ChatGPT

Transformer-based Models (BERT and Beyond)

Transformer architectures (like BERT) use self-attention to capture rich context across a text. BERT (Bidirectional Encoder Representations from Transformers) is a deep, bidirectional transformer model that encodes entire sentences using multi-head attention [1]. In practice, BERT and its variants (RoBERTa, DistilBERT, etc.) are *fine-tuned* on labeled datasets for fake-news or clickbait classification. For example, one study fine-tuned BERT-base and RoBERTa-base on a fake-news dataset and reported high precision/recall: RoBERTa-base achieved 89.23% precision and 90.14% recall (F1 ≈ 89.68%), outperforming BERT-base (F1 ≈ 85.22%) and DistilBERT (F1 ≈ 84.08%) [2]. Generally, transformer models outperform older models on these tasks [3] [4]. Transformer-based classifiers also excel at clickbait detection: Liu et al. note that *fine-tuning* pretrained language models (PLMs) like BERT "has shown superiority in clickbait detection tasks" compared to feature-based methods [5]. (Fine-tuning here means adding a simple output layer on BERT and training on a labeled clickbait dataset.) An overview of performance on one fake-news dataset is shown below:

Model (fine-tuned)	Precision	Recall	F1-score
BERT-base (uncased)	84.12%	86.34%	85.22% [2]
DistilBERT-base (uncased)	84.10%	84.05%	84.08% [2]
RoBERTa-base (uncased)	89.23%	90.14%	89.68% [2]

These results (on a GPT-annotated fake-news test set) show that RoBERTa, a large encoder-only transformer, yielded the best classification metrics [2]. In general, encoder-only transformers like BERT or RoBERTa (with ~100 million parameters) can be fine-tuned with relatively few epochs to achieve strong fake-news detection accuracy [3]. In the clickbait domain, even zero-shot or few-shot large language models (LLMs) are being tested, but fine-tuned PLMs still lead. Wang et al. found that GPT-3.5/ChatGPT gave *stable multilingual* clickbait detection performance, yet zero-shot LLMs underperformed specialized PLMs on title-based clickbait classification [6] [7]. In fact, the first investigation of LLMs for clickbait showed they lag behind fine-tuned models (even though ChatGPT handles multiple languages) [6] [7]. Open-source codebases (often on GitHub) use HuggingFace's Transformers library and PyTorch/TensorFlow backends for fine-tuning these models. For example, a recent clickbait-detection project provides its full code via a GitHub repo [8]. In summary, transformer-based classifiers (especially BERT variants) are among the most effective deep models for fake-news and clickbait tasks [3] [5].

Word Embeddings (Word2Vec, BERT Embeddings, etc.)

Text inputs must be converted to numerical vectors. **Static embeddings** like Word2Vec or GloVe assign each word a fixed vector based on co-occurrence statistics. These capture general semantic similarity but cannot distinguish different uses of the same word. By contrast, **contextual embeddings** (from models like

BERT) produce word vectors that vary with the sentence context. Research shows contextual embeddings usually improve detection. For instance, a study on clickbait in Urdu compared LSTM classifiers using Word2Vec vs. contextual "sentence embeddings." LSTM with Word2Vec achieved ~77% accuracy, but using contextual sentence embeddings raised accuracy to ~85% [9]. This is because richer embeddings capture more syntax/semantics: "sentence embeddings far outperform traditional word embeddings such as Word2Vec and GloVe... They keep more contextual and syntactic relationships" [10].

Simple ML/DL pipelines may use combinations: e.g., TF-IDF or FastText with CNN/RNN. An empirical study on a large mixed news dataset (TruthSeeker) compared TF-IDF, Word2Vec, and FastText embeddings with various models. They found TF-IDF often led to the best performance with models like SVM, but Word2Vec also boosted performance for deep learners. In that work, a CNN with Word2Vec embedding achieved ~94.9% accuracy [11] (see below), while CNNs with TF-IDF got ~99%. Their key conclusion was that *complex embeddings suit deep models*, whereas simpler bag-of-words features (like TF-IDF) worked well with SVM/MLP [12] [13].

- **Word2Vec/FastText:** Static, lower-dimensional vectors from a fixed corpus. Useful as features for RNNs/CNNs or even classic ML. In practice, models using Word2Vec can still reach high accuracy (~94–95% in one CNN experiment [11]).
- **Contextual embeddings (BERT):** Provide dynamic vectors. Fine-tuned BERT embeddings yield richer features. For fake-news, this often means higher precision/recall than static embeddings.
- **Comparative results:** Contextual/sentence embeddings have consistently beaten Word2Vec/GloVe in experiments [9] [10]. For example, replacing Word2Vec with transformer-generated sentence embeddings in an LSTM boosted accuracy from 77% to 85% [9]. Another study notes that adding FastText subword info may introduce noise for some models [12], illustrating that the *choice of embedding* should match the model type: deep networks leverage semantic embeddings, while simpler models handle sparse features.

Sequence Models (LSTM, GRU, BiLSTM)

Recurrent neural networks (RNNs) like LSTMs and GRUs process text in sequence, capturing word order and dependencies. LSTM networks use gates to remember long-range context; GRUs are a streamlined variant. BiLSTMs (bidirectional LSTMs) read text in both directions for fuller context. These models are frequently applied to fake-news and clickbait detection. For example, Padalko *et al.* trained BiLSTM and attention-based BiLSTM models on a Kaggle fake-news dataset; their *attention-BiLSTM* achieved **97.66% accuracy** (and similarly high F1) [14]. In other words, a bidirectional LSTM with learned attention over words can classify fake vs. real news with very high accuracy. In side-by-side comparisons, BiLSTMs typically outperform unidirectional LSTMs/GRUs because they capture both preceding and following context [15] [16].

Despite this power, studies repeatedly show that transformer models often beat RNNs. For instance, a comparative analysis found that a fine-tuned BERT model achieved ~99% accuracy on a fake-news test, substantially higher than LSTM or GRU baselines [17]. Nonetheless, RNNs remain useful especially when compute is limited. GRU-based systems have achieved good performance with fewer parameters. Hybrids (e.g. CNN+GRU) have also been proposed to leverage CNN feature extraction with sequential modeling. Overall, RNNs (LSTM/GRU/BiLSTM) have been **widely used** (often with additional attention layers) and can attain very high recall for fake-news (i.e., catching most fakes) [18] [16], but are typically outperformed by large pre-trained transformers on standard benchmarks [17] [19].

Attention Mechanisms

Attention modules let a model weigh the importance of different words or segments. In the context of fake-news or clickbait detection, two main uses appear:

- **RNN Attention:** Applying an attention layer on top of an LSTM/BiLSTM lets the model "focus" on key tokens. The earlier example (97.66% accuracy) involved an *attention-based BiLSTM*, where the network learned to highlight crucial words in news titles [14]. Attention can significantly boost RNN performance in text classification, as it helps capture salient clues.

- **Self-Attention in Transformers:** Transformers are built on multi-head self-attention, inherently capturing inter-word relations. Research confirms that multi-head attention is effective for these tasks: one study notes that contextual embeddings combined with multi-head attention "capture complex textual patterns" that simpler models miss [20]. Indeed, transformer-based detectors implicitly use attention to weigh all parts of a document against each other.

In practice, adding attention to an RNN often yields state-of-art results for sequence models [14], while transformer architectures rely entirely on stacked attention heads. Attention mechanisms in either case improve the model's ability to identify the subtle, long-range relationships (e.g. contradictory phrases) that often signal fake or clickbait content.

Convolutional Neural Networks (CNNs) for Text

CNNs, though originally designed for images, have been successfully applied to text classification. A text CNN typically slides filters over word embeddings (or n-gram character embeddings), learning local features. In fake-news detection, CNNs (and CNN+RNN hybrids) are popular for efficiently capturing key phrases. For example, a convolutional model called "C-CNN" achieved **99.90% accuracy** on the Kaggle Fake News dataset, and a CNN-LSTM hybrid got 96% on the Fake News Challenge dataset [21]. Similarly, in a CNN study on the TruthSeeker dataset, a three-layer CNN with TF-IDF embeddings achieved ~99% accuracy [11].

CNNs excel at extracting salient n-grams (like "Donald Trump coronavirus video"), and when combined with pooling they create compact features. They are also faster to train than deep RNNs on long texts. However, CNNs typically lack global context: they may miss long-range dependencies. Hence, many systems augment CNNs with RNN or attention (e.g. CNN-BiLSTM). In evaluations, modern studies often find that full transformer models (with self-attention) still edge out pure CNN or CNN-RNN hybrids [3] [21]. Nonetheless, CNNs remain a competitive, lightweight option for sentence- or document-level classification in fake-news pipelines.

Key Datasets for Fake News and Clickbait

Deep-learning models require labeled data. Common **fake-news datasets** include: ISOT's FakeNews dataset (often on Kaggle as "fake-and-real-news", ~44.9K English news articles) [22]; LIAR (13K short political statements, 6-level labels) [23]; the FakeNewsNet collection (100K+ articles from PolitiFact and BuzzFeed) [23]; and COVID-19 misinformation corpora (e.g. a CodaLab COVID-19 fake-news challenge). The **Fake News**

Challenge on Kaggle provided ~21K labeled news articles (binary real/fake) [24]. Many of these datasets use *distant supervision* (labeling by fact-check sites) or crowd annotations [23], and languages are mostly English.

For **clickbait detection**, key datasets include the *Webis Clickbait Corpus 2017* (a.k.a. Clickbait Challenge 2017) with 38,517 annotated tweets from news outlets [25]. More recent shared tasks (SemEval, PAN) and Kaggle collections (news headlines labeled clickbait/not) provide thousands of English headlines. In multilingual contexts, smaller datasets have been created (e.g. Urdu clickbait headlines) [26].

Dataset (example)	Type	Size (approx.)	Language	Notes
ISOT Fake News (Kaggle)	Fake (binary)	~44,900 news articles [22]	English	News titles + bodies
LIAR (PolitiFact statements)	Fake (6-way)	~12,800 statements [23]	English	Short political claims
Kaggle FakeNews dataset	Fake (binary)	~21,000 news articles [24]	English	Kaggle competition data
Webis-Clickbait-17 (Twitter)	Clickbait (binary)	38,517 tweets [25]	English	News tweets (with headlines)
Kaggle Clickbait Headlines	Clickbait (binary)	~32,000 headlines	English	Various news sources

In practice, models are often fine-tuned on a combination of these or on domain-specific corpora. For example, a model might be pre-trained on large news corpora, then fine-tuned on a local set of labeled fake news or clickbait. Importantly, new datasets can be built for regional languages or topics as needed (see *multilingual* below).

Evaluation Metrics

Fake-news and clickbait detection are formulated as classification tasks, so standard metrics apply: **Accuracy** (overall fraction correct), **Precision** (of predicted fakes, fraction that are actually fake), **Recall** (of true fakes, fraction that are detected), and **F1-score** (harmonic mean of precision and recall). Studies routinely report all four [27]. For example, Padalko *et al.* evaluated their attention-BiLSTM model using Recall, Precision, F1, and Accuracy on a held-out set [27]. These metrics can be computed per class or as macro-averages in multiclass scenarios. High recall is often important (catching most fakes) while precision indicates reliability. In comparisons, BERT-like models have achieved both high precision and recall (leading to top F1) on benchmark datasets [2] [3].

Typically, research splits data into train/validation/test (e.g. 80/10/10%) and uses cross-validation for robustness. Table-based reports (like in [28]) show percentage Precision/Recall/F1. In practice, an F1-score above ~0.80 is common for recent transformer methods on standard fake-news tasks.

Tools, Multilingual Adaptation, and Code Availability

Most of the above methods use open-source frameworks. In particular, the Hugging Face *Transformers* library provides easy fine-tuning of BERT/XLM-R/DistilBERT and so on, often under PyTorch or TensorFlow. Attention-augmented RNNs and CNNs can be implemented with Keras or PyTorch (custom layers). Researchers frequently publish code: for example, the ChatGPT-for-Clickbait paper makes its code publicly available [8].

Crucially, many models support multiple languages. Pretrained multilingual transformers (mBERT, XLM-Roberta) and regional BERTs (e.g. Indic-BERT, Malayalam-BERT) allow training on non-English news. For instance, one study fine-tuned **XLM-R**, **mBERT**, **Indic-BERT**, and a Malayalam-specific BERT on a Malayalam fake-news dataset [29]. They found that the domain-specific Malayalam-BERT achieved the highest macro F1 (~0.86) on that task [19], even outperforming generic multilingual models. This demonstrates that the same deep architectures can be adapted: by replacing the pretrained base with a multilingual or language-specific model, teams can train on local datasets in Hindi, Urdu, Chinese, etc., to capture regional vocabulary and style.

In summary, the deep-learning toolkit for fake-news and clickbait detection is well-established and largely open-source. Modern approaches center on pretrained transformers (fine-tuned for the target domain) supplemented by powerful embeddings. Sequence (LSTM/GRU) and CNN models (often with attention) are also effective, especially when resources or data are limited. Across studies from 2021–2025, the trend is clear: contextual embeddings and attention-based architectures yield the most accurate and robust classifiers [3] [5]. By leveraging these models and the available datasets, researchers can build multilingual, adaptable detection systems for new fake-news or clickbait datasets.

Sources: Recent research articles and implementations (2021–2025) as cited above [2] [10] [11] [4] [27] [25] [19].

---

[1] [22] GitHub - wutonytt/Fake-News-Detection: Fake News Detection using a BERT-based Deep Learning Approach
https://github.com/wutonytt/Fake-News-Detection

[2] [3] [23] [28] Fake News Detection: Comparative Evaluation of BERT-like Models and Large Language Models with Generative AI-Annotated Data
https://arxiv.org/html/2412.14276v2

[4] [15] [16] [17] [20] diva-portal.org
http://www.diva-portal.org/smash/get/diva2:1980396/FULLTEXT01.pdf

[5] [6] [7] [8] Clickbait Detection via Large Language Models
https://arxiv.org/html/2306.09597v3

9 10 18 26 Deep learning and sentence embeddings for detection of clickbait news from online content | Scientific Reports
https://www.nature.com/articles/s41598-025-97576-1?error=cookies_not_supported&code=96f1a991-783a-4fc3-b889-1d995b48704c

11 12 13 Enhancing Fake News Detection with Word Embedding: A Machine Learning and Deep Learning Approach
https://www.mdpi.com/2073-431X/13/9/239

14 27 A novel approach to fake news classification using LSTM-based deep learning models - PubMed
https://pubmed.ncbi.nlm.nih.gov/38260054/

19 29 aclanthology.org
https://aclanthology.org/2024.dravidianlangtech-1.30.pdf

21 24 It’s All in the Embedding! Fake News Detection Using Document Embeddings
https://www.mdpi.com/2227-7390/11/3/508

25 Webis Data Webis-Clickbait-17
https://webis.de/data/webis-clickbait-17.html