Old Roots, Fresh Fruits: Clickbait Detection with Effective Model Design Choices on Social Media

Yu-Min Tseng[1] and Cheng-Te Li[2]

[1] Department of Computer Science, Virginia Tech, USA,
ymn.tseng@gmail.com
[2] Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan
chengte@ncku.edu.tw

Abstract. Online clickbait continues to plague social-media platforms, where sensational captions lure users into low-value or misleading content. While prior work has explored individual modeling choices, i.e., sequential encoders, graph-based representations, and simple fusion strategies, no study has systematically compared these design dimensions in the clickbait domain. We address this gap by conducting the first comprehensive analysis of three core axes: (1) how to organize the model streams (treating caption and hashtags jointly vs. separately), (2) how to learn text representations (sequential vs. graph-based), and (3) how to fuse these modalities (concatenation vs. co-attention). Leveraging a large, manually labeled Instagram dataset of short captions paired with hashtags, we implement every combination of these axes to isolate their individual and joint impacts on detection performance. Our experiments reveal clear trends: processing caption and hashtags in parallel streams preserves their distinct semantic patterns and consistently outperforms unified processing; graph-based embeddings capture long-range and corpus-wide co-occurrence structures that sequential models alone miss; and a co-attention fusion mechanism aligns caption and hashtag signals, uncovering subtle mismatches characteristic of clickbait.

Keywords: clickbait detection, dual modeling, graph-based representation, co-attention fusion, short-text classification, social media

1 Introduction

Clickbait has become pervasive in online media, particularly on social networking platforms, as content producers compete for user attention [3,6]. Many publishers and influencers rely on sensational or misleading headlines and captions to lure readers and viewers, driven by strong economic incentives such as advertising revenue and higher engagement [5]. The rise of visually-focused social media (e.g., Instagram, Snapchat, Pinterest) has further shifted the landscape of clickbait: on these platforms an eye-catching image is paired with a short textual caption, and new forms of clickbait have emerged in which the text is often only tangentially related to the image. In particular, targeted clickbait techniques [7] are common, where posts include an array of popular or trending hashtags and buzzwords that are irrelevant to the actual content, solely to broaden reach and attract clicks. This prevalence of misleading content not only degrades user experience and trust but also poses challenges for content moderation. As a result, there is a pressing need for effective automatic clickbait detection. Recent advances in machine learning offer a potential solution by identifying subtle patterns in content that distinguish clickbait from genuine posts [10], but the multimodal and short-text nature of social media posts makes this detection task challenging.

Clickbait detection can be formulated as a binary short-text classification task [3,10,6]: given a piece of content (in this case, a social media post), the goal is to predict whether it is clickbait. The main input in such a scenario is the post's short textual caption, which may be accompanied by auxiliary signals like hashtags, user comments, engagement metrics (likes/shares), or even the post's propagation patterns through a social network. This formulation is analogous to other content integrity problems on social media such as fake news detection [8,4], rumor verification [12,9], and cyberbullying detection [1], which also involve classifying short posts with limited textual data and additional contextual clues.

In this paper, we present an approach towards effective clickbait detection on social media by systematically investigating the aforementioned design dimensions. We formally define the task as determining whether a given social media post is clickbait, where each post consists of a short textual caption and an accompanying set of hashtags (we focus on these textual components, as they are readily available and carry the linguistic cues of clickbait). We then explore three fundamental axes of model design for this problem:

- How to organize the model architecture with respect to input types: either a *unified model* that processes the caption and hashtags together as a single input, or a *dual-model* architecture that handles captions and hashtags in separate streams;
- How to integrate or fuse information from the caption and hashtags: comparing simple *concatenation* of their representations versus *a co-attention* mechanism that allows interactive feature learning between the two
- How to learn the representation of the post's text: contrasting a *sequential encoder* (e.g., Gated Recurrent Unit, which reads the caption and hashtags in order) with a *structural encoder* (e.g., Graph Neural Network, which captures relational structure among words and hashtags in a graph form).

We focus on these three axes because they span the essential ways a model can represent the content, relate the caption with its hashtags, and integrate information from both sources. By systematically varying and evaluating these choices, we aim to identify what combinations yield the most effective detector for social media clickbait.

In summary, our contributions are as follows: (1) **Systematic Architectural Exploration**: we conduct a comprehensive evaluation of key model design dimensions for clickbait detection on social media, examining how different

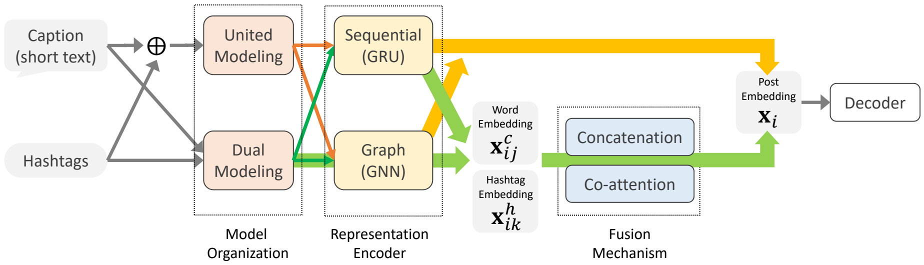

Fig. 1. Model design pipeline for clickbait detection. Captions and hashtags are encoded using either GRU or GNN, combined via united or dual modeling, and fused through concatenation or co-attention before classification.

architectural choices impact performance. (2) **Representation and Fusion Analysis**: we provide an in-depth analysis of how a post's caption and hashtag content should be represented and fused. This analysis sheds light on whether treating hashtags as an integral part of the text or as separate features is more effective, and how best to combine these information sources. (3) **Extensive Empirical Validation**: we present a thorough experimental comparison using a real-world Instagram dataset, benchmarking our approach against prior arts, and showing promising improvement of detection performance.

2 Problem Statement

Let $\mathcal{D} = \{(c_i, H_i, y_i)\}_{i=1}^{N}$ denote the labeled dataset of N social-media posts. For the i-th post: $c_i = (w_{i,1}, w_{i,2}, \dots, w_{i,T_i})$ is the caption, a sequence of T_i tokens drawn from a vocabulary $\mathcal{V}$. $H_i = \{h_{i,1}, h_{i,2}, \dots, h_{i,K_i}\}$ is the set of K_i hashtags (treated as an unordered set). $y_i \in \{0, 1\}$ is the clickbait label, where $y_i = 1$ indicates clickbait and $y_i = 0$ indicates a normal post. We write $\mathbf{x}_i^c \in \mathbb{R}^d$ and $\mathbf{x}_i^h \in \mathbb{R}^d$ for the vector representations of the caption and hashtag set, respectively, produced by an encoder. A fusion function $\mathcal{F}(\mathbf{x}_i^c, \mathbf{x}_i^h)$ then yields a joint vector $\mathbf{x}_i$, and a classifier $g(\mathbf{x}_i; \theta)$ outputs a predicted score. Finally, the clickbait probability is $\hat{y}_i = \sigma\big(g(\mathbf{x}_i; \theta)\big)$, where σ is the sigmoid activation and θ collects all trainable parameters. Clickbait detection is a binary classification problem: we learn θ by minimizing the average binary cross-entropy loss over $\mathcal{D}$ so that, for each post (c_i, H_i), the model correctly predicts its label y_i.

3 Model Design & Choices

We present the clickbait-detection model design pipeline in Figure 1. The pipeline begins by transforming each post's raw caption and hashtag set into continuous vector embeddings. Word embeddings for the caption capture the sequential semantics of short text, while hashtag embeddings encode topical signals from a non-ordered tag set. These dual embedding spaces feed into a representation encoder, which may be either a sequential model (GRU) that preserves word-order information or a structural model (GNN) that captures relational patterns among tokens. This dual-encoding choice acknowledges that captions and hashtags present different structural characteristics: captions rely on sequence, whereas hashtags form a loose graph of co-occurrences. The encoded vectors then enter the model-organization stage, where they are processed either in a single, unified stream (treating caption + hashtags as one concatenated input) or via two parallel streams (separate caption and hashtag encoders). Finally, the resulting features are merged using either simple concatenation or a co-attention mechanism before passing to the decoder for binary classification. This flow, from embedding to encoding, organization, fusion, and decoding, ensures modularity, allowing systematic investigation of how representation choice, input partitioning, and fusion strategy each contribute to overall performance.

Model Organization In the **united modeling** approach, we treat the caption and its associated hashtags as a single input sequence. Formally, given the i-th post's caption $c_i = (w_{i,1}, \ldots, w_{i,T_i})$ and hashtag set $H_i = \{h_{i,1}, \ldots, h_{i,K_i}\}$, we define a concatenated sequence $z_i = (w_{i,1}, \ldots, w_{i,T_i}, h_{i,1}, \ldots, h_{i,K_i})$. A single encoder $\mathcal{E}$ then maps this joint sequence into a unified representation vector $\mathbf{x}_i = \mathcal{E}(z_i)$, which is subsequently fed to the fusion and classification layers. The key idea is to allow the model to learn shared feature interactions between caption tokens and hashtags in one continuous stream. This simplicity often yields strong baselines, as it leverages off-the-shelf sequence encoders directly on all textual inputs. However, it may also conflate the fundamentally different structures and roles of ordered caption words versus unordered tags, potentially diluting specialized signals encoded in hashtags.

By contrast, the **dual modeling** approach maintains two distinct encoding streams for caption and hashtags. Here we deploy separate encoders, $\mathcal{E}_c$ for the caption and $\mathcal{E}_h$ for the hashtag set. We compute $\mathbf{x}_i^c = \mathcal{E}(c_i)$ and $\mathbf{x}_i^h = \mathcal{E}(H_i)$, and pass both vectors to a subsequent fusion module $\mathcal{F}(\mathbf{x}_i^c, \mathbf{x}_i^h)$ before classification. This design explicitly recognizes the different statistical and structural properties of captions and hashtags, enabling each encoder to specialize – $\mathcal{E}_c$ can focus on sequential semantics while $\mathcal{E}_h$ can capture topical co-occurrence in a set. The trade-off is an increase in model complexity and parameter count, as well as the need to carefully balance the two streams. In practice, dual modeling often yields more fine-tuned representations for each modality, improving robustness when hashtags carry distinct clickbait cues. Both united and dual modeling can leverage the same Representation Encoder techniques, i.e., whether a sequential GRU or a structural GNN, to transform raw tokens into dense embeddings.

Representation Encoder A typical approach is to use a **Gated Recurrent Unit (GRU)** to encode the short sequential nature of captions and hashtags. In the unified modeling setting, the concatenated input sequence z_i is mapped by GRU into a single vector $\mathbf{x}_i = \mathrm{GRU}(z_i)$. Under dual modeling, the caption and hashtag streams are processed separately as $\mathbf{x}_i^c = \mathrm{GRU}(c_i)$ and $\mathbf{x}_i^h = \mathrm{GRU}(H_i)$.

The GRU encoder excels at capturing ordered dependencies in language, critical for teasing apart subtle phrasing cues that distinguish clickbait captions from genuine descriptions, but it treats the hashtag set as an artificial sequence, potentially overlooking the unordered, co-occurrence patterns among tags.

To complement this sequential view, we integrate a **Text Graph Convolutional Network (TextGCN)** [11] that explicitly models global token relationships across the entire dataset. We construct an undirected graph $G = (V, E)$ whose node set V comprises both document nodes (one per post) and word/hashtag nodes (one per unique token), and whose weighted adjacency matrix $\mathbf{A} \in \mathbb{R}^{|V|\times|V|}$ encodes (i) document-token TF-IDF edges and (ii) token–token PMI-based co-occurrence edges. Let $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-1/2}$ be the symmetrically normalized adjacency and $\mathbf{H}^0 \in \mathbb{R}^{|V|\times d_0}$ be the initial node features (one-hot or pretrained embeddings). Then two layers of graph convolution yield $\mathbf{H}^{(1)} = \mathrm{ReLU}(\tilde{\mathbf{A}}\mathbf{H}^{(0)}W^{(0)})$ and $\mathbf{H}^{(2)} = \tilde{\mathbf{A}}\mathbf{H}^{(1)}W^{(1)}$, where $W^{(0)}$ and $W^{(1)}$ are trainable weight matrices. For each post i, its document-node embedding $\mathbf{H}^{(2)}_i$ serves as the representation $\mathbf{x}_i$ in the unified model. In the dual scheme, separate subgraphs limited to caption tokens and hashtags respectively can produce $\mathbf{x}^c_i$ and $\mathbf{x}^h_i$. TextGCN captures global, corpus-wide co-occurrence signals that are especially valuable for identifying tag-driven clickbait patterns, though it requires a complete graph construction and does not account for word order.

Both GRU and TextGCN seamlessly integrate with either the united or dual modeling strategies: the GRU naturally ingests linear sequences, while TextGCN leverages graph structure to aggregate contextual signals. By comparing these two encoder paradigms across our pipeline, we can assess how sequential versus structural representations contribute to robust clickbait detection.

Fusion Mechanism We explore two strategies to fuse these textual sources into a single representation for classification: a simple **Concatenation** of the caption and hashtag features, and a **Co-attention** mechanism for fine-grained interaction between caption and hashtags.

Concatenation. The first fusion approach is straightforward vector concatenation. Given the caption representation $\mathbf{x}^c_i \in \mathbb{R}^d$ and hashtag representation $\mathbf{x}^h_i \in \mathbb{R}^d$ for post i, we form the fused post vector by appending one to the other: $\mathbf{x}_i = \mathbf{x}^c_i \oplus \mathbf{x}^h_i \in \mathbb{R}^{2d}$. This concatenation fusion simply stacks the learned caption features and hashtag features into a single vector. It is computationally efficient (no additional parameters beyond those used to obtain $\mathbf{x}^c_i$ and $\mathbf{x}^h_i$) and easy to implement. The intuition is that the model's next layers (the classifier) will use this combined feature vector to make the clickbait prediction. However, concatenation treats caption and hashtag features independently. It does not explicitly model interactions between specific caption words and hashtags. While efficient, this simplicity is a limitation: the network must implicitly learn any relationships between caption and hashtag signals in later layers. For instance, if certain hashtags make a caption more clickbaity, a concatenation-based model relies on the classifier to discover that pattern, rather than highlighting it in the representation itself.

Co-attention. To capture richer interactions between the caption and hashtags, we employ a co-attention fusion mechanism. The co-attention is applied unidirectionally from the caption to the hashtags, allowing the caption context to inform an attentive combination of hashtag features (and vice versa). This means the model learns to emphasize particular hashtag tokens that are most relevant to the caption's content, and simultaneously adjust the caption representation based on those important hashtags. Formally, let the caption encoder produce a matrix of token embeddings $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_T] \in \mathbb{R}^{d \times T}$ for the T words in the caption, and let the hashtag encoder produce $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K] \in \mathbb{R}^{d \times K}$ for the K hashtag tokens. We first compute an attention affinity matrix $\mathbf{F}$ between caption and hashtag features: $\mathbf{F} = \tanh(\mathbf{c}^\top W \mathbf{H})$, where W is a learnable weight matrix. Each element F_{tk} reflects the compatibility between caption word t and hashtag word k. This proximity matrix $\mathbf{F}$ is treated as a feature that bridges the two modalities, allowing us to project information from the caption space into the hashtag space and vice versa.

Next, we transform the caption and hashtag representations by incorporating this cross-modal affinity. We compute $\mathbf{H}^c = \tanh(W_c\mathbf{C} + (W_h\mathbf{H})\mathbf{F}^\top)$ and $\mathbf{H}^h = \tanh(W_h\mathbf{H} + (W_c\mathbf{C})\mathbf{F})$. Here W_c and W_h are learnable weight matrices, producing transformed feature matrices $\mathbf{H}^c$ and $\mathbf{H}^h$. Intuitively, $\mathbf{H}^c$ is an enriched caption representation that has absorbed information from the hashtags (via $\mathbf{F}^\top$), while $\mathbf{H}^h$ is an enriched hashtag representation influenced by the caption content. We then obtain attention weight vectors for caption words and hashtags by applying a softmax over these transformed features: $\alpha^c = \mathrm{softmax}(w_c^\top \mathbf{H}^c)$ and $\alpha^h = \mathrm{softmax}(w_h^\top \mathbf{H}^h)$, where w_c and w_h are learnable weights that produce unnormalized importance scores for each token. The softmax normalizes these scores across the T caption tokens and K hashtag tokens respectively, yielding $\boldsymbol{\alpha}^c = [\alpha_1^c, \ldots, \alpha_T^c]$ and $\boldsymbol{\alpha}^h = [\alpha_1^h, \ldots, \alpha_K^h]$ such that $\sum_{t=1}^{T} \alpha_t^c = 1$ and $\sum_{k=1}^{K} \alpha_k^h = 1$. These attention weights indicate which caption words and which hashtags are most salient *in the presence of each other*.

Using the attention weights, we compute a *co-attended representation* for the caption and for the hashtags by weighted summation of their token embeddings: $\hat{\mathbf{c}} = \sum_{t=1}^{T} \alpha_t^c \mathbf{c}_t$ and $\hat{\mathbf{h}} = \sum_{k=1}^{K} \alpha_k^h \mathbf{h}_k$. The resulting $\hat{\mathbf{c}}$ is a *hashtag-attended caption vector* (a single d-dimensional vector summarizing the caption, with more weight on words that hashtags consider important), and $\hat{\mathbf{h}}$ is the corresponding *caption-attended hashtag vector*. We then concatenate these to form the final fused representation for the post: $\mathbf{x}_i = \hat{\mathbf{c}} \oplus \hat{\mathbf{h}}$.

Through this co-attention process, the model learns a post representation that emphasizes cross-modal cues. For example, if a caption contains sensational language, the co-attention may focus on complementary hashtag terms that reinforce the clickbait nature of the post. This *jointly-attended fusion* is more expressive than simple concatenation, as it allows the network to pinpoint which words and hashtags interact to signal clickbait. The improved capacity to model caption-hashtag interactions can enhance detection performance, especially in cases where neither the caption nor hashtags alone fully indicate clickbait, but their combination does.

Decoder and Training Once the caption and hashtags are fused into a single post representation $\mathbf{x}_i$, we feed this vector into a decoder network to predict the probability that the post is clickbait. Our decoder is a two-layer Multi-Layer Perceptron (MLP): a hidden dense layer (with ReLU activation) followed by an output layer. We apply dropout regularization in the MLP to prevent overfitting. The final output is a single logit which is passed through a sigmoid (equivalently, a 2-class softmax) to produce $\hat{y}_i$, the predicted probability of the post being clickbait. The model is trained end-to-end using a binary cross-entropy loss between $\hat{y}_i$ and the true label y_i. We optimize the parameters using the RMSprop algorithm, which we found effective for this task.

4 Experiments

Data & Settings. We choose the Instagram clickbait dataset from Ha et al.'s large-scale study [3]. There are 7,769 ground-truth posts (3,509 non-clickbait vs. 4,260 clickbait). We leverage solely the textual modality, i.e., each post's *caption* plus its *hashtag* set, both to simplify feature extraction and to isolate the linguistic patterns that signal clickbaiting in visual-centric social media. We randomly split all posts into training, validation, and test sets in a 65:10:25 ratio. Each experiment is repeated 15 times with different random seeds, and we report the mean of all runs. We assess classifier performance using four standard metrics: Accuracy (Acc) to measure overall correctness, Area Under the ROC Curve (AUC) to capture ranking quality, Precision (Pre) to evaluate clickbait-positive prediction reliability, and the F1 score to balance precision and recall.

Analysis Results. We analyze the experimental results (Table 1) to answer three key questions about model design choices for clickbait detection. The table compares models along three axes: model organization (unified vs. dual modeling), representation learning (sequential GRU vs. graph-based TextGCN), and fusion mechanism (simple feature concatenation vs. co-attention).

Q1: (Model Organization) Does Dual Modeling Outperform Unified? Dual modeling consistently outperforms unified modeling for clickbait detection, especially when combined with effective fusion strategies. Splitting the model into separate caption and hashtag streams yields higher F1 and AUC than processing them as a single sequence. The strongest gains appear when dual modeling is paired with co-attention, which better captures complementary signals. However, one outlier, i.e., CoAtt-GRU, shows that dual modeling alone is insufficient; it must be supported by strong representations and carefully aligned fusion. The advantage of dual modeling lies in its ability to learn task-specific embeddings: captions capture "hook" language while hashtags provide supporting or misleading cues. By encoding them independently, the model preserves their semantic differences. When fused, these specialized embeddings can highlight inconsistencies characteristic of clickbait, e.g., a sensational caption not matched by relevant tags. This structural separation improves precision-recall trade-offs and overall ranking, validating dual modeling as an effective strategy.

Table 1. Model comparisons for unified (U) vs. dual (D) modeling, concatenation (Concat) vs. co-attention (CoAtt), and GRU vs. TextGCN representation learning.

		Acc	AUC	Pre	F1
U	GRU [2]	0.8375±0.0094	0.8345±0.0087	0.8193±0.0180	0.8140±0.0095
D	Concat-GRU	0.8415±0.0086	0.8388±0.0083	0.8218±0.0143	0.8190±0.0091
D	CoAtt-GRU	0.8200±0.0084	0.8155±0.0092	0.8063±0.0183	0.7914±0.0115
U	TextGCN [11]	0.8279±0.0039	0.8254±0.0042	0.8268±0.0040	0.8260±0.0041
D	Concat-TextGCN	0.8509±0.0086	0.8482±0.0085	0.8336±0.0144	0.8295±0.0095
D	ConAtt-TextGCN	0.8699±0.0077	0.8683±0.0077	0.8509±0.0205	0.8523±0.0087

Q2: (Representation Learning) Does TextGCN Yield More Informative Representations than GRU? TextGCN-based models consistently outperform or match GRU counterparts, especially under dual modeling with advanced fusion. Even in the unified setting, TextGCN shows advantages in F1 and precision, indicating its ability to capture informative signals comparable to sequential encoders. This advantage grows in dual setups: Concat-TextGCN surpasses Concat-GRU, and co-attention further widens the gap, with TextGCN showing significantly higher F1 and AUC. These trends highlight TextGCN's strength in modeling global word co-occurrences and document-level semantics, which are patterns GRU may overlook due to its local and sequential focus. TextGCN's graph-based structure enables better generalization of clickbait cues, even from rare or weakly associated tokens, which is especially effective when aligning two text streams like captions and hashtags. Its globally contextualized embeddings help the fusion layer more accurately integrate and contrast the inputs, resulting in better precision-recall trade-offs and ranking. In contrast, GRU lacks this broader context and sees diminishing returns in complex settings. Thus, TextGCN proves more effective, especially as model architecture grows in depth and interaction.

Q3: (Fusion Mechanisms) Does Co-attention Improve Performance over Simple Concatenation? The effectiveness of co-attention depends heavily on the strength of underlying text representations. With TextGCN, co-attention consistently outperforms concatenation, achieving the highest F1 and AUC. This indicates that co-attention can align complementary cues between captions and hashtags, such as sensational phrases paired with matching or contrasting tags, enhancing clickbait detection. In contrast, applying co-attention to GRU representations reduces performance across all metrics, suggesting that GRU's sequential, local focus lacks the robustness needed to support co-attention's added complexity, often leading to unstable or misaligned attention. Co-attention only proves beneficial when paired with rich, globally contextual embeddings like those from TextGCN. These allow the model to discover meaningful alignments between caption and hashtags, improving overall predictive power. GRU's localized encodings, especially when input modalities differ in length or vocabulary, make attention alignment more error-prone and prone to overfitting. Hence,

Table 2. Model comparisons for dual modeling that applies co-attention on embeddings derived from different combinations of GRU/TextGCN and Hashtags/Captions.

Hashtags		Captions		Acc	AUC	Pre	F1
GRU	TextGCN	GRU	TextGCN				
	✓		✓	0.8699±0.0077	0.8683±0.0077	0.8509±0.0205	0.8523±0.0087
✓	✓			0.8441±0.0070	0.8422±0.0068	0.8212±0.0196	0.8231±0.0079
✓			✓	0.8405±0.0107	0.8396±0.0106	0.8102±0.0190	0.8209±0.0117
	✓	✓		0.8332±0.0079	0.8282±0.0076	0.8257±0.0151	0.8055±0.0085
✓		✓		0.8200±0.0084	0.8155±0.0092	0.8063±0.0183	0.7914±0.0115
		✓	✓	0.7545±0.0166	0.7513±0.0157	0.7197±0.0260	0.7217±0.0170

co-attention's expressive power is only realized when backed by representation learning capable of capturing cross-modal relationships at a global level.

Q4: Which combination of sequential and graph-based encoders yields the most effective co-attention fusion? Should we allocate representational capacity to captions, hashtags, or both using graph-aware embeddings before applying co-attention? This question is critical because co-attention's effectiveness depends on the richness of the embeddings it aligns. If either input stream is weak or noisy, attention alignment falters, degrading performance. A systematic comparison is necessary to determine where graph convolution yields the most value and whether the added complexity of TextGCN is justified. Table 2 shows that performance peaks only when both captions and hashtags are encoded with TextGCN: this combination uniquely delivers the highest accuracy, AUC, and F1. Mixed setups with one GRU stream perform slightly worse, while GRU-only configurations fall behind significantly, underscoring their limitations in capturing long-range semantics. These trends suggest that clickbait often hinges on subtle inconsistencies between sensational captions and contextual hashtags, patterns best captured through global co-occurrence structures. Co-attention succeeds only when both streams offer equally expressive, graph-enriched embeddings, enabling precise alignment and robust detection.

5 Conclusions

In this work, we revisited the "old roots" of text-and-metadata fusion and the "fresh fruits" of modern graph and attention architectures to advance clickbait detection on social media. Our systematic exploration of three orthogonal design axes, i.e., *model organization* (unified vs. dual streams), *representation learning* (sequential GRU vs. graph-based TextGCN), and *fusion mechanism* (concatenation vs. co-attention), revealed clear patterns: separating caption and hashtag modeling, embedding both streams in a global co-occurrence graph, and then applying a co-attention layer consistently yields the strongest predictive performance. The results show that carefully chosen combinations of well-understood components can surpass more monolithic or heavily engineered alternatives.

Acknowledgements

This work is supported by the National Science and Technology Council (NSTC) of Taiwan under grants 113-2221-E-006-201-MY3, 112-2628-E-006-012-MY3, and 113-2634-F-002-007.

References

1. H.-Y. Chen and C.-T. Li. HENIN: Learning heterogeneous neural interaction networks for explainable cyberbullying detection on social media. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2543–2552. Association for Computational Linguistics, Nov. 2020.
2. K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics, Oct. 2014.
3. Y. Ha, J. Kim, D. Won, M. Cha, and J. Joo. Characterizing clickbaits on instagram. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12, 2018.
4. Y.-J. Lu and C.-T. Li. GCAN: Graph-aware co-attention networks for explainable fake news detection on social media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 505–514. Association for Computational Linguistics, July 2020.
5. K. Munger. All the news that's fit to click: The economics of clickbait media. *Political communication*, 37(3):376–397, 2020.
6. M. Potthast, S. Köpsel, B. Stein, and M. Hagen. Clickbait detection. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*, pages 810–817. Springer, 2016.
7. A. Shrestha, A. Flood, S. Sohrawardi, M. Wright, and M. N. Al-Ameen. A first look into targeted clickbait and its countermeasures: The power of storytelling. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–23, 2024.
8. K. Shu, L. Cui, S. Wang, D. Lee, and H. Liu. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 395–405, 2019.
9. M. Sun, X. Zhang, J. Zheng, and G. Ma. Ddgcn: Dual dynamic graph convolutional networks for rumor detection on social media. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 4611–4619, 2022.
10. Y. Wang, Y. Zhu, Y. Li, J. Qiang, Y. Yuan, and X. Wu. Clickbait detection via prompt-tuning with titles only. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024.
11. L. Yao, C. Mao, and Y. Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377, 2019.
12. J. Zheng, X. Zhang, S. Guo, Q. Wang, W. Zang, and Y. Zhang. Mfan: Multi-modal feature-enhanced attention networks for rumor detection. In *IJCAI*, volume 2022, pages 2413–2419, 2022.