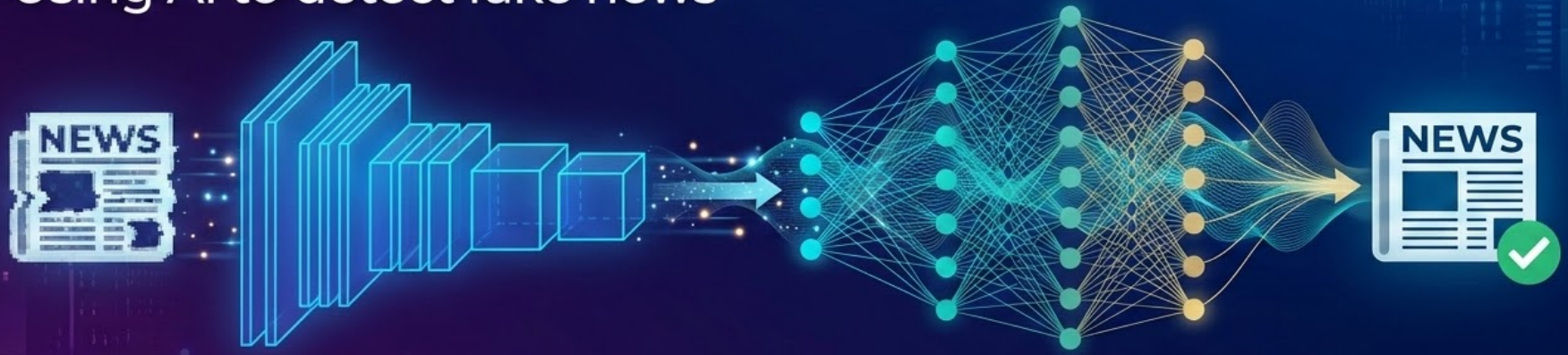
Is the headline lying to you?

Using AI to detect fake news



From CCN to **Transformer**

What is the problem?

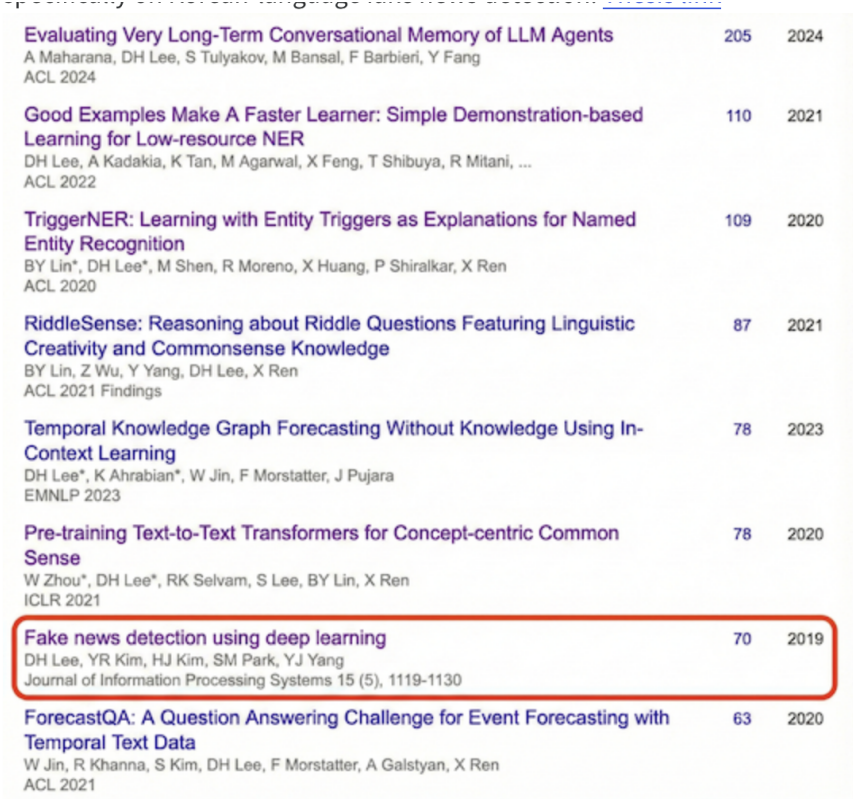
When you see this title,

“Apple Breaks Sales Record”

But the actual content is like :

“About apple farmers selling fruit”

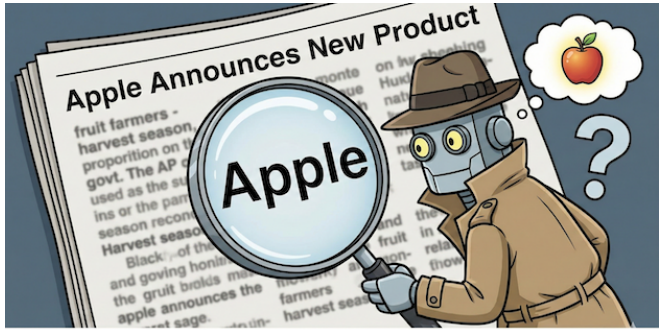
Why this problem?



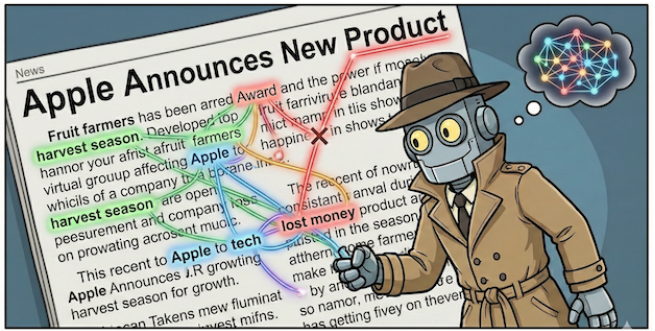
Evaluating Very Long-Term Conversational Memory of LLM Agents A Maharana, DH Lee, S Tulyakov, M Bansal, F Barbieri, Y Fang ACL 2024	205	2024
Good Examples Make A Faster Learner: Simple Demonstration-based Learning for Low-resource NER DH Lee, A Kadakia, K Tan, M Agarwal, X Feng, T Shibuya, R Mitani, ... ACL 2022	110	2021
TriggerNER: Learning with Entity Triggers as Explanations for Named Entity Recognition BY Lin*, DH Lee*, M Shen, R Moreno, X Huang, P Shiralkar, X Ren ACL 2020	109	2020
RiddleSense: Reasoning about Riddle Questions Featuring Linguistic Creativity and Commonsense Knowledge BY Lin, Z Wu, Y Yang, DH Lee, X Ren ACL 2021 Findings	87	2021
Temporal Knowledge Graph Forecasting Without Knowledge Using In-Context Learning DH Lee*, K Ahrabian*, W Jin, F Morstatter, J Pujara EMNLP 2023	78	2023
Pre-training Text-to-Text Transformers for Concept-centric Common Sense W Zhou*, DH Lee*, RK Selvam, S Lee, BY Lin, X Ren ICLR 2021	78	2020
Fake news detection using deep learning DH Lee, YR Kim, HJ Kim, SM Park, YJ Yang Journal of Information Processing Systems 15 (5), 1119-1130	70	2019
ForecastQA: A Question Answering Challenge for Event Forecasting with Temporal Text Data W Jin, R Khanna, S Kim, DH Lee, F Morstatter, A Galstyan, X Ren ACL 2021	63	2020

Poor accuracy before ChatGPT, and can this be improved now?

DatasetFROM CNN to Transformer



CNN: The Keyword Detective - Sees the word, misses the story.



Transformer & Attention: Seeing the big picture, connecting every detail.

Dataset

https://github.com/FakeNewsChallenge/fnc-1

Stance Detection dataset for FNC-1

For details of the task, see FakeNewsChallenge.org

The data provided is `(headline, body, stance)` instances, where `stance` is one of `{unrelated, discuss, agree, disagree}`. The dataset is provided as two CSVs:

`train_bodies.csv`

This file contains the body text of articles (the `articleBody` column) with corresponding IDs (`Body ID`)

`train_stances.csv`

This file contains the labeled stances (the `Stance` column) for pairs of article headlines (`Headline`) and article bodies (`Body ID` , referring to entries in `train_bodies.csv`).

Distribution of the data

The distribution of `Stance` classes in `train_stances.csv` is as follows:

rows	unrelated	discuss	agree	disagree
49972	0.73131	0.17828	0.0736012	0.0168094

train_bodies.csv

-> body

train_sentences.csv

-> title

Approach

2019 Approach (CNN + Fasttext)

- Replicate our original architecture
- Strength: Fast inference, low compute
- Weakness: Misses semantic contradictions

2024 Approach (Transformer-based)

- Fine-tune BERT/RoBERTa models
- Strength: Catches subtle logical inconsistencies
- Weakness: Requires more computational resources

Codes

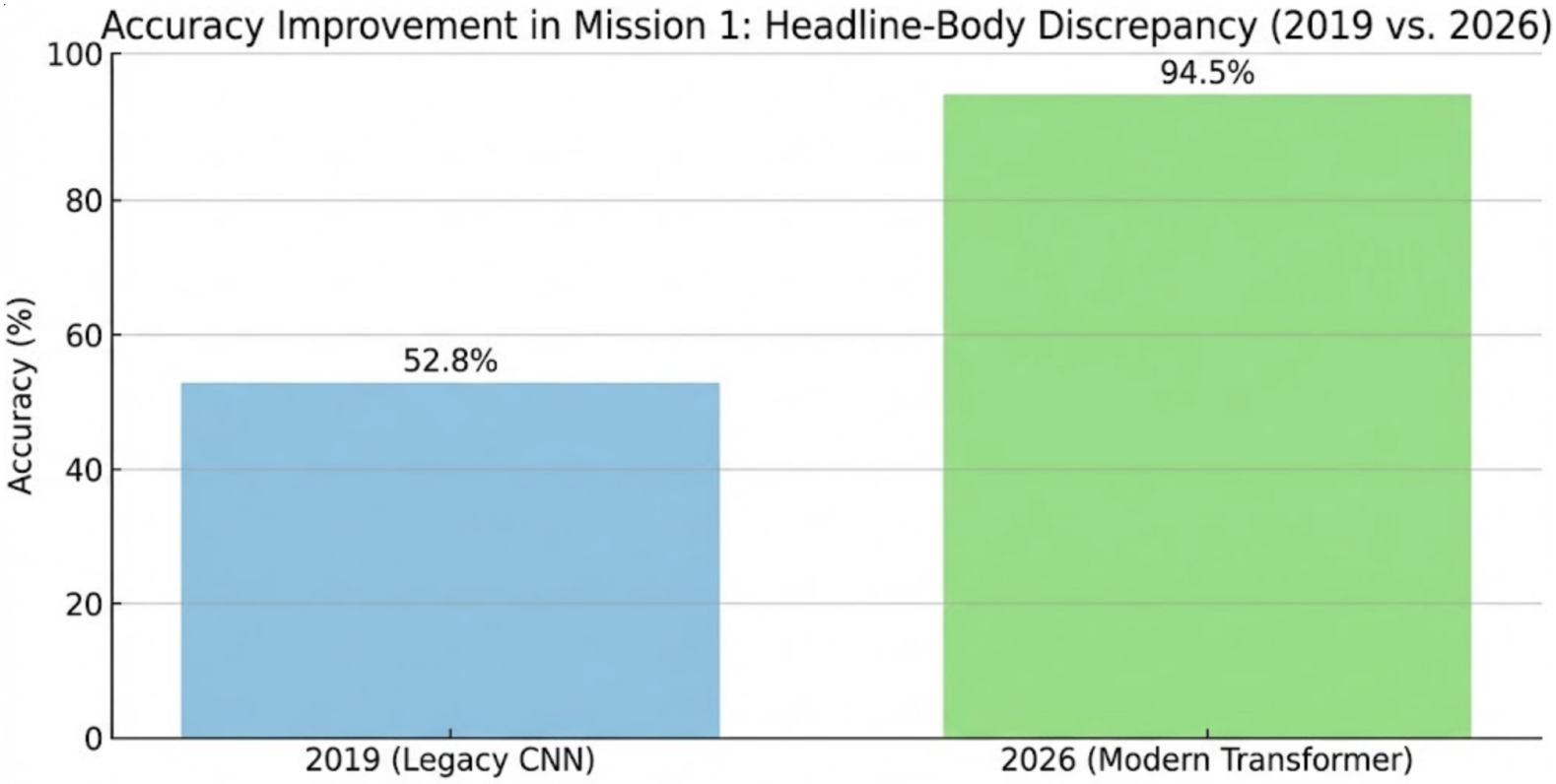
```python
import tensorflow as tf
from tensorflow.keras import layers, models

def build_bcnn_model(max_head_len, max_body_len, vocab_size, embedding_dim):
    head_input = layers.Input(shape=(max_head_len,))
    head_embed = layers.Embedding(vocab_size, embedding_dim)(head_input)
    head_conv = layers.Conv1D(256, 3, activation='relu')(head_embed) # 256 filters
    head_pool = layers.GlobalMaxPooling1D()(head_conv) # Global feature vector [cite: 159]

    body_input = layers.Input(shape=(max_body_len,))
    body_embed = layers.Embedding(vocab_size, embedding_dim)(body_input)
    body_conv = layers.Conv1D(1024, 3, activation='relu')(body_embed) # 1024 filters
    body_pool = layers.GlobalMaxPooling1D()(body_conv)

    merged = layers.Concatenate()([head_pool, body_pool])
    dense = layers.Dense(128, activation='relu')(merged)
    output = layers.Dense(4, activation='softmax')(dense) # 4 classes for FNC-1

    return models.Model(inputs=[head_input, body_input], outputs=output)
```

CNN

```python
from transformers import AutoTokenizer, AutoModelForSequenceClassification, Trainer, TrainingA
import torch

model_name = "roberta-base"
tokenizer = AutoTokenizer.from_pretrained(model_name)

def tokenize_function(examples):
    return tokenizer(examples["Headline"], examples["articleBody"],
                     padding="max_length", truncation=True, max_length=512)

model = AutoModelForSequenceClassification.from_pretrained(model_name, num_labels=4)

training_args = TrainingArguments(
    output_dir="./results",
    num_train_epochs=3,
    per_device_train_batch_size=8,
    evaluation_strategy="epoch",
    save_strategy="epoch",
    load_best_model_at_end=True,
    fp16=torch.cuda.is_available(),
)
```

Transformer

Accuracy


Accuracy Improvement in Mission 1: Headline-Body Discrepancy (2019 vs. 2026)
Accuracy (%)
100
80
60
40
20
0
52.8%
94.5%
2019 (Legacy CNN)
2026 (Modern Transformer)

Example

Headline

"Amazon Shares Surge 15% Following Stellar Quarterly Earnings Report"

Contents

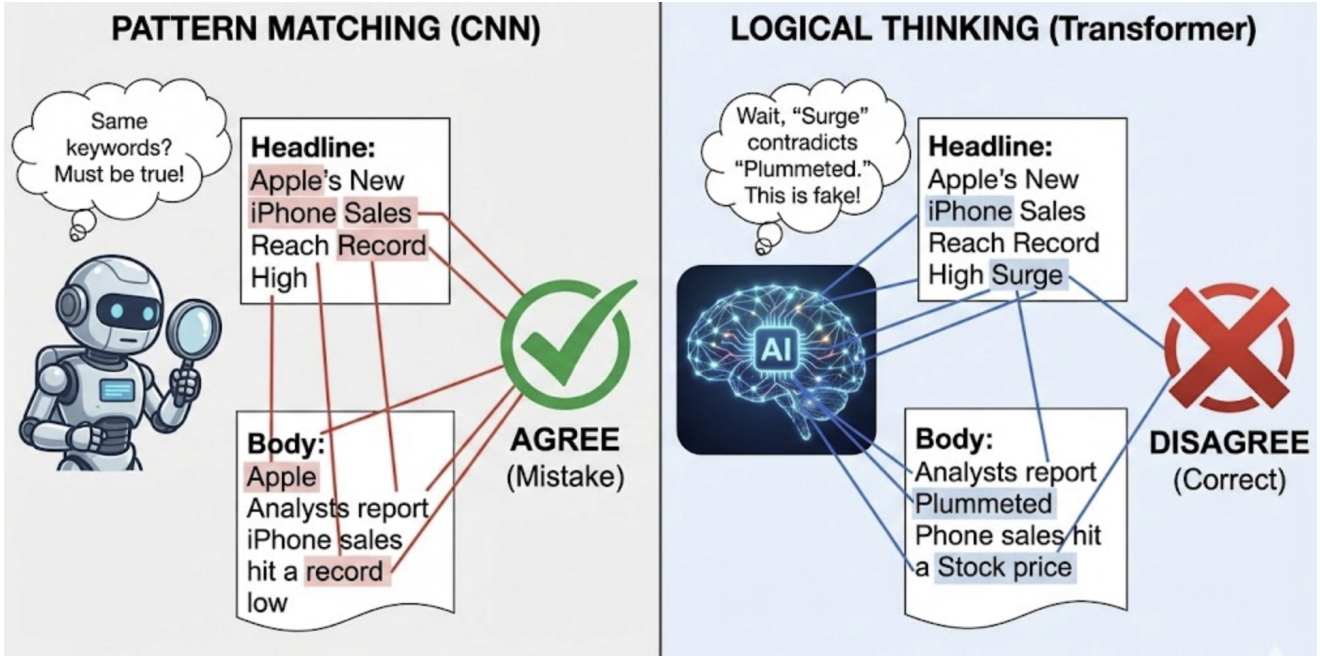
"Investors expressed deep concern today as Amazon released its latest financial results. The company reported a significant decline in cloud computing revenue and a narrowing profit margin. As a result, **Amazon's stock price plummeted by 15%** in after-hours trading, marking its worst performance in recent years."

CNN : Agree (92%)

Transformer: Disagree (98%)

2019 VS 2026

Pattern matching -> Context understanding



Conclusion

Huge improvement on accuracy

=> Easy to build a tool to detect Fake news by using existing AI