence is grammatically correct or binary sentiment prediction, and one semantic textual similarity regression task (Wang et al., 2018). Similar to BERT, RoBERTa uses a [CLS] token for classification and [SEP] for separation.

RoBERTa comes in two model sizes: A base size (L=12, H=768, A=12, Total Parameters = 110M) and a large size (L=24, H=1024, A=16, Total Parameters = 340M) (Lui et al, 2019). Both models are trained on BookCorpus (Zhu et al., 2015), English Wikipedia, CC-News (Nagel, 2016), OpenWebText (Gokaslan and Cohen, 2019) and Stories datasets (Trinh and Le, 2018), which have a combined size of 161GB.

XLNet is an autoregressive language model based on BERT that solves the problem of simultaneously generated predictions by BERT (Yang et al., 2019). BERT learns by predicting masked words simultaneously (Devlin et al., 2019). By predicting words simultaneously the dependencies between these predictions are not learned (Devlin et al., 2019). XLNet has overcome this by introducing a permutational language model while keeping the bidirectionality of BERT (Yang et al., 2019). It learns to predict words by using all possible permutations of the words in a sequence. XLNet, thus, learns in a random order but does so in a sequential and autoregressive way. By doing so it structurally outperformance BERT on the GLUE benchmark, often by 2-13% (Yang et al., 2019).

XLNet also comes in two sizes, XLNet base (L = 12, H=768, A=12, Total Parameters = 110M) and XLNet large size (L=24, H=1024, A=16, Total Parameters=340M) (Yang et al., 2019). The base model is trained on BookCorpus (Zhu et al, 2015) and English Wikipedia dataset (16GB). The larger model adds to that Giga5 (Parker et al, 2011), ClueWeb (Callan et al., 2009) and Commoncrawl datasets totalling about 113GB of training data. For XLNet similar tokens, [CLS] and [SEP], are used for classification and separation respectively (Yang et al, 2019).

2.2.2 Downstream classification models

As language models merely output a sequence representation for classification, a variety of downstream classification models can be combined. The proposed model consists of a transformer as the upstream part and BiLSTM with attention mechanism as the downstream part of the model. Therefore, the underlying mechanisms are briefly discussed.

2.2.2.1 Bidirectional LSTM

Long short-term memory (LSTM) is a type of recurrent neural network that solves the problem of the vanishing gradient and is well equipped for sequential data (Hochreiter, 1998). LSTM can model long-term dependencies by using gates for forgetting irrelevant information of the previous state, storing new

information and outputting relevant information to the next time step (Hochreiter & Schmidhuber, 1997). LSTM is one of the main model architectures used for modelling natural language problems, sound problems, or stock data.

Bidirectional long short-term memory (BiLSTM) extends on the traditional LSTM. It consists of a forward LSTM and a backward LSTM (Schuster & Paliwal, 1997). The backwards LSTM uses the input sequence in the opposite direction. Due to the bidirectionality, it can reap the benefits of both past and future features (Schuster & Paliwal, 1997). This makes BiLSTMs highly interesting for NLP tasks as from both information of previous and future words can be benefitted (Cai et al., 2020).

BiLSTM is often used in NLP classification tasks. Huang et al. (2018) has applied BiLSTM on topic information for binary sentiment classification. Xu et al. (2019) used BiLSTM to effectively capture more semantic information for sentiment classification of Chinese hotel reviews. Recently, Cai et al. (2020) has paired a BiLSTM with BERT embedding for binary sentiment classification of the Chinese energy market. This combination of BERT with BiLSTM outperformed the more traditional BiLSTM with Word2Vec model for sentiment classification of the Chinese energy market. (Cai et al, 2020).

2.2.2.2 Bidirectional LSTM with attention mechanism

Bidirectional LSTM with attention mechanism (BiLSTM-ATT) is a relatively new model for natural language processing. It is a downstream classification model in which the BiLSTM layer is followed by a self-attention layer (Zhou et al., 2018). The function of the self-attention mechanism is to capture different influences of the output of the BiLSTM (Zhou et al., 2018). Due to this attention layer, it often manages to increase the performance of the model compared to traditional BiLSTM models.

Zhou et al. (2018) used BiLSTM-ATT in combination with Word2Vec for classification of sentiment with positive results. BiLSTM-ATT in combination with pre-trained Twitter Word2Vec is effectively applied for emotion classification. Liu and Guo (2019) have enhanced this architecture by adding a convolution layer and matching this downstream classification model with the Skip-gram model of Word2Vec. Their method increased both the semantic understanding and the accuracy for a variety of sentiment analysis tasks (Liu & Guo, 2019).

Recently, highly competitive models use BERT representation in combination with BiLSTM with attention mechanism for competitive results. Lee et al. (2019) use a BiLSTM with a word attention mechanism to capture distinguishing word influences from the BERT sequence representation. Their model achieved third position in a medical text classification challenge (Lee et al., 2019). Baserver et al. (2020) has implemented BERT with BiLSTM with attention for the detection of opinion spam and pushes the performance to a higher level.

2.3 Ensemble learning in natural language processing

Ensemble learning has broadly been researched, primarily with classic machine learning algorithms like Logistic Regression and Decision Trees (Khurshid et al., 2019). In ensemble learning, multiple models are created in order to jointly predict with use of an ensemble strategy (Dietterich, 2000). Ensemble learning has three main distinctive strategies: bagging, boosting and aggregation of classification output. Boosting and bagging are both data sampling and selection techniques in order to ensure diversity in the classifier models (Zhang & Ma, 2012). Aggregation of classification predictions is merely combining classification output of multiple models by assigning a final prediction based on the majority of the class predictions (majority vote) or averaging prediction probabilities (Dietterich, 2000).

Ensemble learning can be highly effective when data is scarce and reduce the impact of a model not finding the global optimum (Dietterich, 2000). When training data is small models can give multiple optimums for the same accuracy. This risk can be averaged out by letting multiple classifiers vote for the final prediction (Dietterich, 2000). Further, neural networks can get stuck in local optima. By ensembling multiple classifiers models this risk can be mitigated (Dietterich, 2000). This can be highly effective for opinion spam detection as training data is limited.

Pre-trained transformer language models, such as BERT, are all different in model architecture, training data and training procedure. This could mean that errors of the pre-trained language models are uncorrelated. Therefore, ensembling might positively impact the performance for text classification as different transformer-based classifiers with the same accuracy might predict different instances correctly. Lee et al. (2019) ensemble BERT and BioBert, a domain-specific BERT, using a simple majority voting strategy for a performance increase of 3% on its classifier components. Multiple BERT models are used with bagging techniques to effectively reduce variance for small data sets (Risch & Kestrel, 2020). Chang et al. (2020) combine XLNet, RoBERTa and BERT in an ensemble model that generates state-of-the-art results for extreme multi-label text classification.

This study proposes an approach, RXB-ensemble, using three transformers, BERT, XLNet and RoBERTa, in combination with BiLSTM with attention mechanism (BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT, XLNet-BiLSTM-ATT). The three classification components are ensembled with use of a majority voting strategy. RXB-Ensemble generates state-of-the-art results for opinion spam detection. Before moving on to the method section, table 1 presents an overview of model names and description further used in this study.

Table 1

Names and description of the models used in this study.

Name	Description
BERT-Naive Bayes	BERT transformer as language presentation and Naive Bayes as classification model
RoBERTa-Naive Bayes	RoBERTa transformer as language presentation and Naive Bayes as classification model
XLNet-Naive Bayes	XLNet transformer as language presentation and Naive Bayes as classification model
BERT-Log. Reg.	BERT transformer as language presentation and Logistic Regression as classification model
RoBERTa-Log. Reg.	RoBERTa transformer as language presentation and Logistic Regression as classification model
XLNet-Log. Reg.	XLNet transformer as language presentation and Logistic Regression as classification model
BERT-BiLSTM	BERT transformer as language presentation and Bidirectional LSTM as classification model
RoBERTa-BiLSTM	RoBERTa transformer as language presentation and Bidirectional LSTM as classification model
XLNet-BiLSTM	XLNettransformer as language presentation and Bidirectional LSTM as classification model
BERT-BiLSTM-ATT	BERT transformer as language presentation and Bidirectional LSTM as classification model
RoBERTa-BiLSTM-ATT	RoBERTa transformer as language presentation and Bidirectional LSTM as classification model
XLNet-BiLSTM-ATT	XLNet transformer as language presentation and Bidirectional LSTM as classification model
RXB-Ensemble	Majority vote ensemble model of BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT and XLNet-BiLSTM-ATT

3. Methods

This section explains the methodology and experimental set-up of this study. Section 3.1 and 3.2 explain the collection of datasets and the preprocessing steps used in this study. Thereafter, Section 3.3 describes the proposed method RXB-Ensemble. Section 3.4 discusses the baseline models. The next section (3.5) explains the experimental set-up and section 3.6 describes the evaluation metrics. Finally, section 3.7 lists the programming language and packages used in this study.

3.1 Data description

The Ott corpus consists of 800 truthful reviews extracted from Tripadvisor and 800 deceptive reviews written by Amazon Mechanical Turkers (Ott et al., 2011). This dataset is the golden-standard within the online opinion spam detection research field.

In addition to the Ott corpus, this study uses the deceptive review dataset, also known as deception dataset (Li et al, 2014). The deception dataset consists of 2,837 reviews across three domains: Hotel (1880), Restaurant (401) and Doctor (556). The dataset has a percentage of deceptive reviews of 57.70%. These deceptive reviews are written either by Amazon Mechanical Turkers or written by experts in the domain field (Li et al, 2014). Below table 2 displays the overview of the datasets.

Table 2

Overview of the number of reviews, the proportion of spam and domains in the datasets.

Dataset	Total number of reviews	Number of spam reviews	Domain
Ott Corpus	1,600	800 (50.00%)	Hotel
Deception dataset	2,837	1,637 (57.70%)	Hotel, Doctor, Restaurant

3.2 Preprocessing

Preprocessing is split into two steps. Preparing the data into the required structure for the transformer model and applying the appropriate truncation method.

3.2.1 Tokenization and attention mask

Transformer models require two inputs for the creation of a sequence representation: the tokenized sequence and an attention mask sequence. The following steps have to be taken to prepare the review for

the transformer model: tokenization, adding special tokens, padding sequence, and creating an attention mask. Figure 1 below presents the conceptual input design structures. Deeper explanation follows in the following paragraphs.

Figure 1

The conceptual structure of the tokenized sequence and attention mask sequence for the transformer models BERT, RoBERTa and XLNet.

Example = "This example sentence is simple. Both obviously will get the point across!?!"

Conceptional Structure

BERT	[<CLS>, <Tokenized sequence>, <SEP>, <PAD>] | Attention_mask = [1,1,1,0]
RoBERTa	[<CLS>, <Tokenized sequence>, <SEP>, <PAD>] | Attention_mask = [1,1,1,0]
XLNet	[<PAD>, <Tokenized sequence>, <SEP>, <CLS>] | Attention_mask = [0,1,1,1]

First, the review has to be tokenized using the transformer specific tokenizer. Every transformer model has its pre-trained vocabulary which it uses to tokenize a sequence to a list of digits. The tokenizer will split the review and tokenize the words and punctuation. Figure 2 shows how an example text is tokenized by the three transformer models applied in this study.

Figure 2

Example of a string of text tokenized by the language models BERT, RoBERTa and XLNet.

Example = "This example sentence is simple. Both obviously will get the point across!?!"

Tokenization

BERT	[2023, 2742, 6251, 2003, 3722, 1012, 2119, 5525, 2097, 2131, 1996, 2391, 2408, 999, 1029, 999]
RoBERTa	[713, 1246, 3645, 16, 2007, 4, 1868, 3334, 40, 120, 5, 477, 420, 328, 17516]
XLNet	[122, 717, 3833, 27, 1369, 9, 1668, 5046, 53, 133, 18, 424, 514, 136, 82, 136]

Secondly, two special tokens are added to the tokenized sentence. A [CLS] token is added for classification purposes. Further, a [SEP] token is added at the end of the review. The [SEP] token normally functions as a separation between question and answer or two sentences in the case of next sentence prediction (Devlin et al., 2019). Since the task at hand is merely a classification task of the complete review

and no separation between segments has to be made, we add the [SEP] token at the end of the review. BERT and RoBERTA add the [CLS] and [SEP] token to the beginning and the end of the tokenized review respectively. XLnet adds both the [SEP] and [CLS] at the end respectively. Below, figure 3 shows the differences in tokenization structure per transformer.

Figure 3

Example of adding the classification and separation token to the tokenized sequence for transformer models BERT, RoBERTa and XLNet. Green indicates the classification token and red the separation token.

Example = "This example sentence is simple. Both obviously will get the point across!?!"

Adding special tokens <CLS> and <SEP>

BERT	[101, 2023, 2742, 6251, 2003, 3722, 1012, 2119, 5525, 2097, 2131, 1996, 2391, 2408, 999, 1029, 999, 102]
RoBERTa	[0, 713, 1246, 3645, 16, 2007, 4, 1868, 3334, 40, 120, 5, 477, 420, 328, 17516, 2]
XLNet	[122, 717, 3833, 27, 1369, 9, 1668, 5046, 53, 133, 18, 424, 514, 136, 82, 136, 4, 3]

Thirdly, padding is added to the tokenized sequence shorter than the maximum sequence length in the dataset. For BERT and RoBERTa padding is added after the tokenized sequence while XLNet inserts padding before the tokenized sequence, as can be seen in figure 4. This step finalizes the process for transforming the review into a tokenized review that functions as the input of the transformer.

Figure 4

Example of adding padding to the tokenized sequence for transformers models BERT, RoBERTa and XLNet. Red indicates padded places in the sequence.

Example = "This example sentence is simple. Both obviously will get the point across!?!"

Padding to max sequence length <PAD>

BERT	[101, 2023, 2742, 6251, 2003, 3722, 1012, 2119, 5525, 2097, 2131, 1996, 2391, 2408, 999, 1029, 999, 102, 0, 0, 0]
RoBERTa	[0, 713, 1246, 3645, 16, 2007, 4, 1868, 3334, 40, 120, 5, 477, 420, 328, 17516, 2, 1, 1, 1]
XLNet	[5, 5, 5, 122, 717, 3833, 27, 1369, 9, 1668, 5046, 53, 133, 18, 424, 514, 136, 82, 136, 4, 3]

Finally, an attention mask sequence is created. The attention mask is merely a binary sequence that indicates where the content of the tokenized sequence is (Devlin et al., 2019). More specifically, it represents the padding as 0 and all other tokens including special tokens as 1. The function of the attention

mask sequence is that the transformer knows where the relevant information is in the tokenized sequence (Devlin et al., 2019). Thus, it focuses on using all digits in the tokenized review sequence and not the padding. Figure 5 presents an example of the attention masks corresponding to the tokenized sequences.

Figure 5

Example of attention mask sequences for transformers BERT, RoBERTa and XLNet.

Example = "This example sentence is simple. Both obviously will get the point across!?!“

Create Attention Mask

BERT	[1,1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1,1,0,0,0]
RoBERTa	[1,1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1,1,0,0,0]
XLNet	[0,0,0,1,1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1,1]

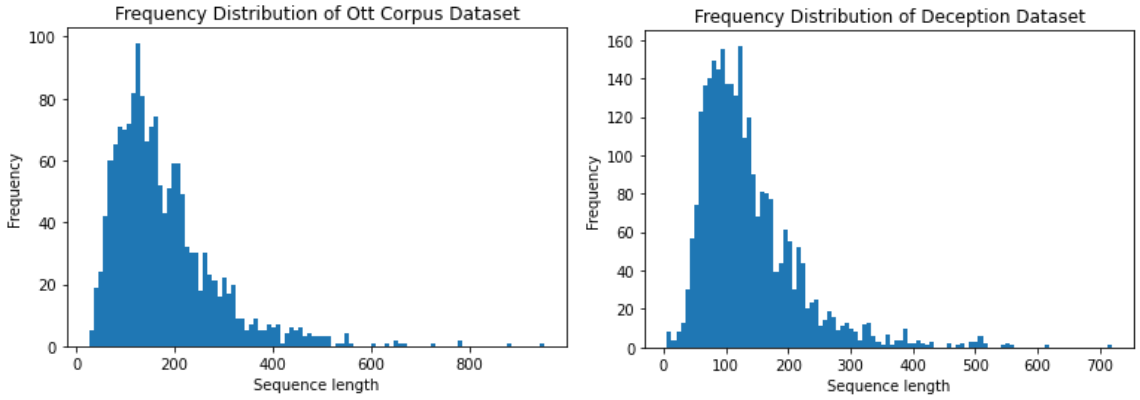
These steps result in two sequences for every review in the dataset. One tokenized review sequence where special tokens [CLS] and [SEP] are added and which is padded. And one attention mask sequence which functions as an indicator for where the non-padded information is in the tokenized review.

3.2.2 Truncation strategy

BERT, RoBERTa and XLNet are all restricted by a maximum sequence length of 512 tokens including special tokens. This means that a review can have a maximum length of 510 tokens (512 minus 2 special tokens). Figure 6 presents the distribution of the number of tokens in a review excluding special tokens for the Ott corpus and deception dataset. The Ott corpus’s average tokenized review length is 178.5 and the maximum is 950. Further, 21 tokenized reviews are longer than the maximum of 510. The deception dataset has an average tokenized length of 136.05, a maximum length of 719 and 10 reviews are longer than the acceptable maximum tokenized sequence length.

Figure 6

Histogram of tokenized review length of the Ott corpus (left) and the deception dataset (right).



Since some reviews are longer than the acceptable input length of the transformers they will be truncated to the acceptable input length. Sun et. al. (2019) conclude that performance can be reliant on the truncation method. In general, essential information of text is either at the beginning or the end. Therefore, the following truncation methods will be tested on the baseline models to quickly assess the impact on performance. Using the first 510 tokens (head), the last 510 tokens (tail) and the first 128 tokens and last 382 tokens (head + tail) of the review. This is merely done for the Ott corpus as the number of sentences above the maximum length for the deception dataset is negligible.

3.3 The proposed method - RXB-Ensemble

This section introduces the proposed ensemble model RXB-Ensemble. First, the neural network architectures of the three classification models (BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT, XLNet-BiLSTM-ATT) used in the ensemble model will be explained. Thereafter, this section concludes by explaining the ensemble strategy used in the proposed model.

The classification models used in the ensemble model are based on Barsever et al. (2020). This classification model consists of a BERT embedding layer, a BiLSTM layer, a self-attention layer and a linear layer (BERT-BiLSTM-ATT). It takes as input a review and outputs a binary class: spam or non-spam. This model, BERT-BiLSTM-ATT, will be used as one of the three classifiers in the ensemble model RXB-Ensemble. For the other two classifiers, the architecture is adapted by modifying the embedding layer with a RoBERTa embedding for one of the classifiers and XLNet for the other classifier. Below every component of the classification models will be explained. Starting with the application of transformers, followed by the downstream classification model its layers. A high-level overview of the classification model architecture is displayed in figure 8 in section 3.3.3.

3.3.1 Embedding layer

Three different transformer models are used in the proposed model for language representation. The specific pre-trained models used in this study are:

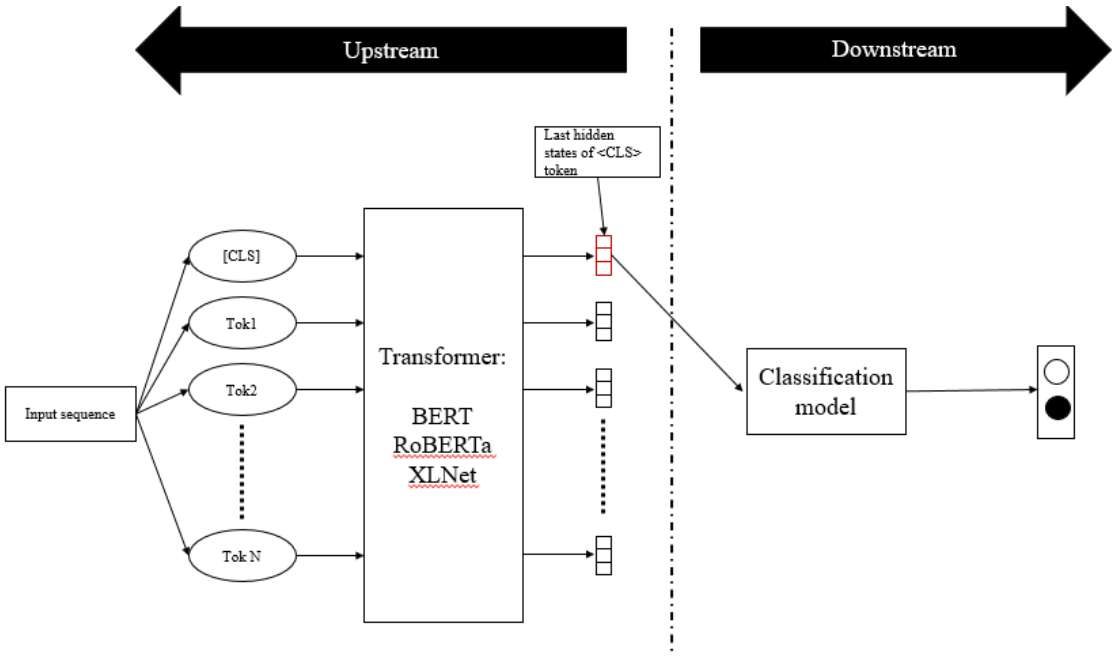
- BERT-Base-uncased : 12-layers, 768-hidden, 12-attention heads, 110M parameters
- RoBERTa-Base: 12-layers, 768-hidden, 12-attention heads, 125M parameters
- XLNet-Base-cased: 12-layers, 768-hidden, 12-attention heads, 110M parameters

To extract the contextual information of the review, each review will be embedded using a transformer. The transformer embedding layer takes the review as input and calculates a representation for every token of the input sequence. A review S_0 will be transformed into token-ids [x1, x2, …, xn] using the pre-trained token vocabulary of the transformer. Special tokens, a classification [CLS] and a separation [SEP] token will be added to the tokenized review which results in S_0 = ($Token_{[CLS]}$, $Token_{x1}$, $Token_{x2}$, …., $Token_{xn}$, $Token_{[SEP]}$). XLNet, RoBERTa and BERT all have a maximum token input size of 512 tokens including special tokens. Tokinized sequences longer than this length will be truncated.

The transformer model will output the hidden states of the final layer of the encoder stacks. The last layer of the transformer model is used for classification tasks as this results in the best model performance for text classification (Sun et al., 2019) Given a sequence S_0 as input, the transformer will output a vector 768 hidden states for every token in S_0. This results in multiple vectors of 768 hidden states equal to the length of tokenized S_0. For classification tasks, merely the hidden states of the [CLS] token is relevant. This vector of 768 hidden states is selected as input for the BiLSTM layer. Figure 7 displays the high-level usage of the transformer in combination with a downstream classification model.

Figure 7

Conceptual use of the transformer as an upstream part of the model. An input sequence will be transformed with the use of a transformer (BERT, RoBERTa or XLNet) to create a representation for every token. The representation of the [CLS] token will function as the input for the downstream classification model.



3.3.2 Bidirectional LSTM Layer and attention layer

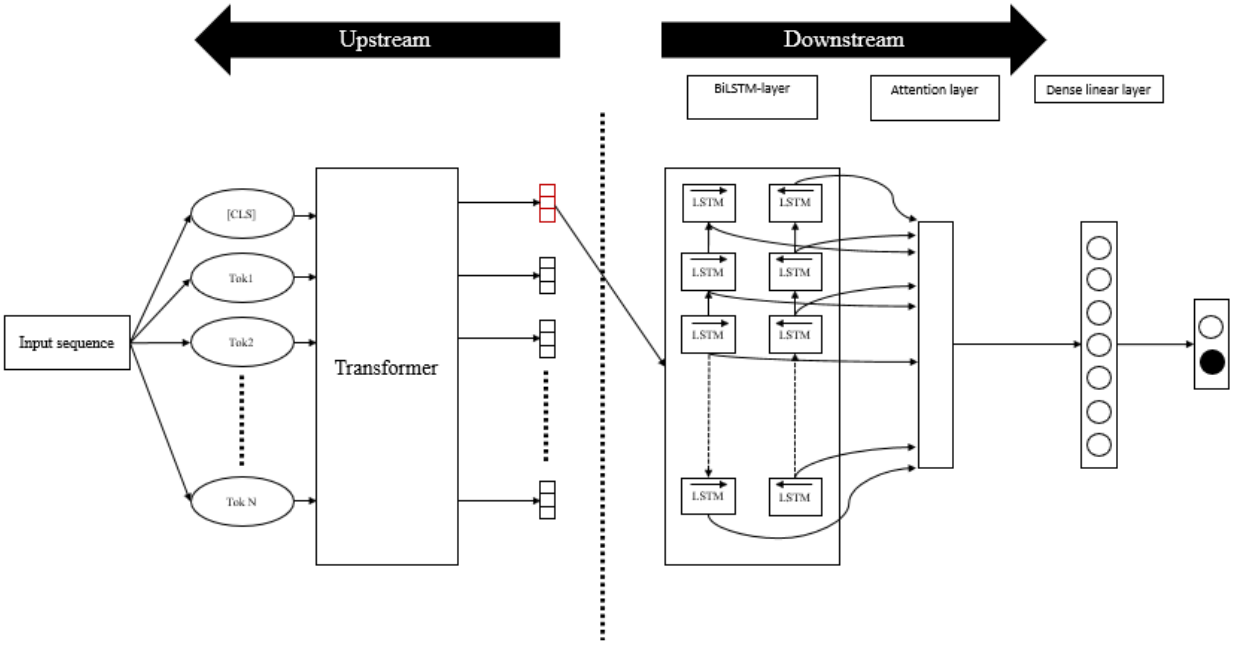
After the embedding layer, the hidden states of the [CLS] token will be fed into the BiLSTM layer. The BiLSTM layer has a hidden size 384 and a dropout of 0.5 is applied to prevent overfitting of the model. Barsever et al. (2020) follow the BiLSTM layer with a self-attention layer. The function of the self-attention mechanism is to identify distinguishing influences of the output of the BiLSTM (Zhou et al., 2018). The output of the self-attention layer, a dense vector, will be fed into a final dense layer.

3.3.3 Dense layer

The output of the self-attention layer is fed into the dense layer of 768 hidden neurons with linear activation. The linear layer outputs a vector with two values, one for every class (non-spam vs spam). The predicted class is determined by taking the maximum of the two values. Figure 8 presents the complete architecture of the classifier in the ensemble model.

Figure 8

The conceptual design of the classifier model. An input sequence will be transformed with the use of a transformer (BERT, RoBERTa or XLNet) to create a representation for every token. The representation of the [CLS] token will function as the input for the BiLTSTM layer. The BiLSTM-layer is followed by a self-attention layer and dense layer with linear action. Finally, the model outputs a binary class.



3.3.4 Model optimization

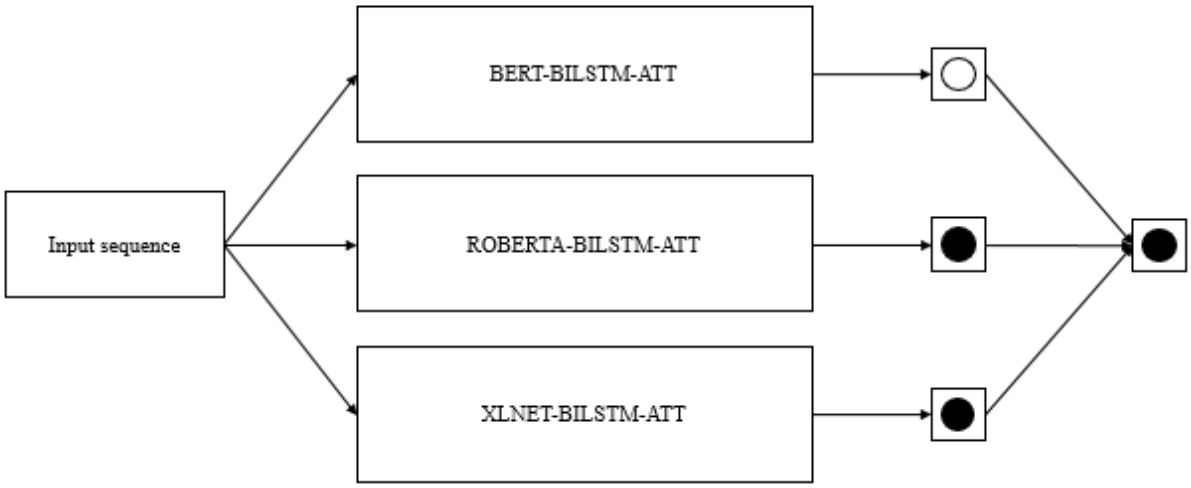
The model is optimized using Adam optimizer with a learning rate of 1e-6 and batch size of 5 (Kingma & Ba, 2017). Further, cross-entropy loss will be used as the loss function.

3.3.5 Ensemble strategy

The three classifiers (BERTt-BiLSTM-ATT, RoBERTa-BiLSTM-ATT and XLNet-BiLSTM-ATT) will be ensembled using a majority voting ensemble strategy. Majority voting is a simple but efficient ensembling strategy in which the final prediction is determined when more than half of the classification models predict a similar class (Dietterich, 2000). The three classifiers {*h1*, *h2*, *h3*} predict a new review *x*. If all classifiers predict identical classes, it will assign that class. E.g. *h1(x)* is spam, *h2(x)* is spam and *h3(x)* is spam the final classification will be spam. If no unanimous vote is registered the final prediction will be that of the majority. E.g. when *h1(x)* is spam *h2(x)* is spam and *h3(x)* is non-spam the final predicted class will be spam. A high-level overview of the ensemble strategy is presented in figure 9.

Figure 9

The input sequence, a review, will be fed into the three classification models (BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT, XLNet-BiLSTM-ATT). All three classifiers output a class prediction whereafter the majority of predictions define the final class.



3.4 Baseline models

The upstream transformers, BERT, RoBERTa, and XLNet will be paired up with three baseline downstream classification models. Below the models will be shortly discussed.

3.4.1 Naive Bayes and Logistic Regression

As downstream baseline classification models, this study uses Naive Bayes and Logistic Regression. Naive Bayes is a probabilistic classification model that uses Bayes theorem (Zheng & Webb, 2000). This algorithm is often applied as a baseline due to its simplistic implementation. It does not have any parameters and, thus, does not require any hyperparameter tuning. Further, this study applies Logistic Regression as another baseline downstream classification model.

Logistic Regression is a binary classification model that models a binary dependent variable using a logistic function (Kuhn & Johnson, 2013). It outputs a probability p belonging to a certain class. It uses the log odds function log(p/(1-p)) to create a certain classification output (Kuhn & Johnson, 2013). Hyperparameters of Logistic Regression can be tuned and this study grid searches for the optimal hyperparameters. The following parameters will be evaluated:

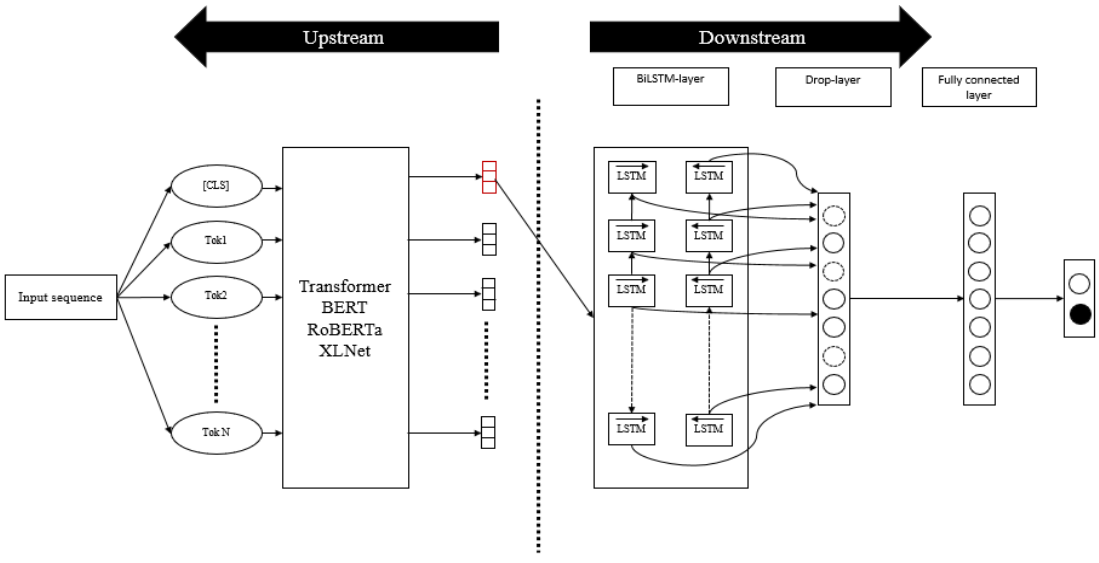
- Solver: [‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’]
- Penalty : [‘none’, ‘l1’, ‘l2’]
- C: [100, 10, 1.0, 0.1, 0.01]

3.4.2 Bidirectional LSTM

As the last baseline, a BiLSTM is used as a downstream classification model. The downstream classification model consists of a BiLSTM with 384 hidden units and 0.5 dropout, a fully connected layer of 768 units with relu as activation function and an output layer with sigmoid activation. Figure 10 displays the model design.

Figure 10

An input sequence will be transformed with the use of a transformer (BERT, RoBERTa or XLNet) to create a representation for every token. The representation of the [CLS] token will function as the input for the BiLTSTM layer with drop-out. Followed by a fully connected layer. Finally, the model outputs a binary class.



A variety of learning rates between 1e-3 and 5e-5 will be tested in combination with a batch size of 16 and 32. The model will be trained with Adam optimization (Kingma and Ba, 2017) and cross-entropy as a loss function. The following hyperparameters will be tuned:

- Batch size: [16,32]
- Learning rate: [1e-3, 5e-3, 1e-4, 5e-4, 1e-5, 5e-5]

3.5 Experimental set-up

The experiments are structured as follows. First, the three different truncation strategies will be evaluated on the Ott Corpus. Thereafter, the hyperparameter selection of the proposed model is discussed and how this study ensures the robustness of the results. Finally, we explain which models are further evaluated on the deception dataset. Across all experiments, a train-validation-test split of 70-10-20 is used.

To assess the impact of the truncation strategy on the performance we combine the transformers with the baseline models and test them on the Ott Corpus. The more complex models will not be included as this increases the computational time unnecessarily. The models are run multiple times on different train-test splits to minimize the impact of a random lucky split. The most beneficial truncation strategy will be applied in further experiments.

The component classifiers of the proposed method RXB-Ensemble are tested on a variety of learning rates in combination with a batch size of 5. The input sequence length is set to a maximum of 512 in order to feed the model the maximum amount of the review. Further, the architecture in terms of layers and the number of neurons is kept similar for comparability. On one train-validation-test split the performance of the models with learning rates [5e-5, 1e-5, 5e-6, 1e-6, 5e-7] is assessed. This learning rate will be used in further experiments.

After the selection of the proper learning rate, the models will be evaluated on the Ott Corpus. On this corpus, the models are run on multiple train-test splits. This is done to create statistically significant results since a random lucky train-test split has a high impact on the performance. All models were run at least 10 times.

After accessing the performance on the benchmark dataset the best models are run on the deception dataset. This entailed running the following four models: BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT, XLNet-BiLSTM-ATT and the proposed RXB-Ensemble model. To ensure statistically significant results the models are run multiple times.

3.6 Performance evaluation

For binary classification problems, there are many kinds of evaluation metrics. Choosing the right indicators based on the problem and existing research is important. This study adopts the following evaluation metrics commonly used in academic circles of opinion spam detection.

Accuracy is the most basic evaluation indicator for binary classification models. Although simple, it has a fatal flaw in cases where the dataset is unbalanced. Accuracy (Eq. 1) is calculated by summing the number correct prediction and dividing it by the total sample size.

$$Accuracy = \frac{n_{Correct}}{n_{total}} \quad (\text{Eq. } 1)$$

Further, the models are scored on precision and recall. Precision (Eq. 2) indicates the ratio of true positive predictions (spam) to the total positively predicted label. Recall (Eq. 3) is the ratio of true predicted positive labels to the total observations in the positive class.

$$Precision = \frac{True\ Positive}{True\ Positive\ +\ False\ Positive} \quad (\text{Eq. } 2)$$

$$Recall = \frac{True\ Positive}{True\ Positve\ +\ False\ Negative} \quad (\text{Eq. } 3)$$

To comprehensively evaluate the classification model, the F1-score is calculated. F1-score is the harmonic mean of precision and recall (Eq. 4). A higher score is preferred.

$$F1 - score = \frac{2\ *\ Precision\ *\ Recall}{Precision\ +\ Recall} \quad (Eq. 4)$$

3.7 Software

The experiments are programmed in Python 3.6.9 in Google Colab to leverage the computational power of the GPU. For data preprocessing and tokenization, the packages Numpy 1.18.5 and Huggingface 3.5.1 are used. Furthermore, the Huggingface 3.5.1 library is used to implement the pre-trained transformers BERT, RoBERTa and XLNet. Logistic regression and Naive Bayes are implemented by using Scikit-learn 0.23.2. For the neural network models, Pytorch 1.7.0 or Tensorflow 2.3.0 is used. Graphs are constructed with Matplotlib 3.2.2.

4. Results

This section presents the results of the experiments. First, the outcome of different truncation methods is displayed. Thereafter, the proper hyperparameter selection for the model components of RXB-Ensemble is discussed followed by the results of all the models on the Ott Corpus. Finally, the performance of the best models on the deception dataset will be discussed.

4.1 Truncation strategy

This study analysed three different truncation methods: using the head, the tail and a combination of the head and tail of the review. Logistic Regression, Naive Bayes and BiLSTM were combined with the three transformers of interest (BERT, RoBERTa, XLNet). Table 3 reports the mean accuracy and standard deviation for every truncation strategy for the Ott corpus.

Table 3

Mean accuracy and standard deviation of the models on the Ott corpus for three different truncation methods: head, tail, head and tail.

Model	Head	Tail	Head+tail	Number of runs
BERT-Log.Reg	**0.7888(0.0200)**	0.7759(0.0206)	0.7871(0.0166)	10
BERT-Naive Bayes	0.6798(0.0221)	0.6809(0.0244)	**0.6880(0.0314)**	30
BERT-BiLSTM	0.7811(0.0217)	**0.7841(0.0221)**	0.7811(0.235)	50
RoBERTa-Log.Reg	0.8650(0.0168)	0.8688(0.0143)	**0.8809(0.0129)**	10
RoBERTa-Naive Bayes	0.7820(0.0517)	0.7807(0.0528)	**0.7826(0.0517)**	30
RoBERTa -BiLSTM	0,8476(0,0228)	**0.8511(0.0213)**	0.8479(0.0226)	50
XLNet -Log.Reg	0.8241 (0.0191)	0.8271(0.0218)	**0.835(0.0218)**	10
XLNet -Naive Bayes	0.6760(0.0646)	**0.6809(0.0682**)	0.6797(0.0665)	30
XLNet-BiLSTM	0.8262(0.0211)	0.8306(0.0192)	**0.8317(0.0224)**	50

Table 3 indicates that using the head and tail of the review is slightly beneficial for the Ott corpus. However, a significant difference in the truncation strategy was not concluded. The small difference could be because the dataset had not that many reviews to truncate. Nonetheless, in further experiments head and tail truncation strategy was applied as the results indicated a preference for head and tail which is a line in research by Sun et al.(2019).

4.2 Hyperparameter selection for RXB-Ensemble components

The subset of classifiers, BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT and XLNet-BiLSTM-ATT, of the proposed ensemble model RXB-Ensemble were tuned on learning rate. The following learning rates [5e-5, 1e-5, 5e-6, 1e-6, 5e-7] were evaluated in combination with a batch size of 5. A learning rate of 1e-6 seemed most appropriate as it converged properly and generated the highest performance. Appendix A presents the training and validation curves for every learning rate and model.

4.3 Model performance on the Ott Corpus

Table 4 reports the results of the models examined on the Ott corpus. The models were run multiple times in order to make the results statistically significant. For all evaluation metrics, accuracy, recall, precision and f1-score, the mean and standard deviation were reported.

Table 4

Performance of models on the Ott Corpus. Both the mean and standard deviation for accuracy, recall, precision and f1 are reported.

Model	Accuracy	Recall	Precision	F1	Number of runs
BERT-Log.Reg.	0.7840(0.0275)	0.7973(0.0442)	0.7767(0.0245)	0.7864(0.0298)	30
RoBERTa-Log.Reg.	0.8748(0.0175)	0.8735(0.0247)	0.8759(0.0170)	0.8745(0.0182)	30
XLNet-Log.Reg.	0.8392(0.0166)	0.8435(0.0306)	0.8368(0.0213)	0.8397(0.0175)	30
BERT-Naive Bayes	0.6807(0.0281)	0.7381(0.0429)	0.6638(0.0327)	0.6979(0.0252)	100
RoBERTa-Naive Bayes	0.7828(0.0218)	0.8097(.0292)	0.7691(0.0262)	0.7884(0.0207)	100
XLNet-Naive Bayes	0.6755(0.0251)	0.8055(0.033)	0.6398(0.0223)	0.7128(0.0218)	100
BERT-BiLSTM	0.7827(0.0235)	0.8012(0.0666)	0.7750(0.0389)	0.7851(0.0298)	50
RoBERTa-BiLSTM	0.8619(0.0200)	0.8746(0.0478)	0.8520(0.0417)	0.8613(0.0222	50
XLNet-BiLSTM	0.8169(0.0209)	0.8097(0.0536)	0.8280(0.0459)	0.8164(0.0238)	50
BERT-BiLSTM-ATT	0.9147(0.0123)	0.9311(0.0142)	0.9046(0.9045)	0.9174(0.0122)	10
RoBERTa-BiLSTM-ATT	0.9194(0.0155)	0.9427(0.0129)	0.9036(0.0284)	0.9225(0.0152)	10
XLNet-BiLSTM-ATT	0.9216(0.0141)	0.9445(0.0229)	0.9060(0.0284)	0.9244(0.0147)	10
RXB-Ensemble	**0.9306(0.0118)**	**0.9525(0.0170)**	**0.9148(0.0199)**	**0.9331(0.0122)**	10

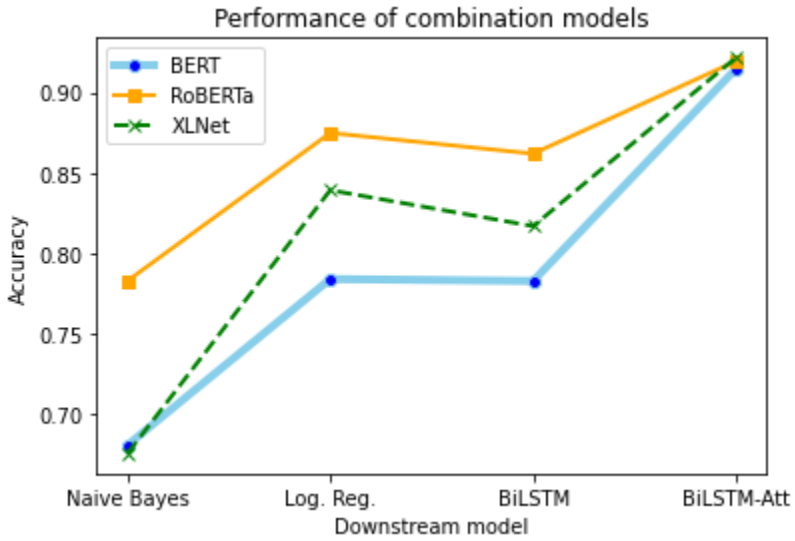
The models with BiLSTM with attention as downstream classification model, in general, outperformed other methods. BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT, XLNet-BiLSTM-ATT achieved an average accuracy of 0.9147, 0.9194, 0.9216 respectively. This was higher than models with BiLSTM as classification model, BERT-BiLSTM (0.7827), RoBERTa-BiLSTM (0.8619) and XLNet-BiLSTM (0.8169). Transformers in combination with Logistic Regression scored slightly better than BiLSTM,

BERT-Log.Reg. (0.7840), RoBERTa-Log.Reg. (0.8748), XLNet-Log.Reg (0.8392). Naive Bayes seemed the worst downstream classification model.

The results also indicated that RoBERTa is the preferred transformer used in a combination model for opinion spam detection. Figure 11 displays the average accuracy of the models split up by downstream classification model. As the yellow line indicates, RoBERTa as a transformer for word representation was most beneficial for opinion spam detection in combination with Naive Bayes, Logistic Regression and BiLSTM. XLNet seemed better than BERT in combination with Logistic regression and BiLSTM. The difference between word representation models minimized when the downstream classification model was BiLSTM with attention.

Figure 11

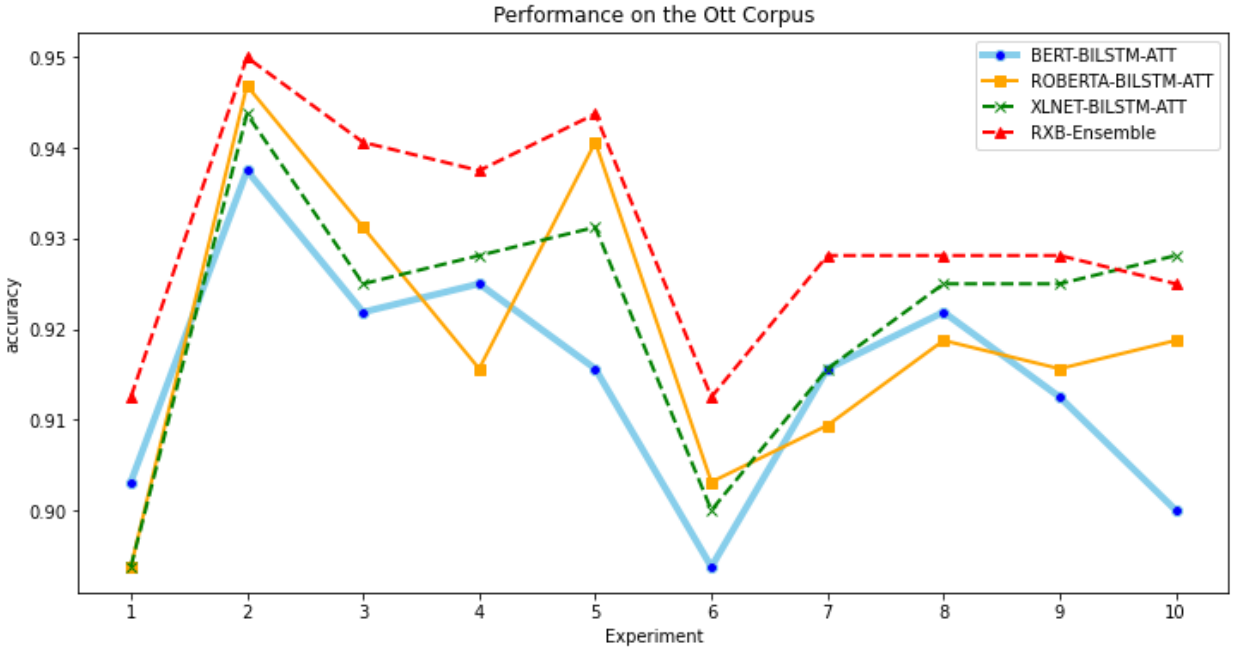
The average accuracy of models categorized by downstream classification model.



The proposed model RXB-Ensemble, a majority voting ensemble model with three different transformers, outperformed all models evaluated in this study. RXB-Ensemble achieved an average accuracy of 0.9306 which was higher than its components BERT-BiLSTM-ATT (0.9147), RoBERTa-BiLSTM-ATT (0.9194), XLNet-BiLSTM-ATT (0.9216). Figure 12 displays the accuracy of the individual classifiers of the ensemble model and RXB-Ensemble for every run. By ensembling predictions of the three classification models the performance was increased above the maximum of the individual components in all but one run.

Figure 12

The accuracy of RXB-Ensemble and its components (BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT and XLNet-BiLSTM-ATT) for every run on the Ott corpus.



4.4 Model performance on deception dataset

Below table 5 shows the results of RXB-Ensemble and its components on the deception dataset. Again the models were evaluated on accuracy, recall, precision and f1-score. All models were run on five different train-test splits.

Table 5

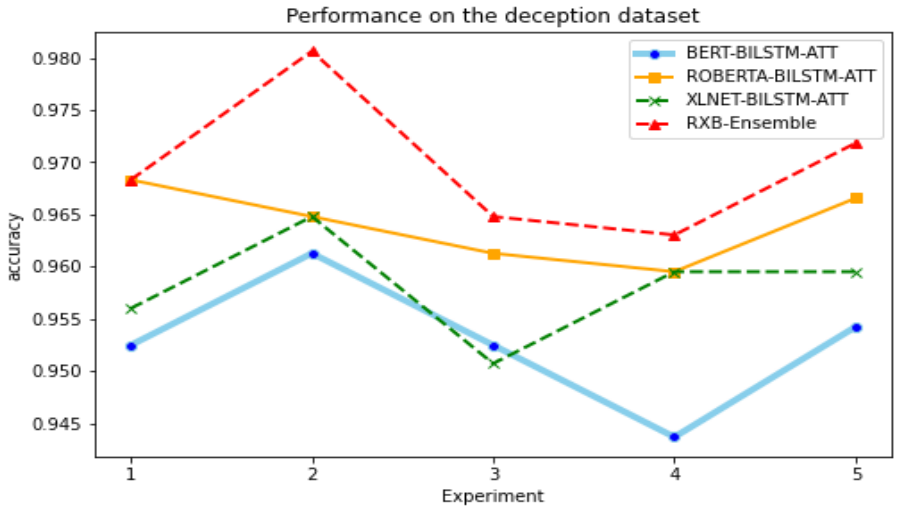
Performance of RXB-Ensemble and its components (BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT and XLNet-BiLSTM-ATT) on the deception dataset. Both the mean and standard deviation for accuracy, recall, precision and f1 are reported.

Model	Accuracy	Recall	Precision	F1	Number of runs
BERT-BiLSTM-ATT	0.9528(0.0056)	0.9697(0.0036)	0.9486(0.0119)	0.9589(0.0055)	5
RoBERTa-BiLSTM-ATT	0.9641(0.0033)	0.9784(0.0159)	0.9597(0.0168)	0.9687(0.0032)	5
XLNet-BiLSTM-ATT	0.9581(0.0046)	0.9780(0.0135)	0.9501(0.0098)	0.9637(0.0041)	5
RXB-Ensemble	**0.9697(0.0062)**	**0.9865(0.0070)**	**0.9615(0.0138)**	**0.9737(0.0056)**	5

RXB-Ensemble achieved an average accuracy of 96.97% on the deception dataset. Its components BERT-BiLSTM-ATT (0.9528), RoBERTa-BiLSTM-ATT (0.9641), XLNet-BiLSTM-ATT (0.9581) achieved lower average accuracies. Figure 13, below, shows the performance of the RXB-Ensemble and its components for every run. RXB-Ensemble performed better on four out of five runs and tied on one run with RoBERTa-BILSTM-ATT.

Figure 13

The accuracy of RXB-Ensemble and its components (BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT and XLNet-BiLSTM-ATT) for every run on the deception dataset.



Although RXB-Ensemble on average performed better on the deception dataset it does not excel in every domain. Table 6, below, presents the results across the three different domains in the dataset. For the doctor and restaurant domain, RXB-Ensemble was able to increase the average accuracy compared to its best component classifier with 1.2% and 2.9% respectively. RXB-Ensemble performed slightly worse than the best model, RoBERTa-BiLLSTM-ATT, in the ensemble for the hotel domain with a difference of 0.17%. This difference was smaller compared to the other domains. For detailed performance per domain, Appendix B displays the performance of the models for every run for every domain in graphs.

Table 6

Performance of RXB-Ensemble and its components (BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT and XLNet-BiLSTM-ATT) on the deception dataset split up by domain. Both the mean and standard deviation for accuracy, recall, precision and f1 are reported.

Domain	Model	Accuracy	Recall	Precision	F1
Doctor	BERT-BiLSTM-ATT	0.9116(0.02000)	0.9444(0.0217)	0.9145(0.0279)	0.9289(0.0184)
	RoBERTa-BiLSTM-ATT	0.9280(0.0383)	0.9771(0.0117)	0.9125(0.0509)	0.9431(0.0309)
	XLNet-BiLSTM-ATT	0.9162(0.0129)	0.9679(0.0235)	0.9822(0.0076)	0.9873(0.0046)
	RXB-Ensemble	**0.9397(0.0223)**	**0.9807(0.0127)**	**0.9256(0.0329)**	**0.8520(0.0187)**
Hotel	BERT-BiLSTM-ATT	0.9773(0.0078)	0.9886(0.0070)	0.9724(0.0172)	0.9803(0.0071)
	RoBERTa-BiLSTM-ATT	**0.9898(0.0048)**	**0.9935(0.0056)**	**0.9888(0.0111)**	**0.9911(0.0043)**
	XLNet-BiLSTM-ATT	0.9853(0.0054)	0.9925(0.0071)	0.9822(0.0076)	0.9873(0.0046)
	RXB-Ensemble	0.9881(0.0063)	0.9962(0.0047)	0.9833(0.0107)	0.9897(0.0055)
Restaurant	BERT-BiLSTM-ATT	0.8995(0.0257)	0.9132(0.0464)	0.8836(0.0438)	0.8969(0.0299)
	RoBERTa-BiLSTM-ATT	0.8993(0.0443)	0.9007(0.0983)	0.8988(0.0683)	0.8938(0.0519)
	XLNet-BiLSTM-ATT	0.8955(0.0115)	0.9193(0.0351)	0.8727(0.0287)	0.8945(0.0136)
	RXB-Ensemble	**0.9284(0.0255)**	**0.9441(0.0295)**	**0.9120(0.0445)**	**0.9269(0.0264)**

5. Discussion

5.1 Findings and contributions

The aim of the study was to research how we can use pre-trained transformers for opinion spam detection. This study proposed a new approach, RXB-Ensemble, that effectively implemented pre-trained transformers for opinion spam detection. First, we empirically researched the optimum combination of pre-trained language models in combination with Logistic Regression, BiLSTM and BiLSTM with attention mechanism. Secondly, we showed that ensembling transformer-based classifiers enhance the performance for opinion spam.

This study described a method for combining pre-trained transformer-based language models with Logistic Regression, BiLSTM and BiLSTM with attention mechanism. Pretrained language models BERT, RoBERTa and XLNet were applied. RoBERTa and XLNet were able to more accurately classify fake reviews. RoBERTa based combination models overall achieved the highest performance. Differences in performance between transformers minimized when transformers were paired up with BiLSTM with attention mechanism as the downstream classification model. Baserver et al.'s (2020) BERT-BiLSTM-Att model did not achieve the claimed state-of-the-art performance of 93,60% for opinion spam detection. In this study, this model achieved a lower score of 91.47%. The difference in model performance could be due to the fact that the models in this study were run multiple times, whereas Barsever et al.'s (2020) model was possibly run only once. Therefore, their high model performance might be due to a random lucky data split. As expected, BiLSTM with attention mechanism seemed to be the superior downstream classification model compared to Logistic Regression and BiLSTM. This is in line with Lee et al.'s (2019) research on the application of BERT-BiLSTM-ATT for multi-class medical text classification.

The proposed model, RXB-Ensemble, effectively showed that ensembling transformer-based classifiers enhance the performance for opinion spam detection. RXB-Ensemble managed to increase the mean accuracy of its component classifiers BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT and XLNet-BiLSTM-ATT on the Ott Corpus with 1.59%, 1.12% and 0.90% respectively. For the deception dataset, mean accuracy also improved for all ensemble components. RXB-Ensemble enhanced the mean accuracy with 1.69%, 0.56%, 1.16% compared to BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT and XLNet-BiLSTM-ATT. The domain results of the deception dataset showed that RXB-Ensemble performed worse than RoBERTa-BilSTM in the hotel domain. RXB-Ensemble, on average, achieved a lower accuracy (-0.17%) than RoBERTa-BiLSTM-ATT in the hotel domain. This underperformance might have occurred because RoBERTa-BiLSTM-ATT scored fairly higher than the other two component classifiers on two out of five runs. Therefore, the votes might have swung in the favour of the two lesser performing classifiers. In the doctor and restaurant domain a minimum improvement of 1.17% and 2.89%, respectively, were concluded. These findings are in line with Chang et al.'s (2020) study where ensembling three classifiers

with XLNet, RoBERTa and BERT enhanced performance for extreme multi-label text classification. Further, ensembling different pre-trained language models in combination with BiLSTM with attention as a downstream classification model was highly effective, as was concluded by Lee et al. (2019).

5.2 Limitations and future research

With these results, some limitations should be considered. Firstly, computational limitations meant that the complex models could not be run more times in order or make the conclusions more robust. Secondly, we acknowledge that selecting parameters of the BiLSTM model with attention on train-validation-test-split was not how theory defines hyperparameter selection but was necessary in order to bring down computation time. This should not have impacted the results enormously. Further, both spam datasets were balanced while spam in real-life is not balanced. Although these datasets are the best the research field has to offer, it should be taken into consideration.

Future research should firstly focus on developing an opinion spam dataset that has a pre-defined testing set. This should create research that is more comparable, robust and mitigates the impact of a random lucky split in the research community. Further, it would reduce the computational need for multiple experiments. Secondly, research on the ensembling of transformers is still lacking. It would be highly beneficial to further explore different transformer combinations and ensemble strategies. Lastly, as these models are computationally expensive it would be interesting to introduce distilled transformer models to lower the computation time to see to what extent the performance will drop at cost of computational power. This is highly interesting for businesses that do not have the budget to get greater amounts of computational power but still want to tackle the problem of opinion spam.

6. Conclusion

This study examined how various pre-trained transformers can be applied in order to detect online opinion spam. Some confusion relating to the optimal downstream classification model is unravelled, as there was no consensus in the literature. Further, this study contributed to existing research by ensembling the best models using a majority voting strategy for a model that generated state-of-the art results for opinion spam detection.

For the detection of opinion spam, XLNet and RoBERTa gave a better performance than BERT in combination with all downstream classification models. These transformers are more refined and, thus, can better represent the information of the review. Further, BiLSTM with attention as a downstream classification model performed better than Logistic Regression and BiLSTM.

Finally, the proposed model RXB-Ensemble showed the potential of ensembling multiple transformers-based classifiers for opinion spam detection. RXB-Ensemble is composed of BERT-BiLSTM-ATT, RoBERTa-BiLSTM-ATT and XLNet-BiLSTM-ATT. By using three different transformers in the classifiers RXB-Ensemble generated state-of-the-art results for online opinion spam detection. RXB-Ensemble contributed to a fairly underdeveloped research field where performance increments can be made by leveraging the best of various transformers. Thoroughly researching the limitations of ensembling transformers could be highly beneficial for natural language processing.

References

Barsever, D., Singh, S., & Neftci, E. (2020). Building a Better Lie Detector with BERT: The Difference Between Truth and Lies. *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–7. https://doi.org/10.1109/ijcnn48605.2020.9206937

Buettner, R. (2016). Predicting user behavior in electronic markets based on personality-mining in large online social networks. *Electronic Markets*, *27*(3), 247–265. https://doi.org/10.1007/s12525-016-0228-z

Cai, R., Qin, B., Chen, Y., Zhang, L., Yang, R., Chen, S., & Wang, W. (2020). Sentiment Analysis About Investors and Consumers in Energy Market Based on BERT-BiLSTM. *IEEE Access*, *8*, 171408–171415. https://doi.org/10.1109/access.2020.3024750

Callan, J., Hoy, M., Yoo, C., & Zhao, L. (2009). *The ClueWeb09 Dataset*. Lemur. https://lemurproject.org/clueweb09

Chang, W.-C., Yu, H.-F., Zhong, K., Yang, Y., & Dhillon, I. S. (2020). Taming Pretrained Transformers for Extreme Multi-label Text Classification. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 3163–3171. https://doi.org/10.1145/3394486.3403368

Daen, G. (2020, September 7). *Amazon deleted 20,000 product ratings after an investigation highlighted paid-for reviews*. Business Insider. https://www.businessinsider.com/amazon-deleted-product-reviews-after-study-highlighted-paid-for-ratings-2020-9?international=true&r=US&IR=T

Devlin, J., & Chang, M. W. (2018, November 2). *Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing*. Google AI Blog. https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. *Association for Computing Machinery, Proceedings of the First International Workshop on Multiple Classifier Systems*, 1–15.

Feng, S., Banerjee, R., & Choi, Y. (2012). Syntactic Stylometry for Deception Detection. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers*, *2*, 171–175. https://www.aclweb.org/anthology/P12-2034.pdf

Gokaslan, A., & Cohen, V. (2019). *Openwebtext corpus.* Github. http://Skylion007.github.io/OpenWebTextCorpus.

Hochreiter, S. (1998). The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *06*(02), 107–116. https://doi.org/10.1142/s0218488598000094

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang, H., Jin, Y., & Rao, R. (2020). SCoEmbeddings: encoding sentiment information into contextualized embeddings for sentiment analysis. *Proceedings of the 17th ACM International Conference on Computing Frontiers*, 261–264. https://doi.org/10.1145/3387902.3394948

Huang, Y., Jiang, Y., Hasan, T., Jiang, Q., & Li, C. (2018). A topic BiLSTM model for sentiment classification. *Association for Computing Machinery*, *Proceedings of the 2nd International Conference on Innovation in Artificial Intelligence*, 143–147. https://doi.org/10.1145/3194206.3194240

Jindal, N., & Liu, B. (2008). Opinion spam and analysis. *Proceedings of the 2008 International Conference on Web Search and Web Data Mining - WSDM 2008*, 219–230. https://doi.org/10.1145/1341531.1341560

Khurshid, F., Zhu, Y., Xu, Z., Ahmad, M., & Ahmad, M. (2019). Enactment of Ensemble Learning for Review Spam Detection on Selected Features. *International Journal of Computational Intelligence Systems*, *12*(1), 387–394. https://doi.org/10.2991/ijcis.2019.125905655

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization., *Proceedings of International Conference on Learning Representations.* arXiv preprint arXiv:1412.6980

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling* (pp. 141-171). Springer Publishing. https://doi.org/10.1007/978-1-4614-6849-3

Lee, L. H., Lu, Y., Chen, P. H., Lee, P. L., & S, K. K. (2019). NCUEE at MEDIQA 2019: Medical Text Inference Using Ensemble BERT-BiLSTM-Attention Model. *Association for Computational Linguistics, Proceedings of the 18th BioNLP Workshop and Shared Task*, 528–532. https://doi.org/10.18653/v1/W19-5058

Li, Y., Pan, Q., Wang, S., Yang, T., & Cambria, E. (2018). A Generative Model for category text generation. *Information Sciences, 450*, 301–315. https://doi.org/10.1016/j.ins.2018.03.050

Li, H., Chen, Z., Liu, B., Wei, X., & Shao, J. (2014). Spotting Fake Reviews via Collective Positive-Unlabeled Learning. *2014 IEEE International Conference on Data Mining*, 899–904. https://doi.org/10.1109/icdm.2014.47

Li, J., Ott, M., Cardie, C., & Hovy, E. (2014). Towards a General Rule for Identifying Deceptive Opinion Spam. *Association for Computational Linguistics*, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1566–1576. https://doi.org/10.3115/v1/p14-1147

Lim, E. P., Nguyen, V. A., Jindal, N., Liu, B., & Lauw, H. W. (2010). Detecting product review spammers using rating behaviors. *Association for Computing Machinery*, *Proceedings of the 19th ACM international conference on Information and knowledge management*, 939–948. https://doi.org/10.1145/1871437.1871557

Liu, G., & Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing*, *337*, 325–338. https://doi.org/10.1016/j.neucom.2019.01.078

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O.., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*

Lozano, M. G., & Fernquist, J. (2019). Identifying Deceptive Reviews: Feature Exploration, Model Transferability and Classification Attack. *2019 European Intelligence and Security Informatics Conference (EISIC)*, 109–116. https://doi.org/10.1109/eisic49498.2019.9108852

McCabe, K. (2020, September 28). *51 Customer Review Statistics to Make You Rethink Using Them*. Learning Hub. https://learn.g2.com/customer-reviews-statistics

Mudambi, & Schuff. (2010). Research Note: What Makes a Helpful Online Review? A Study of Customer Reviews on Amazon.com. *MIS Quarterly*, *34*(1), 185. https://doi.org/10.2307/20721420

Mukherjee, A., Venkataraman, V., Liu, B., & Glance, N. (2013). What Yelp Fake Review Filter Might Be Doing?. *Proceedings of the International AAAI Conference on Web and Social Media*, *7*(1), 409-418.

Nagel, S. (2016, October 4). *Cc-news.* Common Crawl. https://commoncrawl.org/2016/10/newsdataset-available.

Ott, M., Cardie, C., Choi, Y., & Hancock, J.T. (2011). Finding deceptive opinion spam by any stretch of the imagination. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011),* 309–319. https://arxiv.org/abs/1107.4557

Parker, R., Graff, D., Kong, J., Chen, K., & Maeda, K. (2011). English gigaword fifth edition. *Linguistic Data Consortium.* https://doi.org/10.35111/wk4f-qt80

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/d14-1162

Ren, Y., & Ji, D. (2017). Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences*, *385–386*, 213–224. https://doi.org/10.1016/j.ins.2017.01.015

Risch, J., & Krestel, R. (2020). Bagging BERT Models for Robust Aggression Identification. *European Language Resources Association, Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, 55–61. https://www.aclweb.org/anthology/2020.trac-1.9.pdf

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681. https://doi.org/10.1109/78.650093

Stanton, G., & A. Irissappane, A. (2019). GANs for Semi-Supervised Opinion Spam Detection. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 5204–5210. https://doi.org/10.24963/ijcai.2019/723

Trinh, T. H., & Le, Q. V. (2018). A Simple Method for Commonsense Reasoning. https://arxiv.org/abs/1806.02847

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 5998–6008. https://arxiv.org/abs/1706.03762

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2018). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 353–355. https://doi.org/10.18653/v1/w18-5446

Wang, W. Y. (2017). 'Liar, Liar Pants on Fire': A New Benchmark Dataset for Fake News Detection. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 422–426. https://doi.org/10.18653/v1/p17-2067

Xu, G., Meng, Y., Qiu, X., Yu, Z., & Wu, X. (2019). Sentiment Analysis of Comment Texts Based on BiLSTM. *IEEE Access*, *7*, 51522–51532. https://doi.org/10.1109/access.2019.2909919

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 5753–5763.

Zhang, C., & Ma, Y. (2012). *Ensemble Machine Learning* (pp. 1-34). Springer Publishing. https://doi.org/10.1007/978-1-4419-9326-7

Zhang, W., Du, Y., Yoshida, T., & Wang, Q. (2018). DRI-RCNN: An approach to deceptive review identification using recurrent convolutional neural network. *Information Processing & Management*, *54*(4), 576–592. https://doi.org/10.1016/j.ipm.2018.03.007

Zhou, Q., & Wu, H. (2018). NLP at IEST 2018: BiLSTM-Attention and LSTM-Attention via Soft Voting in Emotion Classification. *Association for Computational Linguistics*, *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 189–194. https://doi.org/10.18653/v1/w18-6226

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. *2015 IEEE International Conference on Computer Vision (ICCV)*, 19–27. https://doi.org/10.1109/iccv.2015.11

Zheng, Z., & Webb, G. I. (2000). Lazy Learning of Bayesian Rules. *Machine Learning*, *41*(1), 53–84. https://doi.org/10.1023/a:1007613203719

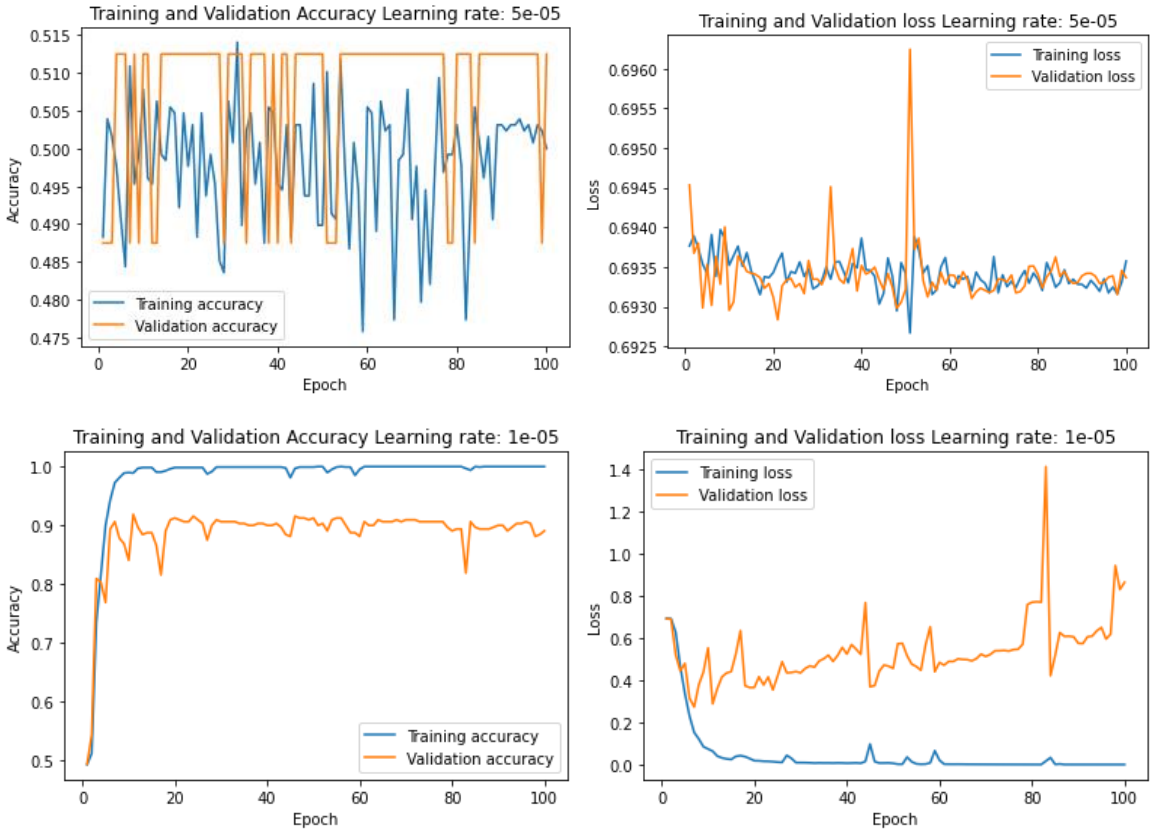
7. Appendices

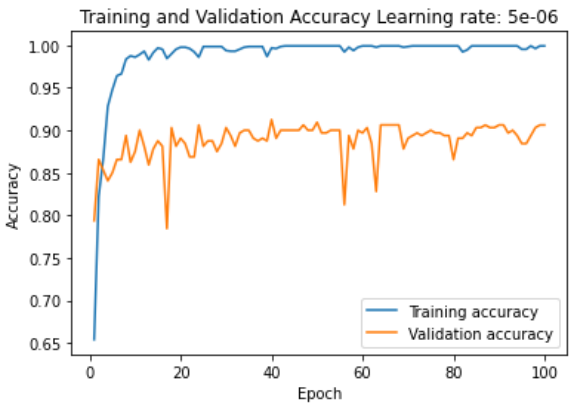
7.1 Appendix A: Accuracy and loss graphs for learning rate selection

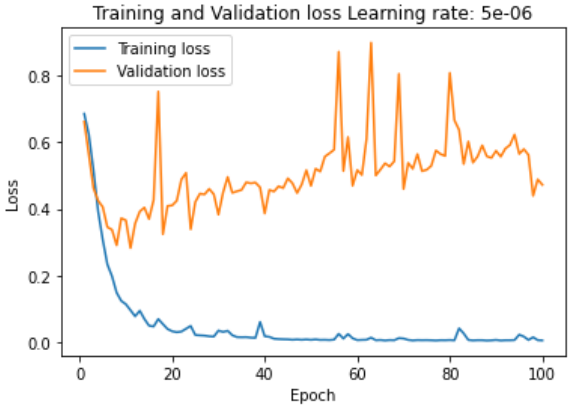
The figures below show the training and validation accuracy and loss for learning rates [5e-5, 1e-5, 5e-6, 1e-6, 5e-7] tested in this study. First the graphs for BERT-BILSTM-ATT (figure 14) are presented followed by RoBERTa-BILSTM-ATT (figure 15) and XLNet-BILSTM-ATT (figure 16).

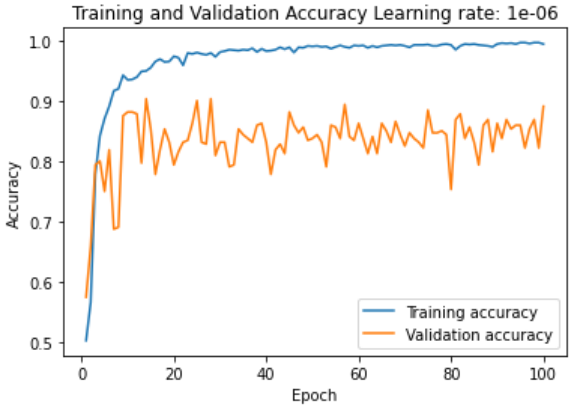
Figure 14

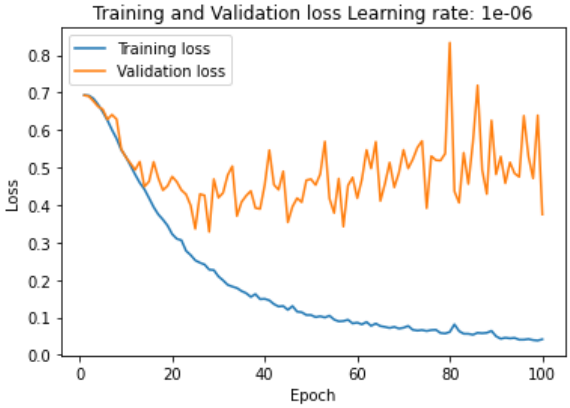
Training-validation curve graphs for BERT-BiLSTM-ATT learning rate selection. Left accuracy graphs and right the corresponding loss graph for learning rates [5e-5, 1e-5, 5e-6, 1e-6, 5e-7].

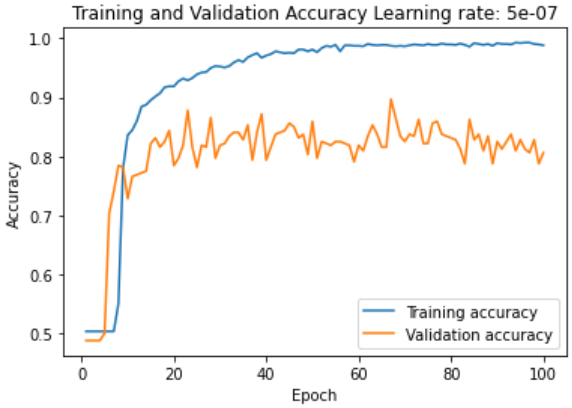



Training and Validation Accuracy Learning rate: 5e-06
Accuracy
Epoch
Training accuracy
Validation accuracy


Training and Validation loss Learning rate: 5e-06
Loss
Epoch
Training loss
Validation loss


Training and Validation Accuracy Learning rate: 1e-06
Accuracy
Epoch
Training accuracy
Validation accuracy


Training and Validation loss Learning rate: 1e-06
Loss
Epoch
Training loss
Validation loss


Training and Validation Accuracy Learning rate: 5e-07
Accuracy
Epoch
Training accuracy
Validation accuracy

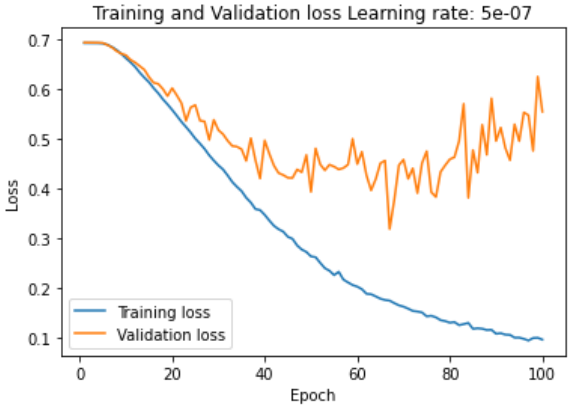
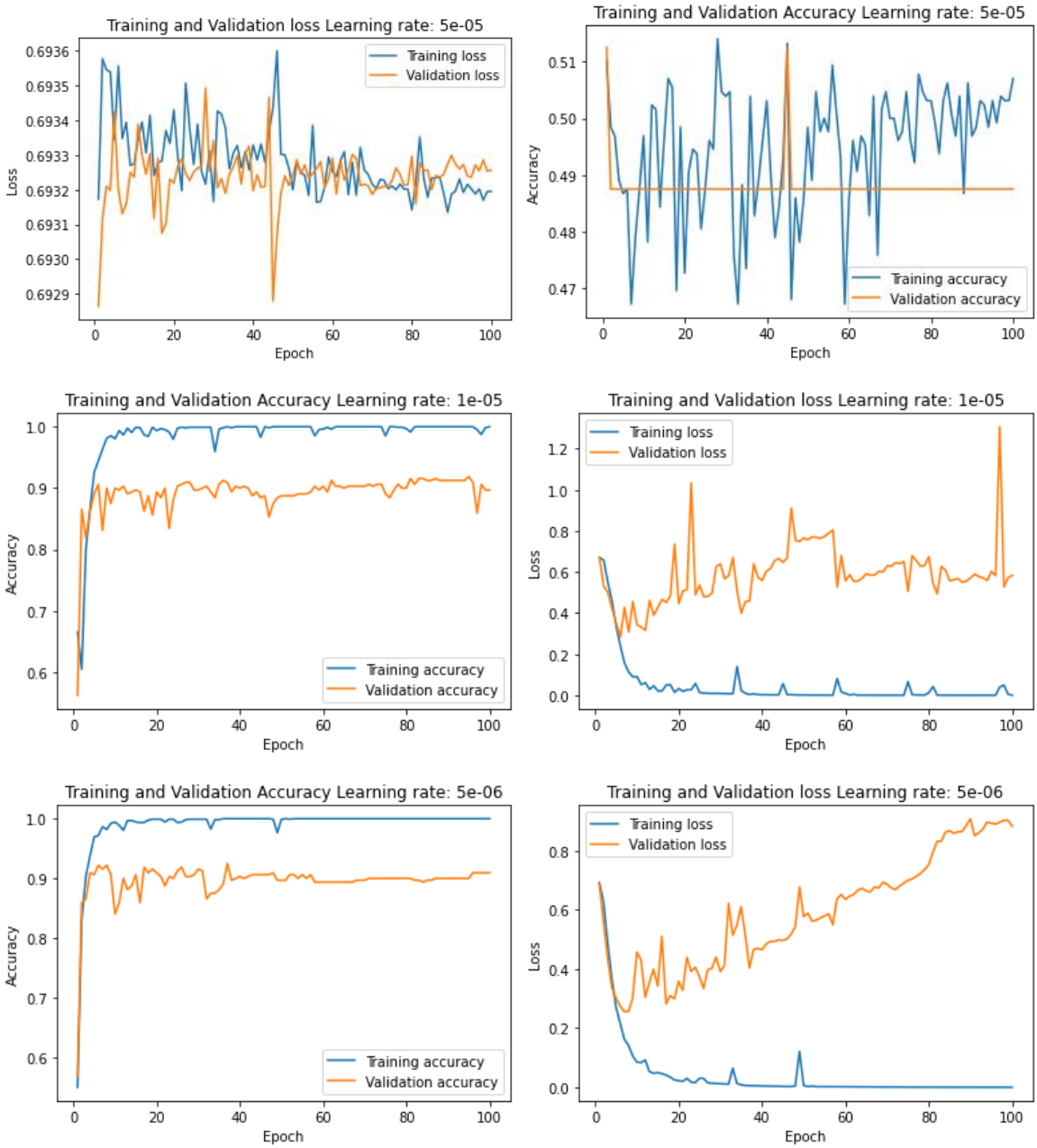

Training and Validation loss Learning rate: 5e-07
Loss
Epoch
Training loss
Validation loss

Figure 15

Training-validation curve graphs for RoBERTa-BiLSTM-ATT learning rate selection. Left accuracy graphs and right the corresponding loss graph for learning rates [5e-5, 1e-5, 5e-6, 1e-6, 5e-7].



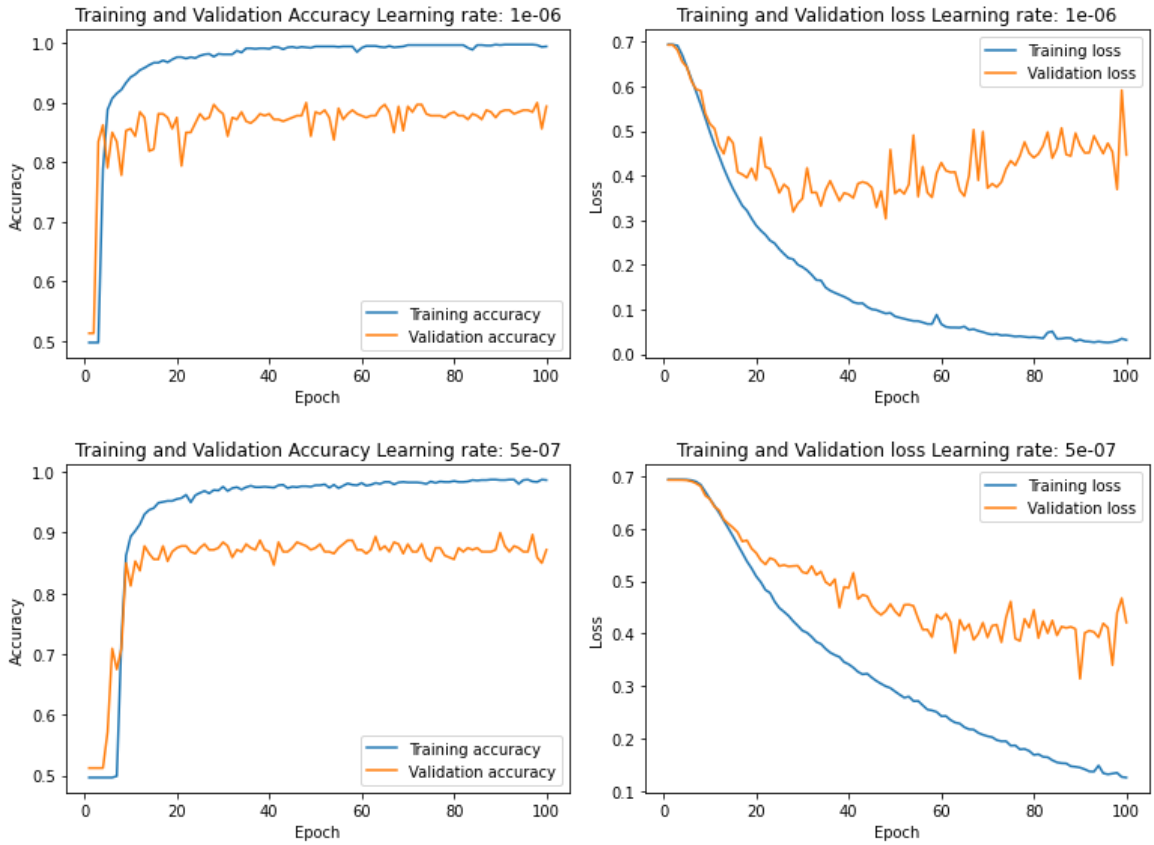
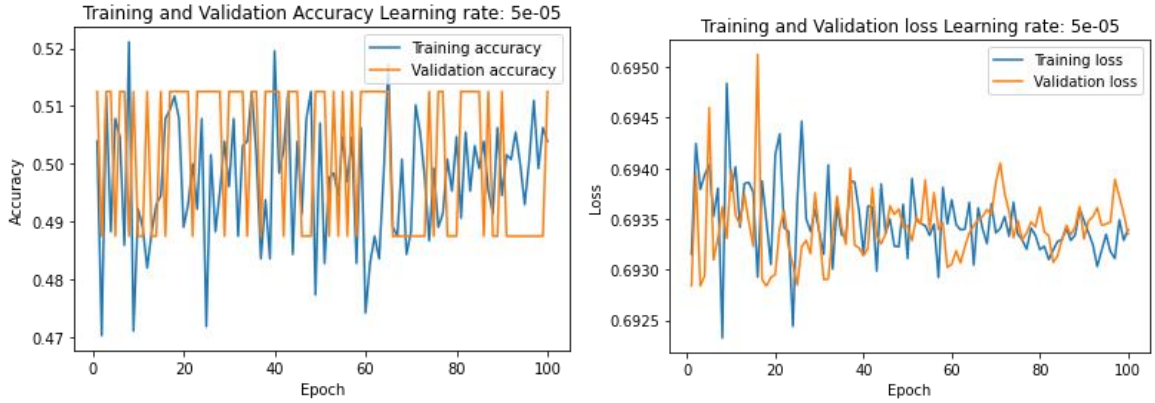
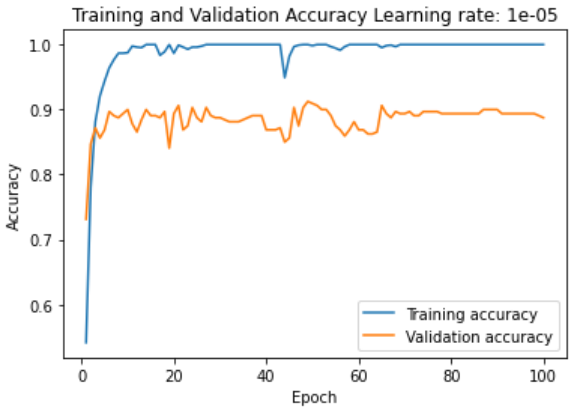
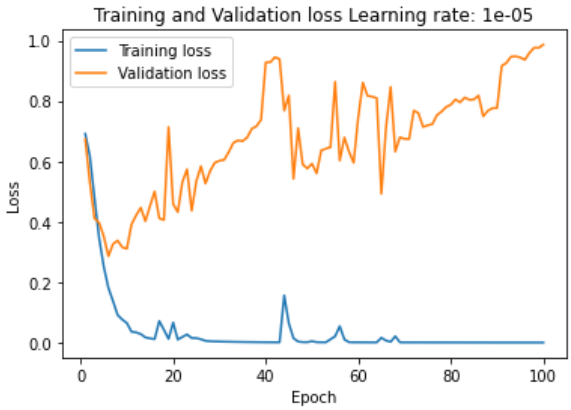


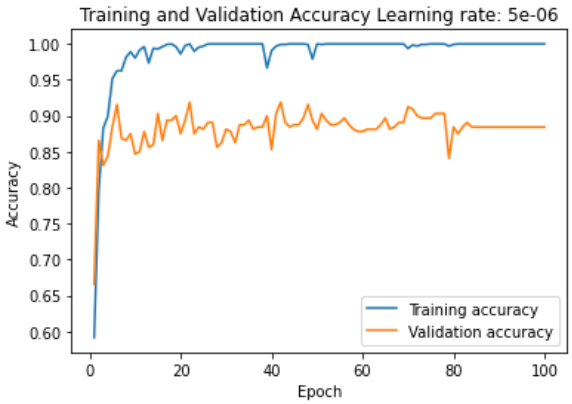
Figure 16

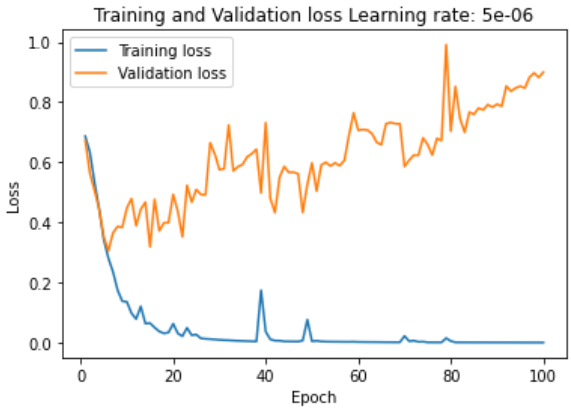
Training-validation curve graphs for XLNet-BiLSTM-ATT learning rate selection. Left accuracy graphs and right the corresponding loss graph for learning rates [5e-5, 1e-5, 5e-6, 1e-6, 5e-7].

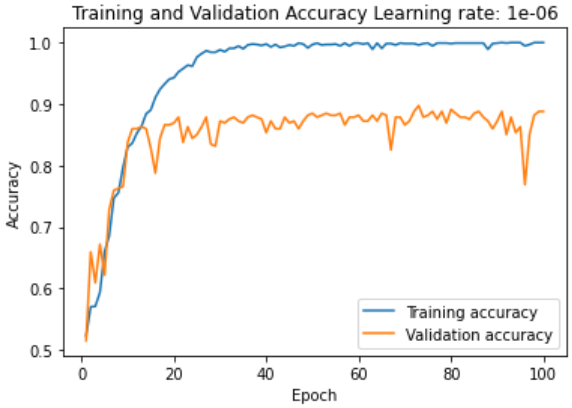


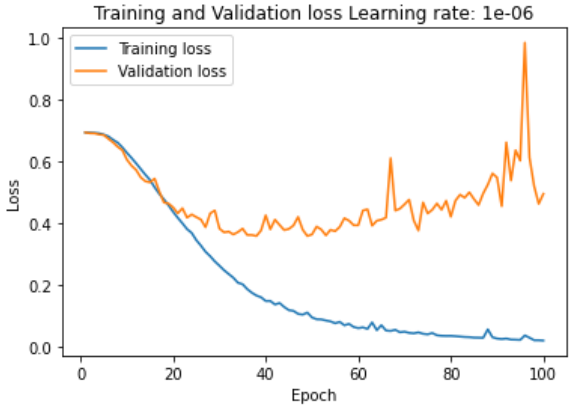

Training and Validation Accuracy Learning rate: 1e-05
Accuracy
Epoch
Training accuracy
Validation accuracy

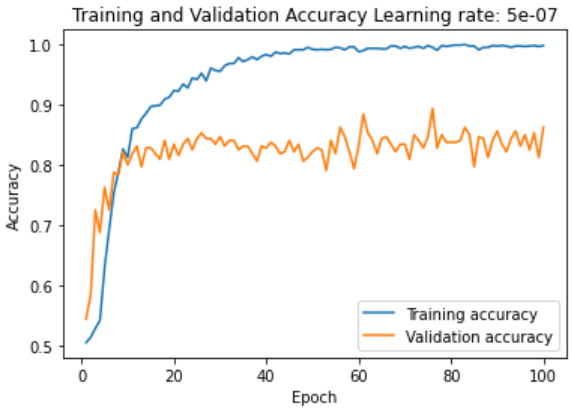

Training and Validation loss Learning rate: 1e-05
Loss
Epoch
Training loss
Validation loss

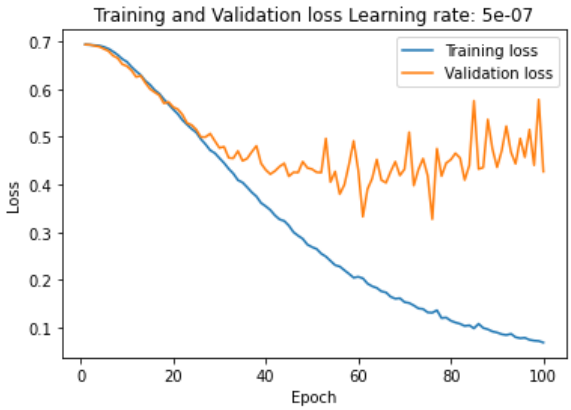

Training and Validation Accuracy Learning rate: 5e-06
Accuracy
Epoch
Training accuracy
Validation accuracy


Training and Validation loss Learning rate: 5e-06
Loss
Epoch
Training loss
Validation loss


Training and Validation Accuracy Learning rate: 1e-06
Accuracy
Epoch
Training accuracy
Validation accuracy


Training and Validation loss Learning rate: 1e-06
Loss
Epoch
Training loss
Validation loss


Training and Validation Accuracy Learning rate: 5e-07
Accuracy
Epoch
Training accuracy
Validation accuracy


Training and Validation loss Learning rate: 5e-07
Loss
Epoch
Training loss
Validation loss

7.2 Appendix B: Accuracy for RXB-ensemble and model component for every run

Below figure 17, 18 and 19 displays the performance of the three models BERT-BiLSTM-ATT, XLNet-BiLSTM-ATT, RoBERTa-BiLSTM-ATT and RXB-Ensemble for every domain in the deception dataset (doctor, hotel and restaurant). The x-axis indicates different train-test splits and the y-axis indicates the accuracy.

Figure 17

Visualization of the accuracy per individual run for the doctor domain.

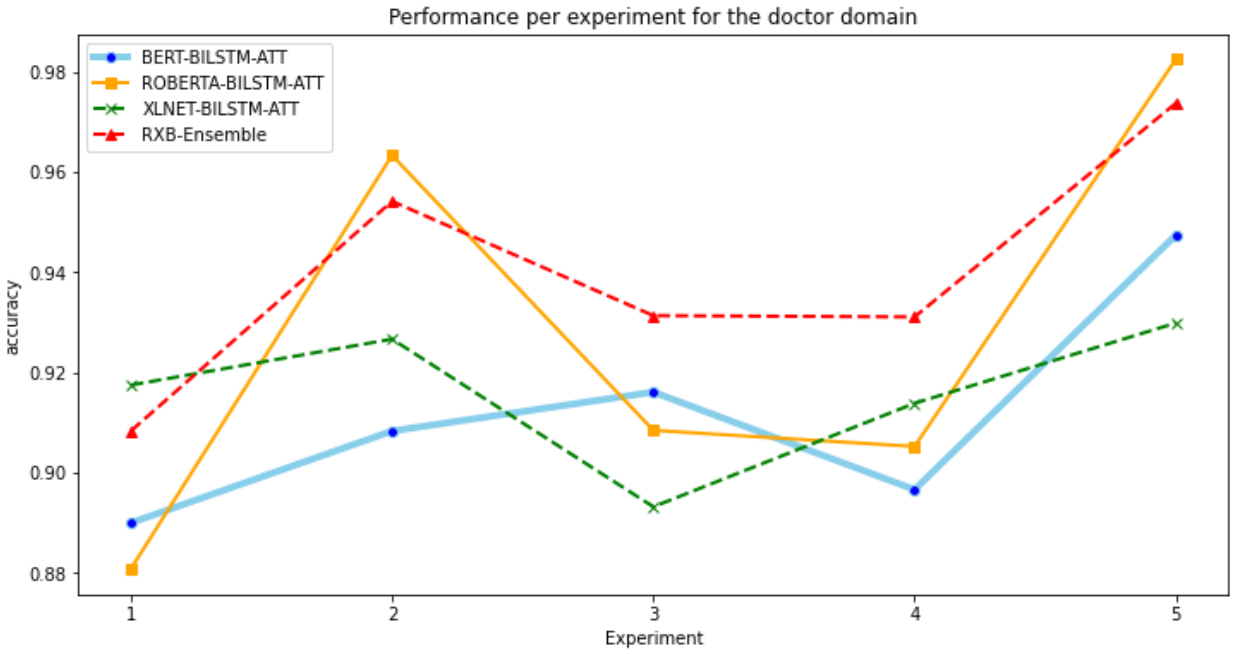


Figure 18

Visualization of the accuracy per individual run for the hotel domain.

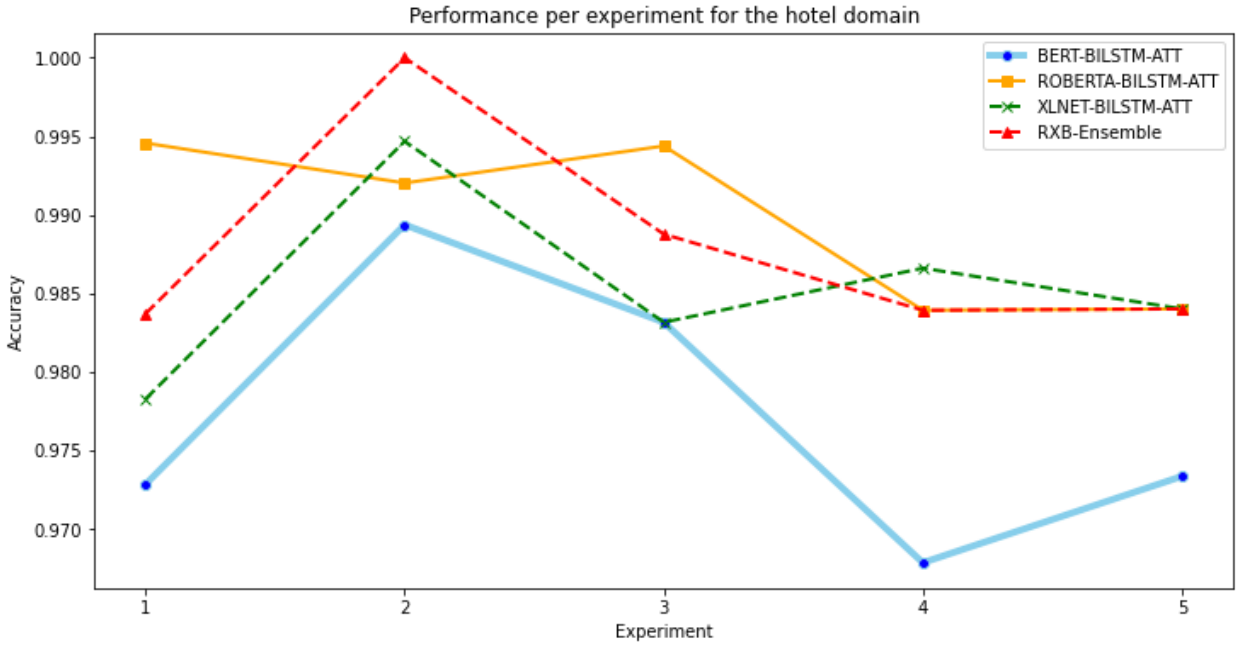


Figure 19

Visualization of the accuracy per individual run for the restaurant domain.