The Impact of Transformer-Based Models on Fake News Detection

Arda Ağaçdelen Yiğit Kaya Bağcı

1 Introduction

1.1 The Spread of Digital Misinformation and the Verification Gap

In today's digital world, the fast spread of news has changed how people talk about public issues. The internet was supposed to democratize information. However, it has increasingly become a channel for misinformation, disinformation, and mal-information. We call this group of false information "fake news."

News spreads very fast on social media platforms like Twitter, Facebook, and Reddit. This creates an "infodemic." This means there is too much information, both true and false, for humans to check alone.

This research identifies a serious problem called the "Verification Gap." Millions of news stories and articles are published on the internet every day. However, humans can only fact-check a limited amount of content. Professional fact-checking organizations use strict methods, but they cannot keep up with the huge amount of content created by bots and fake news farms.

This gap allows fake news to spread without anyone stopping it. This affects elections, public health, and social stability.

The problem is getting worse because fake news creators are becoming smarter. In the past, fake news had bad grammar and used very emotional words. It was easy to find these with simple computer filters. However, modern fake news is different. Creators now copy the style and tone of professional journalists. This is called "syntax mimicking."

They use formal language and scientific words to look real. Because of this, simple computer programs that look for "bad words" cannot tell the difference between a real report and a fake story.

1.2 The Problem with Traditional Machine Learning

For about ten years, the standard way to classify text used traditional machine learning (ML). Algorithms like Logistic Regression, Support Vector Machines (SVM), and Naive Bayes were the main tools. These models mostly used a technique called Bag of Words (BoW) and TF-IDF (enhanced version of BoW). These techniques turns a text into a list of word numbers. [2]

These methods were good starting points. However, they have theoretical limits when detecting fake news. The BoW assumption says that we can understand a document just by counting its words, without looking at the order. This removes the context from the language. Traditional ML models treat words as separate units and do not see how they connect to each other.

For example, "man bites dog" and "dog bites man" mean very different things. But to a BoW model, they look the same because they have the exact same words. Also, traditional models look for statistical patterns, not meaning. They struggle to find complex things like sarcasm or irony. Fake news often uses sarcasm to hide lies.

A traditional model might think a sarcastic sentence is a positive sentence because it uses positive words. This "semantic blindness" (inability to see meaning) makes it hard to improve accuracy against modern fake news.

1.3 The Power of Deep Learning

To fix the Verification Gap, the field of Natural Language Processing (NLP) has moved to Deep Learning, specifically the Transformer architecture. Unlike older models, Deep Learning models analyze the logic, structure, and order of a story.

Models like BERT and its successors, such as DeBERTa (Decoding-enhanced BERT with disentangled attention), changed how machines understand language. These modern architectures use "self-attention mechanisms." This allows the model to decide which words are important in relation to other words. [6]

This helps the model understand words with multiple meanings. For example, the model knows the difference between "run a company" and "run a marathon." Transformers read the whole text at once. They can find hidden patterns that traditional models miss.

Also, these models are "pre-trained" on huge amounts of text (like Wikipedia). This is called "transfer learning." It means models like DeBERTa already know a lot about language. They need less training data to work well compared to building a network from scratch. This is very important for fake news because good datasets are hard to find.

1.4 Research Objectives

This study wants to prove that Transformer-based models are better than traditional machine learning for detecting fake news. The research focuses on these goals:

1. **Theoretical Comparison:** To compare the math limits of the Bag of Words model (problems with word independence) against the power of the Transformer architecture (attention mechanisms).

2. **Architectural Analysis:** To explain the improvements in the DeBERTa architecture. We focus on its Disentangled Attention mechanism and Enhanced Mask Decoder. We explain why these features are good for finding lies in text. [3]

3. **Empirical Evaluation:** To test the models using the ISOT, LIAR, and FakeNewsNet datasets. We compare the DeBERTa v3 model (fine-tuned with Low-Rank Adaptation) against baselines like Logistic Regression, SVM, and Naive Bayes.

4. **Contextual Insight:** To analyze the results to show that the models can understand deep context, like sarcasm and structure. This proves we need deep learning for modern verification systems.

2 Theoretical Framework: From Statistics to Semantics

This section explains the theory behind the study. It compares the math of traditional text models with modern deep learning. It argues that traditional models fail because of a basic problem in how they represent language mathematically.

2.1 Traditional Machine Learning and the Bag of Words (BoW)

2.1.1 Assumption

Before 2018, the main method for text classification was the Bag of Words (BoW) model. In this framework, a document is just a collection of words. It ignores grammar and word order. [1]

2.1.2 Mathematical Formulation of BoW

Let a collection of texts C contain D documents. First, we make a vocabulary $V = \{w_1, w_2, ..., w_M\}$ of all unique words. A specific document d_i is shown as a vector $\mathbf{x}_i \in \mathbb{R}^M$:

$$\mathbf{x}_i = [c_1, c_2, ..., c_M] \tag{1}$$

Here, c_j is the frequency (count) of word w_j in document d_i. This vector exists in a space with M dimensions. To fix the problem where common words (like "the") appear too often but have little meaning, we use Term Frequency-Inverse Document Frequency (TF-IDF):

$$\text{TF-IDF}(w, d) = \text{tf}(w, d) \times \log\left(\frac{N}{\text{df}(w)}\right) \tag{2}$$

N is the total number of documents and $\text{df}(w)$ is the number of documents with word w. This formula gives more weight to rare words.

2.1.3 The Flaws of Orthogonality and Independence

The BoW model is fast, but it has serious theoretical flaws for fake news detection:

1. **Orthogonality of Synonyms:** In a BoW vector space, every word is a separate dimension. Mathematically, the dot product between the vectors for "deceit" and "fraud" is zero. This implies they are not similar at all. Traditional models do not understand that these words have similar meanings unless they appear together often in the training data. The model cannot generalize. If it learns that "deceit" means fake news, it does not automatically know that "fraud" might mean the same thing. [2]

2. **The Independence Assumption:** Algorithms like Naive Bayes assume that the chance of a word appearing is independent of other words. But in reality, language is connected. For example, in politics, the word "white" often predicts the word "house." Ignoring these connections destroys the sentence structure that often reveals lies.

3. **Data Sparsity:** As the vocabulary M gets bigger (often over 50,000 words), the vectors become mostly zeros. In these large spaces, measuring distance (similarity) becomes difficult. This is known as the "Curse of Dimensionality." It makes it hard for models like SVMs to classify data accurately without a huge amount of training examples.

2.2 The Transformer Paradigm: Contextualized Representations

The Transformer architecture arrived in 2017. It changed NLP by using the Attention Mechanism instead of older methods. [5]

2.2.1 The Self-Attention Mechanism

Self-attention lets a model connect each word in a sentence to every other word. It calculates a weighted score that captures context. For input data X, the model learns three weight matrices W^Q, W^K, W^V. These create Query (Q), Key (K), and Value (V) representations. The attention scores are calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

Here, the dot product QK^T measures similarity between a query (the current word) and all keys (other words). The softmax function turns these scores into probabilities. If the word "bank" has a high score with "river," the model understands "bank" in the context of a river. Unlike BoW, where "bank" is always the same, in a Transformer, the meaning of "bank" changes based on the context. [6]

2.2.2 BERT and Bidirectionality

BERT improved this by using Masked Language Modeling (MLM). BERT hides random words in a sentence and tries to guess them using context from both the left and right sides at the same time. This two-way (bidirectional) reading is crucial for fake news. Lies often depend on the relationship between the start and end of a sentence.

2.3 DeBERTa: Architectural Innovations for Nuance

BERT is powerful, but it handles word position in a simple way. It adds a static position number to the word meaning. DeBERTa introduces changes designed to capture subtle language structures better. [3]

2.3.1 Disentangled Attention: Separating Content and Position

DeBERTa argues that the meaning of a word (Content) and its position (Position) are two different things. They should not be added together. Instead, DeBERTa uses two separate vectors for each token: a content vector H and a relative position vector P. The attention weight between token i and token j is calculated using separate matrices. The score has four parts:

1. Content-to-Content: How similar are the meanings of words i and j?
2. Content-to-Position: How is the meaning of i related to the position of j?
3. Position-to-Content: How is the position of i related to the meaning of j?
4. Position-to-Position: (This is often ignored because it has little information).

The formula for disentangled attention is:

$$A_{ij} = \underbrace{H_i H_j^T}_{\text{content-to-content}} + \underbrace{H_i P_{j|i}^T}_{\text{content-to-position}} + \underbrace{P_{i|j} H_j^T}_{\text{position-to-content}} \tag{4}$$

By separating these, DeBERTa distinguishes *what* a word is from *where* it is. This is critical for fake news. The "what" (emotional words) might look like real news, but the "where" (weird sentence structure) often reveals the fake.

2.3.2 Enhanced Mask Decoder (EMD)

Standard BERT models only use position information at the beginning. This information can get lost as it goes through the model layers. DeBERTa adds position information again at the end, right before the final prediction. This forces the model to check the exact position of words when making a decision. This helps detect the structural errors found in fake news.

2.4 Low-Rank Adaptation (LoRA) for Efficient Fine-Tuning

Fine-tuning a big model like DeBERTa (184 million parameters) is expensive. This study uses LoRA (Low-Rank Adaptation). This technique allows for high performance with much less computing power. [7]

LoRA assumes that the changes in weights during training have a low "rank." Instead of changing the full weight matrix W, LoRA keeps W frozen. It adds two small trainable matrices A and B. The math for the forward pass becomes:

$$h = W_0 x + \Delta W x = W_0 x + \frac{\alpha}{r}(BA)x \tag{5}$$

Here, α is a scaling factor. By training only A and B, we reduce the number of parameters by up to 10,000 times. [7] This allows us to use state-of-the-art models on normal hardware.

3 Related Works

The history of automated fake news detection has followed the general progress of machine learning.

3.1 The Era of Traditional Machine Learning

Early efforts used feature engineering. Researchers put linguistic features (like punctuation and readability) into traditional classifiers.

- **Support Vector Machines (SVM):** SVMs were the best models for text before deep learning. Studies showed that SVMs using TF-IDF vectors could separate classes well. However, they were limited by the "bag of words" problem and could not see the sequence of the story.
- **Naive Bayes:** This was a popular baseline because it is simple and fast. It calculates the probability of a document being fake based on its words. However, it assumes words are independent. This is wrong for language, so it performs poorly on complex sentences with negative words (like "not").
- **Logistic Regression:** This is a solid baseline. It gives clear coefficients, so researchers can see which words correlate with "fake" news. However, it struggles with complex relationships between words.

3.2 The Deep Learning Transition

Neural networks solved some of these problems.

- **CNNs and LSTMs:** Convolutional Neural Networks (CNNs) were used to find patterns (like key phrases). Long Short-Term Memory (LSTM) networks could process sequences and connections between words. While effective, LSTMs were slow to train and had the "vanishing gradient" problem. This made it hard for them to understand long articles. [4]
- **The Transformer Breakthrough:** The shift to Transformer models (BERT, RoBERTa) was a huge jump in performance. Literature shows that Transformers are much better than CNNs and LSTMs because they process the whole context at once. DeBERTa is currently one of the best models for understanding text. This study uses DeBERTa for the specific task of fake news detection.

4 Methodology

This section explains the experiment design, the datasets, how we prepared the data, and the model settings.

4.1 Datasets: Characteristics and Challenges

We used three different datasets to test our findings on different types of fake news.

4.1.1 ISOT Fake News Dataset

- **Origin:** Created by the ISOT research lab.
- **Composition:** About 45,000 articles from 2016-2017.
- **Source Methodology:**

  - **Fake (Label 0):** From websites flagged as unreliable. These often have "clickbait" headlines.
  - **Real (Label 1):** From Reuters.com. These have high journalistic standards.
- **Distributions:** The dataset is balanced (>20,000 samples for each class). The text length varies a lot, from short pieces to long reports. This tests if the model can handle long texts.

4.1.2 LIAR Dataset

- **Origin:** A benchmark dataset from PolitiFact.com.
- **Composition:** Contains 12,800 short statements, not full articles.
- **Labeling Complexity:** Originally 6 classes. We mapped them to two classes (Fake and Real) for this study.
- **Challenges:** The mapping creates noise because "barely-true" statements might have some facts. Also, the short text (average 18 words) gives very little context.

4.1.3 FakeNewsNet Dataset

- **Origin:** A repository with news content and social context.
- **Composition:** Data from PolitiFact (politics) and Gossip Cop (entertainment).
- **Significance:** This introduces variety. Celebrity gossip uses informal language even when true. Political news usually tries to sound formal. This tests if the model can tell the difference between "style" and "truth."
- **Sample Size:** We used about 40,000 training samples.

4.2 Data Preprocessing Pipeline

We cleaned the data in a standard way to ensure a fair comparison.

1. **Cleaning:** We removed URLs and special symbols that have no meaning.
2. **Tokenization:**
   - **For Traditional Models (BoW):** We used standard whitespace tokenization. We removed "stop words" (words like "the", "is"). This is normal for BoW, but it destroys sentence structure. This highlights why BoW is limited (e.g., "to be or not to be" disappears if you remove stop words).
   - **For DeBERTa:** We used the DeBERTa v3 tokenizer. This keeps stop words and punctuation because the attention mechanism needs them to understand grammar. It breaks unknown words into smaller parts (e.g., "unaffordability" becomes "un", "afford", "ability").

4.3 Model Architecture and Configuration

4.3.1 Baseline Models (Traditional ML)

We tuned the baselines using Grid Search to make sure they worked as well as possible.

- **Logistic Regression:** Optimized the regularization strength C.
- **Linear SVM:** Used a linear kernel and optimized C.
- **Naive Bayes:** Used the Multinomial version and tuned the smoothing parameter.

4.3.2 The Advanced Model: DeBERTa v3 + LoRA

- **Base Architecture:** DeBERTa v3 Base (184 million parameters).
- **Fine-Tuning Strategy:** We used LoRA adapters on the query, key, value, and output layers.
- **Optimization:** We used Bayesian optimization (Optuna) to find the best settings:
  - Learning Rate: $5e-5$ to $5e-4$.
  - LoRA Alpha and Rank.
- **Training Dynamics:** We used the AdamW optimizer. We used Early Stopping to prevent the model from memorizing the data (overfitting).

4.4 Experimental Environment

Experiments were done on Google Colab Pro+ with an NVIDIA A100 GPU. This hardware was necessary for the complex matrix math in the Transformer model.

5 Results and Analysis

This section shows the results. We look at Accuracy and F1-Score. F1-Score is important because missing a fake story (false negative) is dangerous.

5.1 Quantitative Performance

Table 1 shows the performance on the ISOT dataset.

Table 1: Comparative Performance on ISOT Dataset

Models	Accuracy	F1-Score
DeBERTa v3 + LoRA	**0.9226**	**0.9213**
Linear SVM	0.9058	0.9036
Logistic Regression	0.9012	0.8997
Naive Bayes	0.8619	0.8598

5.1.1 Analysis of Traditional Baselines

- **Naive Bayes (86.19%):** As expected, Naive Bayes was the worst. Its assumption that words are independent does not work for deceptive text. It likely failed on sentences with negations.
- **Logistic Regression (90.12%) & SVM (90.58%):** These models performed surprisingly well. This suggests the ISOT dataset has "lexical leakage." This means specific words correlate strongly with the label (e.g., "breaking news" in fake articles vs. "Reuters" in real ones). The SVM successfully found these specific word features.

5.1.2 Analysis of DeBERTa Performance

- **DeBERTa v3 + LoRA (92.26%):** The Transformer model was the best. It beat the SVM by about 1.7%. This might look small, but at 90% accuracy, it means reducing the error rate by nearly 20%.

- **F1-Score Alignment:** The F1-score is close to the accuracy. This means the model is robust and balances precision and recall effectively.

5.2 Qualitative Error Analysis: Deciphering the "Black Box"

To understand why DeBERTa was better, we looked at specific examples where it was correct.

Case Study 1: The "Reuters" Style (Real News)

- **Text:** "Niger delta leader calls on avengers to hold..."
- **Prediction:** Real (Correct).
- **Analysis:** This headline is complex and neutral. It uses a specific grammatical structure. DeBERTa uses position embeddings to understand this "Subject-Verb-Object" pattern. A BoW model sees words, but DeBERTa sees the journalistic style.

Case Study 2: The Emotional Trigger (Fake News)

- **Text:** "CBS reporter assaulted by cops at cancelled..."
- **Prediction:** Fake (Correct).
- **Analysis:** This headline tries to create an emotional reaction. Fake news often uses violent or emotional words. DeBERTa understood the connection between "assaulted" and "cops."

Case Study 3: The Clickbait Structure (Fake News)

- **Text:** "Terminally ill former miss wi: 'until my last..."
- **Prediction:** Fake (Correct).
- **Analysis:** This uses a quote and a tragic topic. This is typical of clickbait. DeBERTa recognized the structure: "Introductory Clause + Emotional Quote." Traditional models miss this structure completely.

6 Discussion: The Implications of Contextual Intelligence

6.1 The Disentangled Advantage

The results show that DeBERTa's disentangled attention is superior. By separating content from position, the model can tell the difference between real news patterns and fake ones. For example, fake news often puts sensational adjectives before names. Real news uses adjectives differently. DeBERTa learns these subtle position patterns, which Bag of Words models cannot see.

6.2 LoRA and Efficiency

A key finding is that LoRA works very well. We fine-tuned a huge model with very little computing cost. Because we only trained the small matrices (A and B) and got great results, it implies the pre-trained model already "knew" a lot. We just had to guide it to look for fake news patterns. This means small organizations can use powerful AI tools. [7]

6.3 Beyond the "Bag of Words" Ceiling

There is a limit to how good BoW models can be. No matter how much we tune an SVM, it cannot recover the lost information (word order). To improve fake news detection, we must stop using these old methods and use architectures that understand sequences-like Transformers.

7 Limitations and Future Work

7.1 Limitations

1. **Text Only:** The model only looks at text. Modern fake news uses images and videos too.
2. **Label Noise:** Reducing the LIAR dataset to just two classes (Real/Fake) creates confusion for the model.
3. **Static Knowledge:** DeBERTa does not know about events that happened after it was trained. It might fail to check breaking news.

7.2 Future Work

1. **Multimodal Fusion:** Combine DeBERTa with image models to check text and images together.
2. **Continual Learning:** Update the model with new data so it does not forget old information.
3. **Explainability:** Develop ways to visualize what the model is looking at, to build trust.

8 Conclusion

The massive amount of digital information has created a Verification Gap. This study showed that traditional machine learning models are not good enough to fix this. They fail because they use the Bag of Words assumption, which ignores context and structure.

Our experiments proved that Transformer-based models, specifically DeBERTa v3 with LoRA, are a better solution. By disentangling content and position, DeBERTa reached 92.26% accuracy on the custom dataset. The success of Low-Rank Adaptation proves we can use these advanced models efficiently.

Moving from counting words to understanding context is necessary for the "post-truth" era. Fake news writers are copying the style of real journalism. Therefore, our detection systems must look deeper, using the contextual intelligence of Transformers to find the truth.

References

[1] IBM. "What is bag of words?" IBM Topics. [Online]. Available: https://www.ibm.com/think/topics/bag-of-words.

[2] S. Qaiser and R. Ali. "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents." *International Journal of Computer Applications*, vol. 181, no. 1, 2018.

[3] P. He, X. Liu, J. Gao, and W. Chen. "DeBERTa: Decoding-enhanced BERT with Disentangled Attention." *ICLR*, 2021.

[4] J. A. Nasir, O. S. Khan, and I. Varlamis. "Fake News Detection: A Hybrid CNN-RNN Based Deep Learning Approach." *International Journal of Information Management Data Insights*, vol. 1, no. 1, 2021.

[5] A. Vaswani et al. "Attention Is All You Need." *Advances in Neural Information Processing Systems*, 2017. (Contextual citation based on general Transformer theory).

[6] S. H. Hawley, "To Understand Transformers, Focus on Attention," *Dr. Scott Hawley's Blog*, Aug. 21, 2023. [Online]. Available: https://drscotthawley.github.io/blog/posts/Transformers1-Attention.html.

[7] A. Agacdelen and Y. K. Bagci. "LoRA from Scratch: Efficient RoBERTa Fine-Tuning." GitHub Repository, 2025. [Online]. Available: https://github.com/ArdaAgacdelen/lora.