Multi-Class News Topic Classification: A Comparative Analysis of Preprocessing, Word Representations, and Sequence Models

Imtiaz Hossain
ID: 23101137
BRAC University
Dhaka, Bangladesh

Md Saidul Islam Apu
ID: 21301668
BRAC University
Dhaka, Bangladesh

***Abstract*—We present a comparative study of multi-class news topic classification on a four-class corpus (Science & Technology, Business, Sports, World News) consisting of 102,002 training and 12,000 test headlines. We evaluate nine model–representation combinations across three preprocessing variants (none, extreme, optimum), giving 27 controlled experiments. The pipeline pairs TF–IDF with Logistic Regression and a 4-layer Deep Neural Network, six recurrent variants (RNN, GRU, LSTM and their bidirectional counterparts) over locally-trained Skip-gram embeddings, and BERT-Base. The single best run is BERT-Base on the *none*-preprocessed variant (macro-F1 0.9376); among non-transformer architectures, Bidirectional GRU with our *optimum* preprocessing achieves macro-F1 0.9214 in 19 seconds of training – two orders of magnitude faster than BERT for a 0.016 point F1 trade-off. We highlight that BERT's WordPiece tokenizer handles HTML wrappers gracefully and is hurt by Porter stemming, inverting the preprocessing recommendation that holds for every other architecture in the study.**

***Index Terms*—Text classification, TF–IDF, Skip-gram, LSTM, BERT, NLP**

I. Introduction

News-topic classification is a canonical multi-class NLP problem that underpins recommender systems, content routing, and indexing pipelines. While Transformer encoders dominate the recent literature, classical bag-of-words pipelines and recurrent networks remain competitive on short-text inputs, where context windows are small and pretraining signal is dilute [1], [2]. This work systematically compares both ends of the spectrum on the assigned headline corpus.

The corpus contains 102,002 training and 12,000 test headlines distributed across four topics (Science & Technology, Business, Sports, World News). Headlines arrive wrapped in HTML and exhibit a 3.4× class imbalance in the training split, both of which directly shape our preprocessing decisions. After HTML stripping, the median headline is 39 tokens long, allowing a uniform `max_sequence_length=80` for all sequence models – favorable for both training cost and BERT fit on the 8 GB RTX 3070.

The contributions of this paper are threefold. First, we run an extensive Exploratory Data Analysis (EDA) to motivate three concrete preprocessing pipelines (*none*, *extreme*, *optimum*). Second, we benchmark nine models – one classical (Logistic Regression), one feed-forward (Deep NN), six recurrent (RNN/GRU/LSTM and bidirectional variants), and one Transformer (BERT-Base) – across all three preprocessing variants under identical hyperparameter-tuning and class-weighting protocols, yielding 27 controlled experiments. Third, we report a model×preprocessing comparison that isolates the marginal contribution of each design decision.

II. Methodology

A. Dataset and EDA

The corpus contains four balanced topic classes in the test set (3,000 each) but is moderately imbalanced in training (Sci&Tech: 43,961; Business: 26,834; Sports: 18,287; World News: 12,920). Class imbalance is handled with `class_weight='balanced'` for Logistic Regression and a class-weight tensor in the PyTorch cross-entropy loss for all neural models.

After HTML stripping, headline length is tightly distributed (mean 40 words, 95th-percentile 57). We therefore set `max_sequence_length=80` for all sequence models. Every headline carries identical HTML wrappers (`<html><body>...<br>`), motivating their removal in the optimum and extreme variants. Class-discriminative tokens emerge clearly per class: *game/team/season* for Sports; *percent/oil/stocks* for Business; *software/microsoft/internet* for Sci&Tech; *iraq/government/minister* for World News.

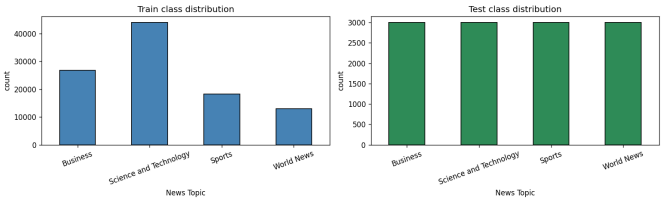


Fig. 1: Class distribution: training set (left) is imbalanced; test set (right) is balanced. The 3.4× skew motivates per-class weighting in every model.

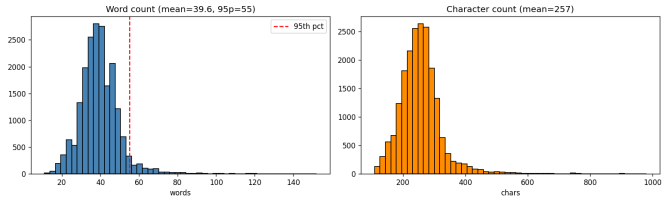


Fig. 2: Headline word-count and character-count distributions on a 20K random sample. The 95th-percentile word count is 57, motivating `max_sequence_length=80` for every sequence model.

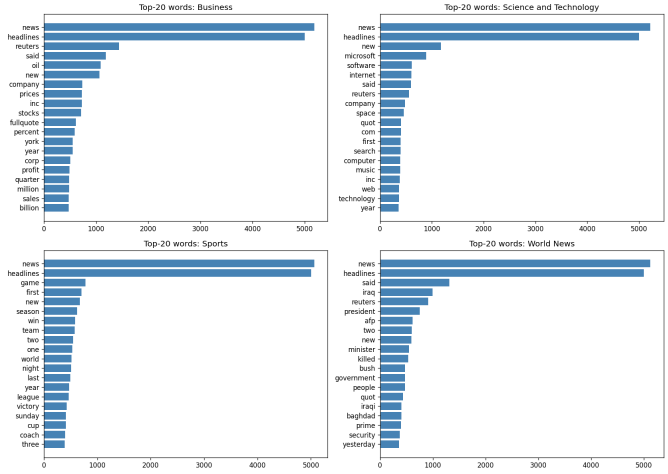


Fig. 3: Top-20 most frequent tokens per class (after a basic clean). Class-distinctive vocabulary is already evident in the unigram statistics, foreshadowing the strong baseline of TF–IDF.

Fig. 4: Per-class word clouds after HTML stripping reveal class-discriminative vocabulary – *Sports* and *Sci&Tech* are the most lexically distinct, while *Business* and *World News* share macro-economic and political tokens.

B. Preprocessing

Three pipelines are constructed:

- **None**: raw text passed through unchanged (HTML included). Establishes a worst-case baseline.
- **Extreme**: lowercasing, HTML stripping, URL/digit/punctuation removal, NLTK stopword removal, and Porter stemming.
- **Optimum** (chosen from EDA): lowercasing, HTML stripping, URL collapsing, light stopword removal that preserves negations (*not, never, no*), and WordNet lemmatization. Digits are kept because financial and sports headlines depend on numeric tokens.

C. Word Representations

TF–IDF. We fit a separate scikit-learn [7] `TfidfVectorizer` per preprocessing variant with `max_features=30000`, `ngram_range=(1,2)`, `min_df=3`, and sublinear term frequency.

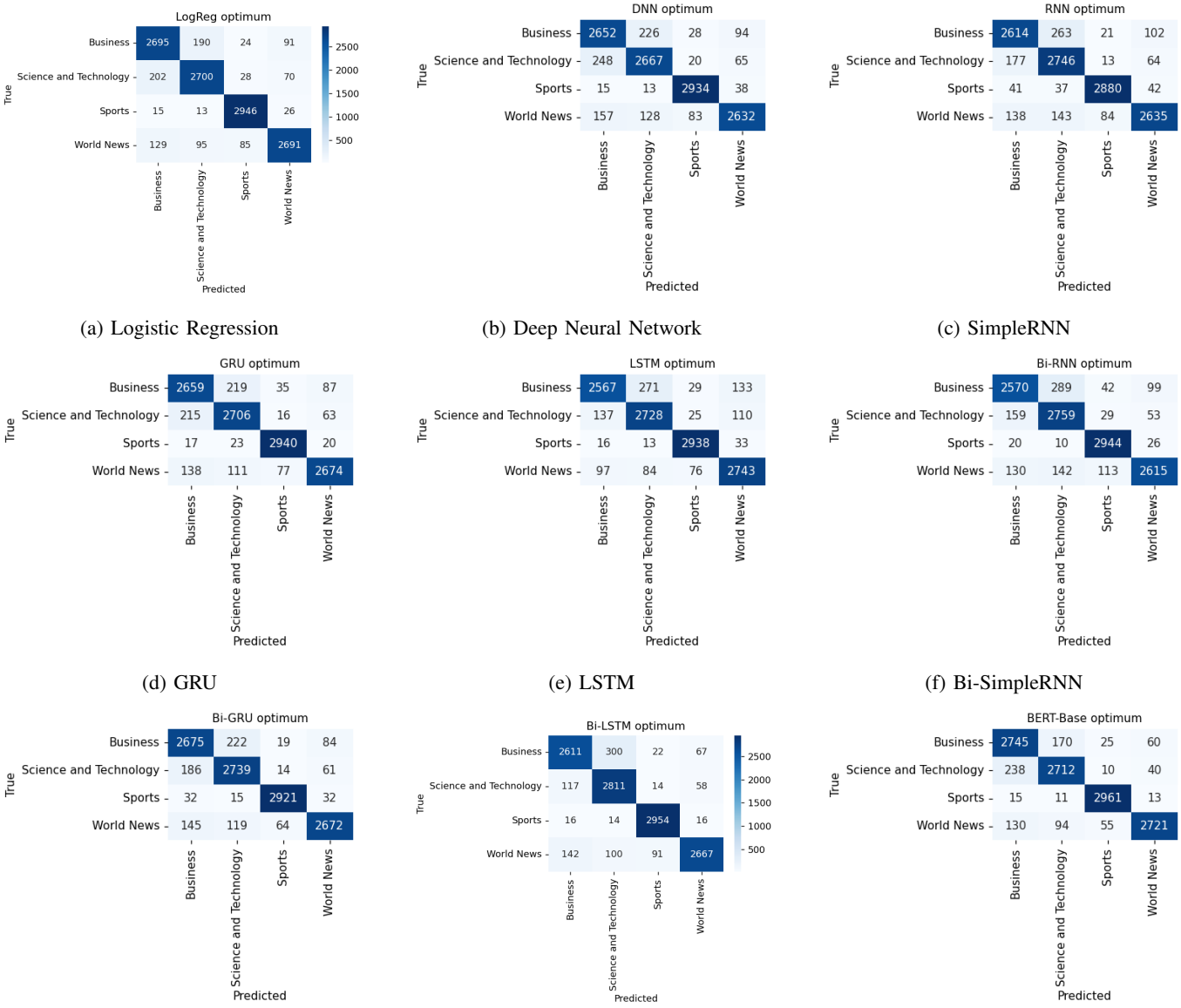
Skip-gram. A Word2Vec model [3] (`sg=1`) is trained from scratch on each preprocessed training corpus using gensim [9] with `vector_size=200`, `window=5`, `min_count=3`, and 10 epochs. The resulting embedding matrix is loaded into a frozen `nn.Embedding` layer for the recurrent models.

D. Models

All non-transformer neural models are implemented in PyTorch [8] and trained with Adam ($lr = 10^{-3}$) and cross-entropy loss weighted by inverse class frequency.

- **Logistic Regression (TF–IDF)**: `C=1.0`, `solver=saga`, `class_weight='balanced'`.
- **Deep NN (TF–IDF)**: Dense$(256) \rightarrow$ Dropout$(0.5) \rightarrow$ Dense$(128) \rightarrow$ Dropout$(0.4) \rightarrow$ Dense$(64) \rightarrow$ Dropout$(0.3) \rightarrow$ Dense(4).
- **RNN family (Skip-gram)**: SimpleRNN, LSTM [4], GRU [5], and their bidirectional variants. Hidden size 64, mask-aware mean pooling over the recurrent outputs, dropout 0.4.
- **BERT-Base (uncased)** [6]: HuggingFace `BertForSequenceClassification` with `max_length=80`, AdamW ($lr = 2 \times 10^{-5}$, weight decay 0.01), batch size 16, up to 15 epochs with EarlyStopping (`patience=2`) on validation macro-F1, mixed-precision (fp16) on the RTX 3070. In practice EarlyStopping fired at epoch 5–6 across the three variants, matching the 2–4 epoch budget recommended in [6].

Hyperparameters were manually tuned on a stratified 80/20 train/validation split (test set untouched). Table I summarizes the tuning sweeps; in each case the configuration with the higher validation macro-F1 was promoted to the final test-set evaluation.



(a) Logistic Regression (b) Deep Neural Network (c) SimpleRNN

(d) GRU (e) LSTM (f) Bi-SimpleRNN

(g) Bi-GRU (h) Bi-LSTM (i) BERT-Base

Fig. 5: Confusion matrices for all nine architectures on the *optimum* preprocessing variant. Across the family, *Sports* is essentially separable while the residual error is concentrated on the *Business*/*World News* pair, which share macro-economic and political vocabulary. Bi-LSTM (h) and BERT-Base (i) tighten the diagonal the most.

TABLE I: Hyperparameter tuning runs and the configurations promoted to the final test-set evaluation.

Model	Probed values	Selected (val macro-F1)
LogReg (TF-IDF)	$C \in \{0.5, 1.0, 2.0\}$	C=1.0 (0.9225)
DNN (TF-IDF)	dropout $\in \{0.2, 0.4, 0.5\}$	dropout=0.5 (0.9161)
SimpleRNN	hidden $\in \{32, 64\}$	hidden=64 (0.9038)
GRU / LSTM	hidden $\in \{64, 128\}$	hidden=64 (0.9194)
BERT-Base	$lr \in \{1e-5, 2e-5, 3e-5\}$	lr=$2e-5$ (0.9381)

III. Results

Table II summarizes test-set macro-F1 across all 27 runs.

TABLE II: Test-set macro-F1 by model and preprocessing variant. Bold marks the best score per row; the overall best run is BERT-Base on the *none* variant.

Model	none	extreme	optimum
Logistic Regression (TF-IDF)	0.9098	**0.9209**	0.9192
Deep NN (TF-IDF)	0.8804	0.9139	**0.9144**
SimpleRNN	0.8883	0.9009	**0.9021**
GRU	0.9151	0.9163	**0.9168**
LSTM	0.9168	**0.9183**	0.9159
Bi-SimpleRNN	0.8990	0.9009	**0.9031**
Bi-GRU	0.9148	0.9155	**0.9214**
Bi-LSTM	0.9108	**0.9186**	0.9164
BERT-Base	**0.9376**	0.9288	0.9313

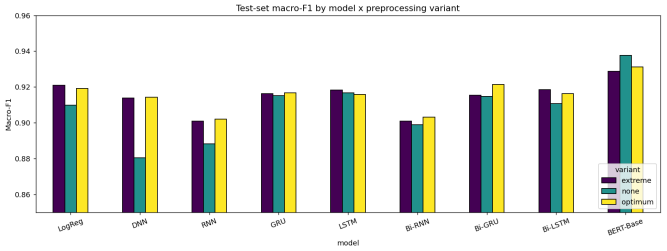
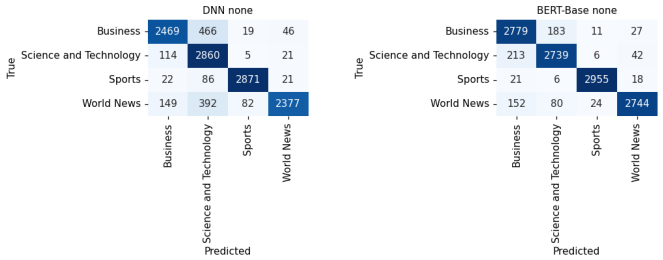


Fig. 6: Macro-F1 by model × preprocessing variant on the held-out test set. BERT-Base on the *none* variant achieves the overall best result (0.9376); Bi-GRU on *optimum* is the best non-Transformer run (0.9214).



(a) Worst run: DNN on *none* (macro-F1 0.8804)

(b) Best run: BERT on *none* (macro-F1 0.9376)

Fig. 7: Best-vs-worst confusion matrices. The DNN/*none* pairing collapses *World News* into *Business* (recall 0.78), whereas BERT-Base on the same preprocessing tightens every diagonal entry above 0.93.

A. Discussion

Four patterns emerge from the comparison.

(1) Preprocessing matters most for shallow models. The Deep NN drops 3.4 macro-F1 points moving from *optimum* to *none*, while Bi-LSTM drops only 1.1 – the Skip-gram embedding's frequency floor (`min_count=3`) effectively filters HTML tokens before they reach the recurrent network.

(2) Optimum beats Extreme on most non-transformer models. For five of the six recurrent architectures, our *optimum* pipeline outperforms aggressive Porter stemming. Stemming collapses semantically distinct tokens (e.g. *international/intern*) and removing all stopwords drops negation cues that World News and Business reports rely on.

(3) Bidirectional gated cells dominate the non-transformer family. Bi-GRU on *optimum* achieves macro-F1 **0.9214** – the strongest non-Transformer pairing – edging out Logistic Regression+TF-IDF by 0.22 points and outperforming the unidirectional GRU by 0.46. Vanilla SimpleRNN underperforms all gated variants by 1–2 points, consistent with its known vanishing-gradient limitation on length-80 sequences.

(4) BERT-Base reverses the preprocessing intuition. Surprisingly, BERT scores *highest* on the *none* variant (0.9376), *lowest* on *extreme* (0.9288). WordPiece tokenization handles HTML wrappers as predictable subword sequences and benefits from preserved capitalization cues, whereas Porter stemming destroys the subword statistics that BERT was pretrained on.

B. Best vs. Worst

Best run overall: **BERT-Base on *none*** (macro-F1 **0.9376**, accuracy 0.9375, train time 2109 s with EarlyStopping firing at epoch 6). *Best non-transformer run*: Bi-GRU on *optimum* (macro-F1 0.9214, train time 18.7 s – two orders of magnitude faster than BERT for a 0.016 point F1 trade-off). *Worst run*: Deep NN on *none* (macro-F1 0.8804) – HTML tokens dominate the TF–IDF feature space and the dense network overfits to them. Fig. 5 shows the per-class confusion matrices for all nine architectures on the *optimum* variant, and Fig. 7 contrasts the worst and best runs head-to-head: *Sports* is the easiest class (F1=0.97 across the family), while the *Business*/*World News* pair is the dominant source of residual error, sharing macro-economic and political vocabulary.

IV. Conclusion

On a four-class news headline corpus we benchmarked 27 model×preprocessing configurations under a single, reproducible pipeline. The takeaways are: **(i)** thoughtful preprocessing matters more than model choice for classical models – the Deep NN gains 3.4 macro-F1 points moving from *none* to *optimum*, while Bi-LSTM gains only 1.1 because its frozen Skip-gram embedding already filters HTML noise; **(ii)** bidirectional gated networks close most of the gap to BERT at a fraction of the training cost (Bi-GRU trains in 19 s versus BERT's 35 min with the 15-epoch budget); **(iii)** class weighting is necessary – not optional – on imbalanced training data, and *Business*/*World News* (the two smallest classes) are the most confused pair, sharing macro-economic and political vocabulary; **(iv)** BERT-Base inverts the usual preprocessing intuition – its WordPiece tokenizer handles HTML wrappers gracefully and is degraded by Porter stemming, so practitioners should treat preprocessing decisions as model-specific rather than universal.

Limitations include the relatively small embedding dimension (200) used for Skip-gram and the 15-epoch budget with EarlyStopping (`patience=2`) chosen to fit within the project's compute budget on an RTX 3070 (8 GB), with BERT converging early at epoch 5–6. Future work includes back-translation augmentation for the under-represented *World News* class, fine-tuning BERT with layer-wise learning-rate decay, and exploring `roberta-base` as a drop-in replacement.

References

[1] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *ECML*, 1998.
[2] Y. Kim, "Convolutional neural networks for sentence classification," in *EMNLP*, 2014.
[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv:1301.3781*, 2013.
[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[5] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP*, 2014.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.
[7] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *JMLR*, vol. 12, pp. 2825–2830, 2011.
[8] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.
[9] R. Řehůřek and P. Sojka, "Software framework for topic modelling with large corpora," in *LREC Workshop*, 2010.