mathematics

MDPI

Article

A Multi-View Text Classification Framework Combining Transformers and Topic Modeling

Filipe M. P. Santos *, João M. C. Sousa and Susana M. Vieira

IDMEC, Instituto Superior Técnico, University of Lisbon, Av. Rovisco Pais 1, 1049-001 Lisbon, Portugal; jmsousa@tecnico.ulisboa.pt (J.M.C.S.); susana.vieira@tecnico.ulisboa.pt (S.M.V.)
* Correspondence: filipempsantos@tecnico.ulisboa.pt

Abstract

Text classification models often rely on transformer-based embeddings or topic-modeling techniques independently, limiting their ability to capture both contextual and thematic information. We propose the Topic-Transformer (TT) framework, a multi-view learning framework that combines transformer-based encoders and topic-modeling methods to improve text classification performance. TT processes text simultaneously through a transformer model and a topic-modeling technique. The resulting contextual embeddings and topic representations are then fed into a classification head. TT is tested using five distinct transformer encoders and three topic-modeling techniques, resulting in fifteen configurations of TT. The tests are conducted on six benchmark datasets and the results are evaluated in terms of macro-average F1-score, precision, recall, and training time. The results show that, for every combination of dataset and transformer, at least one TT configuration outperforms its standalone transformer counterpart. However, the magnitude and statistical significance of these improvements vary across datasets and configurations. In four of the six datasets, at least one TT configuration yields a statistically significant improvement at a significance level of 0.05 after multiple-testing correction. The largest significant gains are observed on 20Newsgroups, where improvements reach 2.8% in macro-average F1-score, with statistically significant improvements found in most transformer–TT comparisons.

Keywords: text classification; topic modeling; transformers; multi-view; information fusion

MSC: 68T50

1. Introduction

Text classification is a fundamental task in Natural Language Processing (NLP), with applications spanning sentiment analysis [1,2], document categorization [3,4], hate speech detection [5], spam detection [6,7], misinformation detection [8], and biomedical text analysis [9]. Over the decades, the evolution of text classification methodologies has closely followed advancements in NLP, integrating increasingly sophisticated techniques for feature extraction and model learning [10,11].

Early approaches relying on handcrafted features derived from word frequency [12] and syntactic structures [13] as primary techniques for representing text data. Models such as Logistic Regression [14], Naive Bayes [15], and Support Vector Machines [12] were commonly employed. With the advent of deep learning, more powerful text representation methods emerged, such as word embeddings [16,17], which provided richer representations of textual data. Recurrent Neural Networks [18], including variants like Gated

Check for updates

Academic Editors: Jing Cheng, Hongping Ren and Ying Li

Received: 24 September 2026
Revised: 26 October 2026
Accepted: 29 October 2026
Published: 2 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Recurrent Units and Long Short-Term Memory networks, enabled sequential processing of text, capturing contextual dependencies more effectively.

In recent years, transformer-based models [19] have significantly advanced the state-of-the-art in text classification by learning deep contextual representations [10,11]. Nowadays, pre-trained Large Language Models such as BERT (Bidirectional Encoder Representations from Transformers) [20], T5 [21], ALBERT [22], RoBERTa [23], XLNet [24], DistilBERT [25], and domain-specific models like FinGPT [26] and Covid-HateBERT [27] have set new state-of-the-art benchmarks. These models excel in understanding complex linguistic structures and contextual nuances, making them effective across a variety of NLP tasks, such as text classification, text generation and translation. More recent work has also explored multilingual cross-modal representation learning, focusing on aligning linguistic representations across languages and visual modalities while improving model efficiency [28].

Topic modeling, on the other hand, is another NLP field which provides an unsupervised approach to uncover latent semantic structures—thematic information—within a corpus [29,30]. This characteristic renders it particularly valuable for exploratory data analysis, document clustering, and enhancing interpretability. Common methods for topic modeling include Latent Dirichlet Allocation (LDA) [29], BERTopic [31], Fuzzy Latent Semantic Analysis (FLSA) [30] and Non-negative Matrix Factorization [32], each offering different ways to extract and represent topics.

All in all, transformer representations are primarily optimized for predictive objectives and token-level contextual modeling, without explicitly modeling corpus-level thematic distributions. In contrast, topic models generate structured document-level representations that explicitly encode global semantic organization and offer enhanced interpretability. Consequently, transformers and topic models capture different aspects of textual information.

According to multi-view learning principles, multiple representations of the same underlying instance can be treated as distinct views that capture complementary information about the same phenomenon [33–35]. However, more views do not necessarily provide more relevant information for a given task; their usefulness depends on whether they contain complementary and discriminative information [36–38]. This perspective applies naturally to textual representations derived from a common input: contextual embeddings and thematic topic distributions constitute two views of the same document, but encode different aspects of its information. Such a formulation corresponds to engineered multi-view learning, where multiple views are constructed from a single input through different representation mechanisms [39]. TFusion [5] applied this principle by explicitly combining topic representations and deep contextual embeddings as complementary views, using a model-level fusion framework [40] to improve classification performance.

Nevertheless, TFusion was only tested on one single configuration, with DistilBERT as the transformer encoder and LDA as the topic model. As a result, it remains unclear whether the reported improvements arise from the intrinsic complementarity between contextual and thematic representations, or from the characteristics of the selected models. Hence, it is yet to be systematically evaluated whether such gains generalize across different transformer architectures, topic-modeling paradigms, and datasets with heterogeneous properties. This limitation restricts our understanding of when and why multi-view integration is beneficial in transformer-based text classification.

To address this gap, we propose Topic-Transformer (TT), a generalized and modular multi-view fusion framework that enables the integration of multiple transformer encoders with diverse topic-modeling techniques. Unlike TFusion, TT decouples representation learning of every component, allowing transformer encoders and topic models to be trained independently and reused across configurations. This design facilitates systematic explo-

ration of cross-model compatibility while significantly reducing computational overhead when evaluating multiple combinations.

The key contributions of this work are as follows:

- A unified multi-view fusion architecture that combines contextual representations from different transformer encoders with thematic representations derived from different topic-modeling methods within a common classification framework.
- A modular training strategy that separates contextual and thematic representation learning, enabling the independent pretraining and reuse of these components across different framework configurations, with only the classification head requiring task-specific training for each combination.
- An empirical evaluation of fifteen framework configurations, comprising five transformer encoders and three topic-modeling methods, across six English benchmark datasets with diverse characteristics, including vocabulary size, sequence length, number of classes, and task type.

By systematically analyzing the interaction between contextual and thematic representations, this work contributes to a deeper understanding of multi-view learning in NLP and clarifies the role of topic modeling in a transformer-dominated environment, both from a performance and architectural perspective.

The remainder of this paper is structured as follows. Section 2 presents a brief literature review on work which concerns both transformers and topic models. Section 3 thoroughly explains the transformer encoders structure. Section 4 describes three distinct topic-modeling techniques. Section 5 describes the proposed Topic-Transformer (TT) framework in detail, including its architecture and fusion strategy. Section 6 discusses the experimental setup, including datasets, evaluation metrics, and baseline models. Finally, Section 7 concludes the paper and outlines potential future research directions.

2. Related Work

Research combining transformers and topic modeling can be broadly divided into two distinct but not mutually exclusive directions: using transformers to support topic discovery, and using topic modeling to support transformer inference. The first focuses on improving textual representations for downstream topic extraction, while the second explores how topic information can contribute to downstream prediction, either through sequential pipelines or through integration into the transformer architecture.

The use of transformers for topic discovery can be understood as an extension of neural topic modeling, in which language-model-based embeddings provide semantic representations beyond traditional bag-of-words features. Transformer-based language models, particularly large language models (LLMs), offer a natural progression of language models through their contextual text representations, positioning them as the semantic foundation for topic discovery [41–44]. A notable example of this is BERTopic [31], which makes use of transformer-generated sentence embeddings as the basis for clustering and topic extraction. Subsequent work investigates how embedding model selection and text preprocessing affect topic coherence and diversity [45], and extends neural topic discovery by enabling topics to be organized within predefined hierarchical structures [46]. Other approaches explore multimodal graph-based refinement [44]. Together, these methods demonstrate how contextual representations can improve the semantic quality, organization, and interpretability of discovered topics, with the primary objective remaining the advancement of topic discovery itself rather than downstream prediction.

The other research direction uses topic modeling to support transformer-based prediction. In these approaches, topic models and transformers can remain as separate components in a sequential pipeline, with the output of one informing the processing or predictions of

the other [42,43,47–52]. Their methodological contribution lies in combining the contextual representations of transformers with the thematic structure provided by topic models, without requiring changes to the transformer internal architecture. For example, BERT-based review filtering followed by BERTopic enables thematic analysis to focus on negative hotel reviews [53], while combining transformer-based sentiment analysis with LDA and importance–performance analysis connects predicted sentiment to topic-level priorities [49]. Other approaches use topic modeling alongside zero-shot classification [48], or transformer-based feature extraction and neural topic models within stacking ensembles [42]. These pipelines demonstrate how topic information can complement predictive outputs and improve the interpretation of text classification results, but the separation between components limits the extent to which topic representations can influence the transformer internal inference process.

A more tightly coupled approach incorporates topic information directly into transformer architectures, allowing thematic representations to participate in prediction through auxiliary objectives, augmentation layers, attention mechanisms, or feature fusion [41,47,54–56]. For instance, TFusion [5] combines topic representations and deep contextual embeddings as complementary views for text classification through a model-level fusion framework. TopicBERT explores two forms of integration: auxiliary topic prediction as an additional learning objective and a topic-augmentation layer that injects LDA-derived representations into the language model [54]. Um and Kim introduce neural topic attention to connect topic representations with transformer-based language modeling, reporting classification performance improvements across ALBERT and ELECTRA configurations [41]. Smoothed-ModernBERT further develops this principle through co-attention between token-level representations and document-level topic distributions derived from a probabilistic topic model, followed by dynamic fusion of the resulting representations [55]. The central methodological distinction is that topic information is no longer restricted to a separate analytical stage or an external predictive feature. It becomes part of the model internal representation and inference process.

The Topic-Transformer framework fits inside the second research line presented. However, in contrast to these approaches, TT enables independent modular representation learning and reuse of components across different configurations, allowing the generalizability of contextual–thematic fusion to be assessed across diverse models and datasets. Moreover, this work focuses on systematically evaluating the compatibility and complementarity of different transformer encoders and topic-modeling techniques.

3. Transformers

Transformers [19] are currently the state-of-the-art in several deep learning fields such as NLP and computer vision. In this section, we define the transformer structure and how it can be applied to generate contextual representations of text. A transformer structure (see Figure 1) has two main components, which consist of a stack of layers:

- Encoder: Responsible for processing input sequences and extracting contextual information. Each encoder layer consists of sublayers, namely:
  - A multi-head self-attention layer,
  - A feedforward neural network (FFN)
- Decoder: Responsible for generating output sequences. Each decoder layer consists of sublayers, namely:
  - A multi-head self-attention layer,
  - A masked multi-head attention layer,
  - A feedforward neural network

Additionally, depending on the application, the complete transformer structure may not be required, and a component may be replaced by a simpler one. In text classification, only the encoder is needed as it is the component which extracts information; and the decoder, in turn, is replaced by a classification head, usually a Multi-layer Perceptron (MLP), (see Figure 1).

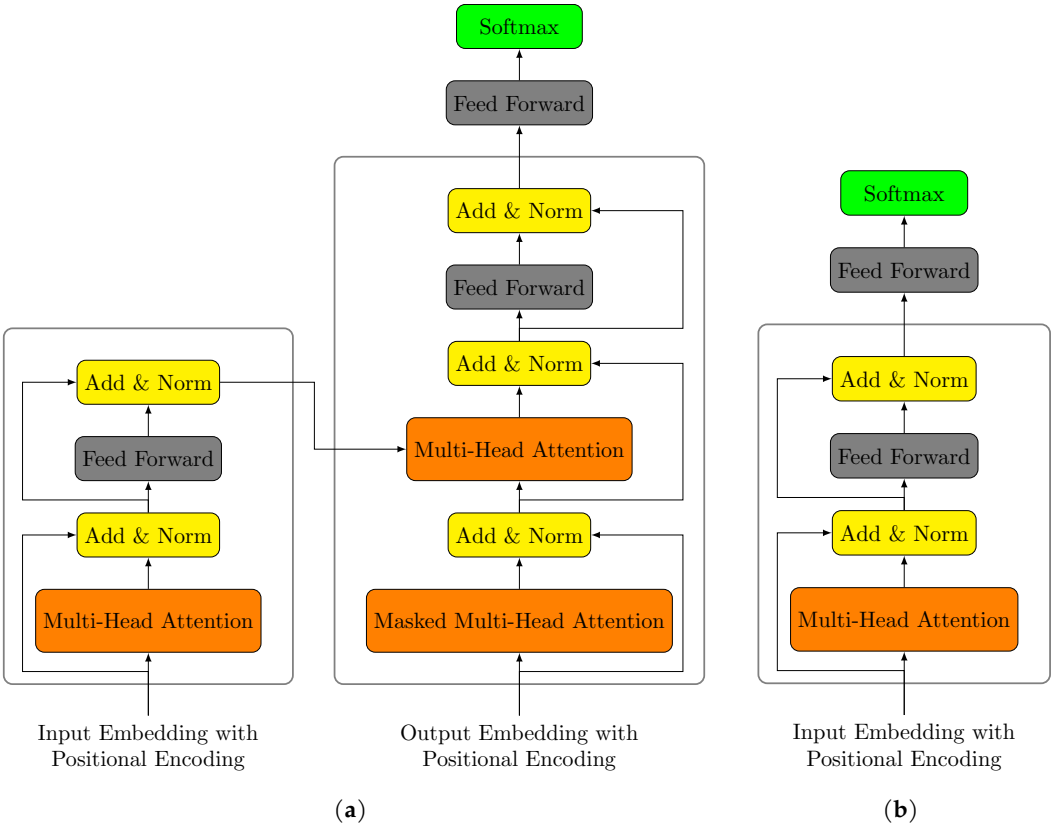


Figure 1. Transformer architecture: (**a**) Encoder–decoder architecture with one encoder layer and one decoder layer; (**b**) Simplified architecture with one encoder layer for text classification.

In the remainder of this section, we focus on the transformer encoder, and how it is used for text classification.

3.1. Input Embedding and Positional Encoding

Before processing the input text, it must first be tokenized into a sequence of discrete units (tokens), typically words or subwords.

After tokenization, each token in the input sequence is mapped to an embedding $\mathbf{e}^* \in \mathbb{R}^f$ in the latent embedding space.

Additionally, transformers do not process data sequentially, thus, they require positional encodings ($\mathbf{e}^p \in \mathbb{R}^f$) to retain order information.

The positional encoding $\mathbf{e}^p$ is then added to the embedding $\mathbf{e}^*$ to inject position information:

$$\mathbf{h}^{(0)} = \mathbf{e}^* + \mathbf{e}^p \tag{1}$$

Having, processed an entire sequence of tokens, the embeddings are fed to the first layer of the encoder.

3.2. Multi-Head Self-Attention

The first sublayer of a transformer encoder is the self-attention layer, this computes attention scores between all pairs of tokens in a sequence. Given an input sequence

$\mathbf{H}^{(l)} = (\mathbf{h}_1^{(l)}, \mathbf{h}_2^{(l)} \ldots)$, the self-attention mechanism computes queries $\mathcal{Q}$, keys $\mathcal{K}$, and values $\mathcal{V}$ as follows:

$$\mathcal{Q}_i^{(l)} = \mathcal{W}_{\mathcal{Q}}^{(l)} \mathbf{H}^{(l-1)}, \quad \mathcal{K}^{(l)} = \mathcal{W}_{\mathcal{K}}^{(l)} \mathbf{H}^{(l-1)}, \quad \mathcal{V}^{(l)} = \mathcal{W}_{\mathcal{V}}^{(l)} \mathbf{H}^{(l-1)} \tag{2}$$

where $\mathcal{W}_{\mathcal{Q}}, \mathcal{W}_{\mathcal{K}}, \mathcal{W}_{\mathcal{V}} \in \mathbb{R}^{f_A \times f}$ are learned weight matrices.

The attention scores are computed using the scaled dot-product:

$$\text{Attention}(\mathcal{Q}^{(l)}, \mathcal{K}^{(l)}, \mathcal{V}_i^{(l)}) = \text{softmax}\left(\frac{\mathcal{Q}^{(l)} \mathcal{K}^{(l)\,T}}{\sqrt{f_A}}\right) \mathcal{V}^{(l)} \tag{3}$$

Multi-head attention extends this by applying several different attention mechanisms in parallel:

$$\mathbf{H}_{SA}^{(l)} = \text{MultiHead}^{(l)}(\mathbf{H}^{(l-1)}) = \mathcal{W}_O^{(l)}[\text{head}_1^{(l)}, \text{head}_2^{(l)}, \ldots] \tag{4}$$

where each head is computed as

$$\text{head}_i^{(l)} = \text{Attention}(\mathcal{W}_{\mathcal{Q}_i}^{(l)} \mathbf{H}^{(l-1)}, \mathcal{W}_{\mathcal{K}_i}^{(l)} \mathbf{H}^{(l-1)}, \mathcal{W}_{\mathcal{V}_i}^{(l)} \mathbf{H}^{(l-1)}) \tag{5}$$

$\mathcal{W}_O$ is the output weight matrix, and [.] indicate concatenation.

This sublayer is followed by layer normalization and a residual connection:

$$\mathbf{H}_{SA}^{(l)} = \text{LayerNorm}(\mathbf{H}^{(l-1)} + \text{MultiHead}^{(l)}(\mathbf{H}^{(l-1)})) \tag{6}$$

This helps stabilize the training process and allows for deeper networks.

3.3. Position-Wise Feedforward Network

After that, the output is passed through a feedforward network:

$$\text{FFN}(\mathbf{H}_{SA}^{(l)}) = \mathcal{W}_2^{(l)} \text{ReLU}(\mathcal{W}_1^{(l)} \mathbf{H}_{SA}^{(l)} + \mathbf{b}_1^{(l)}) + \mathbf{b}_2^{(l)} \tag{7}$$

where $\mathcal{W}_1, \mathcal{W}_2$ are weight matrices, and $\mathbf{b}_1, \mathbf{b}_2$ are bias terms.

Analogously, this sublayer is followed by layer normalization and a residual connection:

$$\mathbf{H}^{(l)} = \text{LayerNorm}(\mathbf{H}_{SA}^{(l)} + \text{FFN}(\mathbf{H}_{SA}^{(l)})) \tag{8}$$

3.4. Token Extraction and Classification Head

After the input sequence is processed through the transformer encoder, the output $\mathbf{H}^*$ is reduced to the fixed size of a single token embedding $\mathbf{h}_{\text{token}}^{(0)}$. There are several approaches to performing this reduction. However, they can be reduced down to two main categorie: selecting one token from the sequence and averaging them.

The reduced embedding is then processed through an MLP neural network where each layer can be formulated as

$$\mathbf{h}_{\text{token}}^{(l)} = \sigma^{(l)}(\mathcal{W}_{\text{token}}^{(l)} \mathbf{h}_{\text{token}}^{(l-1)} + \mathbf{b}_{\text{token}}^{(l)}) \tag{9}$$

where σ is a generic activaction function. Finally, the last layer produces the classification prediction:

$$\hat{\mathbf{y}} = \text{softmax}(\mathcal{W}_{\text{token}} \mathbf{h}_{\text{token}} + \mathbf{b}_{\text{token}}) \tag{10}$$

Regarding TT, the classification head as described in this section is not utilized; instead, a custom classification head is designed to receive as input both $\mathbf{h}_{\text{token}}^{(l)}$ and topic representations. More details regarding the TT custom classification head are provided in Section 5.3.

4. Topic Models

Topic modeling is an unsupervised machine learning technique used to discover hidden thematic structures, topics **T**, within a collection of D documents, **X**. It is widely applied in areas such as document clustering, information retrieval, and text summarization. Topic models can be roughly divided into three groups [52], probabilitic-based models such as LDA; algebraic-based methods such as FLSA-W; and embedding-based models such as BERTopic. In this section, we introduce in detail these three methods. LDA was selected as a well-established probabilistic baseline. BERTopic was selected because it integrates transformer-based document representations into the topic-modeling pipeline. Finally, FLSA-W is also considered as it is a recently introduced algebraic method with promising results.

4.1. Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [29] is a generative probabilistic model used for discovering topics **T** in a collection of D documents. Each document d, where $d \in \{1, 2, \ldots, D\}$, is assumed to be a mixture of (K) topics, while each topic k, where $k \in \{1, 2, \ldots, K\}$, is represented by a probability distribution over words.

LDA assumes document d has a topic distribution:

$$\theta_d \sim \text{Dirichlet}(\alpha) \tag{11}$$

where α is a parameter controlling the sparsity of topics in documents and each topic k has a word distribution:

$$\phi_k \sim \text{Dirichlet}(\beta) \tag{12}$$

where β is a parameter controlling the sparsity of words in topics.

Let $\mathbf{W}$ denote the positional corpus of words, where W_{dn} represents the word at position n in document d. Each document contains N_d word positions, with $n \in \{1, 2, \ldots, N_d\}$. For each word position, LDA first samples a latent topic assignment $\bar{T}_{dn}$ from the document topic distribution:

$$\bar{T}_{dn} \sim \text{Categorical}(\theta_d) \tag{13}$$

then drawing a word from the corresponding topic

$$W_{dn} \sim \text{Categorical}(\phi_{\bar{T}_{dn}}) \tag{14}$$

Given a corpus of D documents consisting of observed words W, we want to estimate the hidden topic distributions θ_d, topic assignments $\bar{T}_{dn}$, and topic-word distributions ϕ_k. Hence, we aim to maximize the marginal likelihood:

$$P(\mathbf{X} \mid \alpha, \beta) = \prod_{d=1}^{D} \int p(\theta_d \mid \alpha) \prod_{n=1}^{N_d} \sum_{\bar{T}_{dn}} P(W_{dn} \mid \bar{T}_{dn}, \beta) P(\bar{T}_{dn} \mid \theta_d) \mathrm{d}\theta_d \tag{15}$$

In this algorithm, θ and ϕ are initialized randomly and then an optimization method, such as Gibbs Sampling [57] or Variational Inference [58], is used to update the distributions over time. Regarding TT, we utilized Variational Inference as our optimization method to obtain topics for LDA.

4.2. BERTopic

BERTopic [31] is a topic-modeling technique that leverages transformer-based embeddings and clustering to identify topics (see Figure 2). The key steps are as follows:

- Embedding Generation: Convert documents to dense vector embeddings.
- Dimensionality Reduction: Reduce the vector embeddings dimensionality to a predefined size.

- Document Clustering: Group similar embeddings.
- Topic Representation: Identify topic keywords.

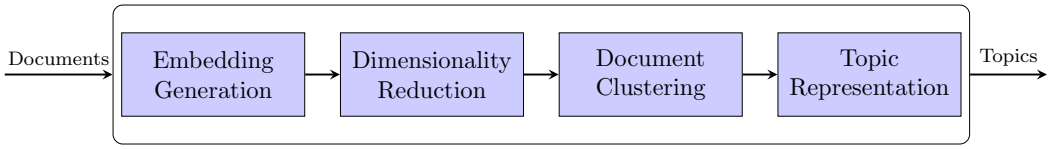


Figure 2. BERTopic block diagram illustrating its stages, the embedding generation, the dimensionality reduction, the document clustering, and the topic representation.

4.2.1. Embedding Generation

Firstly, the collection of D documents, denoted by $\mathbf{X}$, is encoded using the Sentence-BERT (SBERT) [59] transformer encoder as explained in Section 3. Each document is indexed by d, where $d \in \{1, 2, \ldots, D\}$.

$$\mathbf{e}_d = \text{SBERT}(X_d) \tag{16}$$

$\mathbf{e}_d$ corresponds to the embedding of document X_d, where f denotes the dimensionality of the embedding space.

4.2.2. Dimensionality Reduction

Having computed the embeddings for each document, Uniform Manifold Approximation and Projection (UMAP) [60] is used to reduce their dimensionality. UMAP is based on manifold learning, assuming that high-dimensional data lie on a lower-dimensional manifold embedded in a higher-dimensional space. The algorithm consists of two main steps:

- Constructing a weighted k-nearest neighbor (γ-NN) graph in high-dimensional space.
- Optimizing a low-dimensional representation of the data using a force-directed layout.

Given the set of embeddings $E = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_D\} \subset \mathbb{R}^f$, UMAP first defines a weighted graph based on local relationships between embeddings.

For each embedding $\mathbf{e}_d$, we compute pairwise distances using a distance metric $\delta(\mathbf{e}_d, \mathbf{e}_{d'})$. Instead of a hard γ-NN graph, UMAP constructs a fuzzy set representing the probability that an embedding $\mathbf{e}_{d'}$ is a neighbor of $\mathbf{e}_d$:

$$p_{dd'} = \exp\left(-\frac{\delta(\mathbf{e}_d, \mathbf{e}_{d'}) - \rho_d}{s_d}\right), \tag{17}$$

where ρ_d is the distance to the nearest neighbor,

$$\rho_d = \min_{d' \neq d} \delta(\mathbf{e}_d, \mathbf{e}_{d'}) \tag{18}$$

and s_d is a local scaling factor ensuring consistent neighborhood sizes across varying densities. s_d can be obtained by solving the following equation:

$$\sum_{d' \neq d} \exp\left(-\frac{\delta(\mathbf{e}_d, \mathbf{e}_{d'}) - \rho_d}{s_d}\right) = \log_2(\gamma) \tag{19}$$

where γ is a predefined the number of neighbors.

The symmetric edge weight is then computed as

$$p_{dd'}^{sym} = p_{dd'} + p_{d'd} - p_{dd'}p_{d'd}. \tag{20}$$

Having created the γ-NN graph, and calulated their respective probabilities $p_{dd'}^{sym}$, UMAP randomly initializes a low-dimensional embedding $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_D\} \subset \mathbb{R}^s$, where

$s < f$. In the lower-dimensional space, the probability of a connection between embeddings $\mathbf{z}_d$ and $\mathbf{z}_{d'}$ is defined as

$$q_{dd'} = \frac{1}{1 + a\|\mathbf{z}_d - \mathbf{z}_{d'}\|^{2b}}, \tag{21}$$

where a, b are hyperparameters controlling how distances are scaled in the embedding space.

Then, UMAP updates this representations by minimizing the cross-entropy between the high-dimensional and low-dimensional distributions:

$$\mathcal{L} = \sum_{d \neq d'} p^{sym}_{dd'} \log \frac{p^{sym}_{dd'}}{q_{dd'}} + (1 - p^{sym}_{dd'}) \log \frac{1 - p^{sym}_{dd'}}{1 - q_{dd'}}. \tag{22}$$

This loss function is optimized using stochastic gradient descent:

$$\mathbf{z}_d \leftarrow \mathbf{z}_d - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{z}_d}, \tag{23}$$

where η is the learning rate, $\leftarrow$ is the update operator.

4.2.3. Document Clustering

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [61] is used in BERTopic for clustering textual embeddings after dimensionality reduction with UMAP. HDBSCAN extends DBSCAN by allowing clusters of varying densities without requiring a predefined number of topics. This enables a fully data-driven identification of dense topic clusters while filtering out irrelevant documents as noise.

The algorithm begins by computing the mutual reachability distance δ_{reach} between every pair of reduced embeddings $\mathbf{z}_d$ and $\mathbf{z}_{d'}$:

$$\delta_{reach}(\mathbf{z}_d, \mathbf{z}_{d'}) = \max(\delta_{euc}(\mathbf{z}_d, \mathbf{z}_{d'}), \delta_{core}(\mathbf{z}_d), \delta_{core}(\mathbf{z}_{d'})) \tag{24}$$

where the core-distance of an embedding $\delta_{core}(\mathbf{z}_d)$ is defined as

$$\delta_{core}(\mathbf{z}_d) = \max_{\mathbf{z}_{d'} \in \mathcal{N}_\gamma(\mathbf{z}_d)} \delta_{euc}(\mathbf{z}_d, \mathbf{z}_{d'}) \tag{25}$$

where $\mathcal{N}_\gamma(\mathbf{z}_d)$ corresponds to the γ-closest points to $\mathbf{z}_d$

Having calculated δ_{reach} for all pairs of reduced embeddings, a minimum spanning tree (MST) is built based on it. The MST connects all points while minimizing the total edge weight, serving as a backbone for clustering. Hierarchical clustering is then performed by iteratively removing the longest edges in the MST, progressively segmenting the data into clusters and forming a dendrogram.

The resulting dendogram is then condensed by removing clusters that do not meet the minimum cluster size requirement. This step reduces the complexity and focuses on significant clusters.

From the reduced dendogram, the density associated to every relevant connection is calculated as

$$\lambda(\mathbf{z}_d, \mathbf{z}_{d'}) = \frac{1}{\delta_{reach}(\mathbf{z}_d, \mathbf{z}_{d'})} \tag{26}$$

and the stability of each cluster of the dendogram is calculated as

$$S(C) = \sum_{\mathbf{z}_d \in C} \lambda(\mathbf{z}_d, C) - \lambda_{birth}(C) \tag{27}$$

where the density $\lambda_{birth}(C)$ represents the level at which removing connections with corresponding value causes a parent cluster to split into two; and $\lambda(\mathbf{z}_d, C)$ represents the level at which removing connections with corresponding value causes cluster C to split into two (or more clusters), or causes $\mathbf{z}_d$ to leave cluster C, whichever is lower.

Stability measures how long a cluster persists as the density threshold changes, favoring clusters that remain stable over a wide range of densities.

The selected clusters are the ones which maximize the stability for the entire set of embeddings, provided that no point can be in more than one cluster. In this sense, it is important to clarify that it is possible for a point not to be included in any cluster, and be considered as noise. The set of selected clusters will work as topics.

4.2.4. Topic Representation

Class-based Term Frequency-Inverse Document Frequency (c-TF-IDF) is an extension of the traditional TF-IDF weighting scheme. TF-IDF is commonly used in text mining and information retrieval to evaluate the importance of a term in a document relative to a collection of documents. c-TF-IDF adapts this concept by considering entire classes of documents rather than individual documents.

Let $\bar{\mathbf{W}}$ denote the corpus of (M) unique words, where $m \in \{1, 2, \dots, M\}$ indexes each unique word $\bar{W}_m$. Following clustering, let K denote the number of identified topics and $k \in \{1, 2, \dots, K\}$ the topic index. c-TF-IDF is computed as

$$\text{c-TF-IDF}(\bar{W}_m, T_k) = \text{TF}(\bar{W}_m, T_k)\text{IDF}(\bar{W}_m) \tag{28}$$

where:

- $\text{TF}(\bar{W}_m, T_k)$ is the absolute term frequency of term m in class (topic) k.
- $\text{IDF}(\bar{W}_m)$ is the inverse class frequency of term m across all classes (topics).

$$\text{IDF}(\bar{W}_m) = \log\left(\frac{K}{1 + DF(\bar{W}_m)}\right) \tag{29}$$

where K is the total number of classes, topics, and $DF(\bar{W}_m)$ is the number of classes containing term $\bar{W}_m$.

In BERTopic, c-TF-IDF is utilized to enhance the interpretability of the cluster-based topics by associating to each topic probalities to each term. In TT, this step is not utilized, as the topic modeling is used to extract topic representations, which are provided in HDBSCAN.

4.3. Fuzzy Latent Semantic Analysis-Words

Fuzzy Latent Semantic Analysis-Words (FLSA-W) [62] is a topic-modeling technique that extends the traditional FLSA method [30]. This method, FLSA-W, is set apart from other cluster-based topic models since it performs clustering on the corpus of words present in a set of documents rather than clustering the documents themselves.

Given a set of (D) documents $\mathbf{X}$ and a corpus of M unique words $\bar{\mathbf{W}}$, we begin by calculating local term weights (LTW) and global term weights (GTW) matrices. Each document is indexed by $d \in \{1, 2, \dots, D\}$, and each unique word is indexed by $m \in \{1, 2, \dots, M\}$. Having the LTW matrix, GTW can be calculated using metrics such as Entropy [63], Normal [64], Inverse Document Frequency [65], and Probabilistic Inverse Document Frequency [64].

Having calculated both the local and global term weights, the FLSA-W method assumes that the $P(\bar{W}_m, X_d)$ is given by

$$P(\bar{W}_m, X_d) = \text{LTW}(\bar{W}_m, X_d)\text{GTW}(\bar{W}_m). \tag{30}$$

This matrix $P(\bar{\mathbf{W}}, \mathbf{X})$ is decomposed using Singular Value Decomposition [66] into matrices U, Λ, and V. In the FLSA-W algorithm—contrary to other cluster-based topic models which cluster documents to obtain topics—clustering is applied to the words instead. In more detail, the fuzzy c-means algorithm [67] is applied to matrix V to obtain $P(T_k|\bar{W}_m)$, probability of topic k, given word m.

Having obtained $P(T_k|\bar{W}_m)$, the probability of each document, word, and topic are given by

$$P(X_d) = \frac{\sum_{m=1}^{M} P(\bar{W}_m, X_d)}{\sum_{m=1}^{M}\sum_{d=1}^{D} P(\bar{W}_m, X_d)} \tag{31}$$

$$P(\bar{W}_m) = \frac{\sum_{d=1}^{D} P(\bar{W}_m, X_d)}{\sum_{m=1}^{M}\sum_{d=1}^{D} P(\bar{W}_m, X_d)} \tag{32}$$

$$P(T_k) = P(T_k|\bar{W}_m)P(\bar{W}_m). \tag{33}$$

The probability of word m given topic k is, then, given by

$$P(\bar{W}_m, T_k) = P(T_k|\bar{W}_m)P(\bar{W}_m), \tag{34}$$

$$P(\bar{W}_m|T_k) = \frac{P(\bar{W}_m, T_k)}{\sum_{m=1}^{M} P(\bar{W}_m, T_k)}. \tag{35}$$

And finally, the probability of topic k given document d is given by

$$P(\bar{W}_m|X_d) = \frac{P(\bar{W}_m, X_d)}{\sum_{m=1}^{M} P(\bar{W}_m, X_d)}, \tag{36}$$

$$P(X_d|\bar{W}_m) = \frac{P(\bar{W}_m|X_d)P(X_d)}{P(\bar{W}_m)}, \tag{37}$$

$$P(X_d|T_k) = \sum_{m=1}^{M} P(X_d|\bar{W}_m)P(\bar{W}_m|T_k), \tag{38}$$

$$P(T_k|X_d) = \frac{P(X_d|T_k)P(T_k)}{P(X_d)}. \tag{39}$$

A trained FLSA-W model stores the distributions $P(T_k)$, $P(\bar{W}_m, T_k)$, and $P(\bar{W}_m)$. Therefore, when inferring topic distributions for previously unseen test samples, including those in TT, the model only needs to recompute $P(\bar{W}_m, X_d)$ and $P(X_d)$ using Equations (30) and (31) and then to apply Equations (36)–(39) to obtain $P(T_k \mid X_d)$.

5. Topic-Transformer Framework

In this section, the TT architecture is thoroughly presented (see Figure 3). TT has three main components which are the Topic Model (see Section 5.1), where text is converted into topic representations, the Transformer Encoder (see Section 5.2), where the text is encoded into contextual embeddings, and the Classification Head (see Section 5.3), which classifies text based on features the two other components extracted from it.

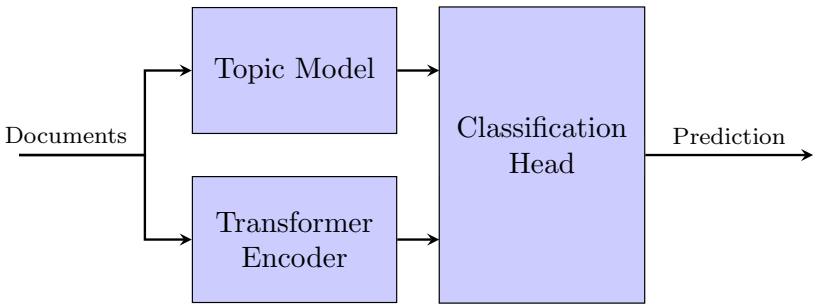


Figure 3. Topic-transformer framework block diagram illustrating its main components, the topic modal, the transformer encoder, and the classification head.

5.1. Topic Modeling

In the TT framework, different topic models can be used to generate a topic representation for a given text. Nevertheless, depending on the selected model, the text may require preprocessing (see Figure 4).

For instance, transformer-based topic models like BERTopic leverage pretrained transformer embeddings which handle tokenization and essential preprocessing. This eliminates the need for additional data treatment, as the topic model handles the text processing internally.

In contrast, topic models like LDA and FLSA-W operate differently. These models typically rely on bag-of-words representations, where words are treated as discrete tokens, and the relationships between them are modeled based on co-occurrence patterns. Although, these models do not inherently require preprocessing, they typically benefit from techniques such as stop word removal and lemmatization, as these reduce vocabulary size, which in turn decreases both dimensionality and sparsity in document–term matrices, consequently lowering memory usage and computational complexity during training and inference [68,69].

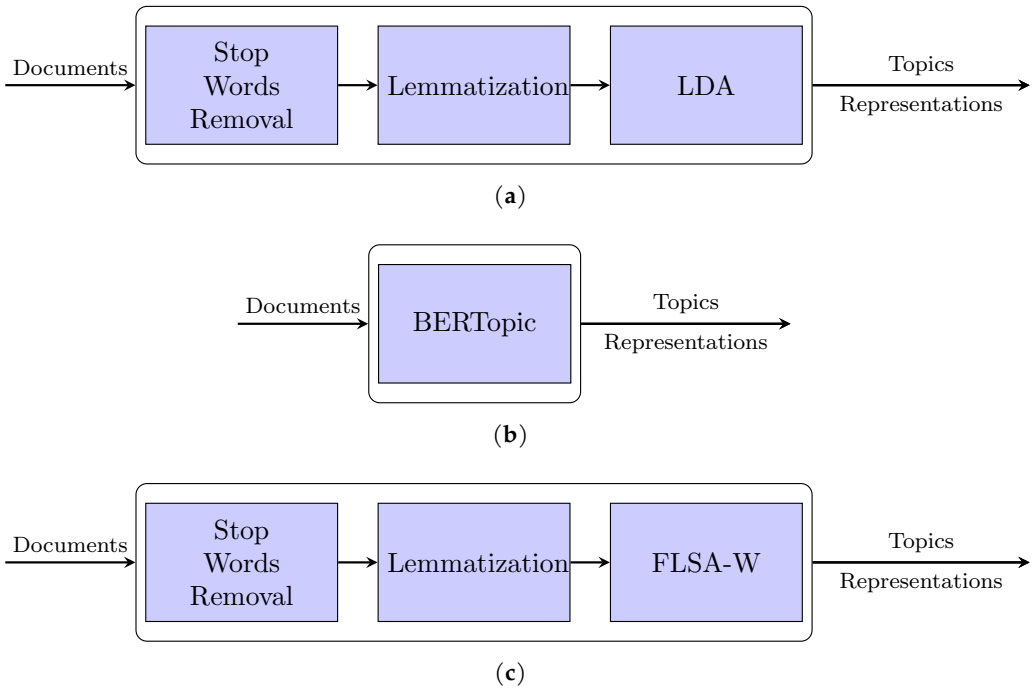


Figure 4. Topic-modeling pipeline for (**a**) Latent Dirichlet analysis; (**b**) BERTopic; (**c**) Fuzzy latent semantic analysis-words.

5.1.1. Stop Words Removal

In NLP, a stop word is a frequently occurring word in a text that does not contribute much to its overall meaning. Common stop words include articles (e.g., "the", "a", "an"), conjunctions (e.g., "and", "or"), and some verbs and their respective conjugations (e.g., "is", "be") [70]. While these words are essential for constructing grammatical sentences, they often do not hold significant value for tasks like topic modeling, where the focus is on uncovering the core subjects of a document. Removing stop words can help reduce the noise in the data, allowing the model to focus on more meaningful terms.

5.1.2. Lemmatization

Lemmatization [71] is another preprocessing step, designed to enhance BoW models. Lemmatization involves reducing words to their base form (or lemma) while considering

the word meaning and grammatical role. For instance, the words "running," "ran," and "runs" are all mapped to the lemma "run," which allows the model to recognize these variations as a single concept. This process not only enhances consistency but also improves model performance by reducing the dimensionality of the feature space, and therefore sparsity. By ensuring that all variations of a word are treated as one, it aids the topic models to identify more coherent and meaningful topics.

5.2. Transformer Encoder

In parallel with topic modeling, the text is also processed by a transformer encoder. As described in Section 3, the text undergoes the following:

- Tokenization: The input text is first tokenized, meaning it is segmented into smaller units, or tokens, according to the vocabulary of the transformer model. These tokens can correspond to words or sub-words. Each token is then mapped to its corresponding embedding.
- Encoding: The token embeddings are processed by the transformer encoder, which generates contextualized representations of the input sequence.
- Dimensionality Reduction: The encoder output is transformed into a fixed-size feature vector, $\mathbf{h}^{(0)}_{\text{token}}$.

5.3. Classification Head

The TT classification head combines information from the transformer-based text encoder and the topic model to classify the corresponding text. As illustrated in Figure 5, it employs model-level fusion to process two distinct views of the input text: (i) the topic representation from the topic model, $\mathbf{h}^{(0)}_{\text{topic}}$, and (ii) the token embedding from the transformer encoder, $\mathbf{h}^{(0)}_{\text{token}}$. Each representation is processed independently through a dedicated multilayer perceptron, allowing the classification head to learn view-specific transformations that capture features relevant to the classification task. Processing the representations separately before fusion preserves their distinct characteristics while enabling each to be mapped into a feature space suited to downstream classification. The resulting representations are then concatenated to form a joint representation that integrates the complementary thematic and contextual information captured by the topic model and the transformer encoder, respectively. This fused representation is subsequently passed through additional dense layers, which learn interactions between the two views and map the integrated features to the target label space to produce the final prediction.

Both inputs are processed through separate stacks of dense layers, where each layer is formulated generically as

$$\mathbf{h}^{(l)}_{\text{topic}} = \sigma^{(l)}(\mathcal{W}^{(l)}_{\text{topic}}\mathbf{h}^{(l-1)}_{\text{topic}} + \mathbf{b}^{(l)}_{\text{topic}}) \tag{40}$$

$$\mathbf{h}^{(l)}_{\text{token}} = \sigma^{(l)}(\mathcal{W}^{(l)}_{\text{token}}\mathbf{h}^{(l-1)}_{\text{token}} + \mathbf{b}^{(l)}_{\text{token}}) \tag{41}$$

where $\mathcal{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the layer-specific weight matrices and bias vectors, respectively.

The resulting representations are concatenated:

$$\mathbf{h}^{(0)}_{\text{fused}} = [\mathbf{h}^{*}_{\text{token}}; \mathbf{h}^{*}_{\text{topic}}] \tag{42}$$

and further processed through another stack of dense layers:

$$\mathbf{h}^{(l)}_{\text{fused}} = \sigma^{(l)}(\mathcal{W}^{(l)}_{\text{fused}}\mathbf{h}^{(l-1)}_{\text{fused}} + \mathbf{b}^{(l)}_{\text{fused}}) \tag{43}$$

Finally, the last layer produces the classification prediction:

$$\hat{\mathbf{y}} = \text{softmax}(\mathcal{W}^{*}_{\text{fused}}\mathbf{h}^{*}_{\text{fused}} + \mathbf{b}^{*}_{\text{fused}}) \tag{44}$$

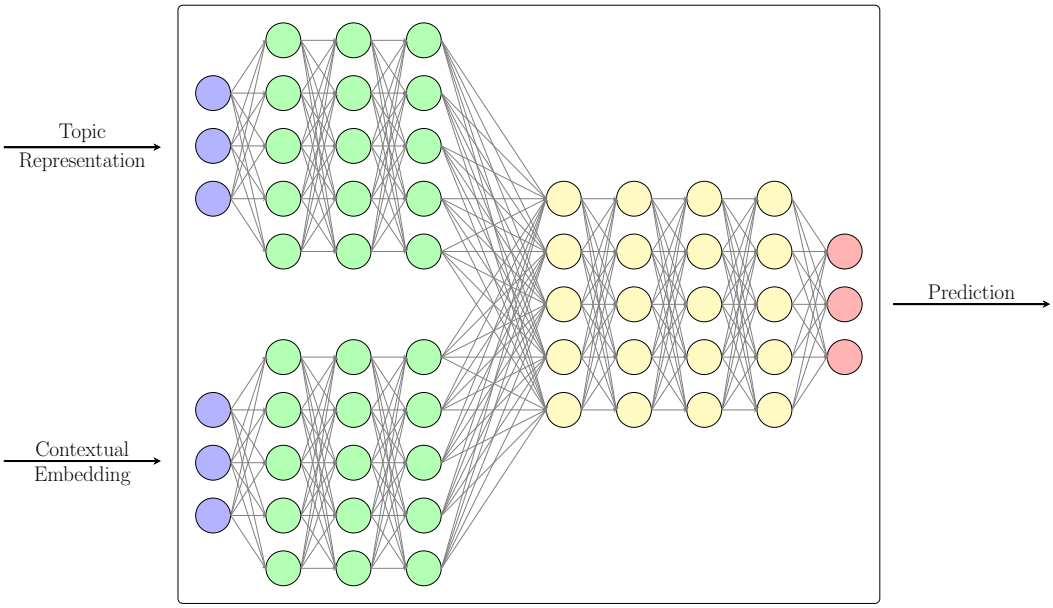


Figure 5. Illustrative Classification Head model-level fusion TT.

6. Experimental Setup and Results

6.1. Data Description

The TT framework was tested on six distinct English datasets, UCI SMS Spam Collection (Sms) [7], Stormfront Hate Speech Dataset (Stormfront) [72], BESSTIE reddit for sentiment analysis (BESSTIE Sent) [73], BESSTIE reddit for sarcasm detection (BESSTIE Sarc) [73], BBC News Dataset [74], and 20Newsgroups (20News) [75]. These datasets were chosen for their diverse characteristics, such as text classification task, number of classes, number of samples, text length, style of writing, dialects variety, and typos. The main characteristics of each dataset are show in Table 1. The remainder of this section includes additional information on each of the datasets.

Table 1. Datasets description.

Dataset	No. Samples	No. Classes	Class Rep. Min	Class Rep. Max	Avg. Text Length	Vocab. Size	Task
Sms	5572	2	13.4%	86.4%	14.5	8713	Spam Det.
Stormfront	10,703	2	11.2%	88.8%	15.4	15,871	Hate Speech Det.
BESSTIE Sent	5064	2	24.8%	75.2%	37.9	17,243	Sentiment Anal.
BESSTIE Sarc	5092	2	26.6%	73.4%	37.3	17,181	Sarcasm Det.
BBC News	2227	5	17.3%	22.9%	370.7	29,275	Document Cat.
20News	18,846	20	3.3%	5.3%	182.7	134,410	Document Cat.

Rep.—Representation, Avg.—Average, Vocab.—Vocabulary, Det.—Detection, Cat.—Categorization, Anal.—Analysis.

The UCI SMS Spam Collection is a publicly available corpus of SMS messages collected from multiple sources. Each message is labeled as either spam or ham (non-spam).

The Stormfront Hate Speech Dataset was created by extracting from 5000 posts from the white supremacist forum Stormfront. These posts were then divided into sentences, resulting in 10,703 sentences. Each sentence was manually annotated as either hateful or non-hateful.

The BESSTIE Reddit datasets (for sentiment analysis and sarcasm detection) each contain approximately 5000 Reddit posts. Both are binary classification datasets and include posts written in three English dialects: British English, Australian English, and Indian

English. These datasets were incorporated to evaluate the framework capabilities in contemporary social media settings, particularly in scenarios where linguistic variation is present.

The BBC News Dataset comprises 2225 news articles categorized into five topics: business, entertainment, politics, sport, and tech. This dataset serves as a benchmark for article categorization tasks, testing a model's ability to distinguish between different types of news content.

The 20Newsgroups dataset contains approximately 20,000 documents, evenly distributed across 20 different newsgroups. This dataset was included to test the framework's capabilities in many-class classification tasks.

6.2. Preliminaries

6.2.1. Topic Specificity

In topic modeling, one crucial aspect of model configuration is determining the optimal number of topics. This decision can significantly affect the quality and interpretability of the generated topics.

Topic specificity, introduced by Rijcken et al. [76], is a descriptive metric that quantifies the specificity of the words characterizing a set of topics. It is based on the assumption that words occurring less frequently throughout a corpus tend to be more specific to particular domains or subject matters than words occurring frequently across different contexts. Consequently, topic specificity can be used to compare topic models and investigate how the number of topics affects the granularity of the resulting representations. Given a prediction task, it is beneficial to maximize topic specificity as the number of topics should be selected based on topic granularity. This approach ensures that the topics generated can capture highly focused themes and provide clear distinctions between different aspects of the text.

Let $\mathbf{T} = \{T_1, T_2, \ldots, T_K\}$ denote the set of topics extracted from the corpus, where each topic T_k is represented by a word-propensity distribution $P(\bar{W}_m \mid T_k)$ over the vocabulary $\bar{\mathbf{W}}$. For each topic, we select the $\mathcal{L}$ words with the highest propensity scores, defining the topic descriptor as

$$T_k^* = \{\bar{W}_m \mid \bar{W}_m \text{ranks in the top } \mathcal{L} \text{ of } P(\bar{W}_m \mid T_k)\} \tag{45}$$

Here, T_k^* denotes the set of descriptor words for topic k.

Considering that a word specificity is inversely related to its frequency in the corpus, words that occur less frequently are treated as more specific. Let $\Phi(\bar{W}_m)$ assign the word its rank in the vocabulary sorted by increasing frequency. The overall topic specificity score TS is calculated as the mean word-specificity score across the M descriptor words of all K topics:

$$\text{TS} = \frac{1}{K\mathcal{L}} \sum_{k=1}^{K} \sum_{\bar{W}_m \in T_k^*} \frac{\Phi(\bar{W}_m)}{M} \tag{46}$$

where $\frac{\Phi(\bar{W}_m)}{M}$ is the word-specificity score.

6.2.2. Evaluation Metrics

Throughout this section, inference performance will be reported using three metrics, F1-score, precision, and recall. Accuracy was not included among the reported metrics because the studied datasets exhibit varying degrees of class imbalance, making accuracy unsuited to characterize classification performance.

Precision measures the proportion of instances predicted as positive that are correctly classified, whereas recall measures the proportion of actual positive instances that are correctly identified. These metrics are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{47}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{48}$$

where TP (true positives) denotes the number of positive instances correctly classified, FP (false positives) denotes the number of negative instances incorrectly classified as positive, and FN (false negatives) denotes the number of positive instances incorrectly classified as negative.

The F1-score combines precision and recall into a single measure by calculating their harmonic mean. It can be expressed in terms of precision and recall as

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}. \tag{49}$$

Nevertheless, it is crucial to consider that these metrics assume binary datasets where one of the classes is considered as the positive one. As a result, using these three metrics without any additional treatment would be nonviable for evaluation. In this study, all reported F1-scores, precision, and recall values correspond to macro-averaged values. Macro-averaging refers to the process of computing each metric for each class separately, considering that class as positive and the remaining classes as negative, and then averaging the results across all classes without assigning weights. This ensures that each class contributes equally to the final score, regardless of its representation in the dataset.

6.3. Experimental Setup

In our experiments, fifteen configurations of the TT framework were evaluated across the six selected datasets. In this sense, a TT configuration is defined as the assembly of a topic model (including any required preceding processing steps), a transformer encoder, and a classification head. These configurations were obtained by combining five different transformer encoders—DistilBERT-base [25], BERT-base [20], RoBERTa-base [23], XLNet-base [24], and the T5 encoder [21]—with three different topic-modeling techniques—LDA [29], BERTopic [31], and FLSA-W [62].

Since LDA and FLSA-W require a predefined number of topics, a preliminary experiment was conducted to determine the appropriate number of topics for each dataset. A grid search was performed, and models were evaluated based on topic specificity [76]. BERTopic was excluded from this step, as it employs HDBSCAN, a hierarchical clustering algorithm that automatically determines the number of topics (clusters) for the given data.

All experiments were conducted using a stratified 10-fold cross-validation approach to ensure robustness on our findings. Additionally, special precaution was taken in regards to the Stormfront Hate Speech Dataset since different sentences from the same post could cause data leakage. To address this issue, it was ensured that all sentences from each post would be contained within the same fold.

Training was performed in a modular manner. Transformer encoders, topic models, and classification heads were trained separately, allowing topic models and encoders to be reused across different TT configurations within each dataset partition. For each fold, only one instance of each topic model and encoder was trained, while each TT configuration had its own classification head. During training, transformer encoders retained their original classification heads, though these were not used in any TT configurations.

In addition to the complete TT configurations, ablation configurations were also evaluated to assess the individual contributions of the transformer encoders and topic representations. These included transformer encoders used with the TT classification head, without topic representations, and topic models followed by the TT classification head. To ensure a fair comparison between the ablation baselines and the complete TT framework, all configurations were trained in an identical manner, inputting zeros in place of missing

signals. Most notably, the transformer-only configurations discarded their original classification heads after the initial training stage, replacing them with TT classification heads that were subsequently trained. This ensured consistency in classification-head architecture, dimensionality, and training procedure, thereby minimizing the influence of these factors on the observed performance differences.

The performance of all TT configurations was evaluated using four metrics: macro-averaged F1-score, macro-averaged precision, macro-averaged recall, and training time. The reported results include the mean and standard deviation of these metrics across the 10 test sets.

Additionally, to assess whether the TT configurations significantly outperformed their corresponding ablation configurations with standalone transformer with the TT classification head, paired-samples t-tests were conducted exclusively on their fold-level F1-scores. To account for multiple hypothesis testing, the Holm correction was applied to the p-values within each dataset. Accordingly, six separate families of hypotheses were considered, one for each dataset, with each family comprising all hypotheses tested for that dataset. Statistical significance was assessed at a significance level of $\alpha = 0.05$ using the Holm-adjusted p-values. In addition, Cohen's d was calculated to quantify the effect size of the observed differences, using the mean and standard deviation of the paired differences, and 95% confidence intervals were computed for the corresponding effect-size estimates.

All neural network-based experiments, including transformer-based models and the SBERT component of BERTopic, were executed on an NVIDIA GeForce RTX 3050 GPU using PyTorch 2.5, while all other models and preprocessing tasks were performed on an 11th Generation Intel(R) Core(TM) i9-11900 CPU. Within this setup, all transformer encoders were fine-tuned using the Adam optimizer with a maximum sequence length of 128 tokens, batch size was fixed to 64, and random seeds were fixed to 42. The TT classification head is composed of two parallel branches followed by a joint block. The topic branch processes the topic representation through three dense layers of 768 neurons each, while the transformer branch processes the extracted token embedding through two dense layers of 768 neurons each. The outputs of both branches are then concatenated and passed through three additional dense layers of 768 neurons each, and a final softmax output layer produces the class probabilities. All hidden dense layers use ReLU activation as its standard in transformer classification heads. Additionally, dropout is also used with a rate of 0.1. Stop word removal was handled with spaCy, and the FLSA-W and LDA implementations were taken from the fuzzyTM [77] and scikit-learn libraries, respectively. Additional hyperparameters, such as learning rates, number of epochs, and other topic model hyperparameters (BERTopic, LDA, and FLSA-W), were set per dataset; all values are available in our public repository at https://doi.org/10.6084/m9.figshare.34346568.

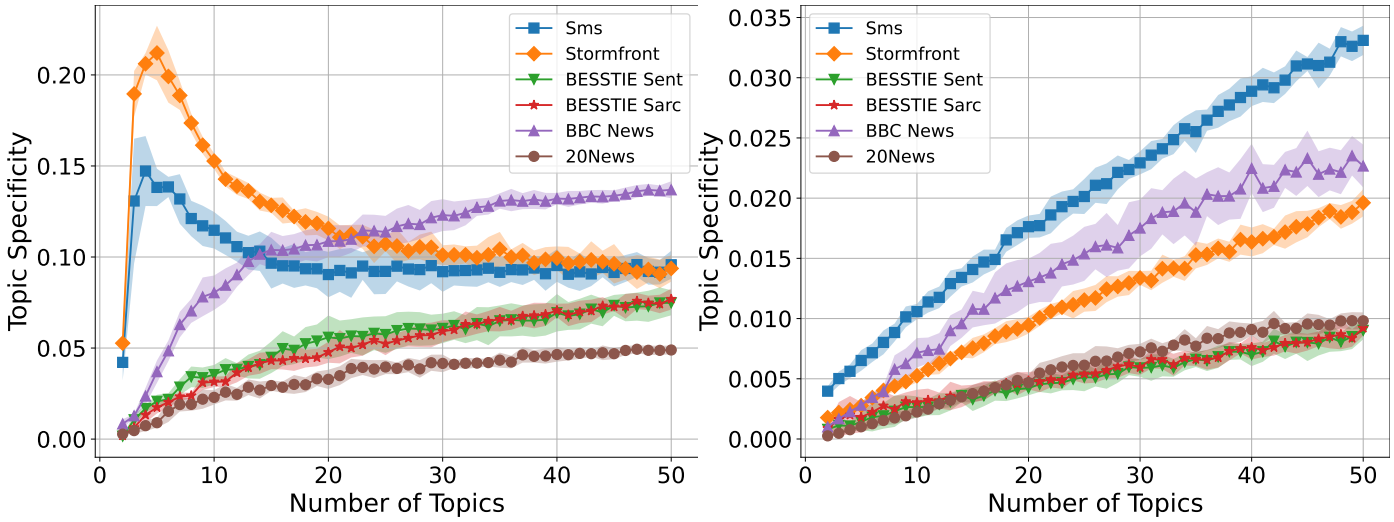
6.4. Number of Topics Selection

We trained several LDA and FLSA-W models, performing a grid search from 2 to 50 topics for each dataset, and evaluated them in terms of topic specificity with $\mathcal{L} = 10$. The obtained results can be found in Figure 6.

The obtained results show that, in general, as the number of topics increase, the topic specificity tends to increase as well. That is in line with what is expected, as the more topics a model has, the more specific they become.

Nevertheless, LDA and FLSA-W exhibit very distinctive behavior as the number of topics increase. Regarding the BESSTIE Sent, BESSTIE Sarc, the BBC News, and the 20News datasets, the observed behavior is similar since as the number of topics increase, the slope of the curve decreases, still more significantly in FLSA-W than in LDA. Regarding the two other datasets, very different behavior can be observed comparing the two models, as the LDA

curves are mostly monotonous, whereas the FLSA-W are not, clearly having a maximum within the studied domain.



(**a**) (**b**)

Figure 6. Topic results (mean ± standard deviation): (**a**) Fuzzy latent semantic analysis-words; (**b**) Latent Dirichlet allocation

Finally, the number of topics was selected by balancing two competing factors. On one hand, higher topic specificity is desirable, and as shown in Figure 6, topic specificity generally increases with the number of topics. On the other hand, a larger number of topics implies a higher computational demand. Taking both factors into account, the number of topics was selected manually for each combination of fold and topic model (LDA and FLSA-W) by inspecting the topic specificity curve obtained for that specific fold on its training partition only. The combined curves in Figure 6 are shown for illustrative purposes only and were not used for selection, as doing so would leak information from the test into the choice of hyperparameters. The resulting numbers of topics are reported in Table 2.

Table 2. Selected number of topics.

Dataset	**Topic Model**	**Fold**									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Sms	FLSA-W	6	4	3	5	3	6	5	4	5	4
	LDA	48	49	46	50	44	48	46	49	50	48
Stormfront	FLSA-W	6	4	5	5	6	4	5	5	5	5
	LDA	39	49	47	50	50	50	48	50	47	50
BESSTIE Sent	FLSA-W	42	27	24	22	45	45	40	49	40	20
	LDA	49	47	44	42	50	41	47	45	45	49
BESSTIE Sarc	FLSA-W	32	32	21	24	45	37	43	44	35	38
	LDA	48	48	48	44	49	48	50	44	50	47
BBC	FLSA-W	38	29	29	31	27	30	36	33	24	24
	LDA	40	45	49	47	41	49	40	33	45	49
20News	FLSA-W	22	27	29	48	22	37	22	25	35	47
	LDA	46	45	50	36	42	42	42	42	48	49

6.5. Results

Having selected the number of topics LDA and FLSA-W, all models, including transformers, topic models, and classification heads, were trained separately, and then assembled as explained in Section 6.3. The obtained results are shown in Tables 3–8. The resulting *p*-values are reported in Table 9; significant values (lesser or equal than 0.05) are highlighted in bold.

Table 3. Results on Sms (mean ± standard deviation).

Transfomer	Topic Model	F1 (%)	Precision (%)	Recall (%)	Training Time (s)
-	BERTopic	**81.8 ± 4.3**	78.3 ± 4.8	**89.9 ± 2.3**	**10.0 ±0.3**
	FLSA-W	44.0 ± 6.8	47.5 ± 3.1	48.8 ± 3.0	29.4 ± 3.8
	LDA	80.3 ± 2.9	**79.3 ± 3.8**	81.9 ± 2.7	25.9 ± 0.1
BERT	-	98.0 ± 1.0	97.7 ± 1.1	**98.3 ± 1.6**	**399.8 ± 1.6**
	BERTopic	**98.3 ± 1.0**	**98.6 ± 1.2**	98.0 ± 1.5	408.2 ± 1.5
	FLSA-W	98.1 ± 1.1	98.4 ± 1.0	97.9 ± 1.7	427.6 ± 3.9
	LDA	97.9 ± 1.5	97.9 ± 2.2	98.0 ± 1.6	424.1 ± 1.7
DistilBERT	-	97.9 ± 0.8	97.7 ± 1.0	**98.2 ± 1.1**	**205.2 ± 0.4**
	BERTopic	**98.1 ± 0.7**	**98.4 ± 1.0**	97.9 ± 1.4	217.6 ± 0.4
	FLSA-W	**98.1 ± 0.7**	98.3 ± 0.8	97.8 ± 1.2	236.9 ± 4.0
	LDA	98.0 ± 0.7	98.0 ± 0.9	98.0 ± 1.3	233.5 ± 0.4
RoBERTa	-	**98.6 ± 0.8**	98.6 ± 0.9	**98.7 ± 1.4**	**417.1 ± 51.2**
	BERTopic	**98.6 ± 0.8**	98.9 ± 0.6	98.2 ± 1.4	425.5 ± 51.2
	FLSA-W	**98.6 ± 0.8**	**99.0 ± 0.6**	98.4 ± 1.4	444.9 ± 51.3
	LDA	**98.6 ± 0.8**	**99.0 ± 0.7**	98.3 ± 1.4	441.4 ± 51.2
T5	-	98.2 ± 1.2	**99.0 ± 0.7**	97.5 ± 1.9	**977.3 ± 24.7**
	BERTopic	**98.4 ± 0.9**	98.9 ± 0.7	97.9 ± 1.4	985.7 ± 24.6
	FLSA-W	**98.4 ± 0.9**	98.9 ± 0.7	**98.0 ± 1.4**	1005.1 ± 24.2
	LDA	**98.4 ± 0.9**	98.9 ± 0.7	97.9 ± 1.4	1001.6 ± 24.7
XLNet	-	97.9 ± 1.0	97.7 ± 1.0	**98.1 ± 1.5**	**1553.7 ± 54.5**
	BERTopic	**98.3 ± 1.0**	98.6 ± 0.8	97.9 ± 1.5	1562.1 ± 54.5
	FLSA-W	98.1 ± 0.9	98.4 ± 0.6	97.9 ± 1.5	1581.5 ± 54.8
	LDA	**98.3 ± 1.0**	**98.8 ± 0.6**	97.9 ± 1.5	1578.0 ± 54.5

- refers to the absence of a TT component, resulting in an ablation configuration.

Table 4. Results on Stormfront (mean ± standard deviation).

Transfomer	Topic Model	F1 (%)	Precision (%)	Recall (%)	Training Time (s)
-	BERTopic	43.3 ± 11.7	40.5 ± 12.3	**50.0 ± 0.0**	**25.3 ± 8.8**
	FLSA-W	**47.0 ± 0.0**	**44.4 ± 0.1**	**50.0 ± 0.0**	49.3 ± 1.2
	LDA	39.7 ± 15.6	36.7 ± 16.4	**50.0 ± 0.0**	50.0 ± 0.2
BERT	-	76.6 ± 2.6	73.1 ± 2.5	83.3 ± 2.9	**611.8 ± 32.3**
	BERTopic	**76.7 ± 2.0**	73.1 ± 1.9	**83.4 ± 2.6**	634.8 ± 32.0
	FLSA-W	76.6 ± 2.2	73.1 ± 2.2	83.3 ± 2.7	658.8 ± 32.4
	LDA	**76.7 ± 2.1**	**73.2 ± 1.8**	83.2 ± 3.1	659.5 ± 32.2
DistilBERT	-	75.3 ± 2.0	71.5 ± 1.7	**84.2 ± 2.4**	**308.9 ± 0.4**
	BERTopic	**76.2 ± 2.1**	**72.6 ± 2.0**	83.6 ± 2.1	328.5 ± 8.6
	FLSA-W	**76.2 ± 1.9**	72.5 ± 1.8	83.6 ± 2.5	352.5 ± 1.5
	LDA	**76.2 ± 2.2**	**72.6 ± 2.2**	83.7 ± 2.3	353.2 ± 0.4
RoBERTa	-	76.8 ± 1.8	73.3 ± 2.1	84.0 ± 2.3	**675.3 ± 82.9**
	BERTopic	**77.4 ± 1.7**	**73.7 ± 1.6**	**84.5 ± 1.9**	698.2 ± 82.1
	FLSA-W	77.3 ± 1.6	73.6 ± 1.5	84.4 ± 1.9	722.2 ± 83.0
	LDA	77.3 ± 1.6	73.6 ± 1.5	84.3 ± 1.8	722.9 ± 82.8

Table 4. *Cont.*

Transfomer	Topic Model	F1 (%)	Precision (%)	Recall (%)	Training Time (s)
T5	-	78.0 ± 2.2	**78.9 ± 5.7**	79.2 ± 4.7	**1845.9 ± 56.7**
	BERTopic	**78.8 ± 1.3**	77.4 ± 2.8	81.0 ± 2.2	1871.5 ± 55.2
	FLSA-W	**78.8 ± 1.4**	77.1 ± 2.6	81.3 ± 2.1	1895.5 ± 56.4
	LDA	78.7 ± 1.5	77.0 ± 2.9	**81.5 ± 2.1**	1896.2 ± 56.7
XLNet	-	75.7 ± 2.5	73.1 ± 2.0	**80.0 ± 3.8**	**5931.4 ± 24.6**
	BERTopic	**76.4 ± 2.4**	**74.9 ± 1.9**	79.1 ± 4.8	5953.6 ± 31.1
	FLSA-W	76.2 ± 2.8	74.6 ± 2.3	78.9 ± 4.7	5977.6 ± 23.8
	LDA	76.3 ± 3.0	74.7 ± 2.5	78.8 ± 4.6	5978.2 ± 24.4

- refers to the absence of a TT component, resulting in an ablation configuration.

Table 5. Results on BESSTIE Sent (mean ± standard deviation).

Transfomer	Topic Model	F1 (%)	Precision (%)	Recall (%)	Training Time (s)
-	BERTopic	**64.7 ± 2.9**	**64.8 ± 3.0**	**66.7 ± 2.0**	**16.4 ± 11.9**
	FLSA-W	38.2 ± 9.6	41.6 ± 9.1	50.1 ± 0.2	43.9 ± 6.2
	LDA	46.7 ± 6.3	54.2 ± 1.7	55.2 ± 2.2	35.6 ± 0.2
BERT	-	79.5 ± 2.2	79.1 ± 3.1	80.9 ± 1.9	**287.1 ± 0.2**
	BERTopic	80.5 ± 1.8	79.2 ± 2.0	82.3 ± 1.5	303.6 ± 11.8
	FLSA-W	**80.6 ± 1.5**	79.3 ± 1.6	82.4 ± 1.3	331.1 ± 6.1
	LDA	**80.6 ± 1.9**	**79.4 ± 2.0**	**82.5 ± 1.6**	322.9 ± 0.4
DistilBERT	-	78.8 ± 2.2	77.4 ± 2.2	**81.3 ± 2.2**	**153.3 ± 0.5**
	BERTopic	**79.1 ± 2.4**	**77.8 ± 2.5**	81.2 ± 2.2	170.9 ± 12.2
	FLSA-W	78.7 ± 2.7	77.4 ± 2.8	81.1 ± 2.4	198.3 ± 6.1
	LDA	**79.1 ± 2.4**	**77.8 ± 2.5**	**81.3 ± 2.2**	190.1 ± 0.4
RoBERTa	-	82.2 ± 1.5	80.8 ± 1.5	84.2 ± 1.7	**314.6 ± 0.2**
	BERTopic	82.3 ± 1.4	80.9 ± 1.4	84.2 ± 1.7	325.9 ± 12.0
	FLSA-W	82.3 ± 1.7	80.9 ± 1.8	**84.3 ± 1.6**	353.4 ± 6.2
	LDA	**82.4 ± 1.4**	**81.1 ± 1.4**	**84.3 ± 1.5**	345.2 ± 0.4
T5	-	83.2 ± 1.4	**84.6 ± 1.9**	82.1 ± 1.7	**973.2 ± 50.7**
	BERTopic	**83.4 ± 1.3**	83.2 ± 2.0	**83.7 ± 1.3**	992.7 ± 45.6
	FLSA-W	83.3 ± 1.3	83.0 ± 1.9	**83.7 ± 1.4**	1020.1 ± 49.4
	LDA	83.3 ± 1.3	83.0 ± 1.9	**83.7 ± 1.2**	1011.9 ± 50.8
XLNet	-	79.3 ± 1.9	77.9 ± 1.9	81.9 ± 1.9	**1741.4 ± 88.6**
	BERTopic	80.1 ± 1.8	78.7 ± 1.8	82.4 ± 1.5	1757.7 ± 80.2
	FLSA-W	80.1 ± 2.3	78.8 ± 2.6	82.3 ± 2.0	1785.2 ± 92.4
	LDA	**80.4 ± 1.4**	**79.0 ± 1.3**	**82.5 ± 1.8**	1777.0 ± 88.7

- refers to the absence of a TT component, resulting in an ablation configuration.

Table 6. Results on BESSTIE Sarc (mean ± standard deviation).

Transfomer	Topic Model	F1 (%)	Precision (%)	Recall (%)	Training Time (s)
-	BERTopic	**49.2 ± 4.7**	**56.0 ± 3.0**	**56.7 ± 3.3**	**13.6 ± 6.5**
	FLSA-W	42.3 ± 0.5	36.7 ± 0.7	50.0 ± 0.0	45.3 ± 4.0
	LDA	47.5 ± 4.2	51.2 ± 2.5	50.8 ± 2.5	35.7 ± 0.4
BERT	-	63.9 ± 2.8	64.2 ± 2.5	66.1 ± 2.0	**289.0 ± 0.3**
	BERTopic	65.4 ± 2.6	65.0 ± 2.4	**66.3 ± 3.1**	310.8 ± 6.4
	FLSA-W	65.3 ± 2.6	65.0 ± 2.3	66.2 ± 3.1	342.5 ± 4.1
	LDA	**65.6 ± 2.4**	**65.4 ± 2.4**	66.1 ± 2.6	332.8 ± 0.6

Table 6. *Cont.*

Transfomer	Topic Model	F1 (%)	Precision (%)	Recall (%)	Training Time (s)
DistilBERT	-	63.6 ± 2.5	64.2 ± 2.9	**64.5 ± 2.7**	**146.8 ± 0.3**
	BERTopic	64.2 ± 2.3	64.2 ± 2.8	64.4 ± 2.1	168.5 ± 6.5
	FLSA-W	**64.3 ± 2.2**	64.2 ± 2.5	**64.5 ± 2.1**	200.3 ± 4.0
	LDA	**64.3 ± 2.7**	**64.3 ± 2.9**	**64.5 ± 2.6**	190.6 ± 0.5
RoBERTa	-	63.4 ± 7.1	65.9 ± 2.5	66.6 ± 2.3	**420.8 ± 0.6**
	BERTopic	66.2 ± 2.4	66.1 ± 2.6	67.7 ± 2.9	434.5 ± 6.5
	FLSA-W	**66.9 ± 2.4**	**66.7 ± 2.5**	**67.9 ± 3.1**	466.2 ± 3.8
	LDA	66.7 ± 2.7	66.6 ± 2.6	67.5 ± 3.2	456.6 ± 0.5
T5	-	65.1 ± 2.5	65.9 ± 2.8	65.7 ± 2.9	**1066.5 ± 64.0**
	BERTopic	**67.1 ± 2.7**	**66.8 ± 2.7**	**67.5 ± 2.7**	1088.3 ± 64.0
	FLSA-W	66.7 ± 2.8	66.5 ± 2.7	67.3 ± 2.9	1120.1 ± 62.6
	LDA	67.0 ± 2.6	**66.8 ± 2.6**	**67.5 ± 2.7**	1110.4 ± 64.0
XLNet	-	63.0 ± 5.6	65.7 ± 3.3	66.0 ± 3.8	**3132.3 ± 55.8**
	BERTopic	65.2 ± 2.0	65.2 ± 2.7	65.9 ± 2.2	3146.0 ± 55.7
	FLSA-W	**66.0 ± 1.6**	**66.1 ± 1.8**	66.5 ± 2.0	3177.8 ± 54.3
	LDA	65.9 ± 2.1	65.5 ± 2.1	**66.7 ± 2.1**	3168.1 ± 55.7

- refers to the absence of a TT component, resulting in an ablation configuration.

Table 7. Results on BBC News (mean ± standard deviation)

Transfomer	Topic Model	F1 (%)	Precision (%)	Recall (%)	Training Time (s)
-	BERTopic	92.6 ± 2.7	93.0 ± 2.6	92.7 ± 2.7	**10.3 ± 0.6**
	FLSA-W	7.3 ± 1.3	4.7 ± 1.5	20.1 ± 0.4	115.7 ± 7.4
	LDA	**93.7 ± 1.3**	**93.9 ± 1.3**	**93.7 ± 1.3**	89.1 ± 0.7
BERT	-	96.9 ± 1.2	96.9 ± 1.3	97.0 ± 1.2	**170.1 ± 0.4**
	BERTopic	**98.0 ± 0.9**	**98.0 ± 0.9**	**98.1 ± 1.0**	179.7 ± 0.7
	FLSA-W	97.6 ± 0.9	97.6 ± 1.0	97.6 ± 0.9	285.1 ± 7.4
	LDA	**98.0 ± 0.9**	**98.0 ± 0.9**	98.0 ± 0.9	258.4 ± 0.9
DistilBERT	-	96.7 ± 1.3	96.6 ± 1.3	96.7 ± 1.3	**90.7 ± 1.1**
	BERTopic	97.7 ± 1.2	97.8 ± 1.1	97.8 ± 1.2	100.3 ± 1.3
	FLSA-W	**98.0 ± 0.9**	**98.0 ± 1.0**	**98.1 ± 0.9**	205.6 ± 7.9
	LDA	97.7 ± 0.9	97.7 ± 0.9	97.7 ± 0.9	179.0 ± 1.1
RoBERTa	-	97.5 ± 0.6	97.5 ± 0.6	97.5 ± 0.7	**187.1 ± 36.1**
	BERTopic	97.6 ± 1.0	97.6 ± 0.8	97.6 ± 1.1	197.5 ± 35.8
	FLSA-W	**97.9 ± 0.9**	**97.9 ± 0.8**	**98.0 ± 0.9**	302.8 ± 35.0
	LDA	97.7 ± 0.8	97.8 ± 0.8	97.8 ± 0.9	276.2 ± 36.0
T5	-	97.1 ± 0.8	97.1 ± 0.8	97.3 ± 0.7	**384.3 ± 10.9**
	BERTopic	**97.9 ± 1.0**	**97.9 ± 1.0**	**97.9 ± 1.0**	394.4 ± 10.9
	FLSA-W	97.6 ± 1.0	97.7 ± 1.0	97.6 ± 1.0	499.8 ± 13.1
	LDA	97.6 ± 0.8	97.7 ± 0.9	97.7 ± 0.8	473.2 ± 10.8
XLNet	-	97.2 ± 1.2	97.2 ± 1.1	97.3 ± 1.2	**611.9 ± 5.7**
	BERTopic	**97.7 ± 0.9**	**97.7 ± 0.8**	97.7 ± 0.9	621.5 ± 5.6
	FLSA-W	97.4 ± 0.6	97.5 ± 0.6	97.5 ± 0.7	726.9 ± 9.2
	LDA	**97.7 ± 0.9**	**97.7 ± 0.9**	**97.8 ± 0.9**	700.3 ± 5.6

- refers to the absence of a TT component, resulting in an ablation configuration.

Table 8. Results on 20News (mean ± standard deviation).

Transfomer	Topic Model	F1 (%)	Precision (%)	Recall (%)	Training Time (s)
-	BERTopic	**50.4 ± 3.1**	**52.7 ± 3.3**	**51.8 ± 2.6**	**66.8 ± 7.3**
	FLSA-W	1.1 ± 0.5	1.2 ± 1.2	5.0 ± 0.7	568.4 ± 37.2
	LDA	44.8 ± 2.2	46.5 ± 2.3	46.4 ± 2.2	400.4 ± 3.5
BERT	-	67.9 ± 1.1	69.1 ± 1.7	68.2 ± 1.1	**1119.9 ± 3.2**
	BERTopic	**70.2 ± 1.2**	**71.3 ± 1.1**	**70.2 ± 1.2**	1181.2 ± 7.7
	FLSA-W	69.1 ± 1.5	70.3 ± 1.5	69.2 ± 1.4	1683.0 ± 35.1
	LDA	69.9 ± 1.0	71.1 ± 1.1	69.8 ± 1.0	1515.0 ± 5.9
DistilBERT	-	66.1 ± 1.3	66.7 ± 1.5	66.6 ± 1.1	**589.5 ± 2.9**
	BERTopic	**68.9 ± 1.1**	**70.3 ± 1.1**	**68.9 ± 1.1**	650.8 ± 8.4
	FLSA-W	68.0 ± 0.4	69.4 ± 0.5	68.1 ± 0.4	1152.6 ± 37.9
	LDA	68.6 ± 0.9	69.7 ± 0.9	68.7 ± 0.9	984.6 ± 4.1
RoBERTa	-	70.1 ± 0.9	70.8 ± 0.9	70.2 ± 0.8	**1362.7 ± 120.5**
	BERTopic	**70.9 ± 0.9**	**72.1 ± 0.9**	**70.9 ± 0.7**	1424.0 ± 117.5
	FLSA-W	70.4 ± 1.0	71.4 ± 1.1	70.4 ± 0.8	1925.8 ± 115.8
	LDA	70.7 ± 1.0	71.9 ± 1.3	70.7 ± 0.8	1757.8 ± 121.9
T5	-	70.2 ± 0.8	71.2 ± 0.9	70.4 ± 0.8	**3265.4 ± 98.6**
	BERTopic	**71.8 ± 0.9**	**72.8 ± 1.0**	**71.6 ± 0.9**	3331.5 ± 97.9
	FLSA-W	71.4 ± 0.8	72.2 ± 1.1	71.4 ± 0.8	3833.3 ± 94.8
	LDA	**71.8 ± 0.9**	**72.8 ± 1.0**	**71.6 ± 0.9**	3665.3 ± 97.4
XLNet	-	69.1 ± 0.9	69.8 ± 1.1	69.3 ± 0.9	**6389.9 ± 56.3**
	BERTopic	**71.5 ± 0.9**	**72.6 ± 0.8**	**71.4 ± 1.0**	6451.2 ± 53.6
	FLSA-W	70.6 ± 1.1	71.9 ± 1.0	70.5 ± 1.1	6953.0 ± 83.2
	LDA	70.9 ± 1.2	72.2 ± 1.0	70.9 ± 1.3	6785.0 ± 55.3

- refers to the absence of a TT component, resulting in an ablation configuration.

The results confirm that contextual embeddings from transformers outperform features from topic models, which aligns with the expectation since transformers represent the state-of-the-art in text classification. Across all datasets, the standalone transformer models exhibit largely comparable performance in terms of Macro F1-score, precision, and recall, with the largest discrepancy observed in Table 8 (20News), where the gap between DistilBERT and T5 reaches 4.9% in Macro F1-score. Even in this case, the difference remains relatively modest despite being significantly smaller and less computationally demanding, as reflected in shorter training times and lower memory requirements. Additionally, it is also noteworthy that BERT sometimes achieves better performance than more recent transformer architectures (RoBERTa, XLNet, and T5) in our experiments. Although these models are often regarded as state of the art, this designation typically refers to their performance on broad, generic NLP benchmarks rather than guaranteeing superior results for every specific task. Differences in pretraining objectives, architectural design, and optimization strategies can make certain models better aligned with particular datasets, which helps explain why BERT can outperform larger or more complex alternatives. In this context, increased architectural sophistication does not necessarily translate into proportional gains in text classification performance.

In contrast, classification heads which only rely on topic model-generated features exhibit significantly lower classification performance, often comparable to a random model or marginally better. This suggests that standalone topic models are insufficient feature extractors for the nuances of text classification.

Table 9. Statistical tests comparing the results (macro F1-score) obtained by transformers to the ones obtained by TT configurations where the same transformer is included.

Transformer	TT Configuration	Sms		Stormfront		BESSTIE Sent	
		p_{adj}	e.s. [95% CI]	p_{adj}	e.s. [95% CI]	p_{adj}	e.s. [95% CI]
BERT	BERTopic+(...)	1.00	0.4 [−0.3, 1.0]	1.00	0.1 [−0.5, 0.7]	0.81	0.6 [−0.1, 1.3]
	FLSA-W+(...)	1.00	0.2 [−0.4, 0.9]	1.00	0.0 [−0.6, 0.7]	0.71	0.7 [0.0, 1.4]
	LDA+(...)	1.00	0.0 [−0.6, 0.6]	1.00	0.1 [−0.5, 0.8]	0.81	0.7 [0.0, 1.3]
DistilBERT	BERTopic+(...)	1.00	0.5 [−0.2, 1.2]	0.15	1.0 [0.2, 1.8]	1.00	0.2 [−0.4, 0.8]
	FLSA-W+(...)	1.00	0.5 [−0.2, 1.1]	**0.05**	1.2 [0.4, 2.1]	1.00	−0.1 [−0.7, 0.6]
	LDA+(...)	1.00	0.2 [−0.4, 0.8]	0.33	0.8 [0.1, 1.6]	1.00	0.3 [−0.3, 0.9]
RoBERTa	BERTopic+(...)	1.00	−0.1 [−0.7, 0.5]	1.00	0.4 [−0.2, 1.0]	1.00	0.2 [−0.5, 0.8]
	FLSA-W+(...)	1.00	0.1 [−0.6, 0.7]	1.00	0.3 [−0.3, 1.0]	1.00	0.2 [−0.4, 0.8]
	LDA+(...)	1.00	0.0 [−0.7, 0.6]	1.00	0.4 [−0.3, 1.0]	1.00	0.4 [−0.3, 1.0]
T5	BERTopic+(...)	1.00	0.3 [−0.3, 0.9]	1.00	0.4 [−0.2, 1.1]	1.00	0.2 [−0.5, 0.8]
	FLSA-W+(...)	1.00	0.4 [−0.3, 1.0]	1.00	0.4 [−0.3, 1.0]	1.00	0.1 [−0.5, 0.7]
	LDA+(...)	1.00	0.3 [−0.3, 0.9]	1.00	0.3 [−0.3, 0.9]	1.00	0.1 [−0.5, 0.7]
XLNet	BERTopic+(...)	1.00	0.5 [−0.1, 1.2]	1.00	0.5 [−0.2, 1.1]	0.71	0.7 [0.0, 1.4]
	FLSA-W+(...)	1.00	0.5 [−0.2, 1.1]	1.00	0.3 [−0.3, 0.9]	0.94	0.6 [−0.1, 1.3]
	LDA+(...)	1.00	0.6 [−0.1, 1.3]	1.00	0.4 [−0.3, 1.0]	0.77	0.7 [0.0, 1.4]
Transformer	**TT Configuration**	**BESSTIE Sarc**		**BBC**		**20News**	
		p_{adj}	e.s. [95% CI]	p_{adj}	e.s. [95% CI]	p_{adj}	e.s. [95% CI]
BERT	BERTopic+(...)	0.50	0.7 [0.0, 1.4]	0.18	1.0 [0.2, 1.7]	**0.00**	2.4 [1.1, 3.6]
	FLSA-W+(...)	0.54	0.7 [0.0, 1.4]	0.38	0.8 [0.0, 1.5]	**0.01**	1.4 [0.5, 2.3]
	LDA+(...)	0.21	0.9 [0.2, 1.7]	0.20	0.9 [0.1, 1.6]	**0.00**	2.4 [1.2, 3.7]
DistilBERT	BERTopic+(...)	1.00	0.3 [−0.4, 0.9]	**0.01**	1.7 [0.7, 2.7]	**0.00**	2.1 [0.9, 3.2]
	FLSA-W+(...)	1.00	0.3 [−0.4, 0.9]	**0.00**	1.9 [0.8, 3.0]	**0.00**	1.7 [0.7, 2.7]
	LDA+(...)	1.00	0.3 [−0.3, 0.9]	0.12	1.0 [0.2, 1.8]	**0.00**	2.4 [1.1, 3.6]
RoBERTa	BERTopic+(...)	1.00	0.4 [−0.3, 1.0]	1.00	0.1 [−0.5, 0.7]	**0.01**	1.4 [0.5, 2.3]
	FLSA-W+(...)	1.00	0.4 [−0.2, 1.1]	0.99	0.5 [−0.2, 1.1]	0.16	0.6 [−0.1, 1.3]
	LDA+(...)	1.00	0.4 [−0.2, 1.1]	1.00	0.3 [−0.4, 0.9]	0.16	0.6 [−0.1, 1.3]
T5	BERTopic+(...)	**0.03**	1.4 [0.5, 2.2]	0.43	0.7 [0.0, 1.4]	**0.00**	1.9 [0.8, 3.0]
	FLSA-W+(...)	0.12	1.0 [0.2, 1.8]	0.99	0.5 [−0.1, 1.2]	**0.00**	1.6 [0.6, 2.6]
	LDA+(...)	**0.01**	1.6 [0.7, 2.6]	0.99	0.5 [−0.2, 1.2]	**0.00**	2.3 [1.1, 3.5]
XLNet	BERTopic+(...)	1.00	0.5 [−0.2, 1.1]	0.99	0.5 [−0.2, 1.1]	**0.00**	3.0 [1.5, 4.5]
	FLSA-W+(...)	0.68	0.6 [−0.1, 1.3]	1.00	0.2 [−0.4, 0.9]	**0.00**	1.7 [0.7, 2.7]
	LDA+(...)	0.68	0.6 [−0.1, 1.3]	0.99	0.5 [−0.2, 1.1]	**0.00**	2.2 [1.0, 3.3]

(...) refers to the same transformer presented in column Transformer.

Despite this, the vast majority of TT configurations improve upon its corresponding standalone transformer on every dataset, although this does not hold for every individual configuration. Moreover, the magnitude of the differences varies considerably across datasets. On Sms (see Table 3), where the standalone transformers already achieved near-perfect scores (around 98% F1-score), the best gain was 0.4%, obtained by XLNet combined with FLSA-W or BERTopic. On Stormfront (see Table 4), the best gain was 0.9%, obtained by every DistilBERT-based TT configuration. On BESSTIE Sent (see Table 5), the best gain was 1.1%, obtained by BERT combined with FLSA-W or LDA, and by XLNet combined with LDA. On BESSTIE Sarc (see Table 6), the best gain was 3.5%, obtained by RoBERTa coupled with FLSA-W, the highest obtained, though it must be noted that the standalone RoBERTa results have very high standard deviations. On BBC News (see Table 7), the best gain was 1.3%, obtained by DistilBERT combined with FLSA-W. The most consistent gains were observed on 20News (see Table 8), where F1-score increased up to 2.8%, a result obtained by DistilBERT combined with BERTopic.

Table 9 shows that statistically significant improvements are less pronounced. After multiple-testing correction, at least one TT configuration is significant in four of the six datasets, namely, Stormfront, BESSTIE Sarc, BBC, and 20News. On Stormfront, only one configuration did so (DistilBERT combined with FLSA-W). On BBC, two configurations did (DistilBERT combined with FLSA-W and with BERTopic). On BESSTIE Sarc, T5 was the only standalone transformer significantly outperformed by its configurations with BERTopic and LDA. The strongest evidence is on 20News, where all configurations built on BERT, DistilBERT, T5, and XLNet significantly outperformed their standalone counterparts, as did the RoBERTa configuration with BERTopic. Importantly, it should be noted that the largest observed gain in Tables 3–8, namely, from RoBERTa to RoBERTa combined with FLSA-W, was not considered significant, with a p-value of 1.00 due to the high standard deviation of the standalone RoBERTa.

While standalone components train faster than full architectures, the complete TT framework—as its total training time is the sum of its transformer, topic model, and classification head components—remains competitive. In this regard, it is important to note that architectures incorporating FLSA-W require noticeably longer training times, likely due to its less optimized implementation compared to other topic models.

Overall, these findings build upon the results previously reported for TFusion [5], which was evaluated in a similar experimental setting but only on the Stormfront dataset using DistilBERT and LDA. The modular design of TT enabled us to extend this evaluation to six datasets, five transformer encoders, and three topic-modeling techniques. The results suggest that using TT to integrate deep learning with topic-based representations can be beneficial in some settings. The observed improvements cannot be explained solely by the standalone predictive performance of the topic representations. For instance, FLSA-W achieves performance close to that of a random classifier when used in isolation, yet some TT configurations incorporating FLSA-W achieve higher macro-average F1-scores than their corresponding standalone transformers. One possible explanation is that topic representations provide thematic information that becomes useful when combined with contextual embeddings. This interpretation is consistent with discussions of complementarity between views in multiview learning [78] and multimodal integration [79,80]. However, this explanation remains a hypothesis rather than an established finding of the present study.

Ultimately, the authors acknowledge that TT versatility is constrained by the least versatile component in the pipeline. Although modular, each component must independently suit the target task. For instance, if the dataset at hand is multilingual, all components must support multilingual processing. This particularly affects transformer-based components, including the main encoder and embedding models used within topic-modeling approaches (e.g., SBERT in BERTopic), as well as language-dependent preprocessing steps such as lemmatization and stop-word removal. Scalability imposes a similar constraint; if a dataset is too large for a standalone method, any TT configuration incorporating it will also be impractical. Thus, despite its architectural flexibility, TT practical applicability is ultimately bounded by its individual components.

Beyond these architectural constraints, some limitations of our analysis should be noted. The contribution of topic representations to the observed performance improvements has not been conclusively established. Although our ablation experiments compare the complete TT framework with configurations that exclude topic representations, controlled experiments using random or shuffled topic representations were not conducted. Such comparisons would help determine whether the observed gains arise from meaningful topic information or may instead be partly attributable to other effects of the fusion architecture. In addition, all experiments were conducted using a fixed random seed, and evaluations across multiple random seeds were not performed. Consequently, the sensitivity of the

observed improvements to random initialization and the extent to which these results are reproducible across different random seeds remain uncertain. Further experiments incorporating these controls are therefore necessary to better isolate the contribution of topic representations and assess the robustness of the observed performance gains.

7. Conclusions

This paper proposed a versatile and modular text classification framework, the Topic-Transformer (TT) framework. TT combines transformer-based encoders and topic-modeling methods through model-level fusion, allowing contextual embeddings and topic representations to be jointly used for classification. TT was tested on six English benchmark datasets, namely, UCI SMS Spam Collection, Stormfront Hate Speech Dataset, BESSTIE Sent, BESSTIE Sarc, BBC News Dataset, and 20Newsgroups. Additionally, these tests were conducted using fifteen distinct configurations, corresponding to five distinct transformers and three topic models.

The results show that, for every combination of dataset and transformer, at least one TT configuration achieves a higher macro-average F1-score than its standalone transformer counterpart. However, the magnitude and statistical significance of the improvements vary across datasets and configurations. In four of the six datasets, at least one TT configuration yields a statistically significant improvement at a significance level of 0.05 after multiple-testing correction. The largest significant gains are observed on 20Newsgroups, where improvements reach 2.8% in macro-average F1-score.

Nevertheless, it must be noted that our experiments do not establish whether the observed gains result specifically from information provided by topic representations or from other aspects of the fusion architecture. Future work should investigate the contribution of topic information independently of architectural effects and evaluate the framework on a broader range of datasets.

Author Contributions: Conceptualization, F.M.P.S.; methodology, F.M.P.S.; software, F.M.P.S.; validation, F.M.P.S., J.M.C.S. and S.M.V.; formal analysis, F.M.P.S., J.M.C.S. and S.M.V.; investigation, F.M.P.S.; resources, F.M.P.S.; data curation, F.M.P.S.; writing—original draft preparation, F.M.P.S.; writing—review and editing, F.M.P.S., J.M.C.S. and S.M.V.; visualization, F.M.P.S.; supervision, J.M.C.S. and S.M.V.; project administration, J.M.C.S. and S.M.V.; funding acquisition, J.M.C.S. and S.M.V. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge Fundação para a Ciência e a Tecnologia (FCT) for its financial support via LAETA (project https://doi.org/10.54499/UID/50022/2025). F.M.P.S.' work was supported by the Ph.D. Scholarship 2022.12077.BDANA from FCT.

Data Availability Statement: The original data presented in the study are openly available in FigShare at https://doi.org/10.6084/m9.figshare.34346568.

Acknowledgments: Although not required by the MDPI guidelines, the authors disclose the use of ChatGPT 5 for text editing to assist with the readability and overall clarity of the manuscript. The authors have reviewed and revised the manuscript as necessary and take full responsibility for its content.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

TT	Topic-Transformer framework
NLP	Natural Language Processing
BERT	Bidirectional Encoder Representations from Transformers
LDA	Latent Dirichlet Allocation
FLSA	Fuzzy Latent Semantic Analysis
FFN	Feedforward Neural Network
MLP	Multi-Layer Perceptron

References

1. Ahamad, R.; Mishra, K.N. Exploring sentiment analysis in handwritten and E-text documents using advanced machine learning techniques: A novel approach. *J. Big Data* **2025**, *12*, 11. [CrossRef]
2. Martins, M.; Santos, F.; Almeida, A.; Vieira, S.M.; Sousa, J.M. Using Information Fusion to Predict Customer Sentiment. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 235–245. [CrossRef]
3. Hasib, K.M.; Azam, S.; Karim, A.; Al Marouf, A.; Shamrat, F.J.M.; Montaha, S.; Yeo, K.C.; Jonkman, M.; Alhajj, R.; Rokne, J.G. Mcnn-lstm: Combining cnn and lstm to classify multi-class text in imbalanced news data. *IEEE Access* **2023**, *11*, 93048–93063. [CrossRef]
4. Chafi, S.; Kabil, M.; Kamouss, A. Distributed CV classification with attention mechanisms. *Int. J. Speech Technol.* **2024**, *27*, 1149–1157. [CrossRef]
5. Santos, F.; Sousa, J.M.; Vieira, S.M. Topic Modeling for Enhancing Transformers Hate Speech Detection. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 223–234. [CrossRef]
6. Guo, Y.; Mustafaoglu, Z.; Koundal, D. Spam detection using bidirectional transformers and machine learning classifier algorithms. *J. Comput. Cogn. Eng.* **2023**, *2*, 5–9. [CrossRef]
7. Almeida, T.A.; Hidalgo, J.M.G.; Yamakami, A. Contributions to the study of SMS spam filtering: New collection and results. In *Proceedings of the 11th ACM Symposium on Document Engineering*; Association for Computing Machinery: New York, NY, USA, 2011; pp. 259–262. [CrossRef]
8. Kumari, R.; Ashok, N.; Ghosal, T.; Ekbal, A. What the fake? Probing misinformation detection standing on the shoulder of novelty and emotion. *Inf. Process. Manag.* **2022**, *59*, 102740. [CrossRef]
9. Lu, H.; Ehwerhemuepha, L.; Rakovski, C. A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance. *BMC Med. Res. Methodol.* **2022**, *22*, 181. [CrossRef]
10. Gasparetto, A.; Marcuzzo, M.; Zangari, A.; Albarelli, A. A survey on text classification algorithms: From text to predictions. *Information* **2022**, *13*, 83. [CrossRef]
11. Minaee, S.; Kalchbrenner, N.; Cambria, E.; Nikzad, N.; Chenaghlu, M.; Gao, J. Deep learning–based text classification: A comprehensive review. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–40. [CrossRef]
12. Greevy, E.; Smeaton, A.F. Classifying racist texts using a support vector machine. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*; Association for Computing Machinery: New York, NY, USA, 2004; pp. 468–469. [CrossRef]
13. Chen, Y.; Zhou, Y.; Zhu, S.; Xu, H. Detecting offensive language in social media to protect adolescent online safety. In *Proceedings of the 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*; IEEE: New York, NY, USA, 2012; pp. 71–80. [CrossRef]
14. Melville, P.; Gryc, W.; Lawrence, R.D. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2009; pp. 1275–1284. [CrossRef]
15. Kim, S.B.; Han, K.S.; Rim, H.C.; Myaeng, S.H. Some effective techniques for naive bayes text classification. *IEEE Trans. Knowl. Data Eng.* **2006**, *18*, 1457–1466. [CrossRef]
16. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1532–1543.
17. Czajka, M.M.; Kubacka, D.; Świetlicka, A. Embedding representation of words in sign language. *J. Comput. Appl. Math.* **2025**, *465*, 116590. [CrossRef]

18. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent convolutional neural networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Austin, TX, USA, 2015; Volume 29. [CrossRef]
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
20. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186. [CrossRef]
21. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
22. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
23. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
24. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
25. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
26. Liu, X.Y.; Wang, G.; Yang, H.; Zha, D. Fingpt: Democratizing internet-scale data for financial large language models. *arXiv* **2023**, arXiv:2307.10485.
27. Li, M.; Liao, S.; Okpala, E.; Tong, M.; Costello, M.; Cheng, L.; Hu, H.; Luo, F. COVID-HateBERT: A pre-trained language model for COVID-19 related hate speech detection. In *Proceedings of the 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*; IEEE: New York, NY, USA, 2021; pp. 233–238. [CrossRef]
28. Weng, Y.; Yang, K.; Liu, Z.; He, W.; Tang, X. LGVLM-mIoT: A Lightweight Generative Visual-Language Model for Multilingual IoT Applications. *IEEE Internet Things J.* **2025**, *12*, 13311–13326. [CrossRef]
29. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
30. Karami, A.; Gangopadhyay, A.; Zhou, B.; Kharrazi, H. Fuzzy approach topic discovery in health and medical corpora. *Int. J. Fuzzy Syst.* **2018**, *20*, 1334–1345. [CrossRef]
31. Grootendorst, M. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv* **2022**, arXiv:2203.05794.
32. Lee, D.; Seung, H.S. Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Process. Syst.* **2000**, *13*.
33. Yan, X.; Hu, S.; Mao, Y.; Ye, Y.; Yu, H. Deep multi-view learning methods: A review. *Neurocomputing* **2021**, *448*, 106–129. [CrossRef]
34. Qin, Y.; Zhang, X.; Yu, S.; Feng, G. A survey on representation learning for multi-view data. *Neural Netw.* **2025**, *181*, 106842. [CrossRef]
35. Yu, Z.; Dong, Z.; Yu, C.; Yang, K.; Fan, Z.; Chen, C.P. A review on multi-view learning. *Front. Comput. Sci.* **2025**, *19*, 197334. [CrossRef]
36. Yu, H.S.; Zhao, S.K.; Zhang, Y. Cross-Fitted Multi-View Neural Risk Augmentation for Interpretable University Dropout Prediction. *Mathematics* **2026**, *14*, 3147. [CrossRef]
37. Wen, Y.; Li, H. Cluster Complementarity and Consistency Mining for Multi-View Representation Learning. *Mathematics* **2025**, *13*, 2521. [CrossRef]
38. Tan, H.; Wan, Y.; Luo, G.; Sun, Z. Hierarchical Multi-View Representation Learning via Generalized Deep Non-Negative Matrix Factorization. *Mathematics* **2026**, *14*, 1442. [CrossRef]
39. Ngo, D. Pseudo-Multiview Learning Using Subjective Logic for Enhanced Classification Accuracy. *Mathematics* **2025**, *13*, 2085. [CrossRef]
40. Baltrušaitis, T.; Ahuja, C.; Morency, L.P. Multimodal machine learning: A survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 423–443. [CrossRef]
41. Um, T.; Kim, N. A study on performance enhancement by integrating neural topic attention with transformer-based language model. *Appl. Sci.* **2024**, *14*, 7898. [CrossRef]
42. Habbat, N.; Nouri, H. Unlocking travel narratives: A fusion of stacking ensemble deep learning and neural topic modeling for enhanced tourism comment analysis. *Soc. Netw. Anal. Min.* **2024**, *14*, 82. [CrossRef]
43. Le, T.K.H.; Anh, D.T.K.; Huyen, N.T.T.; Kim, L.N.H.; Quan, H.L.D.; Huong, P.T.T. Uncovering customer insights from online hotel reviews in Vietnam: A bilingual NLP study using sentiment analysis and topic modelling. *Asia Pac. Manag. Rev.* **2026**, *31*, 100450. [CrossRef]
44. Egger, R. TriTopic: Tri-Modal Graph-Based Topic Modeling with Iterative Refinement and Archetypes. *arXiv* **2026**, arXiv:2602.19079.

45. Yang, C.; Kim, Y. Enhancing topic coherence and diversity in document embeddings using LLMs: A focus on BERTopic. *Expert Syst. Appl.* **2025**, *281*, 127517. [CrossRef]
46. Niruti, B.; Sujith, D.; Returi, D. TAXONOMY ALIGNED TOPIC MODELING IN MULTIMODAL SYSTEMS USING TACTM++: A TRANSFORMER-BASED TOPIC MODELING APPROACH. *Int. J. Appl. Math.* **2025**, *38*, 35–49. [CrossRef]
47. Ding, F.; Kang, X.; Ren, F. Neuro or Symbolic? Fine-Tuned Transformer with Unsupervised LDA Topic Clustering for Text Sentiment Analysis. *IEEE Trans. Affect. Comput.* **2024**, *15*, 493–507. [CrossRef]
48. Alcoforado, A.; Ferraz, T.P.; Gerber, R.; Bustos, E.; Oliveira, A.S.; Veloso, B.M.; Siqueira, F.L.; Costa, A.H.R. Zeroberto: Leveraging zero-shot text classification by topic modeling. In *Proceedings of the International Conference on Computational Processing of the Portuguese Language*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 125–136. [CrossRef]
49. Cheng, X.; Chen, Y.; Wang, P.; Zhou, Y.; Wei, X.; Luo, W.; Duan, Q. A novel ChatGPT-based multimodel framework for tourism review mining: A case study on China's five sacred mountains. *J. Hosp. Tour. Technol.* **2024**, *15*, 592–609. [CrossRef]
50. Chen, B.; Chen, Y.; Feng, B.; Xu, X.; Chen, R.; Chen, W. Research on Sentiment Analysis of Tourism Reviews Based on Deep Transfer Learning and Multi-source UGC Mining—A Case Study of Shantou. In *Proceedings of the 2025 2nd Symposium on Big Data, Neural Networks, and Deep Learning*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 9–17. [CrossRef]
51. Çelik, E.; Dalyan, T. Unified benchmark for zero-shot Turkish text classification. *Inf. Process. Manag.* **2023**, *60*, 103298. [CrossRef]
52. Voskergian, D.; Jayousi, R.; Yousef, M. Topic selection for text classification using ensemble topic modeling with grouping, scoring, and modeling approach. *Sci. Rep.* **2024**, *14*, 23516. [CrossRef]
53. Wang, B.; Ge, J.; Lu, F. From review filtering to topic modeling: A deep learning-based approach for analyzing negative online reviews of five-star hotels. *J. Hosp. Tour. Technol.* **2025**, 1–25. [CrossRef]
54. Zhou, Y.; Liao, L.; Gao, Y.; Wang, R.; Huang, H. TopicBERT: A Topic-Enhanced Neural Language Model Fine-Tuned for Sentiment Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 380–393. [CrossRef]
55. Ojo, A.O.; Bouguila, N. Smoothed-ModernBERT: Cross-Modal Gated Attention of Probabilistic Topic Models and ModernBERT Through Dynamic Fusion. In *Proceedings of the International Conference on Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2026; pp. 335–348. [CrossRef]
56. Gopal, G.L.; Gopal, G.R.; Gupta, D.; Punithaasree, K.S.; Kiran, A.; Durairaj, M. Enhancing Advanced Text Classification Algorithms for Topic Modelling in English Literature. In *Proceedings of the 2024 International BIT Conference (BITCON)*; IEEE: New York, NY, USA, 2024; pp. 1–5. [CrossRef]
57. Porteous, I.; Newman, D.; Ihler, A.; Asuncion, A.; Smyth, P.; Welling, M. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2008; pp. 569–577. [CrossRef]
58. Jordan, M.I.; Ghahramani, Z.; Jaakkola, T.S.; Saul, L.K. An introduction to variational methods for graphical models. *Mach. Learn.* **1999**, *37*, 183–233. [CrossRef]
59. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
60. McInnes, L.; Healy, J.; Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv* **2018**, arXiv:1802.03426.
61. Campello, R.J.; Moulavi, D.; Sander, J. Density-based clustering based on hierarchical density estimates. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 160–172. [CrossRef]
62. Rijcken, E.; Scheepers, F.; Mosteiro, P.; Zervanou, K.; Spruit, M.; Kaymak, U. A comparative study of fuzzy topic models and LDA in terms of interpretability. In *Proceedings of the 2021 IEEE Symposium Series on Computational Intelligence (SSCI)*; IEEE: New York, NY, USA, 2021; pp. 1–8. [CrossRef]
63. Dumais, S. *Enhancing Performance in Latent Semantic Indexing (LSI) Retrieval*; Bellcore Technical Memorandum TM-ARH-017527; Bellcore: Morristown, NJ, USA, 1992.
64. Papineni, K. Why inverse document frequency? In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics*; Association for Computational Linguistics: Pittsburgh, PA, USA, 2001.
65. Kolda, T.G. *Limited-Memory Matrix Methods with Applications*; University of Maryland: College Park, MD, USA, 1997.
66. Abdi, H. Singular value decomposition (SVD) and generalized singular value decomposition. *Encycl. Meas. Stat.* **2007**, *907*, 44.
67. Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203. [CrossRef]
68. Manning, C.D. *Introduction to Information Retrieval*; Syngress Publishing: Burlington, MA, USA, 2008; pp. 79–80.
69. Seddah, D.; Chrupała, G.; Çetinoğlu, Ö.; Van Genabith, J.; Candito, M. Lemmatization and lexicalized statistical parsing of morphologically-rich languages: The case of french. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*; Association for Computational Linguistics: Los Angeles, CA, USA, 2010; pp. 85–93.
70. Ferilli, S.; Esposito, F.; Grieco, D. Automatic learning of linguistic resources for stopword removal and stemming from text. *Procedia Comput. Sci.* **2014**, *38*, 116–123. [CrossRef]

71. Piskorski, J.; Wieloch, K.; Sydow, M. On knowledge-poor methods for person name matching and lemmatization for highly inflectional languages. *Inf. Retr.* **2009**, *12*, 275–299. [CrossRef]
72. De Gibert, O.; Perez, N.; García-Pablos, A.; Cuadros, M. Hate speech dataset from a white supremacy forum. *arXiv* **2018**, arXiv:1809.04444.
73. Srirag, D.; Joshi, A.; Painter, J.; Kanojia, D. Besstie: A benchmark for sentiment and sarcasm classification for varieties of english. In *Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025*; Association for Computational Linguistics: Vienna, Austria, 2025; pp. 8413–8429. [CrossRef]
74. Greene, D.; Cunningham, P. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd International Conference on Machine Learning*; Association for Computing Machinery: New York, NY, USA, 2006; pp. 377–384. [CrossRef]
75. Lang, K. Newsweeder: Learning to filter netnews. In *Proceedings of the Machine Learning Proceedings 1995*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1995; pp. 331–339. [CrossRef]
76. Rijcken, E.; Zervanou, K.; Mosteiro, P.; Scheepers, F.; Spruit, M.; Kaymak, U. Topic specificity: A descriptive metric for algorithm selection and finding the right number of topics. *Nat. Lang. Process. J.* **2024**, *8*, 100082. [CrossRef]
77. Rijcken, E.; Mosteiro, P.; Zervanou, K.; Spruit, M.; Scheepers, F.; Kaymak, U. FuzzyTM: A software package for fuzzy topic modeling. In *Proceedings of the 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*; IEEE: New York, NY, USA, 2022; pp. 1–8. [CrossRef]
78. Zhao, J.; Xie, X.; Xu, X.; Sun, S. Multi-view learning overview: Recent progress and new challenges. *Inf. Fusion* **2017**, *38*, 43–54. [CrossRef]
79. Liang, P.P.; Cheng, Y.; Fan, X.; Ling, C.K.; Nie, S.; Chen, R.; Deng, Z.; Allen, N.; Auerbach, R.; Mahmood, F.; et al. Quantifying & modeling multimodal interactions: An information decomposition framework. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 27351–27393. [CrossRef]
80. Liang, P.P.; Zadeh, A.; Morency, L.P. Foundations & trends in multimodal machine learning: Principles, challenges, and open questions. *ACM Comput. Surv.* **2024**, *56*, 1–42. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.