

# Deep Learning for News Topic Identification in Limited Supervision and Unsupervised Settings

*Arezoo Hatefi*



DOCTORAL THESIS, APRIL 2024  
DEPARTMENT OF COMPUTING SCIENCE  
UMEÅ UNIVERSITY  
SWEDEN

Department of Computing Science  
Umeå University  
SE-901 87 Umeå, Sweden

*arezooh@cs.umu.se*

Copyright © 2024 by Arezoo Hatefi

Except Paper I, © Association for Computing Machinery, 2021

Paper III, © Association for Computational Linguistics, 2023

**ISBN 978-91-8070-342-0 (print)**

**ISBN 978-91-8070-343-7 (digital)**

**ISSN 0348-0542**

**UMINF 24.04**

Cover design by Arezoo Hatefi and Marlene Lahti

Printed by UmU Print Service, Umeå University, 2024

امروز نہ آغاز نہ انجام جہان است  
ای بس غم و شادی کہ پس پردہ نہان است  
کہ مرد رہی غم مخور از دوری و دیری  
دانی کہ رسیدن ہر گام زمان است

سایہ

*Today is neither the beginning nor the end of the world,  
Oh, what sorrow and joy lie hidden behind the curtain!  
If you are on the path, do not grieve over the distance or  
delay,  
You know that reaching it takes the step of time.  
Saaye (Shadow)*



# Abstract

In today’s world, following news is crucial for decision-making and staying informed. With the growing volume of daily news, automated processing is essential for timely insights and in aiding individuals and corporations in navigating the complexities of the information society. Another use of automated processing is contextual advertising, which addresses privacy concerns associated with cookie-based advertising by placing ads solely based on web page content, without tracking users or their online behavior. Therefore, accurately determining and categorizing page content is crucial for effective ad placements. The news media, heavily reliant on advertising to sustain operations, represent a substantial market for contextual advertising strategies.

Inspired by these practical applications and the advancements in deep learning over the past decade, this thesis mainly focuses on using deep learning for categorizing news articles into topics of varying granularity. Considering the dynamic nature of these applications and the limited availability of relevant labeled datasets for training models, the thesis emphasizes developing methods that can be trained effectively using unlabeled or partially labeled data. It proposes semi-supervised text classification models for categorizing datasets into predefined coarse-grained topics, where only a few labeled examples exist for each topic, while the majority of the dataset remains unlabeled. Furthermore, to better explore coarse-grained topics within news archives and streams and overcome the limitations of predefined topics in text classification the thesis suggests deep clustering approaches that can be trained in unsupervised settings.

Moreover, to address the identification of fine-grained topics, the thesis introduces a novel story discovery model for monitoring event-based topics in multi-source news streams. Given that online news reporting often incorporates diverse modalities like text, images, video, and audio to convey information, the thesis finally initiates an investigation into the synergy between textual and visual elements in news article analysis. To achieve this objective, a text-image dataset was annotated, and a baseline was established for event-topic discovery in multimodal news streams. While primarily intended for news monitoring and contextual advertising, the proposed models can, more generally, be regarded as novel approaches in semi-supervised text classification, deep clustering, and news story discovery. Comparison with state-of-the-art baseline models

demonstrates their effectiveness in addressing the respective objectives.

# Sammanfattning

I dagens samhälle är nyhetsbevakning avgörande för beslutsfattande och för att hålla sig informerad. Med den ständigt växande mängden av dagliga nyheter ger automatiserad bearbetning insikter som hjälper individer och företag att navigera den moderna världens komplexitet. Ett annat användningsområde för automatisk bearbetning är innehållsbaserad (eller kontextuell) reklam, ett koncept för onlinereklam som undviker integritetsproblemen kopplade till cookie-baserad reklam genom att enbart använda sig av webbplatsers innehåll för att placera annonsen, utan att spåra användare och deras onlinebeteende. Därför är korrekt identifiering och kategorisering sidinnehåll viktigt för effektiv annonsplacering. Nyhetsmedia är starkt beroende av reklamintäkter för att upprätthålla sin verksamhet, och representerar en stor marknad för kontextuella reklamstrategier.

Inspirerad av dessa praktiska tillämpningar och det senaste decenniets framsteg inom djupinlärning fokuserar denna avhandling främst på användandet av djupinlärning för att ämneskategorisera nyhetsartiklar på varierande nivåer av granularitet. Med avseende på hur dynamiska dessa applikationer är och den begränsade tillgängligheten av relevant annoterad data att träna på, betonar avhandlingen utvecklingen av metoder som effektivt kan tränas med oannoterad eller delvis annoterad data. Avhandlingen föreslår semi-övervakade textklassificeringsmodeller för att kategorisera datamängder i fördefinierade ämnen på hög nivå, där endast ett fåtal annoterade exempel finns för varje ämne, medan största delen av datamängden saknar annotering. För att bättre utforska ämnen lämpliga för nyhetsarkiv och strömmad data, samt adressera begränsningarna som fördefinierade ämnen för textklassificering medför, föreslås djupa klustringsmetoder som kan tränas i oövervakade miljöer.

Utöver detta, och för att förbättra identifieringen av detaljerade ämnen, introducerar avhandlingen en ny modell för upptäckt av berättelser för att övervaka händelsebaserade ämnen i nyhetsströmmar med flera källor. Med tanke på att onlinerapportering av nyheter ofta använder sig av en kombination av olika modaliteter som text, bilder, video och ljud för att förmedla information, undersöker avhandlingen också samverkan mellan textuella och visuella element i analys av nyhetsartiklar. För att uppnå detta mål annoterades en text-bild-datamängd, och en referensmodell för upptäckt av händelserelaterade ämnen i multimodala nyhetsströmmar utvecklades. Även om de i första hand är

avsedda för nyhetsövervakning och kontextuell reklam, kan de föreslagna modellerna merallmänt betraktas som nya tillvägagångssätt för semi-övervakad textklassificering, djup klustring och upptäckt av nyheter. En jämförelse med state-of-the-art visar modellernas effektivitet.



# Preface

This thesis contains the following papers.

- Paper I     Arezoo Hatefi, Xuan-Son Vu, Monowar Bhuyan, and Frank Drewes. Cformer: Semi-Supervised Text Clustering Based on Pseudo Labeling. *In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM)*, pp. 3078-3082, 2021.
- Paper II    Arezoo Hatefi, Xuan-Son Vu, Monowar Bhuyan, and Frank Drewes. The Efficiency of Pre-training with Objective Masking in Pseudo Labeling for Semi-Supervised Text Classification. *Submitted to the Northern European Journal of Language Technology (NEJLT)*, 2023.
- Paper III   Arezoo Hatefi, Xuan-Son Vu, Monowar Bhuyan, and Frank Drewes. ADCluster: Adaptive Deep Clustering for Unsupervised Learning from Unlabeled Documents. *In Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP)*, pp. 68-77, Association for Computational Linguistics, 2023.
- Paper IV    Arezoo Hatefi, Anton Eklund, and Mona Forsman. PromptStream: Self-Supervised News Story Discovery Using Topic-Aware Article Representations. *Accepted to Appear in the Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, 2024.
- Paper V     Arezoo Hatefi, Johanna Björklund, Xuan-Son Vu, and Frank Drewes. METHOD: A Dataset and Baseline for Multimodal Discovery of Event-Based News Topics. *Submitted to the International Journal of Multimedia Information Retrieval*, 2024.



# Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, Frank Drewes and Johanna Björklund for their support, guidance and trust throughout the journey of my PhD. Thanks for believing in me and helping me grow as a researcher and also as an individual. Frank, I deeply appreciate all the insightful discussions and valuable feedback you provided me with. Johanna, you inspired me with your knowledge, experience, management, and personality. I would also like to extend my heartfelt thanks to my co-advisor, Xuan-Son Vu, for his guidance and assistance in finding my research direction when I was at a crossroads. Additionally, I am grateful to my reference person, Patrik Eklund, as well as Monowar Bhuyan for their valuable guidance and contributions.

This thesis was made possible through the funding and support provided by the Industrial Doctoral School at Umeå University and Aeterna Labs. Their contributions have been instrumental in the successful completion of this thesis, and I am grateful for the opportunities they have provided me with. I want to extend special thanks to the amazing team at Aeterna Labs, particularly Mona and Anton. Thanks for the great collaboration and friendly environment you provided me with to peruse my ideas and do my research.

Next, I want to express my gratitude to all my colleagues and friends in the Foundations of Language Processing group at Umeå University for their support and the memorable moments we shared during launches, fikas, and other enjoyable events. Special thanks to Anna and Adam for their assistance in translating the abstract of the thesis into Swedish. I am also thankful to all my colleagues and friends at the Department of Computing Science.

I am deeply indebted to my parents, as well as my extended family, including my in-laws, for all the love and support during all these years. I am immensely grateful to my parents for their unwavering support throughout my entire life, especially during my PhD journey, where their emotional support was a constant source of strength. They have always been there for me, and I deeply appreciate their presence and guidance. To my dear sister and brother, Atefe and Reza, I am incredibly thankful to have you in my life.

Lastly, to my beloved Abbas for endless support and love and for standing by me through the emotional ups and downs of my PhD journey. My life is worth nothing without you. I Love you dearly.



To my beloved Abbas,  
and to my parents, Iran and Ali,  
and to all the courageous Iranian women #Mahsa-Amini.



# Abbreviations

Table 1: List of terminologies and abbreviations used in the thesis

#	Abbreviation	Full form
1	NLP	Natural Language Processing
2	IR	Information Retrieval
3	AI	Artificial Intelligence
4	ML	Machine Learning
5	MLP	Multilayer Perceptron
6	CNN	Convolutional Neural Network
7	RNN	Recurrent Neural Network
8	MLM	Masked Language Modeling
9	LM	Language Model
10	PLM	Pre-trained Language Model
11	LLM	Large Language Model
12	BERT	Bidirectional Encoder Representations from Transformers [28]
13	MPL	Meta Pseudo Labels [115]
14	TF-IDF	Term Frequency-Inverse Document Frequency
15	UDA	Unsupervised Data Augmentation [171]
16	PCA	Principal Components Analysis [165]
17	UMAP	Uniform Manifold Approximation and Projection [97]
18	LDA	Latent Dirichlet Allocation [13]
19	DEC	Deep Embeded Clustering [169]
20	InfoNCE	Normalized Cross-Entropy with Information Maximization [108]
21	TDT	Topic Detection and Tracking Allan [1]
22	CLIP	Contrastive Language-Image Pre-Training [123]





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Automatic News Analysis . . . . .	1
1.2	Contextual Advertising . . . . .	2
1.3	Research Problem and Questions . . . . .	3
1.4	Thesis Outline . . . . .	5
<b>2</b>	<b>Deep Learning</b>	<b>7</b>
2.1	A Brief Introduction to Deep Learning . . . . .	7
2.2	Learning Paradigms . . . . .	9
2.3	Data Dependency in Deep Learning . . . . .	11
2.3.1	Transfer Learning . . . . .	11
2.3.2	Data Augmentation . . . . .	14
2.3.3	Pseudo-labeling and Teacher-Student Architecture . . . . .	15
<b>3</b>	<b>Coarse-Grained News Topic Identification</b>	<b>17</b>
3.1	News Classification . . . . .	17
3.1.1	Classification . . . . .	18
3.1.2	Text Classification . . . . .	20
3.1.3	Semi-Supervised Text Classification . . . . .	21
3.2	News Clustering . . . . .	22
3.2.1	Clustering . . . . .	22
3.2.2	Traditional Clustering Algorithms . . . . .	22
3.2.3	Dimension Reduction . . . . .	23
3.2.4	Deep Clustering . . . . .	24
<b>4</b>	<b>Event-based Topic Discovery in News Streams</b>	<b>29</b>
4.1	Topic Detection and Tracking . . . . .	29
4.2	Online Clustering . . . . .	32
4.3	Multimodal News Streams . . . . .	34
4.3.1	Deep Learning for Multimodal Data . . . . .	35

<b>5</b>	<b>Summary of Contributions</b>	<b>37</b>
5.1	Paper I . . . . .	38
5.2	Paper II . . . . .	39
5.3	Paper III . . . . .	41
5.4	Paper IV . . . . .	42
5.5	Paper V . . . . .	43
	<b>Paper I</b>	<b>63</b>
	<b>Paper II</b>	<b>76</b>
	<b>Paper III</b>	<b>118</b>
	<b>Paper IV</b>	<b>139</b>
	<b>Paper V</b>	<b>159</b>

# Chapter 1

## Introduction

This chapter motivates the development of systems capable of effectively processing and interpreting the wealth of information provided by news sources. It introduces *contextual advertising* as the real-world application that served as the motivation behind this thesis. Furthermore, it highlights the objectives of the thesis and concludes with an overview of the thesis structure.

### 1.1 Automatic News Analysis

In today's world, news serves as a cornerstone for information, awareness, and societal cohesion. It shapes public opinions, guides decision-making, and reflects the evolving global landscape. News monitoring is crucial for individuals, aiding informed decision-making and crisis awareness. Similarly, it is valuable for companies, providing insights into competitors, industry trends, and market developments, supporting strategic planning, positioning, risk management, and market intelligence.

A substantial and continually increasing volume of daily news articles, exemplified by Reuters' production of approximately 5,000 articles from 2,500 journalists<sup>1</sup>, underscores the information flow. Additionally, social media has emerged as a significant news channel [58], especially during crises, influencing public discourse. In this extensive news landscape, automated processing becomes crucial, swiftly navigating through vast content to keep individuals and companies updated on evolving events. Also, by leveraging different sources, automated news processing can offer a broader range of perspectives, giving a well-rounded understanding of events.

In Natural Language Processing (NLP) and Information Retrieval (IR), numerous research lines are dedicated to the study of news, including:

- **News Topic Modeling:** techniques for automatically identifying and

---

<sup>1</sup><https://www.reutersagency.com/en/about/about-us/>

categorizing topics within news articles to understand and organize the content effectively [86, 69, 161, 65, 118].

- **Topic Detection and Tracking:** methods for automatically identifying, monitoring, and organizing emerging topics or events from a continuous flow of news [1, 72, 102, 137, 182, 34, 59].
- **Sentiment Analysis:** methods to determine sentiment and opinions expressed in news articles, helping to understand public reactions and attitudes towards different topics [103, 27, 6, 178].
- **Fake News Detection:** methods to identify misinformation and fake news, including the development of algorithms that assess the credibility and reliability of news sources [168, 195, 194, 179, 193, 67, 98].
- **Personalized News Recommendation:** algorithms for personalized news recommendations based on individual preferences, browsing history, and user behavior [106, 167, 3, 56, 116].
- **Summarization:** techniques to automatically generate concise and informative summaries of news articles, focusing on both extractive and abstractive summarization [11, 146, 174, 190].
- **Multimodal Analysis in News:** research exploring the integration of text with other modalities (images, videos) in news analysis to provide a more comprehensive understanding of news content [22, 65, 161, 105, 117].

The central focus of this doctoral thesis is on the first and second research domains, specifically the identification of topics within news articles at different levels of granularity. This endeavor encompasses the classification of news into broader categories such as “sports” and “politics,” as well as more detailed event-based topics like “a plane crash in Malaysia.” The research includes analyzing both static collections of news articles and continuous streams of news content. The motivation behind this research stems from the application of *contextual advertising*. Furthermore, there exists an interest in multimodal news processing, exploring the correlation between news text and images for the purpose of news topic identification.

## 1.2 Contextual Advertising

Traditional automated advertising relies on cookies and users’ browsing and shopping histories. However, growing privacy concerns have prompted advertisers to explore alternative approaches. Contextual advertising, as a privacy-friendly and less invasive alternative to cookie-based advertising, has emerged in response to these concerns. Contextual advertising, also known as cookie-less advertising, involves placing ads on web pages based solely on their content,

without tracking users and their online behavior. For instance, this could mean displaying ads for an online Artificial Intelligence (AI) course on a news article about AI. Figure 1.1 illustrates a contrast between conventional and contextual advertising methods.

Despite the advantages of cookie-based behavioral advertising, which enables deeper personalization and utilizes browsing history as a strong indicator of buying readiness, many companies are now shifting their advertising approaches toward contextual advertising. An essential advantage is that companies can avoid dealing with the constantly evolving regulations, legislation, and shifting attitudes toward privacy associated with tactics that use cookies to track user online behavior. For instance, General Data Protection Regulation (GDPR) [128] has introduced strict regulations on user consent and data privacy, significantly affecting the use of cookies for advertising purposes. This makes it increasingly challenging for companies to rely solely on cookie-based advertising. The process of obtaining valid consent can be complex, and user rejection of cookies due to privacy concerns is on the rise. Moreover, companies have concerns about brand safety and reputation. Contextual advertising allows them to select the specific contexts in which they want their ads to appear or not appear, providing greater control over the environments associated with their brand. Additionally, contextual advertising prioritizes current context over past behavior, ensuring that personalized ads align with users' immediate interests and needs. By focusing on immediate relevance, it offers a certain level of appropriateness without compromising privacy.

Contextual advertising identifies relevant content for ads by aligning the advertising campaign's keywords or topics with the central theme of the webpages. In this thesis, our focus is on news websites, presenting different approaches for news topic identification. It is important to note that while our motivation stems from an industrial application, the proposed methods can be viewed as general algorithms for document topic identification.

### 1.3 Research Problem and Questions

As outlined in earlier sections, the primary objective of this thesis is to categorize news articles into distinct topics, spanning various levels of granularity, for subsequent application in contextual programmatic advertising. News articles typically convey their messages through a multimodal approach, utilizing diverse modalities such as text, image, and video. Understanding the interplay between these modalities and understanding their respective contributions to topic identification is also an aspect studied in this thesis.

The accessibility of data intended for topic identification varies across different situations. In some cases, all data is readily accessible, while in others, the setting is more dynamic, presenting data as a continuous stream gradually accessible to the model. The dynamic scenario introduces the potential for various topic behaviors, such as emergence, disappearance, distribution shift,

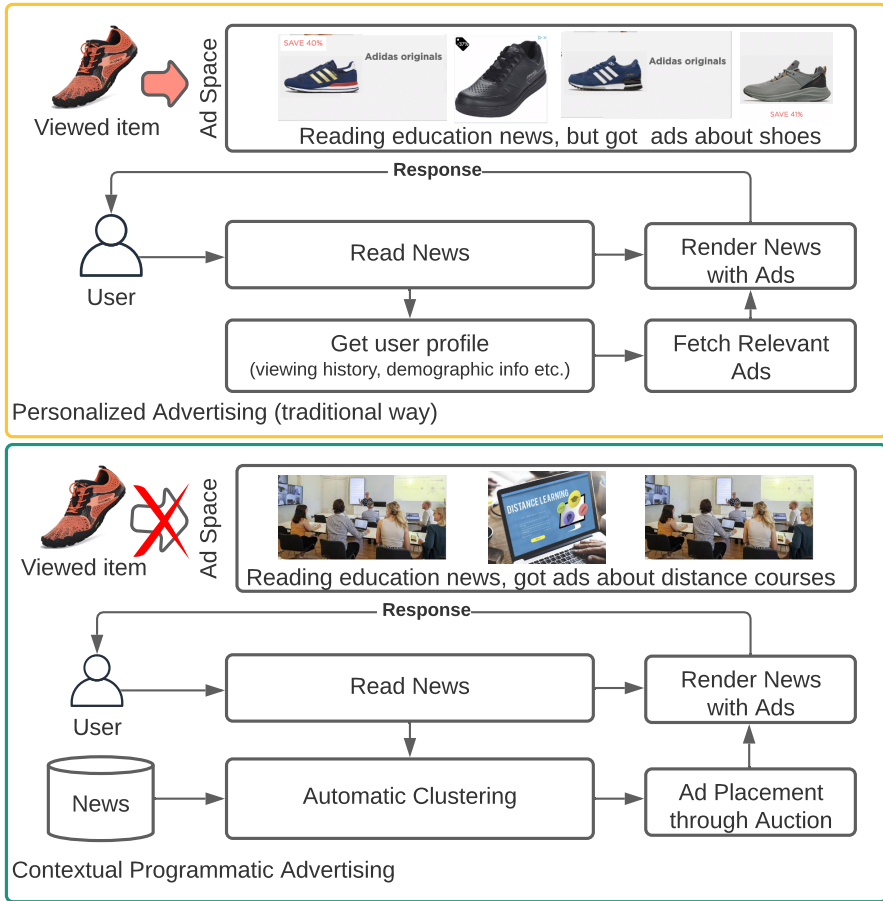


Figure 1.1: A conceptual comparison between cookie-based advertising and contextual advertising within a programmatic advertising framework. The former relies on personal information, while the latter solely utilizes news content.

splitting, and merging, thereby complicating the task at hand. In such dynamic situations, the model must dynamically adapt to changes in the data stream to effectively predict future topics.

Deep learning, a specialized area of machine learning, is primarily focused on representation learning and has shown remarkable success across various domains in recent years. Its accomplishments include achieving state-of-the-art results, often leveraging the computational power of GPUs and TPUs to train deep neural networks on large datasets. Motivated by the effectiveness of deep learning, this thesis seeks to investigate its use for topic identification, especially in situations where labeled data is limited or unavailable.

Consequently, the thesis addresses four pivotal research questions:

**RQ1** How can news topics be automatically identified across various granularity levels?

**RQ2** What effective methodologies can be employed to integrate deep learning into the investigation of news topics when labeled data is scarce or unavailable?

**RQ3** How can deep learning techniques be utilized for topic identification in news streams while effectively addressing challenges associated with changes in topic focus and evolution over time?

**RQ4** What is the interrelation between different modalities within multimodal news, and how can these modalities be harnessed for the purpose of topic identification?

## 1.4 Thesis Outline

The thesis is structured into four main chapters. Chapter 2 provides a brief introduction to deep learning and learning paradigms. Moreover, it discusses the data-dependency of deep learning methods as a challenge in the field and presents some solutions, such as transfer learning, data augmentation, and pseudo-labeling, to address this challenge. These techniques have been utilized in various parts of the thesis for training models in semi-supervised and unsupervised settings. Chapter 3 is centered on coarse-grained topic identification for news articles. It introduces classification and clustering tasks, along with other relevant preliminaries essential for comprehending **Papers I, II, and III**, which focus on addressing research questions **RQ1** and **RQ2** concerning coarse-grained topics. **Paper I** and **Paper II** introduce semi-supervised classification models utilizing deep learning for coarse-grained news topic identification, while **Paper III** suggests deep clustering in unsupervised settings for the same purpose. Chapter 4, is focused on identification of fine-grained event-based topics in streams of news articles and provides the background for **Papers IV** and **V**, which tackle research questions **RQ3** and **RQ4**, respectively. Chapter 5

concludes the thesis by summarizing the research contributions. Furthermore, the thesis includes five papers related to the research.



# Chapter 2

# Deep Learning

## 2.1 A Brief Introduction to Deep Learning

Machine Learning (ML), encompasses algorithms designed to learn from data presented as feature vectors and predict outcomes for new data. These feature vectors encompass features denoting distinct aspects or attributes of the data, carefully crafted by human experts through a process known as feature engineering. For example, when predicting the appropriate food category for recipes, a comprehensive feature set might encompass a range of ingredients and cooking techniques such as grilling, baking, frying, steaming, and beyond.

Deep Learning, a subfield of Machine Learning, specializes in data representation learning, automating the feature extraction process by eliminating the need for human intervention. Deep learning algorithms ingest unstructured raw data, such as text and images, and autonomously infer the crucial features for decision making [76]. The cornerstone of deep learning lies in deep neural networks, which are composed of multiple interconnected layers of artificial modules many of which compute non-linear input–output mappings.

Deep Feedforward Networks, also known as Multilayer Perceptrons (MLPs), serve as fundamental modules in deep learning, characterized by multiple layers of neurons comprising an input layer, one or more hidden layers, and an output layer. The addition of extra hidden layers enhances the model’s predictive capability, particularly with abundant training data. In an MLP, each neuron in a layer connects to every neuron in the subsequent layer, forming a fully connected network structure. Transitioning from one layer to the next, each neuron calculates a weighted sum of its inputs from the preceding layer and passes the result through a non-linear function such as the Rectified Linear Unit (ReLU), defined as the half-wave rectifier ( $f(z) = \max(z, 0)$ ). When a neuron’s output surpasses its threshold, it becomes activated, transmitting data to the subsequent layer. Conversely, if the output falls below the threshold, no data is transmitted. This sequential computation process across the network is termed forward propagation.

During the training process, the learning algorithm fine-tunes the weights of the network using a method known as backpropagation [134, 132]. This technique, rooted in the chain rule for derivatives [148], computes the gradient of the training objective function with respect to the network’s weights. Essentially, it quantifies how each weight contributes to the overall prediction error. Backpropagation operates by recursively applying the chain rule to propagate gradients backward through the network layers, starting from the output and moving towards the input. These gradients serve as crucial information for optimization algorithms like variants of gradient descent [133] to adjust the network weights. Through this iterative refinement process, the network steadily improves its predictive performance.

MLPs have been widely used in various fields, including image recognition, natural language processing, and financial forecasting. Despite their simplicity compared to more complex architectures, MLPs remain powerful tools in machine learning and serve as the foundation for many deep learning models.

In addition to MLPs as the most basic form of deep neural networks, deep learning encompasses diverse neural network architectures tailored to tackle specific challenges or datasets.

**Convolutional Neural Networks (CNNs)** are a type of deep learning models primarily designed for processing and analyzing visual data, such as images and videos [76]. They consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers use filters to extract features from input images, while pooling layers reduce spatial dimensions. CNNs leverage parameter sharing and hierarchical feature learning to efficiently extract meaningful patterns from the data, making them highly effective for tasks such as image classification, object detection, and image segmentation.

**Recurrent Neural Networks (RNNs)** are a type of neural network designed to work with sequential data, where the order of the data points matters [134]. Unlike feedforward neural networks, which process each input independently, RNNs maintain a memory of previous inputs by using loops within the network architecture. This memory enables RNNs to capture temporal dependencies and patterns in sequential data, making them well-suited for tasks such as time series prediction, natural language processing, speech recognition, and handwriting recognition. RNNs are characterized by their ability to handle inputs of varying lengths and their capability to learn from past information to make predictions about future data points. However, traditional RNNs suffer from the vanishing gradient problem, which can hinder their ability to learn long-range dependencies. To address this issue, variants of RNNs such as Long Short-Term Memory (LSTM) networks [52] and Gated Recurrent Units (GRUs) [24] have been developed, which incorporate mechanisms to better retain and update information over long sequences.

**Autoencoders** aim to learn a compressed representation of input data by first encoding it into a lower-dimensional space (encoder) and then reconstructing it from this representation (decoder) [48]. Autoencoders are commonly used

for tasks like data denoising, dimensionality reduction, and anomaly detection.

**The Transformer architecture**, introduced in [162], is a groundbreaking model for natural language processing. It has an encoder-decoder architecture. Initially, the encoder processes the input sequence, converting it into a set of contextualized representations. Subsequently, the decoder utilizes these representations to produce the output sequence, attending to relevant sections of the input sequence through cross-attention mechanism. While the encoder and decoder are often used together in sequence-to-sequence tasks like machine translation, they can also be used separately for specific tasks that only require encoding or decoding functionality. For instance, encoder models can be utilized separately for tasks like text classification or named entity recognition. Similarly, decoder models can be employed independently for tasks such as text generation or language modeling. The Transformer utilizes self-attention to capture long-range dependencies between words in a sentence efficiently. Self-attention empowers the network to consider every other word and determine the significance it should assign to different words when generating representation of a word in the input sequence. Other key components are multi-head attention for focusing on different aspects of the input, positional encodings to provide sequential order information, feedforward neural networks for complex feature extraction, and layer normalization with residual connections for stable training. Unlike recurrent models, Transformers process the entire input sequence in parallel, making them highly efficient for both training and inference. Transformers have revolutionized NLP by outperforming older models like RNNs and CNNs, making them widely adopted in both research and industry for various tasks such as machine translation and text generation.

## 2.2 Learning Paradigms

The supervised, unsupervised, semi-supervised, and reinforcement learning paradigms are different approaches to training machine learning models, each with its own characteristics and applications.

- **Supervised Learning** involves training a model on a dataset consisting of input-output pairs, where the input data is associated with corresponding labels or target values. The goal is to learn a mapping from inputs to outputs based on the provided examples, enabling the model to make predictions on new, unseen data [12]. Common tasks in supervised learning include classification (predicting discrete labels) and regression (predicting continuous values). Examples of supervised learning algorithms include decision trees [119], support vector machines [26], and neural networks [47].
- **Unsupervised Learning** involves training the model on input data without explicit labels or target values. Instead, the model learns patterns, hidden structures, or representations inherent in the data without

guidance [45]. Clustering algorithms, dimensionality reduction techniques like Principal Component Analysis (PCA) [63], and generative models like Gaussian mixture models [130] are common examples of unsupervised learning algorithms. Self-supervised learning is a special case of unsupervised learning that has gained popularity in deep learning as a way to leverage large amounts of unlabeled data for representation learning and for pre-training models which can then be fine-tuned on smaller labeled datasets for specific tasks. In self-supervised learning, a model is trained using supervision signals that are automatically generated from the input data itself, without requiring manually labeled data. This typically involves creating auxiliary tasks or objectives that are related to the main task of interest but do not require explicit human annotation. Examples of deep networks trained in a self-supervised manner include Autoencoders [49], generative models such as Generative Adversarial Networks (GANs) [39] and Variational Autoencoders (VAEs) [68], as well as Pre-trained Language Models (PLMs) [28].

- **Semi-Supervised Learning** combines elements of both supervised and unsupervised learning paradigms [21]. In this approach, the model is trained on a dataset that contains a small amount of labeled data along with a larger amount of unlabeled data. Semi-supervised learning is particularly useful when obtaining labeled data is expensive or time-consuming, as it allows using abundant unlabeled data to improve model performance. Unlabeled data can contribute to the learning process in several ways. It can be used for regularization purposes, particularly in consistency training [171]. Consistency regularization techniques encourage the model to produce consistent predictions for similar examples in the unlabeled data. By penalizing inconsistencies between predictions on different perturbations of the same input, the model learns to generalize better and become more robust. Unlabeled data can also be used as a form of pseudo-labeled data [78, 115]. In this approach, the model generates predictions for the unlabeled data, treating these predictions as pseudo-labels. The model is then trained on a combined dataset consisting of both labeled and pseudo-labeled data.
- **Reinforcement Learning (RL)** is a learning paradigm which is commonly used in scenarios where explicit supervision is not available, and the agent must learn through trial and error. The agent learns by interacting with the environment, receiving feedback in the form of rewards or penalties, and adjusting its actions accordingly to achieve its goals. Reinforcement learning has applications in various domains, including robotics, game playing, autonomous driving, and recommendation systems. This line of algorithms are not the focus of this thesis. However, interested readers are recommended to see Sutton and Barto [151] for detailed information.

## 2.3 Data Dependency in Deep Learning

The data dependency challenge in deep learning refers to the reliance of deep neural networks on large amounts of labeled data for effective training. Deep learning models often require massive datasets to learn complex patterns and achieve high performance on various tasks. However, collecting labeled data can be costly, time-consuming, and sometimes impractical. Over the years, researchers have devised numerous methods to address this challenge. Here, we explain some of these techniques that are relevant to the field of NLP including transfer learning, data augmentation, and pseudo labeling.

### 2.3.1 Transfer Learning

In transfer learning, knowledge gained from solving one problem is applied to a different but related problem. In the context of deep learning, transfer learning involves taking a model trained on one task and fine-tuning it on a different task. This allows the model to leverage knowledge learned from the first task to improve its performance on the second task, especially when the second task has limited training data. In the context of natural language processing, word embeddings such as Word2Vec [101], GloVe [110], and FastText [15], where words are represented as dense vectors, were used to transfer knowledge from large text corpora to downstream NLP tasks however, pre-trained language models like BERT [28] have popularized and significantly advanced the application of transfer learning in NLP.

**Pre-trained Language Models (PLMs)** serve as a foundation for transfer learning in NLP. The key idea is acquiring a general, latent representation of language through a generic task, then using this knowledge for various NLP tasks. Language modeling, where the model predicts a word based on its context, serves as one such generic task due to the abundance of self-supervised text available for training. The process of training a deep neural network with a language modeling objective on a large corpus is termed *pre-training*. However, to effectively utilize the pre-trained model for downstream NLP tasks, further training or task adaptation is typically required. Existing task adaptation methods include *fine-tuning* the PLMs for the specific task, *prompting* the PLMs to execute the desired task, or reformulating the task as a *text generation* problem.

Almost all widely-used PLMs, such as those from the GPT series [16, 120, 121], BERT [28] and its variants, BART [79], and T5 [125], are built upon the Transformer architecture [162]. A Transformer-based language model can fall into one of three architectures: decoder-only (e.g., GPT [120] and Gopher [124]), encoder-only (e.g., BERT [28] and XLM-R [25]), or encoder-decoder (e.g., BART [79], T5 [125], and T0 [136]). Furthermore, models can be trained using different objectives: autoregressive training (predicting the next word based on preceding context), masked language modeling (MLM) (filling in the missing word, i.e., predicting the masked word given surrounding context), or

various denoising tasks where the model must undo some form of corruption in the original sequence, such as sentence permutation, token deletion, or span deletion. Typically, though not necessarily, decoder-only models are trained with an autoregressive objective, encoder-only models utilize MLM for training, and encoder-decoder architectures are trained on denoising tasks or MLM.

While autoregressive models process input sequentially, masked language models predict a masked word based on all other words in the sequence offering greater contextual information. During MLM training, a random subset of tokens in the input text sequence is masked using a special token [MASK], and the model is trained to predict these masked tokens based on both left and right contexts. Hence, the training goal is to optimize the log-likelihood:

$$\sum_i m_i \log(P(x_i|x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n); \theta_T)$$

where  $m_i \in \{0, 1\}$  indicates whether  $x_i$  is masked or not, and  $\theta_T$  denotes the model parameters. MLMs incorporate multiple transformer encoder layers [162] to progressively acquire meaningful representations. Prominent examples include BERT [28], RoBERTa [90], and XLM-R [25].

Lately, researchers have tried to enhance the adaptability of PLMs to specific tasks by further pre-training them with task-specific masking techniques. This method, referred to as *objective masking*, seeks to incorporate downstream task-related information into general PLMs through masking [64, 188, 155, 145, 41]. Gu et al. [41] proposed a three-stage framework for text classification by adding a task-guided pre-training stage with selective masking between general pre-training and fine-tuning stages. The method first finds the important words for the downstream task in the in-domain unsupervised dataset using a binary classifier trained on the supervised task-specific dataset (which has already been annotated automatically with word importance information) and then masks them in the task-guided pre-training stage. In **Paper II**, we utilize objective masking to incorporate topical information, collected in an unsupervised way based on statistical information from the unlabeled data, into the PLM for topic classification.

**Fine-tuning** adjusts the contextual representations obtained during pre-training for distinct NLP tasks. Typically, for classification tasks such as sentiment analysis, natural language inference, and semantic similarity, one or two feed-forward classification layers, referred to as prediction heads [166], are appended on top of the PLM. The classification head transforms the contextualized embeddings generated by the language model into predictions for desired classes. Both the output layers and the PLM undergo training simultaneously in an end-to-end setup, with the major computational load allocated to fine-tuning the LM. It is crucial to carefully select the learning rate for the weights of the feed-forward layer(s) and for the PLM in this configuration. Given that the PLM is already extensively trained, a low learning rate is advisable, especially for smaller datasets. Conversely, the feed-forward layer weights, which are initialized randomly, necessitate considerable training. The word embeddings are

derived either directly from the top layer of the language model or through a concatenation or weighted average of the top  $n$  (typically  $n = 4$ ) layers [113]. The text representation can then be computed by taking a weighted average of the word embeddings or the representation of the special [CLS] token. It is worth noting that certain tasks, such as parsing tasks, demand significant additional architecture atop a PLM [183]. In such instances, ample training data and computational resources are essential to train both the task-specific architecture and effectively fine-tune the PLM.

**Prompting** is the practice of inserting natural language text or continuous vectors in the input to guide PLMs in executing particular tasks. The objective of prompting is to simplify the downstream task for the language model by utilizing the prompts as contextual cues. Prompting serves as a knowledge probing technique for PLMs, enabling evaluation of their acquired knowledge for specific tasks [114]. Two common approaches to prompt learning are *template-based learning* and *in-context learning*.

*Template-based learning* reformulates NLP tasks into tasks resembling pre-training tasks of language models, such as MLM, by employing templates. This strategy effectively utilizes the knowledge learned during pre-training, resulting in a significant reduction in the number of task-specific training examples needed, which is particularly advantageous in scenarios with limited data [74]. Le Scao and Rush [74] conducted an extensive analysis to quantify the advantages of prompts in classification tasks. Their study involved controlled fine-tuning across various tasks and data sizes, demonstrating that the use of prompts consistently enhances performance compared to relying solely on traditional fine-tuning methods. For supervised template-based prompt learning, labeled examples are converted into “natural” text using carefully crafted templates with open slots. Subsequently, solving the tasks becomes a matter of filling these slots with words or phrases using PLMs and then mapping these outputs to task-specific labels via a verbalizer. *Cloze-style* templates introduced in [138] are one of the widely used templates. Table 2.1 shows examples of this approach, for text classification, sentiment classification, textual entailment, and probing for facts.

Task	Cloze-style template	PLM’s output	Task-specific class
Topic classification	— News: [Article Text]	Politics	1
Sentiment classification	[Movie Review]. Overall, it was —.	Disappointing	Negative
Textual entailment	The cat is on the table? —, the cat is under the table.	No	FALSE
Probing for facts	Obama was the president of the —.	U.S.	-

Table 2.1: Examples of close-style prompting for different NLP tasks.

In **Paper IV** we used cloze-style prompting to make topic-aware document representation for topic identification.

*In-context learning* or learning from instructions and demonstrations is particularly efficient when applied to large generative PLMs, known as Large

Language Models (LLMs). These models were initially introduced in [121], featuring tens to hundreds of billions of parameters. They have demonstrated significant performance across various NLP tasks in zero-shot and few-shot settings. LLMs such as GPT-3 [16] exhibit the capability to handle diverse NLP tasks in a few-shot setting through in-context learning. In-context learning provides LLMs with instructions and a few input-output examples for a specific task, allowing them to produce desired outputs for new inputs without the need for gradient updates. In contrast to the easy implementation, there are certain limitations to consider for this prompting approach. Firstly, its success in few-shot tasks heavily relies on the sheer size of LLMs, limiting its applicability. Moreover, insights from [112] indicate that the few-shot performance of PLMs is very sensitive to the choice of prompts which limits the robustness of the approach.

### 2.3.2 Data Augmentation

Data augmentation was first introduced in computer vision to enhance the quantity and diversity of training examples by modifying the original data while maintaining its semantic significance [75, 143]. This process is relatively straightforward in computer vision, where techniques such as cropping, rotating, flipping, or color jittering can be applied to generate new examples without changing the underlying subject matter [141].

However, in NLP, due to the discrete nature of language, augmenting text while maintaining its meaning poses greater challenges. Even minor alterations can significantly change the meaning of the text. Nonetheless, researchers have proposed several promising methods for text augmentation including token-level random perturbation operations such as random insertion, deletion, and swap [164], back translation (translating the text into another language and then back into the source language) [140], replacing words with synonyms [187], utilizing PLMs to substitute words based on their contextual surroundings [71], and guided generation using large-scale generative language models [89, 88]. Another example is TMix [23], inspired by MixUp in computer vision [186], which interpolates two or more text instances and their labels in their respective hidden space. In **Papers I** and **II**, we used PLMs to generate augmented text for semi-supervised text classification.

Data augmentation methods are employed to expand labeled datasets when training data is insufficient, as exemplified by Augmented SBERT [154]. Additionally, these techniques can introduce noise to data to enhance model robustness, particularly in consistency training. Xie et al. [171] proposed to replace the traditional noise injection methods by high quality data augmentation such as back translation of textual data.



### 2.3.3 Pseudo-labeling and Teacher-Student Architecture

To tackle the data dependency issue in deep learning methods, one effective approach is to train the deep neural network in a semi-supervised manner. Semi-supervised learning encompasses various techniques like self-training [163, 2] and temporal ensembling [73]. Among these methods, pseudo labeling using a teacher-student architecture stands out as a commonly utilized approach.

The teacher-student architecture was initially used for knowledge distillation from a large teacher to a light-weight student while maintaining comparable performance with the teacher [50, 152]. Recently, the teacher-student architecture has found widespread application in various types of knowledge learning objectives including knowledge expansion [170, 144], knowledge adaptation [96, 156], and multi-task learning [37, 176].

Knowledge expansion aims to train a student model with superior generalizability and performance compared to the teacher model by leveraging the vast amount of unlabeled data in a semi-supervised fashion. In this approach, the capacity of the student model is either the same as or larger than that of the teacher model. To achieve this goal through an offline approach, the teacher model is initially trained or fine-tuned using labeled data. Subsequently, it generates predictions, termed pseudo-labels, for the unlabeled dataset. Both the labeled and pseudo-labeled datasets are then utilized to train the student model. This process facilitates the transfer of knowledge from the teacher to the student, potentially leading to improved performance as the student benefits from pseudo-labeled data and the application of regularization techniques such as data augmentation [170]. Despite the simplicity and computational efficiency of the offline learning scheme, it has a drawback: if the pseudo-labels predicted by the teacher network are inaccurate, confirmation bias may arise, as the student network may reinforce existing inaccuracies [4]. Consequently, the student may not surpass the performance of the teacher significantly. To mitigate the confirmation bias problem, Pham et al. [115] proposed Meta Pseudo Labels (MPL), an iterative training approach for both the teacher and student networks, resulting in enhanced performance for both networks. In each iteration, the teacher receives feedback from the student in the form of the student’s performance on the gold-labeled data and adjusts itself accordingly to predict more accurate pseudo-labels in the subsequent iteration. In **Papers I** and **II**, we employed pseudo-labeling for text classification, and in **Paper III**, it was used for clustering.



## Chapter 3

# Coarse-Grained News Topic Identification

This chapter provides the background knowledge for **Papers I, II, and III**, which are focused on addressing research questions **RQ1** and **RQ2**. **Paper I** and **Paper II** propose semi-supervised classification models using deep learning for news topic identification. This is particularly relevant when predefined coarse-grained topics are of interest and there is insufficient labeled data available to effectively train the classifier. **Paper III** advocates for deep clustering in scenarios where a set of predefined classes is absent, yet there is a desire to explore coarse-grained topics within the news dataset. This chapter provides an introduction to both classification and clustering tasks, as well as other relevant preliminaries.

### 3.1 News Classification

Classification plays a crucial role in various applications within the News domain, such as fake news detection, sentiment analysis, and topic identification. To categorize news articles into topics, standard taxonomies like Interactive Advertising Bureau (IAB) tags<sup>1</sup> and IPTC media topics<sup>2</sup> are commonly utilized, offering predefined classes organized into hierarchical structures. The IAB provides a standardized categorization system tailored for classifying news content, aiding advertisers and publishers in effectively categorizing news articles. This taxonomy covers a wide range of topics, from “politics” and “sports” to “entertainment” and “technology”, with each category further subdivided for enhanced granularity. By ensuring consistency and clarity in labeling and organization, the IAB taxonomy facilitates targeted advertising and efficient content monetization for publishers.

---

<sup>1</sup><https://www.iab.com/guidelines/content-taxonomy/>

<sup>2</sup><https://iptc.org/standards/media-topics/>

Utilizing deep learning for categorizing news articles into predefined classes, such as those found in one of the layers of the IAB taxonomy, typically requires a substantial labeled dataset, which is often unavailable or requires significant manual effort to compile. The extensive range of potential topics further complicates the task of gathering training documents to construct a supervised classification model. Consequently, semi-supervised approaches have gained popularity in this field.

Prior studies in contextual advertising have explored the creation of labeled datasets for training classifiers using class-specific keywords and knowledge bases. Jin, Wanvarie, and Le [62] proposed a method to model contextual targeting as a lightly-supervised one-class classification problem. Their algorithm takes unlabeled documents and labeled keywords for the target class  $c$  as input, generating a classifier  $M_c$  specifically designed to identify documents belonging to class  $c$ . However, the dynamic nature of contextual advertising poses challenges in preparing effective keywords for each class and maintaining one-class classifiers, especially for large-scale applications. In a related study, Jin, Kadam, and Wanvarie [61] automated the process of mapping categories in the IAB taxonomy to category nodes in the Wikipedia category graph. Through label propagation across the graph, they obtained a list of labeled Wikipedia documents for training purposes.

To the best of our knowledge, no previous study has explored the utilization of pre-trained language models and their implicit knowledge, alongside deep learning techniques, for categorizing news articles specifically for contextual advertising purposes.

### 3.1.1 Classification

Classification is one of the fundamental and challenging problems in machine learning, with applications in various fields such as natural language processing, computer vision, and speech recognition. It involves categorizing a set of data instances into predefined classes. In classification tasks, the model aims to learn patterns and relationships in the data that distinguish between different classes, enabling it to accurately classify unseen instances based on their features or attributes. This process involves training the model on labeled data, where each data instance is associated with a known class label, and then evaluating its performance on unseen data to assess its ability to generalize to new examples [12]. Common evaluation metrics for classification tasks include:

- Accuracy: the proportion of correctly classified instances out of the total number of instances. It is a simple and intuitive metric but can be misleading in the presence of imbalanced classes.
- Precision: The proportion of true positive predictions out of all positive predictions made by the model. It measures the accuracy of positive predictions and is useful when the cost of false positives is high.

- Recall (Sensitivity): the proportion of true positive predictions out of all actual positive instances in the dataset. It measures the ability of the model to identify all positive instances and is important when the cost of false negatives is high.
- F1 Score: the harmonic mean of precision and recall. It provides a balance between precision and recall and is particularly useful when there is an imbalance between the classes.
- Area Under the ROC Curve (AUC-ROC): the area under the receiver operating characteristic (ROC) curve, which plots the true positive rate against the false positive rate at various threshold settings. AUC-ROC summarizes the performance of a classifier across all possible threshold settings and is particularly useful for imbalanced datasets.

These metrics provide different perspectives on the performance of a classification model and are chosen based on the specific requirements and characteristics of the dataset and task at hand. In this thesis *accuracy* has been used to measure classification performance.

Classification plays a crucial role in numerous real-world applications, including sentiment analysis [51, 33, 184, 139], document classification [181, 60], fraud detection [30, 111, 35], stock market prediction [175, 38, 46], face recognition [99, 91], disease diagnosis [131, 66], and so much more, where accurately categorizing data instances into meaningful classes is essential for decision-making and problem-solving.

Classification problems are typically categorized into four distinct types: binary, multi-class, multi-label, and imbalanced [12]. Binary classification involves tasks with two class labels. Multi-class classification pertains to tasks with more than two class labels. Multi-label classification refers to tasks where each example may have two or more class labels predicted. Imbalanced classification addresses tasks where the distribution of examples across classes is uneven. The classification problems in this thesis belong to the category of *multi-class classification*.

There are many classification algorithms in traditional machine learning including: naive bayes, decision trees, Support Vector Machines (SVM) and k-Nearest Neighbors (kNN) [12]. Also many deep learning models such as CNNs [76], RNNs [134], transformer Models (e.g., BERT [28], GPT [120]), and Autoencoders [48] can be used for classification tasks. Classic algorithms are often simpler and more interpretable, while deep learning algorithms tend to be more complex and capable of learning intricate patterns from large-scale data. The choice of algorithm depends on factors such as the size and nature of the dataset, computational resources, and the specific requirements of the classification task. Given the success of deep learning over traditional classification algorithms in NLP tasks, this thesis employs deep learning methodologies for news classification.

### 3.1.2 Text Classification

Text classification poses a significant challenge because it requires an effective representation of text capable of distinguishing between various classes. Initially, text was represented using the Term Frequency-Inverse Document Frequency (TF-IDF) approach, treating it as a bag of words [135]. This method involves creating a vocabulary, after which each piece of text is represented by a vector showcasing the TF-IDF values of the vocabulary words. The TF-IDF of word  $t$  in document  $d$  is computed as a product of the term frequency  $tf(t, d)$  and the inverse document frequency  $idf(t, d)$ .  $tf(t, d)$  is the relative frequency of word  $t$  in document  $d$  and  $idf(t, d)$  is a measure of how much the word is common or rare across all documents. Various methods exist for computing these statistics. This representation has been commonly used with classical machine learning algorithms for text classification. However, the TF-IDF representation has limitations, such as its inability to account for sequential word orders and contextual information. Additionally, these vectors often have high dimensionality, resulting in computationally expensive operations and the curse of dimensionality problem [10].

To address the limitations of TF-IDF representations, word embeddings were introduced. Traditional word embeddings are static representations of words in a continuous vector space, where each word is assigned a fixed vector regardless of its context. Examples of traditional word embedding models include Word2Vec [101], GloVe [110], and FastText [15]. These models are typically learned using unsupervised learning techniques on large text corpora, analyzing the co-occurrence patterns of words within a context window in the training data. The underlying concept is that words appearing in similar contexts are likely to have similar meanings and should thus be close to each other in the vector space. These representations have commonly been employed with deep neural architectures such as LSTMs [52] and Autoencoders [48] for text classification tasks.

Following the rapid improvement of deep learning in the last decade, NLP has witnessed a drastic improvement in word and text representations resulted from emergence of pretrained language models. PLMs offer highly effective general-purpose contextual word embeddings that can be fine-tuned for specific domains. Contextual word embeddings are word representations that capture the meaning and context of words based on their surrounding context in a sentence or document. Unlike traditional word embeddings, which assign a single fixed vector to each word regardless of context, contextual word embeddings generate dynamic embeddings that vary based on the context in which the word appears. This allows contextual word embeddings to capture nuances in meaning and polysemy, as well as syntactic and semantic relationships between words. Today, encoding text with these language models has become standard practice as the initial step in text classification.

### 3.1.3 Semi-Supervised Text Classification

This section reviews prior research on semi-supervised text classification, exploring methodologies that use unlabeled data to enhance classification performance. Most of these methods have been used as baselines in **Paper I** and **Paper II**.

Several recent semi-supervised learning methods leverage consistency training on extensive amounts of unlabeled data [73, 153]. These techniques regularize model predictions to remain unaffected by minor levels of noise. Xie et al. [171] explored the impact of noise injection in consistency training and introduced Unsupervised Data Augmentation (UDA) as an alternative approach, replacing traditional noise injection with high-quality data augmentation techniques such as back translation for textual data.

Chen, Yang, and Yang [23] introduced TMix, a text augmentation technique interpolating two texts within their semantic hidden space. TMix promotes linear behavior across the training dataset. Additionally, they presented MixText, a new semi-supervised learning approach for text classification leveraging TMix. MixText employs a BERT-based text encoder equipped with TMix, followed by a linear classifier. During training iterations, it initially predicts labels for unlabeled data using the current model and subsequently trains the model with pseudo labeled data using TMix augmentation.

FLiText, introduced by Liu et al. [85], is a lightweight model designed for scenarios where resources are limited. Initially, an inspirer network, based on a transformer model, is trained using both labeled and unlabeled data. Following this, the inspirer network is distilled into a smaller CNN-based model using output-based distillation, which relies on the inspirer’s output, and feature-based distillation, utilizing the layer weights of the inspirer. FLiText significantly enhanced inference speed while maintaining or surpassing the state-of-the-art performance of lightweight models.

Xu, Liu, and Abbasnejad [172] proposed a novel approach to leverage the matching capability inherent in pre-trained language models like BERT for classification tasks. They identified class keywords as words with high attention weights during fine-tuning of a BERT classifier on class samples, thereby creating class semantic representations (CSRs). These CSRs are integrated with sentences and fed into the encoder. A matching classifier is added on top of the BERT encoder alongside a conventional K-way classifier that compares sentences with CSRs. Both classifiers are jointly trained, and CSRs are progressively improved using the updated language model. This method achieved state-of-the-art performance on text datasets, particularly in scenarios with limited labeled data.

Yang et al. [180] introduced prototype-guided pseudo-labeling (PGPL) for semi-supervised text classification. For each class, they selected the  $k$  nearest samples to the corresponding class prototype for the subsequent training iteration to ensure a balanced training process. Additionally, they trained the model with prototype-anchored contrasting, pushing samples toward their

respective class prototypes and away from others. This approach effectively alleviates underfitting near class decision boundaries and enhances text classification performance.

## 3.2 News Clustering

The dynamic nature of news presents a challenge for creating classifiers capable of effectively capturing and categorizing new articles. In scenarios where predefined classes are lacking and the objective is to explore the content of news articles, clustering emerges as a suitable tool. Clustering provides a robust means to organize vast collections of data without the need for predefined categories. In **Paper III**, we explore innovative ways to uncover patterns and structures within the ever-evolving landscape of news articles, utilizing the power of language models and clustering techniques.

### 3.2.1 Clustering

Clustering is a technique used in unsupervised machine learning to group similar data points together based on certain characteristics. The goal of clustering is to partition a dataset into distinct groups, or clusters, where data points within the same cluster are more similar to each other than to those in other clusters [12]. Clustering is commonly used in data analysis, pattern recognition, image segmentation, and recommendation systems, among other applications.

### 3.2.2 Traditional Clustering Algorithms

Many traditional strategies for clustering arbitrary sets of data points in an  $n$ -dimensional space have been proposed. These algorithms can generally be categorized into partitional, density-based, hierarchical, grid-based, and model-based categories based on their underlying principles and methodologies [173]. Each category of clustering algorithms has its own advantages and limitations, and the choice of algorithm depends on factors such as the nature of the data, the desired cluster structure, and computational considerations. In this thesis, partitional, and density-based clustering algorithms have been used.

Partitional clustering algorithms divide the dataset into a set of disjoint clusters, where each data point belongs to exactly one cluster. These algorithms typically require specifying the number of clusters in advance. Popular examples include K-Means [93] and K-Medoids [127] algorithms.

Density-based Clustering algorithms identify clusters based on the density of data points in the feature space. Clusters are formed around regions of high density, separated by regions of low density. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [32] is a popular density-based clustering algorithm.



### 3.2.3 Dimension Reduction

As the number of features or dimensions in a dataset increases, several challenges arise, such as sparsity of data, increased computational complexity, overfitting, and difficulty in visualization. These difficulties are collectively referred to as the curse of dimensionality [10]. Dimensionality reduction algorithms are methods utilized to decrease the number of features dimensions within a dataset while retaining its vital information. Widely employed in machine learning and data analysis, they serve to combat the curse of dimensionality [10].

Numerous algorithms are available for dimensionality reduction, falling into two main classes: matrix factorization-based methods and manifold learning-based methods. Matrix factorization methods drawn from the field of linear algebra, seek to derive a lower-dimensional representation of the data by decomposing the original high-dimensional matrix into lower-dimensional components. Popular methods in this category include Principal Components Analysis (PCA) [63], Singular Value Decomposition (SVD) [29], and Non-Negative Matrix Factorization (NMF) [77]. PCA [63], for example, is a commonly used linear dimensionality reduction technique that identifies principal components, which are directions along which data variation is most pronounced. Ultimately, PCA projects the original data onto selected principal components, effectively reducing dimensionality while preserving maximum variance.

Manifold learning methods utilize the geometric structure of the data, often represented as a neighborhood graph, to find a lower-dimensional embedding that preserves the local relationships between data points. Some of the more popular methods in this category include Spectral Embedding [9], t-distributed Stochastic Neighbor Embedding (t-SNE) [158], and Uniform Manifold Approximation and Projection (UMAP) [97]. UMAP, in particular, stands out as a state-of-the-art nonlinear dimensionality reduction approach widely adopted for visualizing and analyzing high-dimensional data. UMAP performs dimension reduction by constructing a low-dimensional embedding that captures both local and global relationships between data points, making it particularly adept at capturing complex patterns and relationships.

Furthermore, Autoencoders [48], as neural network architectures, offer a robust technique for dimensionality reduction by learning compact representations from high-dimensional data in an unsupervised manner. At their core, Autoencoders consist of two main components: an encoder and a decoder. The encoder compresses the input data into a lower-dimensional representation, while the decoder attempts to reconstruct the original input from this compressed representation. The lower-dimensional representation, also known as the latent space or encoding, captures the essential information present in the data while discarding noise and redundant features. It can then be leveraged for various downstream tasks such as data visualization, clustering, or classification.

In **Paper III**, PCA and UMAP have been used for dimension reduction.

### 3.2.4 Deep Clustering

Traditional clustering algorithms which typically assume that data is represented as feature vectors, exhibit poor performance when faced with large and high-dimensional datasets. This is primarily due to the curse of dimensionality problem [10] and the associated high computational complexity. With the remarkable success of deep learning, particularly deep unsupervised learning, various representation learning techniques have emerged in the past decade. These techniques convert unstructured data, such as text and images, into a latent space that is typically lower-dimensional and contains richer information compared to conventional feature vectors.

To cope with the challenges posed by clustering high-dimensional data, researchers have explored the use of deep representations in clustering instead of traditional feature vectors. For instance, Guan et al. [42] employed pre-trained LSTM-based text encoders, and Subakti, Murfi, and Hariadi [150] utilized BERT for text encoding. Afterwards, they normalized these representations and applied traditional clustering algorithms to them. In a related research direction, researchers have recently begun directly clustering dimension-reduced embeddings created with pre-trained language models for topic modeling [40, 189, 31] instead of relying on complex statistical models such as LDA [13]. BERTopic [40] is an example of this approach, generating document embeddings with pre-trained Transformer-based language models, clustering these embeddings, and ultimately producing topic representations using the class-based TF-IDF procedure. However, utilizing deep representations for clustering in this manner is not always optimal. These deep representations have typically been trained on general domain data for generic tasks, so they are not inherently optimized for clustering tasks and often require adaptation. In fact, deep representation learning methods struggle to integrate potential clustering information to improve the quality of learned representations, primarily due to a lack of mutual enhancement between clustering and representation learning.

To overcome these challenges, the concept of *deep clustering* has arisen, with the goal of optimizing representation learning and clustering simultaneously. The deep clustering model consists of a representation learning module that takes in raw data and generates a low-dimensional representation, commonly referred to as an embedding. Furthermore, it includes a clustering module that takes these low-dimensional representations as input and produces either cluster labels for hard clustering or probabilities for cluster assignment in soft clustering. The parameters of both modules are trained simultaneously using certain objective functions including clustering loss.

Initially, numerous deep clustering algorithms were introduced in the computer vision domain for clustering image datasets [177, 20, 36, 54, 55]. However, their applicability to other data types, such as text data, was constrained by the image-specific techniques employed, such as CNN architectures [76] and data augmentations. The Autoencoder [48] is a general structure that can be

customized for different data types. Hence, in the early general-purpose deep clustering approaches, Autoencoders were used as the representation learning component for different data types such as text and image data.

Xie, Girshick, and Farhadi [169] introduced the Deep Embedded Clustering (DEC) method which simultaneously updates the data points’ representations, initialized from a pre-trained Autoencoder, and cluster centers initially computed through K-Means clustering. They introduced the self-training technique to the deep clustering task, initiating an active branch of methods referred to as *self-training deep clustering*. More specifically, the model’s parameters are optimized by minimizing the KL-divergence [70] between the soft cluster assignments and an auxiliary distribution derived from these assignments. The assignment distribution  $\mathbb{Q}$  for each data point  $x_i$  is determined by calculating the Student’s t-distribution [149] similarity between the data point representation  $h_i$  and the cluster centroids:

$$q_{ik} = \frac{(1 + \|h_i - \mu_k\|_2^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_j^K (1 + \|h_i - \mu_j\|_2^2/\alpha)^{-\frac{\alpha+1}{2}}}$$

where  $q_{ik}$  represents the probability of instance  $x_i$  belonging to cluster  $k$ ,  $K$  is the total number of clusters,  $\mu_k$  denotes the representation of cluster  $k$ , and  $\alpha$  signifies the degree of freedom of the Student’s t-distribution. The auxiliary distribution  $\mathbb{P}$  is a modified version of the assignment distribution  $\mathbb{Q}$  computed as follows:

$$p_{ik} = \frac{q_{ik}^2/f_k}{\sum_j^K q_{ij}^2/f_j}$$

where  $f_k = \sum_i^N q_{ik}$  are soft cluster frequencies. Raising  $q_i$  to the second power helps the model prioritize learning from instances with higher confidence, effectively reducing the influence of low-confidence instances during training. Moreover, normalizing by frequency per cluster regulates the contribution of clusters with varying sizes in the loss function, thereby mitigating the risk of degenerate solutions where all instances are assigned to a single cluster.

Accordingly, the objective function utilized for training the deep clustering model is computed as follows:

$$L = KL(\mathbb{P} \parallel \mathbb{Q}) = \sum_i \sum_k p_{ik} \log \frac{p_{ik}}{q_{ik}}$$

It is worth mentioning that DEC utilized TF-IDF vectors as text features for the input of the Autoencoder. DEC is highly significant in deep clustering and has been employed as a baseline in numerous studies including our research presented in **Paper III**.

Some studies have explored integrating various clustering losses or additional loss functions into the optimization process. IDEC was subsequently introduced by Guo et al. [43] as an enhancement to DEC, incorporating the Autoencoder’s reconstruction error into the objective function. Deep K-Means

[104] employs a general form of the K-Means objective function along with the Autoencoder reconstruction loss, as an alternative to the DEC loss function.

Moreover, some studies tried to customize DEC for text data. Instead of TF-IDF vecotrs used in DEC, Hadifar et al. [44] utilized Smooth Inverse Frequency (SIF) embeddings [5] which were considered more suitable representations for short text than TF-IDF. Motivated by the great success of PLMs in NLP, Huang et al. replaced the Autoencoder component of the DEC architecture with BERT [28], fine-tuning it simultaneously with masked language modeling loss and DEC clustering loss.

It is important to note that while Autoencoders succeed at dimensionality reduction, their primary focus may limit their ability to comprehensively capture the underlying data distribution within the latent space. Specifically, these representations are learned in an instance-wise manner, overlooking the interrelations among different instances. Consequently, the resulting embeddings may fail to effectively discriminate between instances in the embedding space, thereby leading to suboptimal clustering performance.

Inspired by the success of contrastive representation learning, contrastive learning has also been introduced into deep clustering. Contrastive learning has emerged as a highly popular unsupervised representation learning technique in recent years. Its fundamental principle involves bringing positive pairs of instances closer together while pushing negative pairs further apart, a concept often referred to as *instance discrimination*. At the core of contrastive learning lies the Normalized Cross-Entropy with Information Maximization (InfoNCE) loss [108] which for a set of  $N$  random samples is formulated as:

$$L_{InfoNCE} = -\log \sum_{i=1}^N \frac{\exp(f(h_i, h_i^\tau)/\tau)}{\sum_{j=1}^N \exp(f(h_i, h_j^\tau)/\tau)}$$

where  $h_i$  is the representation of anchor sample,  $h_i^\tau$  and  $h_j^\tau$  are the representations of the positive and negative samples respectively,  $f$  is a similarity function such as cosine similarity, and  $\tau$  is a temperature parameter that controls the smoothness of the probability distribution. Positive samples are typically generated through data augmentation, which may vary depending on the data type and the specific task at hand. The vision-language model CLIP [123] is one of the very successful applications of contrastive learning, where it is used to learn a joint representation of images and text.

Similar to contrastive representation learning, the objective of *contrastive deep clustering* is to pull positive pairs closer while pushing the negative pairs away. However, the distinction lies in how positive and negative pairs are defined. Similar to other deep clustering methods, contrastive deep clustering has its origins in the field of computer vision. SCAN [159], GCC [191], SwAV [19], MiCE [157], and LNSCC [92] have recently demonstrated state-of-the-art clustering performance on image datasets through contrastive learning techniques. SCAN utilizes the nearest neighbors in the k-Nearest Neighbors (kNN) graph, suggesting that a sample and its nearest neighbors should be grouped into the

same cluster. In contrast, GCC assumes that the transformation of an image and its neighbors’ transformation should exhibit similarity, thereby enhancing clustering performance on image data. Based on the insight that neighboring samples in a kNN graph might not consistently share the same category and that distinguishing between positive and negative pairs can be challenging in a naïve kNN setup, LNSCC proposes a *soft contrastive clustering* approach. This method assigns positivity and negativity scores to each pair of samples to capture their similarity and dissimilarity, thereby addressing ambiguity at cluster boundaries and yielding clearer distinctions between clusters.

Researchers in NLP have also explored the integration of contrastive learning into text deep clustering. SCCL [185] was among the first to utilize instance-level contrastive learning in text deep clustering. It employs Sentence-BERT [129] as the text encoder and combines the DEC [169] clustering objective with the contrastive InfoNCE loss [108] for optimizing the model parameters. However, while instance-level contrastive learning is successful at learning general feature representation, it overlooks semantic-level correlations within the same cluster, leading to suboptimal clustering outcomes and sparse cluster spaces. DACL [81] tackles this issue by smoothly shifting the loss weight of the model from contrastive learning to clustering throughout training and filtering negative samples in contrastive learning using pseudo-labels generated by clustering.

Pseudo-labeling, already discussed in Section 2.3.3 in the context of teacher-student models, has recently found its way into the domain of deep clustering. The proposed methods typically involve an iterative process where a clustering module and a classification module mutually enhance each other, resulting in notable performance gains. For instance, DeepCluster [18] employs an iterative approach wherein image features extracted by a convolutional neural network are clustered using a standard algorithm like K-Means. The resulting assignments, serving as hard pseudo-labels, are used for updating network’s weights. Pseudo labeling has extended the capabilities of semi-supervised learning to unsupervised clustering tasks. However, its effectiveness heavily depends on the quality of the pseudo-labels used for training the classifier, which are influenced by model capacity and hyperparameter tuning. While existing methods [107, 159] have addressed this challenge by incorporating pre-training as an initial step before pseudo-labeling, further attention is needed in this area.

The exploration of pseudo-labeling in deep clustering for text data has been limited. Rakib et al. [126] proposed an iterative method where a Multinomial Logistic Regression classifier is trained using cluster labels from non-outlier samples. This classifier is then employed to correct the clustering outcome by reclassifying outliers, with the resulting set of clusters serving as input for the next iteration. However, this approach relies on fixed TF-IDF representations for clustering, potentially limiting its generalizability.



## Chapter 4

# Event-based Topic Discovery in News Streams

This chapter provides the groundwork for **Papers IV** and **V**, which tackle research questions **RQ3** and **RQ4**, respectively. **Paper IV** introduces a novel model for story discovery in news streams. **Paper V** explores the relationship between news text and images and introduces a multimodal dataset for event-based topic discovery in multimodal news streams, along with a baseline model tailored for this task.

### 4.1 Topic Detection and Tracking

Topic Detection and Tracking (TDT) is a prevalent technique in the field of information retrieval (IR) and is pivotal for exploring, mining, and organizing news stories across various media sources. Introduced by Allan [1], TDT aims to identify and monitor real-world events within a multi-source news stream. In the context of TDT, a news story is a report on a particular event, and a topic is characterized by a collection of news stories discussing different aspects of the same event. When a *plane crashes in Malaysia*, it serves as the seminal event that initiates the topic. Any stories detailing the crash cause, death toll, rescue efforts, survivors, and so on are all considered part of the topic. Stories covering *a separate plane crash in a different country on the same day* or an *earthquake in Japan* would not typically fall under the same topic. However, in some instances in the literature, the terms news story and topic have been used interchangeably.

It is essential to distinguish an event topic from the conventional notion of topic found in information organization research. While the latter typically embodies the theme or subject of a text, such as “sports” or “politics” in news classification, event topics focus specifically on the triggering event of a story. Furthermore, topics in TDT evolve over time and may encompass stories that

are not necessarily related in subject matter.

TDT comprises five core tasks. *Story Segmentation* involves breaking down continuous news texts, such as transcriptions of news shows, into individual stories. However, this task becomes meaningless when processing streamed news from websites where stories are already separated. *First Story Detection* focuses on identifying stories that are not associated with previously recognized events, potentially signaling the beginning of a new event topic. *Cluster Detection* aims to allocate new stories within a streaming dataset to relevant topics in real-time, either by linking them to existing topic clusters or by creating new ones as needed. *Tracking* is closely linked to cluster detection and involves monitoring existing topics while continually seeking out additional stories to enrich them over time. *Story Link Detection* involves determining whether two given stories are related and belong to the same topic or not. Some of these tasks are closely interconnected, and their collective contributions enable the functionality of TDT. However, numerous studies in the literature often tackle these tasks within their proposed methods without explicitly specifying individual components for each task.

In the literature, TDT has been framed as a non-parametric topic modeling problem [192] which falls outside the scope of this thesis. Alternatively, TDT has been approached as a stream clustering problem. These works refer to this task as *online story discovery* or *news stream clustering*. It is noteworthy that in these studies, the terms *event topic* and *news story* are used interchangeably, deviating slightly from their definitions in the TDT task outlined by Allan [1].

Early attempts at news story discovery relied on sparse document representations such as keywords and TF-IDF vectors. Laban and Hearst [72] extracted article keywords and constructed a graph of articles spanning a window of  $N$  days such that articles sharing more keywords than a specified threshold were connected. Local topic clusters were then identified using the Louvain community detection algorithm [14]. The window was moved along the news stream, and if a topic continued to receive new articles across overlapping graphs, all of these articles were linked to the same topic. This process enabled the story to develop and grow over time. For longer-term stories, topics from non-overlapping windows were combined if their similarity exceeded a certain threshold. Staykovski et al. further improved this approach by utilizing TF-IDF vectors instead of keywords.

Miranda et al. [102] investigated a multilingual news stream. Their monolingual article representation comprised TF-IDF subvectors for words, word lemmas, and named entities extracted from different document sections: the title, the body, and a combination of both, totaling nine subvectors. Additionally, they developed a cross-lingual version of these vectors. Their methodology involves computing similarities between the monolingual TF-IDF subvectors of an article and those of monolingual clusters, which are the aggregated subvectors of their members. These similarities are then aggregated using a Rank-SVM model. The decision to merge the document with an existing cluster or create a new cluster is determined by another SVM classifier. Both SVM



models undergo training using a supervised training set. Furthermore, they incorporated article timestamps to prevent recent documents from merging with older clusters. In this setup, a crosslingual cluster comprises several monolingual clusters in different languages. During stream processing, after updating monolingual clusters, adjustments are made to crosslingual clusters accordingly.

With the emergence of dense document representations containing richer semantic information, researchers have begun exploring their potential in news story discovery. Staykovski et al. [147] conducted a comparison between TF-IDF and doc2vec representations for this purpose and concluded that sparse representations are more effective. In a more recent study, Saravanakumar et al. [137] adopted a methodology similar to that of Miranda et al. [102] for news story discovery but they used a combination of sparse and dense representations for articles. They demonstrated that incorporating contextual BERT representations alongside TF-IDF representations could enhance performance in this task. This enhancement was achieved through fine-tuning BERT on event similarity using a triplet network architecture [53] and incorporating external entity knowledge.

The weaker performance of dense representations like BERT (without being fine-tuned) in comparison to sparse representations, in news story discovery may be attributed to the low uniformity of their embedding space. Alignment and uniformity, as discussed in [160], are fundamental attributes of any embedding space. For the task of news story discovery, alignment refers to how closely articles related to the same story are positioned within the embedding space, while uniformity is a measure of the uniform distribution of random articles throughout that space. Lack of uniformity poses a challenge in distinguishing between two articles that share a common theme but pertain to different events.

In recent years, contrastive learning has proven highly effective across various language processing and computer vision tasks. This effectiveness primarily arises from its capacity to improve the alignment and uniformity of embedding spaces, as demonstrated by Wang and Isola [160]. A notable example of this success in news story discovery is shown in the study by Yoon et al. In this work, with the idea that not all sentences in the article have the same significance for its story, a story-indicative article representation is made by aggregating the sentence representations, derived from a pre-trained Sentence-Transformer, via a single transformer layer. Subsequently, these representations are compared with existing cluster representations within the current window to either identify the best match or create a new cluster. Once clusters are defined, the representations are further refined to adapt to the recent context through cluster-level contrastive learning. This research demonstrated that these dense representations outperform sparse alternatives.

Despite numerous efforts to discover effective article representations that facilitate the seamless identification of news stories and differentiate between various stories, this remains an active research domain.

## 4.2 Online Clustering

Another aspect worth exploring in previous studies on news story discovery is the clustering methodology employed. Given that TDT is an online task and articles in the news stream require real-time clustering, opting for an online clustering algorithm is a natural choice. However, there is variation among previous works regarding their approach to online data processing. Many of them favor a non-parametric version of online K-Means clustering.

The online K-Means algorithm is a modification of the traditional K-Means algorithm, allowing for continuous learning and cluster updating as new data points emerge over time. In the non-parametric version, the number of clusters is not fixed and can expand indefinitely. This characteristic aligns well with the demands of story discovery, as the clustering problem is inherently non-parametric, and each document in the stream could potentially initiate a new event cluster.

Here is a breakdown of how the non-parametric online K-Means operates:

- **Data Streaming:** instead of processing the entire dataset simultaneously, the online K-Means algorithm handles data points one at a time as they are received.
- **Assignment:** upon receiving a data point, the algorithm assigns it to the nearest centroid based on a chosen distance or similarity metric, such as Euclidean distance or cosine similarity. Alternatively, if the data point is not sufficiently close to any existing cluster, the algorithm creates a new cluster for it. In the literature, this decision has been made through both supervised and unsupervised approaches. For instance, Miranda et al. [102] and Saravanakumar et al. [137] utilized trained classifiers to determine when a new cluster should be created, using labeled training datasets. Alternatively, a similarity threshold can be employed, with optimal values determined via grid search if supervised data is available. However, obtaining supervised data is not always feasible due to the high cost associated with acquiring human annotations and the challenge of keeping it up-to-date. Therefore, an unsupervised approach is often more practical and suitable for evolving news article streams.
- **Centroid Update:** following the assignment of a data point to a cluster, the algorithm adjusts the centroid of that cluster to incorporate the new data point and adapt to the evolving data distribution.

Algorithm 1 shows the pseudocode of this clustering algorithm. As previously emphasized, time plays a crucial role in news stories. Some studies employing this online clustering algorithm [102, 137], incorporated timestamp features for each cluster. During document-cluster comparisons, such methods assess not only textual representations but also timestamp features, making decisions based on a combination of these comparisons. A significant disparity between the publishing time of an article and the timestamp features of a

---

**Algorithm 1:** The non parametrik online K-Menas clustering algorithm for data stream clustering

---

**Data:**  $\mathbb{D}$ : a news article stream

$f$ : document representation generation function

$\theta$ : article-story similarity threshold

**Result:** A set  $\mathbb{S}$  of stories in stream  $\mathbb{D}$

```

1  $\mathbb{S} \leftarrow \emptyset$ 
2 for every new article  $d \in \mathbb{D}$  do
3    $R_d \leftarrow f(d)$ 
4   if  $\max(\{sim_{d,s_j} | s_j \in \mathbb{S}\}) > \theta$  then
5     Assign article  $d$  to corresponding  $s_j$ 
6     Update  $R_{s_j}$  with  $R_d$ 
7   else
8      $s_{|\mathbb{S}|+1} \leftarrow \{d\}$ 
9      $R_{s_{|\mathbb{S}|+1}} \leftarrow R_d$ 
10     $\mathbb{S} \leftarrow \mathbb{S} \cup \{s_{|\mathbb{S}|+1}\}$ 
11  end
12 end
13 return  $\mathbb{S}$ 

```

---

cluster serves as a valuable indicator that the article may not belong to that cluster, even if the textual features exhibit a reasonable match between the incoming article and the cluster representation. Yoon et al. [182] adopted a different approach, employing a sliding window mechanism along the stream. In this approach, documents are compared only with the active clusters within the time frame of the sliding window, eliminating the need to explicitly consider temporal features for the clusters. Moreover, incorporating the window to the online algorithm enhances the algorithm’s efficiency and speed for large-scale datasets with numerous topics. In **Paper IV** and **Paper V**, we also utilize this approach for online clustering of news articles within the news stream.

Additionally, some studies in the literature employed a two-step clustering approach. They analyze a collection of articles gathered over a specific time frame, such as  $N$  consecutive days, to form local clusters. Subsequently, these local clusters are linked over time to track the progression of stories. For instance, Laban and Hearst [72] and Staykovski et al. [147] construct a graph of articles spanning a window of  $N$  days based on the similarity of article representations. They then apply a Louvain community detection algorithm [14] to identify local topics within the current window. As the window moves along the news stream, if a topic consistently receives new articles across overlapping windows, all these articles are associated with the same topic. This mechanism facilitates the gradual development and expansion of the story over time. For longer-term stories, topics from non-overlapping windows are compared based on their keyword distributions, and if their similarity surpasses a certain threshold, they are merged. Linger and Hajaiej [84] introduced a similarity-

based *replaying* strategy to connect local topics into cohesive stories. For a new batch of articles at time  $t$ , they calculate similarities between all new articles and all topics from the previous time  $t - 1$ . If a topic from  $t - 1$  exhibits a similarity with a new article at time  $t$  surpassing a predefined threshold, all articles associated with that topic are included in the current batch. This allows them to be considered during the subsequent round of topic detection at time  $t$ . These two-step algorithms do not utilize explicit time features; rather, time is implicitly incorporated through the batch/window procedure.

Unlike algorithms that process articles one by one, batch processing algorithms are better suited for handling large-scale streams where scalability is a key concern. Additionally, they facilitate the emergence of various topic behaviors such as splitting and merging over time. However, detecting such topic behaviors across batches/windows and tracking stories may pose challenges and add complexity to the stream clustering model.

### 4.3 Multimodal News Streams

News websites have evolved to incorporate a diverse array of presentation modalities, such as text, images, diagrams, and videos, strategically designed to engage readers and convey messages effectively. Each modality offers unique advantages and constraints, and their integration can enrich the user experience, making it more immersive and engaging. In this section of the thesis, *multimodal* specifically refers to the combination of images and text, while other modalities are not considered within the scope of this study.

Analyzing multimodal news poses significant challenges due to the varied interrelationships between information from different modalities. News article texts typically contain an abundance of details ranging from timing and content to location and individuals involved in reported events. In contrast, the role of accompanying images in news articles is diverse. Images may serve as decorative elements, provide supplementary information, or, at times, present potential sources of misinformation. For instance, imagine a news article highlighting a specific action by Trump, accompanied by an image solely featuring Trump himself.

Studying the relationship between text and images, especially in the context of news analysis, is an interdisciplinary research question that has garnered significant attention from various fields, including communication science, media studies, journalism, machine learning, and multimodal analysis. Several taxonomies of image-text relations, sometimes specifically for analyzing news articles, have been proposed in media studies [8, 17] and semiotics [7, 94, 95]. Among these, Barthes' work [7] stands out as pioneering. He categorizes text-image relations into three main types: (1) Anchorage, where text describes the image; (2) Illustration, where the image visually represents information from the text; and (3) Relay, where text and image share an equal relationship, such as complementarity or interdependence.

However, there is limited work in computational approaches for multimodal news analysis that attempts to model and utilize the relationship between images and text. Müller-Budack et al. [105] introduced an unsupervised approach that measures the cross-modal consistency of entity relations between image and text modalities in news articles. Oostdijk et al. [109] investigated the relationship between text and images in news articles for flooding-event detection. They identified four cross-modal relations: images visualizing what the text describes, images visualizing people referred to in the text, images visualizing a situation as it existed before while the text describes or suggests how a similar situation might arise (flood threat), and images visualizing a situation as it exists now but which will be affected by developments described in the text (e.g., an image of an elephant in an area that will be flooded once a dam is ready). Nonetheless, the scope of their investigation is restricted, and its generalizability might be limited. In a recent study, Cheema et al. [22] introduced a framework for the computational analysis of multimodal news. Drawing from real examples of news reports, they outlined a set of image-text relationships and multimodal news values, exploring their implementation through computational methods. Yet, there has been no research exploring the relationships between images and text for story discovery in news streams, so the extent to which multimodal information aids in the online story discovery task remains an open question.

Prior research in multimodal news analysis has primarily concentrated on two main areas: thematic classification of news [161, 65, 118] and fake news detection [168, 195, 194, 179]. The only instance of multimodal work in topic detection and tracking, to our knowledge, is by Li et al. [82], who specifically explored topic detection and tracking within video news.

### 4.3.1 Deep Learning for Multimodal Data

Advancements in multimodal deep learning have empowered vision-language models such as CLIP [123], BLIP2 [80], and LLaVA [87] to comprehend fundamental relationships between modalities, such as correlations between words and phrases and their visual representations. While these developments have fueled significant progress in tasks like image captioning, text-to-image generation, and visual question answering, they are inadequate for generating multimodal representations for complex objects like multimodal news articles, thereby restricting their capacity to interpret the overall multimodal message.

Many existing studies in multimodal news analysis utilize diverse fusion models to integrate image and text, creating a multimodal representation [161, 65, 118, 168, 194, 100]. Typically, these studies employ modality-specific encoders to generate embeddings for each modality. These embeddings are then projected into a shared space to enable comparison between different modalities before being fused for the downstream task.

Various fusion approaches, including early fusion and late fusion, have been proposed to leverage heterogeneous data and modalities. Early fusion, also

known as feature-level fusion, aggregates all features, including textual and visual features, into a single feature vector which serves as the multimodal representation. This can be achieved through concatenation [161, 65], or employing attention mechanisms [118, 168, 194, 100]. The resulting representation is then used for downstream tasks. In late fusion, modalities are merged at the decision level. In classification tasks, this usually entails combining the posterior probabilities derived from classifiers for each class. Rather than directly predicting labels, these classifiers produce probabilities for various classes [83].

It is worth mentioning that CLIP [123] has been utilized in literature to measure cross-modal similarity [195, 194]. CLIP, a multimodal model trained on diverse image-text pairs, is capable of predicting relevant text snippets for given images and vice versa. This integration enables CLIP to embed texts and images into a unified latent space, facilitating the calculation of cross-modal correlations. Consequently, the cosine similarity between CLIP representations of text and image modalities indicates the extent to which the article text and image are aligned, serving as a criterion to adjust the contribution of the image modality in the overall multimodal representation of the article. This technique has been used in **Paper V**.

## Chapter 5

# Summary of Contributions

This thesis aims to answer five pivotal research questions:

**RQ1** How can news topics be automatically identified across various granularity levels?

**RQ2** What effective methodologies can be employed to integrate deep learning into the investigation of news topics when labeled data is scarce or unavailable?

**RQ3** How can deep learning techniques be utilized for topic identification in news streams while effectively addressing challenges associated with changes in topic focus and evolution over time?

**RQ4** What is the interrelation between different modalities within multimodal news, and how can these modalities be harnessed for the purpose of topic identification?

In response to **RQ1**, the thesis proposes classification and clustering for coarse-grained topic identification and event-topic discovery in news streams for fine-grained topic identification. Additionally, it addresses **RQ2** by proposing semi-supervised deep classification and deep clustering approaches for topic identification in cases where supervision is limited or absent, respectively. **Paper I**, **Paper II**, and **Paper III** focus on addressing research questions **RQ1** and **RQ2** for coarse-grained topics. **Paper I** and **Paper II** propose the development of semi-supervised classification models using deep learning for news topic identification in cases where pre-defined coarse-grained topics are of interest and there is insufficient labeled data available to effectively train a classifier in a fully supervised manner. Additionally, **Paper III** proposes deep clustering in scenarios where a set of predefined classes is absent, yet there is a desire to explore coarse-grained topics within the news dataset.

**Paper IV** and **Paper V** address research questions **RQ3** and **RQ4**, respectively. **Paper IV** introduces a novel model for story discovery in news

streams, while **Paper V** initiates the study of using both news text and images. In the discovery of event-based topics, in particular, it introduces a multimodal dataset for event-based topic discovery in multimodal news streams, along with a baseline model tailored for this task.

## 5.1 Paper I

**Arezoo Hatefi**, Xuan-Son Vu, Monowar Bhuyan, and Frank Drewes. Cformer: Semi-Supervised Text Clustering Based on Pseudo Labeling. *In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM)*, pp. 3078-3082, 2021.

### Paper Contributions

This paper is motivated by a scenario in contextual advertising where the number of classes is known, but there are only a few labeled examples available for each class, while the majority of the dataset remains unlabeled. To tackle this issue, the paper introduces Cformer, a semi-supervised approach that utilizes the teacher-student architecture employed in pseudo-labeling.

Cformer adapts the MPL method proposed by Pham et al. [115] from the computer vision field for semi-supervised text classification, incorporating necessary modifications to suit text data. This architecture aims to mitigate the confirmation bias inherent in pseudo-labeling methods by iteratively training the teacher and student models. Feedback from the student to the teacher in each iteration informs the teacher about the quality of the generated pseudo-labels, facilitating self-improvement. Consequently, the teacher is trained using a supervised loss computed for the labeled dataset, a consistency loss calculated based on the unlabeled dataset and an augmented version of it, and feedback from the student, represented by the loss value of the student for the labeled dataset. Additionally, the student undergoes supervised training using the pseudo-labels generated by the teacher for the unlabeled data.

In Cformer, the teacher and student share the same architecture, which consists of a BERT encoder followed by an MLP for performing the classification task. Furthermore, the paper proposes a version of Cformer called Distill-Cformer, in which a DistilBERT model is used as the text encoder in the student. After being trained, this student is better suited for resource-limited environments.

The experiments demonstrated that Cformer could surpass state-of-the-art semi-supervised text classification methods when a reasonable amount of labeled data for each class is available. Additionally, despite its smaller size, Distill-Cformer exhibited performance on par with Cformer.



## Author Contributions

As the main author, I contributed to formulating the problem, implementing the code and experiments, analyzing the results, and leading the writing of the first draft. Xuan-Son Vu offered valuable guidance and support throughout the process, especially in formulating the problem, analyzing the results, and incorporating them into the first draft. Frank Drewes fulfilled advisory roles, engaging in discussions concerning problem formulation, experiments, and result presentations. Additionally, he made significant contributions to the writing of the first draft by writing the introduction section and reviewing and providing feedback on other sections. Monowar Bhuyan engaged in discussions and provided feedback on the draft.

## 5.2 Paper II

**Arezoo Hatefi**, Xuan-Son Vu, Monowar Bhuyan, and Frank Drewes. The Efficiency of Pre-training with Objective Masking in Pseudo Labeling for Semi-Supervised Text Classification. *Submitted to the Northern European Journal of Language Technology (NEJLT), 2023.*

### Paper Contributions

This paper proposes CformerM, an extension of the Cformer introduced in Paper I. CformerM incorporates an unsupervised pre-training phase, further training the text encoders of the teacher and student models on the unlabeled data using objective masking. Objective masking prioritizes masking topic words from a topic word list, supplemented by random word masking if necessary, to mask a total of 15% of the words of the text. This masking objective aims to enhance the text encoder ability to grasp the underlying topics in the dataset and recognize its topical information.

To create the topic word list, the dataset undergoes LDA [13] topic modeling with an appropriate number of topics. The number of topics is selected based on the coherence scores of various topic models with differing numbers of topics. Then, the  $N$  most relevant words for each topic are extracted using the relevance measure introduced by Sievert and Shirley [142] and compiled into a list. This measure includes a parameter  $\lambda$  that allows for the selection of the specificity of the topic words. When  $\lambda$  is small, the method prioritizes words strongly associated with the topic but less common in other topics, resulting in distinct topics but potentially neglecting relevant words shared across topics. Conversely, with a higher  $\lambda$ , the approach concentrates on words prevalent within the topic and also across other topics, capturing more general aspects of the topics. Additionally, the optimal value for  $N$  is determined through assessing the coherence of different lists with varying values for  $N$ .

In extensive experiments conducted on datasets in English and Swedish, CformerM was compared with numerous baselines, including Cformer, various

state-of-the-art semi-supervised classifiers, and a variant of CformerM achieved by employing random masking instead of objective masking. The experimental results indicated that CformerM outperforms Cformer and other baselines in most cases across all datasets. However, the influence of objective masking on classification accuracy is more notable when the amount of supervised data for classification is limited.

In comparing CformerM and its variant created with random masking, it was demonstrated that when the dataset significantly deviates from the BERT training data and includes domain-specific information, such as medical documents, the difference in the impact of objective masking and random masking on the classification performance becomes more noticeable.

Moreover, a comparison was made between the proposed LDA-based method for generating topic word lists and a simpler technique that uses TF-IDF to identify topic words within the corpus. Specifically, words are sorted based on their average TF-IDF scores across all documents, and the top words are selected. It was found that creating the topic word list based on TF-IDF instead of LDA is less effective, particularly when the labeled data is severely limited. It was hypothesized that this superiority of CformerM could be attributed to the fact that the topic model considers the underlying structure of the dataset, whereas TF-IDF relies on individual documents. Additionally, the LDA-based method offers flexibility in choosing between highly topic-specific words and more general ones, addressing the specific needs of the analysis, while the TF-IDF method offers less control over the generated lists.

Last but not least, a qualitative analysis conducted indicated that pre-training with objective masking enhances the reliability and interpretability of the model, resulting in more accurate classification results. Additionally, experiments conducted in a zero-shot setting demonstrated that the proposed pre-training of the language model with objective masking could enhance the language model's ability to recognize examples of classes that had not been seen before.

## Author Contributions

As the main author, I contributed significantly to formulating the problem, implementing the code and experiments, analyzing the results, and leading the writing of the first draft. Xuan-Son Vu provided valuable guidance and support throughout the process, particularly in formulating the problem, analyzing the results and incorporating them into the first draft, and enhancing the illustrations. Frank Drewes fulfilled advisory roles, engaging in discussions concerning problem formulation, experiments, and result presentations. Additionally, he made significant contributions to the writing of the first draft by writing the introduction section and reviewing and providing feedback on other sections. Monowar Bhuyan engaged in discussions and provided feedback on the draft.

## 5.3 Paper III

**Arezoo Hatefi**, Xuan-Son Vu, Monowar Bhuyan, and Frank Drewes. AD-Cluster: Adaptive Deep Clustering for Unsupervised Learning from Unlabeled Documents. *In Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP)*, pp. 68-77, Association for Computational Linguistics, 2023.

### Paper Contributions

Paper III introduces ADCluster, a deep clustering approach based on pseudo-labeling, for document clustering. ADCluster comprises a clustering and a classification component that iteratively promoted each other, leading to significant performance improvements. During each iteration, K-Means clusters the document representations generated via the language model and predicts pseudo-labels. Subsequently, these labels are utilized to train the classifier, consisting of the language model encoder followed by an MLP for classification, in a supervised manner. This iterative adaptation, referred to as *inner adaptation*, allows the PLM to adjust to the clustering task and generate more clustering-friendly representations, thereby enhancing K-Means clustering in subsequent epochs.

The paper also explores the adaptation power of ADCluster over time to growing sets of documents, a process referred to as *outer adaptation*. Outer adaptation resumes the inner adaptation when a significant amount of new data becomes available, either by considering the entire dataset (accumulative outer adaptation) or using only the new data (non-accumulative outer adaptation). In this dynamic setting, the assumption is made that the number of clusters over time remains constant, with new samples being received. In this scenario, distribution shift only occurs within clusters. This setup is motivated by a scenario in which there is a steady stream of content, such as news articles, centering around a fixed set of topics, albeit with changing focus over time. For instance, during major sports events like the FIFA World Cup, sports news primarily revolves around this event, even though this may not have been the case previously.

Extensive experiments conducted on various short and long text datasets demonstrated ADCluster’s superiority over established document clustering techniques, particularly on medium and long-text documents, by a significant margin. Furthermore, the proposed approach surpassed well-established baseline methods in both accumulative and non-accumulative outer adaptation scenarios.

### Author Contributions

As the main author, I contributed to formulating the problem, implementing the code and experiments, analyzing the results, and leading the writing of

the first draft. Xuan-Son Vu provided valuable guidance and support throughout the process, particularly in formulating the problem and improving the first draft particularly the algorithm and illustrations. Frank Drewes fulfilled advisory roles, engaging in discussions concerning problem formulation, experiments, and result presentations. Additionally, he made significant contributions to the writing parts of the first draft and reviewing and providing feedback on the other parts. Monowar Bhuyan engaged in discussions and provided feedback on the draft.

## 5.4 Paper IV

**Arezoo Hatefi**, Anton Eklund, and Mona Forsman. PromptStream: Self-Supervised News Story Discovery Using Topic-Aware Article Representations. *Accepted to Appear in the Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, 2024.

### Paper Contributions

The paper introduces a methodology for discovering news stories within a news stream known as PromptStream. PromptStream utilizes an online clustering approach to assign articles in the stream to their relevant stories. This involves employing a sliding window that traverses the stream, serving as a representation of the time frame of interest. As the window progresses, new news articles are sequentially clustered into news stories based on their temporal order. To assign a news article to a suitable cluster, PromptStream compares it only with the existing clusters within the time frame of the sliding window. If the article’s resemblance to a story exceeds the similarity threshold, it is clustered into that cluster; otherwise, a new story is initiated with this news article.

PromptStream generates topic-aware document representations by combining a prompt-based representation with the output of a mean pooling layer applied to the last layer of the PLM. The prompt-based representation is constructed using a cloze-style template that prompts the model about the topic of the given text:

`[ topic : <mask> ] <title> <body>`

where `<title>` and `<body>` represents the title and body of the news article, respectively. This representation extracts topic-specific information from the text by prioritizing attention to topic-related tokens and entities. Conversely, mean pooling provides a broader representation of the entire document. By integrating these two representations, the approach effectively leverages both the detailed, contextually rich information acquired from cloze-based prompting and the global context captured through mean pooling.

Moreover, the text encoder remains consistently updated to reflect the latest context within the news stream through continual learning techniques. A memory is maintained, filled with the most confident clustering results based on the resemblance of the articles to the stories they are clustered into for a certain duration (e.g., 10 days). At the end of this period, these samples are replayed to update the encoder using cluster-level contrastive learning. This process encourages articles to move closer to the center of their respective clusters while simultaneously being pushed away from other cluster centers, resulting in enhanced uniformity and alignment of the embeddings. Given a batch  $\mathbb{B}$  of positive article-story pairs  $(d, s) \in \mathbb{B}$  the cluster-level contrastive loss function is computed as follows:

$$L_{cts} = - \sum_{(d,s) \in \mathbb{B}} \log \frac{\exp(\cos(R_d, R_s)/\tau)}{\sum_{s' \in \mathbb{S}_W} \exp(\cos(R_d, R_{s'})/\tau)}$$

where  $\tau$  is a temperature parameter and  $\mathbb{S}_W$  is the set of existing stories in window  $W$ . Through extensive experiments, PromptStream was compared with state-of-the-art methods, demonstrating its superior performance across three news stream datasets.

## Author Contributions

As the main author, I played a vital role in formulating the problem, implementing the code and experiments, analyzing the results, and leading the writing of the first draft. Anton Eklund engaged in discussions concerning problem formulation and experiment designs. Additionally, he conducted a qualitative analysis on the method’s results, presenting them in the “Qualitative Analysis” section of the paper. Moreover, he made significant contributions to improving the first draft and enhancing the illustrations. Mona Forsman fulfilled advisory roles, engaging in discussions concerning problem formulation, experiments, and result presentations. Additionally, she reviewed and commented on the draft.

## 5.5 Paper V

**Arezo Hafei**, Johanna Björklund, Xuan-Son Vu, and Frank Drewes. *METHOD: A Dataset and Baseline for Multimodal Discovery of Event-Based News Topics. Submitted to the International Journal of Multimedia Information Retrieval, 2024.*

## Paper Contributions

Given that online news reporting typically integrates various modalities such as text, images, video, audio, and other data types to convey information, this paper proposes event-based topic discovery in a stream of multimodal news

articles as a significant and challenging problem within the broader field of topic discovery. To address the lack of an appropriate dataset for this task, the authors annotated a dataset of image-text news articles from the New York Times, named METOD, enabling researchers to develop and evaluate methods for this task.

Event-based topics typically have a limited lifespan. For instance, in the case of a sudden event like an earthquake, the initial articles usually appear shortly after the event, and coverage gradually diminishes over time. Considering the temporal aspect of event-based topics, this paper defines some characteristics for such topics that could be used for evaluating the performance of event-based topic discovery algorithms. These characteristics include topic size, topic duration, article frequency, temporal irregularity, disconnectedness index, suddenness, specificity, and image informativeness. Except for the last characteristic, others are relevant for only-text news streams as well. Additionally, the values of these characteristics are computed for the topics in the METOD dataset to the extent possible.

Moreover, the Multimodal EventTracker, a baseline model for event-based topic discovery in multimodal news streams, is introduced and its performance on the METOD dataset is analyzed. Multimodal EventTracker bears similarity to PromptStream introduced in **Paper IV** from the online clustering aspect. However, its encoder differs in that it is tailored to produce a robust representation for text-image data. Additionally, the encoder remains fixed and does not undergo continual learning. To generate the representation for the image-text data, both text and image are initially encoded using text-specific and image-specific encoders. Subsequently, they are combined with the similarity of the text and image representations generated with CLIP [122] serving as the weight of the image representation.

## Author Contributions

I, Johanna Björklund, and Frank Drewes contributed equally to the problem formulation, conceptualization, paper writing, and dataset development. In addition, I developed and implemented the baseline model, and designed and conducted the experiments. Moreover, Xuan-Son Vu engaged in the discussions regarding problem formulation, and design and implementation of the baseline model. He also made the first version of Figure 1, wrote the “Datasets in news clustering” part of the “Related Work” section, and reviewed and commented on the other parts of the manuscript.

# Bibliography

- [1] James Allan. “Introduction to Topic Detection and Tracking”. In: *Topic Detection and Tracking: Event-based Information Organization*. 2002, pp. 1–16.
- [2] Massih-Reza Amini et al. “Self-training: A survey”. In: *arXiv preprint arXiv:2202.12040* (2022).
- [3] Mingxiao An et al. “Neural News Recommendation with Long- and Short-term User Representations”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 336–345.
- [4] Eric Arazo et al. “Pseudo-labeling and confirmation bias in deep semi-supervised learning”. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020, pp. 1–8.
- [5] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. “A simple but tough-to-beat baseline for sentence embeddings”. In: *International conference on learning representations*. 2017.
- [6] Alexandra Balahur et al. “Sentiment Analysis in the News”. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10)*. European Language Resources Association (ELRA), 2010.
- [7] R. Barthes and S. Heath. *Image, Music, Text*. Fontana Press, 1977.
- [8] Monika Bednarek. “Investigating evaluation and news values in news items that are shared through social media”. In: *Corpora* 11.2 (2016), pp. 227–257.
- [9] Mikhail Belkin and Partha Niyogi. “Laplacian eigenmaps and spectral techniques for embedding and clustering”. In: *Advances in neural information processing systems* 14 (2001).
- [10] Richard Bellman. “Dynamic programming”. In: *Science* 153.3731 (1966), pp. 34–37.

- [11] Prithwiraj Bhattacharjee et al. “Bengali Abstractive News Summarization (BANS): A Neural Attention Approach”. In: *Proceedings of International Conference on Trends in Computational and Cognitive Engineering*. Springer Singapore, 2021, pp. 41–51.
- [12] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [13] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. “Latent dirichlet allocation”. In: *Journal of Machine Learning Research* 3 (2003), pp. 993–1022.
- [14] Vincent D Blondel et al. “Fast unfolding of communities in large networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (2008), P10008.
- [15] Piotr Bojanowski et al. “Enriching word vectors with subword information”. In: *Transactions of the association for computational linguistics* 5 (2017), pp. 135–146.
- [16] Tom Brown et al. “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.
- [17] Helen Caple, Changpeng Huan, and Monika Bednarek. *Multimodal News Analysis across Cultures*. Elements in Corpus Linguistics. Cambridge University Press, 2020.
- [18] Mathilde Caron et al. “Deep clustering for unsupervised learning of visual features”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 132–149.
- [19] Mathilde Caron et al. “Unsupervised learning of visual features by contrasting cluster assignments”. In: *Advances in neural information processing systems* 33 (2020), pp. 9912–9924.
- [20] Jianlong Chang et al. “Deep adaptive image clustering”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5879–5887.
- [21] O. Chapelle, B. Schölkopf, and A. Zien, eds. *Semi-Supervised Learning*. MIT Press, 2006.
- [22] Gullal S. Cheema et al. “Understanding image-text relations and news values for multimodal news analysis”. In: *Frontiers in Artificial Intelligence* 6 (2023).
- [23] Jiaao Chen, Zichao Yang, and Diyi Yang. “MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020, pp. 2147–2157.



- [24] Kyunghyun Cho et al. “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2014, pp. 1724–1734.
- [25] Alexis Conneau et al. “Unsupervised Cross-lingual Representation Learning at Scale”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, July 2020, pp. 8440–8451.
- [26] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines: And other kernel-based learning methods*. Cambridge University Press, 2000.
- [27] Justina Deveikyte et al. “A sentiment analysis approach to the prediction of market volatility”. In: *Frontiers in Artificial Intelligence* 5 (2022), p. 836809.
- [28] Jacob Devlin et al. “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, 2019, pp. 4171–4186.
- [29] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. “Kernel k-means: spectral clustering and normalized cuts”. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004, pp. 551–556.
- [30] Yingtong Dou et al. “Enhancing graph neural network-based fraud detectors against camouflaged fraudsters”. In: *Proceedings of the 29th ACM international conference on information & knowledge management*. 2020, pp. 315–324.
- [31] Anton Eklund and Mona Forsman. “Topic Modeling by Clustering Language Model Embeddings: Human Validation on an Industry Dataset”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Abu Dhabi, UAE: Association for Computational Linguistics, 2022, pp. 635–643.
- [32] Martin Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996, pp. 226–231.
- [33] Feifan Fan, Yansong Feng, and Dongyan Zhao. “Multi-grained Attention Network for Aspect-Level Sentiment Classification”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018, pp. 3433–3442.

- [34] Wentao Fan et al. “Clustering-Based Online News Topic Detection and Tracking Through Hierarchical Bayesian Nonparametric Models”. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2021, pp. 2126–2130.
- [35] Kang Fu et al. “Credit Card Fraud Detection Using Convolutional Neural Networks”. In: *Neural Information Processing*. Springer International Publishing, 2016, pp. 483–490.
- [36] Kamran Ghasedi Dizaji et al. “Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5736–5745.
- [37] Golnaz Ghiasi et al. “Multi-Task Self-Training for Learning General Representations”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 8836–8845.
- [38] Pushpendu Ghosh, Ariel Neufeld, and Jajati Keshari Sahoo. “Forecasting directional movements of stock prices for intraday trading using LSTM and random forests”. In: *Finance Research Letters* 46 (2022), p. 102280.
- [39] Ian Goodfellow et al. “Generative adversarial networks”. In: *Communications of the ACM* 63.11 (2020), pp. 139–144.
- [40] Maarten Grootendorst. “BERTopic: Neural topic modeling with a class-based TF-IDF procedure”. In: *arXiv preprint arXiv:2203.05794* (2022).
- [41] Yuxian Gu et al. “Train No Evil: Selective Masking for Task-Guided Pre-Training”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 6966–6974.
- [42] Renchu Guan et al. “Deep Feature-Based Text Clustering and its Explanation”. In: *IEEE Transactions on Knowledge and Data Engineering* 34.8 (2022), pp. 3669–3680.
- [43] Xifeng Guo et al. “Improved Deep Embedded Clustering with Local Structure Preservation”. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. 2017, pp. 1753–1759.
- [44] Amir Hadifar et al. “A Self-Training Approach for Short Text Clustering”. In: *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 194–199.
- [45] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.
- [46] Arezoo Hatefi Ghahfarrokhi and Mehrnoush Shamsfard. “Tehran stock exchange prediction using sentiment analysis of online textual opinions”. In: *Intelligent Systems in Accounting, Finance and Management* 27.1 (2020), pp. 22–37.

- [47] S. Haykin. *Neural networks: A comprehensive foundation*. Prentice Hall, 1999.
- [48] G. E. Hinton and R. R. Salakhutdinov. “Reducing the Dimensionality of Data with Neural Networks”. In: *Science* 313.5786 (2006), pp. 504–507.
- [49] G. E. Hinton and R. R. Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *Science*. Vol. 313. 5786. American Association for the Advancement of Science, 2006, pp. 504–507.
- [50] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. “Distilling the knowledge in a neural network”. In: *arXiv preprint arXiv:1503.02531* (2015).
- [51] Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. “Aspect-Based Sentiment Analysis using BERT”. In: *Proceedings of the 22nd Nordic Conference on Computational Linguistics*. Linköping University Electronic Press, 2019, pp. 187–196.
- [52] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.
- [53] Elad Hoffer and Nir Ailon. “Deep metric learning using triplet network”. In: *Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*. Springer, 2015, pp. 84–92.
- [54] Chih-Chung Hsu and Chia-Wen Lin. “CNN-Based Joint Clustering and Representation Learning with Feature Drift Compensation for Large-Scale Image Data”. In: *IEEE Transactions on Multimedia* 20.2 (2018), pp. 421–429.
- [55] Weihua Hu et al. “Learning discrete representations via information maximizing self-augmented training”. In: *International conference on machine learning*. PMLR, 2017, pp. 1558–1567.
- [56] Yunfan Hu, Zhaopeng Qiu, and Xian Wu. “Denoising Neural Network for News Recommendation with Positive and Negative Implicit Feedback”. In: *Findings of the Association for Computational Linguistics: NAACL 2022*. Association for Computational Linguistics, 2022, pp. 2320–2329.
- [57] Shaohan Huang et al. “Unsupervised Fine-tuning for Text Clustering”. In: *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Dec. 2020, pp. 5530–5534.
- [58] Ipsos. *Media and News Survey 2023*. Available at <https://europa.eu/eurobarometer/surveys/detail/3153>. EB-ID: FL012EP | Fieldwork: 18/10/2023 – 24/10/2023 | Conducted by Ipsos European Public Affairs. 2023.
- [59] Hang Jiang et al. “Topic detection and tracking with time-aware document embeddings”. In: *arXiv preprint arXiv:2112.06166* (2021).

- [60] Bernal Jimenez Gutierrez et al. “Document Classification for COVID-19 Literature”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, 2020, pp. 3715–3722.
- [61] Yiping Jin, Vishakha Kadam, and Dittaya Wanvarie. “Bootstrapping Large-Scale Fine-Grained Contextual Advertising Classifier from Wikipedia”. In: *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*. Association for Computational Linguistics, 2021, pp. 1–9.
- [62] Yiping Jin, Dittaya Wanvarie, and Phu Le. “Combining Lightly Supervised Text Classification Models for Accurate Contextual Advertising”. In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Asian Federation of Natural Language Processing, 2017, pp. 545–554.
- [63] I. T. Jolliffe. *Principal component analysis*. Springer, 2011.
- [64] Mandar Joshi et al. “Spanbert: Improving Pre-training by Representing and Predicting Spans”. In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 64–77.
- [65] Namgyu Jung et al. “News Category Classification via Multimodal Fusion Method”. In: *Proceedings of the 2023 International Conference on Research in Adaptive and Convergent Systems*. Association for Computing Machinery, 2023.
- [66] Sharif Amit Kamran et al. “Optic-Net: A Novel Convolutional Neural Network for Diagnosis of Retinal Diseases from Optical Tomography Images”. In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. 2019, pp. 964–971.
- [67] Hamid Karimi and Jiliang Tang. “Learning Hierarchical Discourse-level Structure for Fake News Detection”. 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, 2019, pp. 3432–3442.
- [68] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).
- [69] Mykyta Kretinin and Giang Nguyen. “Topic Modeling on News Articles using Latent Dirichlet Allocation”. In: *2022 IEEE 26th International Conference on Intelligent Engineering Systems (INES)*. 2022, pp. 000249–000254.
- [70] S. Kullback and R. A. Leibler. “On Information and Sufficiency”. In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86.

- [71] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. “Data Augmentation using Pre-trained Transformer Models”. In: *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*. Association for Computational Linguistics, 2020, pp. 18–26.
- [72] Philippe Laban and Marti Hearst. “newsLens: building and visualizing long-ranging news stories”. In: *Proceedings of the Events and Stories in the News Workshop*. Association for Computational Linguistics, 2017, pp. 1–9.
- [73] Samuli Laine and Timo Aila. “Temporal Ensembling for Semi-supervised Learning”. In: *arXiv preprint arXiv:1610.02242* (2016).
- [74] Teven Le Scao and Alexander Rush. “How many data points is a prompt worth?” In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021, pp. 2627–2636.
- [75] Y. Lecun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.
- [76] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.
- [77] Daniel D Lee and H Sebastian Seung. “Learning the parts of objects by non-negative matrix factorization”. In: *Nature* 401.6755 (1999), pp. 788–791.
- [78] Dong-Hyun Lee. “Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks”. In: *ICML 2013 Workshop on Challenges in Representation Learning (WREPL)* (2013).
- [79] Mike Lewis et al. “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020, pp. 7871–7880.
- [80] Junnan Li et al. “Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models”. In: *arXiv preprint arXiv:2301.12597* (2023).
- [81] Ruihui Li and Hongbin Wang. “Clustering of Short Texts Based on Dynamic Adjustment for Contrastive Learning”. In: *IEEE Access* 10 (2022), pp. 76069–76078.
- [82] Weixin Li et al. “Joint Image-Text News Topic Detection and Tracking by Multimodal Topic And-Or Graph”. In: *IEEE Transactions on Multimedia* 19.2 (2017), pp. 367–381.
- [83] Xinhang Li et al. “IMF: Interactive Multimodal Fusion Model for Link Prediction”. In: *Proceedings of the ACM Web Conference 2023*. Association for Computing Machinery, 2023, pp. 2572–2580.

- [84] Mathis Linger and Mhamed Hajaiej. “Batch clustering for multilingual news streaming”. In: *arXiv preprint arXiv:2004.08123* (2020).
- [85] Chen Liu et al. “FLiText: A Faster and Lighter Semi-Supervised Text Classification with Convolution Networks”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 2481–2491.
- [86] Dairui Liu, Derek Greene, and Ruihai Dong. “A Novel Perspective to Look At Attention: Bi-level Attention-based Explainable Topic Modeling for News Classification”. In: *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, 2022, pp. 2280–2290.
- [87] Haotian Liu et al. “Visual instruction tuning”. In: *Advances in neural information processing systems* 36 (2024).
- [88] Ruibo Liu, Guangxuan Xu, and Soroush Vosoughi. “Enhanced offensive language detection through data augmentation”. In: *ICWSM Data Challenge* (2020).
- [89] Ruibo Liu et al. “Data Boost: Text Data Augmentation Through Reinforcement Learning Guided Conditional Generation”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2020, pp. 9031–9041.
- [90] Yinhan Liu et al. “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692* (2019).
- [91] Jiali Ma et al. “Invariant Feature Regularization for Fair Face Recognition”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2023, pp. 20861–20870.
- [92] Xin Ma and Won Hwa Kim. “Locally Normalized Soft Contrastive Clustering for Compact Clusters”. In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. International Joint Conferences on Artificial Intelligence Organization, 2022, pp. 3313–3320.
- [93] James MacQueen et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. 1967, pp. 281–297.
- [94] Emily E. Marsh and Marilyn Domas White. “A taxonomy of relationships between images and text”. In: *Journal of Documentation* 59.6 (2003), pp. 647–672.
- [95] Radan Martinec and Andrew Salway. “A system for image–text relations in new (and old) media”. In: *Visual Communication* 4.3 (2005), pp. 337–371.

- [96] Tambat Matiisen et al. “Teacher–Student Curriculum Learning”. In: *IEEE Transactions on Neural Networks and Learning Systems* 31.9 (2020), pp. 3732–3740.
- [97] Leland McInnes, John Healy, and James Melville. “Umap: Uniform manifold approximation and projection for dimension reduction”. In: *arXiv preprint arXiv:1802.03426* (2018).
- [98] Nikhil Mehta, Maria Leonor Pacheco, and Dan Goldwasser. “Tackling Fake News Detection by Continually Improving Social Context Representations using Graph Neural Networks”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2022, pp. 1363–1380.
- [99] Yuxi Mi et al. “Privacy-Preserving Face Recognition Using Random Frequency Components”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2023, pp. 19673–19684.
- [100] Md Messal Monem Miah, Adarsh Pyarelal, and Ruihong Huang. “Hierarchical Fusion for Online Multimodal Dialog Act Classification”. In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 2023, pp. 7532–7545.
- [101] Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).
- [102] Sebastiao Miranda et al. “Multilingual Clustering of Streaming News”. In: Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 4535–4544.
- [103] Saloni Mohan et al. “Stock Price Prediction Using News Sentiment Analysis”. In: *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*. 2019, pp. 205–208.
- [104] Maziar Moradi Fard, Thibaut Thonet, and Eric Gaussier. “Deep k-Means: Jointly clustering with k-Means and learning representations”. In: *Pattern Recognition Letters* 138 (2020), pp. 185–192.
- [105] Eric Müller-Budack et al. “Multimodal news analytics using measures of cross-modal entity and context consistency”. In: *International Journal of Multimedia Information Retrieval* 10.2 (2021), pp. 111–125.
- [106] Ashutosh Nayak, Mayur Garg, and Rajasekhara Reddy Duvvuru Muni. “News Popularity Beyond the Click-Through-Rate for Personalized Recommendations”. In: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2023, pp. 1396–1405.
- [107] Chuang Niu, Hongming Shan, and Ge Wang. “Spice: Semantic pseudo-labeling for image clustering”. In: *IEEE Transactions on Image Processing* 31 (2022), pp. 7264–7278.

- [108] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv:1807.03748* (2018).
- [109] Nelleke Oostdijk et al. “The Connection between the Text and Images of News Articles: New Insights for Multimedia Analysis”. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. European Language Resources Association, 2020, pp. 4343–4351.
- [110] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.
- [111] Ronald DR Pereira and Fabrício Murai. “How effective are Graph Neural Networks in Fraud Detection for Network Data?” In: *arXiv preprint arXiv:2105.14568* (2021).
- [112] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. “True few-shot learning with language models”. In: *Advances in neural information processing systems* 34 (2021), pp. 11054–11070.
- [113] Matthew E. Peters et al. “Deep Contextualized Word Representations”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018, pp. 2227–2237.
- [114] Fabio Petroni et al. “Language Models as Knowledge Bases?” In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019, pp. 2463–2473.
- [115] Hieu Pham et al. “Meta pseudo labels”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 11557–11568.
- [116] Tao Qi et al. “News Recommendation with Candidate-aware User Modeling”. In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2022, pp. 1917–1921.
- [117] Shengsheng Qian et al. “Multi-Modal Event Topic Model for Social Event Analysis”. In: *IEEE Transactions on Multimedia* 18.2 (2016), pp. 233–246.
- [118] Shengsheng Qian et al. “Open-World Social Event Classification”. In: *Proceedings of the ACM Web Conference 2023*. Association for Computing Machinery, 2023, pp. 1562–1571.
- [119] J. Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.



- [120] Alec Radford et al. *Improving language understanding by generative pre-training*. OpenAI. 2018. URL: [https://cdn.openai.com/research-covers/language-unsupervised/language\\_understanding\\_paper.pdf](https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf).
- [121] Alec Radford et al. “Language models are unsupervised multitask learners”. In: *OpenAI blog* 1.8 (2019), p. 9.
- [122] Alec Radford et al. “Learning Transferable Visual Models From Natural Language Supervision”. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. PMLR, 2021, pp. 8748–8763.
- [123] Alec Radford et al. “Learning transferable visual models from natural language supervision”. In: *International conference on machine learning*. 2021, pp. 8748–8763.
- [124] Jack W Rae et al. “Scaling language models: Methods, analysis & insights from training gopher”. In: *arXiv preprint arXiv:2112.11446* (2021).
- [125] Colin Raffel et al. “Exploring the limits of transfer learning with a unified text-to-text transformer”. In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 5485–5551.
- [126] Md Rashadul Hasan Rakib et al. “Enhancement of Short Text Clustering by Iterative Classification”. In: *Natural Language Processing and Information Systems*. Springer International Publishing, 2020, pp. 105–117.
- [127] LKPJ Rduseeun and P Kaufman. “Clustering by means of medoids”. In: *Proceedings of the statistical data analysis based on the L1 norm conference, neuchatel, switzerland*. Vol. 31. 1987.
- [128] *Regulation (EU) 2016/679 General Data Protection Regulation*. Official Journal of the European Union. 2016. URL: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679>.
- [129] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 2019, pp. 3982–3992.
- [130] D. A. Reynolds. “Gaussian mixture models”. In: *Encyclopedia of Biometrics*. Springer, 2015, pp. 829–832.
- [131] Johannes Rieke et al. “Visualizing Convolutional Networks for MRI-Based Diagnosis of Alzheimer’s Disease”. In: *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*. Springer International Publishing, 2018, pp. 24–31.
- [132] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, 1962.

- [133] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *Computing Research Repository* arXiv:1609.04747 (2016).
- [134] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *Nature* 323.6088 (1986), pp. 533–536.
- [135] Gerard Salton and Christopher Buckley. “Term-weighting approaches in automatic text retrieval”. In: *Information Processing & Management* 24.5 (1988), pp. 513–523.
- [136] Victor Sanh et al. “Multitask Prompted Training Enables Zero-Shot Task Generalization”. In: *International Conference on Learning Representations*. 2022.
- [137] Kailash Karthik Saravanakumar et al. “Event-Driven News Stream Clustering using Entity-Aware Contextual Embeddings”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, 2021, pp. 2330–2340.
- [138] Timo Schick and Hinrich Schütze. “Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, 2021, pp. 255–269.
- [139] Martin Schmitt et al. “Joint Aspect and Polarity Classification for Aspect-based Sentiment Analysis with End-to-End Neural Networks”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2018, pp. 1109–1114.
- [140] Rico Sennrich, Barry Haddow, and Alexandra Birch. “Improving Neural Machine Translation Models with Monolingual Data”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 86–96.
- [141] Connor Shorten and Taghi M. Khoshgoftaar. “A survey on Image Data Augmentation for Deep Learning”. In: *Journal of Big Data* 6.1 (2019), p. 60.
- [142] Carson Sievert and Kenneth Shirley. “LDAvis: A Method for Visualizing and Interpreting Topics”. In: *Proceedings of the workshop on interactive language learning, visualization, and interfaces*. 2014, pp. 63–70.
- [143] P.Y. Simard, D. Steinkraus, and J.C. Platt. “Best practices for convolutional neural networks applied to visual document analysis”. In: *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings*. 2003, pp. 958–963.
- [144] Kihyuk Sohn et al. “A simple semi-supervised learning framework for object detection”. In: *arXiv preprint arXiv:2005.04757* (2020).

- [145] Tiberiu Sosea and Cornelia Caragea. “eMLM: A New Pre-training Objective for Emotion Related Tasks”. In: *ACL*. 2021.
- [146] Lukas Stankevičius and Mantas Lukoševičius. “Generating Abstractive Summaries of Lithuanian News Articles Using a Transformer Model”. In: *Information and Software Technologies*. Springer International Publishing, 2021, pp. 341–352.
- [147] Todor Staykovski et al. “Dense vs. Sparse Representations for News Stream Clustering”. In: *Proceedings of Text2Story - 2nd Workshop on Narrative Extraction From Texts, co-located with the 41st European Conference on Information Retrieval, Text2Story@ECIR 2019, Cologne, Germany, April 14th, 2019*. Vol. 2342. CEUR-WS.org, 2019, pp. 47–52.
- [148] James Stewart. *Calculus*. Cengage Learning, 2015.
- [149] Student. “The probable error of a mean”. In: *Biometrika* 6.1 (1908), pp. 1–25.
- [150] Alvin Subakti, Hendri Murfi, and Nora Hariadi. “The performance of BERT as data representation of text clustering”. In: *Journal of Big Data* (2022).
- [151] R. S. Sutton and A. G. Barto. “Reinforcement learning: An introduction”. In: *MIT Press* (1998).
- [152] Raphael Tang et al. “Distilling task-specific knowledge from bert into simple neural networks”. In: *arXiv preprint arXiv:1903.12136* (2019).
- [153] Antti Tarvainen and Harri Valpola. “Mean Teachers are Better Role Models: Weight-averaged Consistency Targets Improve Semi-supervised Deep Learning Results”. In: *arXiv preprint arXiv:1703.01780* (2017).
- [154] Nandan Thakur et al. “Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021, pp. 296–310.
- [155] Hao Tian et al. “SKEP: Sentiment Knowledge Enhanced Pre-training for Sentiment Analysis”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020, pp. 4067–4076.
- [156] Yi-Hsuan Tsai et al. “Learning to Adapt Structured Output Space for Semantic Segmentation”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7472–7481.
- [157] Tsung Wei Tsai, Chongxuan Li, and Jun Zhu. “Mi{CE}: Mixture of Contrastive Experts for Unsupervised Image Clustering”. In: *International Conference on Learning Representations*. 2021.
- [158] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.11 (2008).

- [159] Wouter Van Gansbeke et al. “SCAN: Learning to Classify Images Without Labels”. In: *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X*. Springer-Verlag, 2020, pp. 268–285.
- [160] Tongzhou Wang and Phillip Isola. “Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere”. In: *Proceedings of the 37th International Conference on Machine Learning*. JMLR.org, 2020.
- [161] Zhen Wang et al. “N24News: A New Dataset for Multimodal News Classification”. In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. European Language Resources Association, 2022, pp. 6768–6775.
- [162] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.
- [163] Colin Wei et al. “Theoretical analysis of self-training with deep networks on unlabeled data”. In: *arXiv preprint arXiv:2010.03622* (2020).
- [164] Jason Wei and Kai Zou. “EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019, pp. 6382–6388.
- [165] Svante Wold, Kim Esbensen, and Paul Geladi. “Principal component analysis”. In: *Chemometrics and intelligent laboratory systems 2* (1987), pp. 37–52.
- [166] Thomas Wolf et al. “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 2020, pp. 38–45.
- [167] Fangzhao Wu et al. “MIND: A Large-scale Dataset for News Recommendation”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020, pp. 3597–3606.
- [168] Yang Wu et al. “Multimodal fusion with co-attention networks for fake news detection”. In: *Findings of the association for computational linguistics: ACL-IJCNLP 2021*. 2021, pp. 2560–2569.
- [169] Junyuan Xie, Ross Girshick, and Ali Farhadi. “Unsupervised deep embedding for clustering analysis”. In: *International conference on machine learning*. PMLR. 2016, pp. 478–487.

- [170] Qizhe Xie et al. “Self-Training With Noisy Student Improves ImageNet Classification”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 10684–10695.
- [171] Qizhe Xie et al. “Unsupervised Data Augmentation for Consistency Training”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 6256–6268.
- [172] Haiming Xu, Lingqiao Liu, and Ehsan Abbasnejad. “Progressive Class Semantic Matching for Semi-supervised Text Classification”. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2022, pp. 3003–3013.
- [173] Rui Xu and Donald Wunsch. “Survey of clustering algorithms”. In: *IEEE Transactions on neural networks* 16.3 (2005), pp. 645–678.
- [174] Xinnuo Xu et al. “MiRANews: Dataset and Benchmarks for Multi-Resource-Assisted News Summarization”. In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, 2021, pp. 1541–1552.
- [175] Yumo Xu and Shay B. Cohen. “Stock Movement Prediction from Tweets and Historical Prices”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2018, pp. 1970–1979.
- [176] Chenxiao Yang et al. “Cross-task knowledge distillation in multi-task recommendation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 4. 2022, pp. 4318–4326.
- [177] Jianwei Yang, Devi Parikh, and Dhruv Batra. “Joint unsupervised learning of deep representations and image clusters”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* . 2016, pp. 5147–5156.
- [178] Ming Yang et al. “News Text Mining-Based Business Sentiment Analysis and Its Significance in Economy”. In: *Frontiers in Psychology* 13 (2022).
- [179] Sin-han Yang et al. “Entity-Aware Dual Co-Attention Network for Fake News Detection”. In: *Findings of the Association for Computational Linguistics: EACL 2023*. Association for Computational Linguistics, 2023, pp. 106–113.
- [180] Weiyi Yang et al. “Prototype-Guided Pseudo Labeling for Semi-Supervised Text Classification”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2023, pp. 16369–16382.

- [181] Zichao Yang et al. “Hierarchical Attention Networks for Document Classification”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2016, pp. 1480–1489.
- [182] Susik Yoon et al. “SCStory: Self-Supervised and Continual Online Story Discovery”. In: *Proceedings of the ACM Web Conference 2023*. Association for Computing Machinery, 2023, pp. 1853–1864. ISBN: 9781450394161.
- [183] Ahmet Üstün et al. “UDapter: Language Adaptation for Truly Universal Dependency Parsing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020, pp. 2302–2315.
- [184] Jianfei Yu et al. “Improving Multi-label Emotion Classification via Sentiment Classification with Dual Attention Transfer Network”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018, pp. 1097–1102.
- [185] Dejiao Zhang et al. “Supporting Clustering with Contrastive Learning”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021, pp. 5419–5430.
- [186] Hongyi Zhang et al. “mixup: Beyond Empirical Risk Minimization”. In: *International Conference on Learning Representations*. 2018.
- [187] Xiang Zhang, Junbo Zhao, and Yann LeCun. “Character-level Convolutional Networks for Text Classification”. In: *Advances in Neural Information Processing Systems*. Vol. 28. Curran Associates, Inc., 2015.
- [188] Zhengyan Zhang et al. “ERNIE: Enhanced Language Representation with Informative Entities”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 1441–1451.
- [189] Zihan Zhang et al. “Is Neural Topic Modelling Better than Clustering? An Empirical Study on Clustering with Contextual Embeddings for Topics”. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2022, pp. 3886–3893.
- [190] Chao Zhao et al. “Read Top News First: A Document Reordering Approach for Multi-Document News Summarization”. In: *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, 2022, pp. 613–621.

- [191] Huasong Zhong et al. “Graph Contrastive Clustering”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 9224–9233.
- [192] Deyu Zhou, Haiyang Xu, and Yulan He. “An Unsupervised Bayesian Modelling Approach for Storyline Detection on News Articles”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2015, pp. 1943–1948.
- [193] Xinyi Zhou et al. “ReCOVery: A Multimodal Repository for COVID-19 News Credibility Research”. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. Association for Computing Machinery, 2020, pp. 3205–3212.
- [194] Yangming Zhou et al. “Multi-modal Fake News Detection on Social Media via Multi-grained Information Fusion”. In: *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval*. Association for Computing Machinery, 2023, pp. 343–352.
- [195] Yangming Zhou et al. “Multimodal fake news detection via clip-guided learning”. In: *2023 IEEE International Conference on Multimedia and Expo (ICME)*. 2023, pp. 2825–2830.





---

## **Cformer: Semi-Supervised Text Clustering Based on Pseudo Labeling**

Arezoo Hatefi, Xuan-Son Vu, Monowar Bhuyan, and Frank Drewes

*In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM), pp. 3078-3082, 2021.*



# Cformer: Semi-Supervised Text Clustering Based on Pseudo Labeling\*

Arezoo Hatefi, Xuan-Son Vu, Monowar Bhuyan, Frank Drewes

*Department of Computing Science, Umeå University, Umeå, Sweden*

*arezooh@cs.umu.se, sonvx@cs.umu.se, monowar@cs.umu.se, drewes@cs.umu.se*

**Abstract:** We propose a semi-supervised learning method called *Cformer* for automatic clustering of text documents in cases where clusters are described by a small number of labeled examples, while the majority of training examples are unlabeled. We motivate this setting with an application in contextual programmatic advertising, a type of content placement on news pages that does not exploit personal information about visitors but relies on the availability of a high-quality clustering computed on the basis of a small number of labeled samples.

To enable text clustering with little training data, *Cformer* leverages the teacher-student architecture of Meta Pseudo Labels. In addition to unlabeled data, *Cformer* uses a small amount of labeled data to describe the clusters aimed at. Our experimental results confirm that the performance of the proposed model improves the state-of-the-art if a reasonable amount of labeled data is available. The models are comparatively small and suitable for deployment in constrained environments with limited computing resources.

**Key words:** meta pseudo clustering, semi-supervised learning, pseudo labeling

## 1 Introduction

Clustering in its purest form refers to unsupervised methods for dividing a set of  $n$  data points into  $k$  so-called clusters, groups of closely related points. For this, a similarity measure between data points is required. When the objective is to cluster text documents, using the similarity of document vectors given by some standard model usually does not work very well because of the high dimensionality of these vector spaces [1]. Furthermore, downstream tasks often require the clusters to carry meaning. An application area in which

---

\*The paper has been published in the *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM '21)*, and has been re-typeset to match the thesis style.

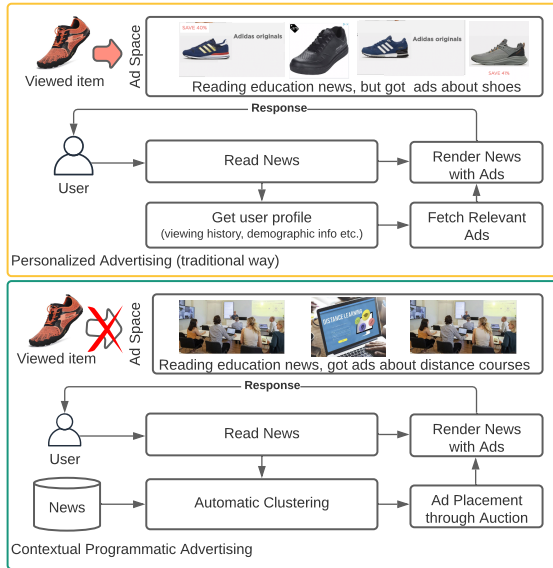


Figure 1: A conceptual comparison of personalized and contextual advertising. The former exploits personal information, the latter uses only the news content, thus being less intrusive.

this is the case is the one that motivated this work: contextual programmatic advertising. To make clusters reflect intended meaning, one would ideally want the clustering approach to be trained on labeled data. Unfortunately, this area is also one in which large amounts of labeled data are hard to come by.

In programmatic advertising [12], the aim is to fill ad space on, e.g., a news page, in real time with suitable ads when a visitor of the site accesses the page. To accomplish this, programmatic advertising platforms conduct auctions for ad space the moment pages are accessed. Software agents representing advertisers place their bids according to their notion of how much the advertising space is worth, and the ad space goes to the one who wins the auction. The worth of the ad space is traditionally estimated based on personal information about the visitor, such as their viewing history. Contextual advertising is a comparatively new idea that challenges this model. It avoids the use of personal information for both privacy and efficiency reasons by focusing on the content of the news page to decide how well the ad fits it.<sup>1</sup>

Here, “fitting” often does not simply mean that the contents of news article and ad align. Companies often conduct advertising campaigns during which they want their ads to be seen (or not to be seen) in contexts that promote a

<sup>1</sup>If someone has recently bought new shoes but they are currently looking at a news page about self-education, they might not be interested in buying yet another pair of shoes, but would perhaps be inclined to sign up for online courses (see Figure 1).

certain image, regardless of the specific product being advertized.

Abstractly, each desired context can be understood as a cluster. These clusters and their descriptions change over time as campaigns are canceled and new ones are set up. Most importantly, as campaigns may focus on arbitrary aspects, there is typically little labeled data available. To cope with this situation, we propose *Cformer*, a semi-supervised clustering approach that makes use of a small amount of labeled documents (news articles provided as typical example contexts for a given advertising campaign) and a larger number of unlabeled documents (uncategorized news articles).

Cformer is inspired by the recent work of Pham et al. [11] on meta pseudo labels, an extension of pseudo labeling. The latter is a successful semi-supervised learning method which resulted in state-of-the-art performance in many computer vision tasks. It works by having a pair of networks, a *student* and a *teacher*. The teacher model predicts labels for unlabeled data, so-called pseudo labels. Then both pseudo labeled data and the original labeled data are used to train the student. To tackle the confirmation bias (the student learns to confirm the teacher), the idea of meta pseudo labeling is to train teacher and student in parallel, letting the teacher use the performance of the student on labeled data to predict better pseudo labels. We transfer this idea to the realm of text clustering. Also, our Distill-Cformer model departs from using identical teacher and student architectures. This speeds up training, which is important for contextual advertising due to the frequently changing campaigns.

The main contributions of the present work are:

- The proposed Cformer model utilizes meta pseudo labels for document clustering. The architecture is adaptable to similar tasks such as document classification and document retrieval.
- We further introduce Distill-Cformer to confirm the effectiveness of our proposed architecture on a much smaller neural model (i.e., DistilBert [13]) for the student, thus considerably reducing the overall training time.
- We conduct performance tests with two benchmark datasets. The results of these experiments indicate that Cformer and Distill-Cformer outperform the state of the art in most cases.

## 2 Related Work

Previous work on contextual advertising tried to exploit prior knowledge (usually in the form of labeled words for each class) or generating labeled data automatically. Jin, Wanvarie, and Le [6] model contextual targeting as a lightly-supervised one-class classification problem. Their algorithm takes unlabeled documents and the labeled keywords for the target class  $c$  as input and returns a classifier  $M_c$  identifying documents that belong to class  $c$ .

Jin, Kadam, and Wanvarie [5] automatically map the categories in the Interactive Advertising Bureau (IAB) taxonomy to category nodes in the Wikipedia

category graph and propagate labels across the graph to obtain a list of labeled Wikipedia documents for training purposes.

We tackle document clustering with limited labeled data by semi-supervised learning. Such methods add more flexibility to supervised approaches by needing only a very small portion of the dataset to be labeled. Many of the recent approaches in semi-supervised learning use consistency training on a large amount of unlabeled data [7, 14]. These methods regularize model predictions to be invariant to small levels of noise. Data augmentation methods are used to enlarge labeled datasets in supervised learning cases when training data is not sufficient, e.g., in Augmented SBERT [15]. Further, these methods can be used to inject noise to data. Xie et al. [16] investigate the role of noise injection in consistency training and propose Unsupervised Data Augmentation (UDA) to replace the traditional noise injection methods by high quality data augmentation such as back translation of textual data. Chen, Yang, and Yang [3] propose the data augmentation method TMix that takes in two text instances and interpolates them in their corresponding hidden space. Based on TMix they propose MixText, a semi-supervised learning method for text classification and clustering. MixText predicts labels for unlabeled data and then uses TMix to interpolate between labeled and unlabeled data to impose a regularization on the model.

### 3 Methodology

Figure 2 shows our proposed approach for semi-supervised clustering. Following the architecture of [11], we have a *teacher* model  $T$  with learnable parameters  $\Theta_T$  (left side in Figure 2) and a *student* model  $S$  with learnable parameters  $\Theta_S$  (right side in Figure 2) that are trained in parallel. The teacher is trained with the Unsupervised Data Augmentation (UDA) objective [16] and feedback from the student [11]. The UDA objective consists of supervised loss on labeled data and consistency loss between unlabeled and augmented data. Additional feedback is the performance of the student on labeled data (which is assumed to be correctly labeled). The student is trained with supervised loss on the pseudo labeled data provided by the teacher. Augmented data is built by applying text augmentation techniques (e.g., word substitution with the most suitable word found by ContextualWordEmbsAug [9]) on unlabeled data. As Figure 2 shows, both the student and the teacher consist of encoders that map documents to their distributed representations (transformer and a mean pooling module that computes the average of the transformer outputs in different positions) followed by a classifier.

In a first training step, a batch of labeled data  $(x_l, y_l)$  (track ① in Figure 2), a batch of unlabeled data  $x_u$  (track ③), and its augmented version  $x_a$  (track ②) are fed to the teacher. The cross entropy loss is computed between labels

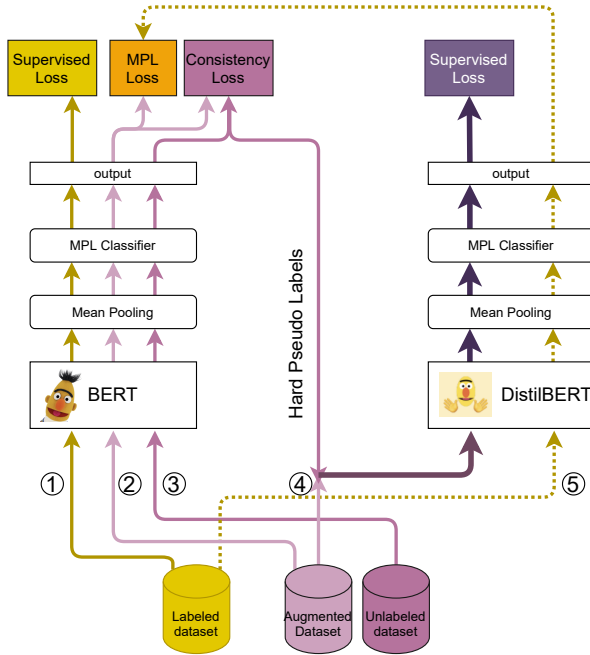


Figure 2: The teacher(left network)-student(right network) architecture of Distill-Cformer.

$y_l$  and teacher outputs for  $x_l$ :

$$Loss_T^l = CrossEntropy(y_l, T(x_l; \theta_T)) .$$

Unsupervised or consistency loss is computed using  $x_u$  and  $x_a$ . The consistency loss constrains the model predictions to be invariant to input noise by forcing augmented samples to have the same labels as the original data samples. Moreover, to encourage the model to predict confident low-entropy labels for unlabeled data, we use a sharpening function over soft predictions for  $x_u$  denoted as  $y_u^{soft}$ . We utilize the sharpening function used in Chen, Yang, and Yang [3]. Given *soft pseudo labels*  $y_u^{soft}$  and a temperature hyper-parameter  $Temp$

$$y_u^{soft} = T(x_u; \Theta_T)$$

$$y_u^{sharp} = sharpen(y_u^{soft}, Temp) = \frac{(y_u^{soft})^{\frac{1}{T}}}{\|(y_u^{soft})^{\frac{1}{T}}\|}$$

where  $\|\cdot\|$  is the  $l_1$ -norm of the vector. So, the teacher unsupervised loss is

$$Loss_T^u = CrossEntropy(y_u^{sharp}, T(x_a; \Theta_T)) .$$

We found it helpful to mask out examples that the current model is not confident about. So, in each batch, the consistency loss term is computed only on

examples whose highest probability among clustering categories is greater than an experimentally determined threshold  $\beta$ .

In a second step, the student model learns from pseudo labeled data annotated by the teacher. The augmented batch  $x_a$  (as a regularization to make the student insensitive to noise) and *hard pseudo labels*  $y_u^{hard}$  (cross-point ④ in Figure 2) are fed to the student. The student tries to minimize the cross entropy loss between the hard pseudo labels and its own predictions. The hard pseudo labels  $y_u^{hard}$  are generated by considering the clusters with the highest values among the soft pseudo labels  $y_u^{soft}$  as the correct clusters. Therefore:

$$y_u^{hard} = j : y_{u,j}^{soft} = \max_i(y_{u,i}^{soft})$$

$$Loss_S^l = CrossEntropy(y_u^{hard}, S(x_a; \Theta_S)) .$$

In parallel, the teacher learns from the reward signal of how well the student performs on labeled data  $(x_l, y_l)$  (dotted line ⑤) from student to teacher in Figure 2). This loss is called *Meta Pseudo Labels (MPL) loss*. Using the parameters of the student after updating with  $Loss_S^l$  as  $\Theta'_S$ :

$$Loss_T^{MPL} = \nabla_{\Theta_T} CrossEntropy(y_l, S(x_l; \Theta'_S)) .$$

To see how this loss is exactly computed and its derivation equations, we refer to Pham et al. [11].

Combining the three losses, we get the overall objective function of the teacher:

$$Loss_T = Loss_T^l + \lambda_u * loss_T^u + Loss_T^{MPL}$$

where  $\lambda_u$  is the contribution coefficient of the consistency loss.

Finally, as the student only learns from unlabeled data with pseudo labels generated by the teacher, we fine-tune the student (that has converged after training with pseudo labels) on labeled data to improve its accuracy. Moreover, to increase the generalization capability of both student and teacher, we use label smoothing [10] when computing supervised losses  $Loss_T^l$  and  $Loss_S^l$  to prevent the model from overfitting to labeled data.

## 4 Experiments and Result Analysis

We perform experiments with two English text classification benchmark datasets: AG News [17] and Yahoo! answers [2]. For Yahoo! answers, we obtain the text to be clustered by concatenating the question title, question content and best answer; for AG News we only utilize the news content (without titles). To be comparable with our baselines, we randomly sample the same amount of data as in [3] from the original training sets for our unlabeled and validation sets and used the original test sets. The dataset statistics and splits are available in Figure 1. To generate augmented data from unlabeled data,



Table 1: Dataset statistics and dataset split. The number of sentences and words are denoted by  $\#s$  and  $\#w$ , respectively. The number of unlabeled, dev, and test data items are given in terms of the number of data items per class.

Dataset	Classes	Documents	Average $\#s$	Max $\#s$	Average $\#w$	Max $\#w$	Vocabulary	Unlabeled	Dev	Test
Yahoo! answers	10	1,450,000	6.4	515	108.4	4002	1,554,607	5000	5000	6000
AG News	4	120,000	1.7	20	36.2	212	94,443	5000	2000	1900

Table 2: Experimental results of our proposal models (Cformer & Distill-Cformer) in comparison with SoTA models. Bold values indicate the highest performance per column.

Dataset	Model	10	200	2500	Dataset	Model	10	200	2500
AG News	BERT	69.5	87.5	90.8	Yahoo! answers	BERT	56.2	69.3	73.2
	UDA [16]	84.4	88.3	91.2		UDA [16]	63.2	70.2	73.6
	MixText [3]	88.4	89.2	91.5		MixText [3]	<b>67.6</b>	71.3	74.1
	Cformer (Ours)	<b>88.7</b>	89.9	91.8		Cformer (Ours)	66.8	<b>72.0</b>	<b>74.5</b>
	Distill-Cformer (Ours)	88.0	<b>90.0</b>	<b>91.9</b>		Distill-Cformer (Ours)	65.2	71.9	74.3

we use the library *nlpaug* [9]. We substitute text words based on contextual word embeddings with probability 0.9.

We use MixText [3] together with two of its baselines (BERT [4] and UDA [16]) as our baseline models and compare our results against the results for these models as reported in [3]. The BERT baseline is a BERT-base-uncased model fine-tuned only with the labeled data for text classification. It consists of a two-layer MLP (as in our model) on top of the BERT encoder. The UDA baseline is a PyTorch version of the original UDA model implemented for GPU by the inventors of MixText. We consider two variations of our proposed architecture with different encoder components:

1. **Cformer** model: the student and the teacher models both use the BERT-base-uncased model.
2. **Distill-Cformer** model: the teacher is the same as in Cformer but the student uses DistilBERT-base-uncased.

The teacher and student models in Cformer have 109.58 million parameters; the student in Distill-Cformer has 66.46 million parameters. We use the BERT-based-uncased tokenizer to tokenize the text, average pooling over the output of the transformer to aggregate word embeddings into document embedding, and a two-layer MLP with a 128 hidden size and hyperbolic tangent as its activation function (the same as in MixText) to predict the labels. Documents are truncated to their first 256 tokens. Like UDA and MixText, in all experiments, the labeled and unlabeled batch sizes are 4 and 8, respectively. Both models are trained with the AdamW optimizer [8]. We train our models for 7000 steps (including 50 warm-up steps) and evaluate them every 500 steps. To avoid overfitting, we use early stopping with delta 5E-3 and patience 4. We set the learning rate of the transformer and classifier components in both models to 1E-5 and 1E-3 respectively. After training both models, we fine-tune the stu-

dent on the labeled dataset using the AdamW optimizer with a fixed learning rate of 5E-6 and a batch size of 32, running for 10 epochs. The temperature  $T$  for sharpening is set to 0.5 for Yahoo answers and 0.3 for AG News. The confidence threshold  $\beta$  is set to 0.9 and the label smoothing parameter is 0.15 for both datasets. For the contribution coefficient of unsupervised loss in the teacher loss function  $\lambda_u$ , we start from 0 and increase it linearly for 6000 steps until it reaches 1. All experiments are run using 4 GPU V100 32GB. With small batch sizes, the model can be trained using other regular GPUs. Since we kept the same batch size as previous work, the training process only occupies 16GB of memory per GPU.

## 4.1 Result Analysis

Table 2 presents our results with Cformer and Distill-Cformer in comparison with other methods.

**Overall performance of Cformer.** In comparison with the current state-of-the-art models, we can observe that ours yield good performance across the considered datasets. First, our model outperformed UDA in all experiments. In fact, Cformer achieves better accuracy than UDA from 0.6% to more than 4% across these datasets. Since the teacher in our model is trained with the UDA objective function, this shows the effectiveness of using pseudo labels and knowledge distillation from teacher to student. Second, in comparison to MixText, Cformer stably works better on both datasets unless the number of labeled samples is very small. For 10-shot cases, Cformer achieves better performance on AG News but worse on Yahoo! answers. In this regard, it is worth observing that the AG News dataset is easier to learn than the Yahoo! answers dataset, due to its smaller vocabulary and the smaller number of documents. Therefore, less labeled data is required to learn how to classify AG news than for the Yahoo! answers dataset.

**Cformer vs. Distill-Cformer.** Given the requirement of getting high performance in constrained environments, we are especially interested in analyzing Distill-Cformer. Generally, it exhibits on-par performance compared to Cformer even though its student model is considerably smaller. The gap in performance is less than 0.2% in most cases. This indicates that the size of the model is not a bottleneck as long as the knowledge distillation works effectively. Moreover, Distil-Cformer offers faster inference time than Cformer since its architecture is smaller in size. Specifically, testing on the Test set of *Yahoo! answers* dataset with one GPU, the inference time of Distil-Cformer was *143.5 seconds*, which is 2 times faster than that of Cformer (*287.0 seconds*). This result again confirmed the finding of Sanh et al. [13] pointing out that “DistilBERT retains 97% of the performance with 40% fewer parameters”.

## 5 Conclusion

We have presented Cformer, a teacher-student architecture for semi-supervised text clustering in contexts where clusters are given by a limited number of labeled samples. An example application is dynamic content placement on contextual advertising platforms. In general, we expect the technique to be useful for all downstream tasks which require text classification based on partially labeled training data, especially when the labels and the amount of labeled data change over time (as in the case of advertising campaigns).

Cformer showcases a new approach in dealing with both short and long texts datasets. It effectively performs better on both short text data (AG News) and long text data (Yahoo! answers) by integrating the knowledge distillation into the learning process. Moreover, the proposed models work effectively on both full-size BERT and DistillBERT as the encoders. Cformer outperforms the state-of-the-art approaches with various settings, especially when sufficient labeled data is available. For applications such as content placement on web pages, a useful extension would be a multimodal version (e.g., Zong et al. [18] on multimodal clustering).

## References

- [1] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. “On the surprising behavior of distance metrics in high dimensional space”. In: *International conference on database theory*. Springer. 2001, pp. 420–434.
- [2] Ming-Wei Chang et al. “Importance of Semantic Representation: Dataless Classification.” In: *Aaai*. Vol. 2. 2008, pp. 830–835.
- [3] Jiaao Chen, Zichao Yang, and Diyi Yang. “MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 2147–2157. DOI: 10.18653/v1/2020.acl-main.194. URL: <https://aclanthology.org/2020.acl-main.194>.
- [4] Jacob Devlin et al. “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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: <https://aclanthology.org/N19-1423>.

- [5] Yiping Jin, Vishakha Kadam, and Dittaya Wanvarie. “Bootstrapping Large-Scale Fine-Grained Contextual Advertising Classifier from Wikipedia”. In: *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*. Mexico City, Mexico: Association for Computational Linguistics, June 2021, pp. 1–9. URL: <https://aclanthology.org/2021.textgraphs-1.1>.
- [6] Yiping Jin, Dittaya Wanvarie, and Phu Le. “Combining lightly-supervised text classification models for accurate contextual advertising”. In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2017, pp. 545–554.
- [7] Samuli Laine and Timo Aila. “Temporal Ensembling for Semi-Supervised Learning”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017. URL: <https://openreview.net/forum?id=BJ6o0fqge>.
- [8] Ilya Loshchilov and Frank Hutter. “Decoupled Weight Decay Regularization”. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. 2019. URL: <https://openreview.net/forum?id=Bkg6RiCqY7>.
- [9] Edward Ma. *NLP Augmentation*. <https://github.com/makcedward/nlpaug>. 2019.
- [10] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. “When does label smoothing help?”. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach et al. 2019, pp. 4696–4705.
- [11] Hieu Pham et al. “Meta Pseudo Labels”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 11557–11568.
- [12] Anthony Samuel et al. “Programmatic advertising: An exegesis of consumer concerns”. In: *Computers in Human Behavior* 116 (2021), p. 106657.
- [13] Victor Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: (2019). URL: <http://arxiv.org/abs/1910.01108>.
- [14] Antti Tarvainen and Harri Valpola. “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 2017, pp. 1195–1204.

- [15] Nandan Thakur et al. “Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 296–310. DOI: 10.18653/v1/2021.naacl-main.28. URL: <https://aclanthology.org/2021.naacl-main.28>.
- [16] Qizhe Xie et al. “Unsupervised Data Augmentation for Consistency Training”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, December 6-12, 2020, virtual*. 2020.
- [17] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. “Character-level Convolutional Networks for Text Classification”. In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. Ed. by Corinna Cortes et al. 2015, pp. 649–657.
- [18] Linlin Zong et al. “Multimodal Clustering via Deep Commonness and Uniqueness Mining”. In: *The 29th ACM International Conference on Information and Knowledge Management (CIKM), Virtual Event, October 19-23, Ireland*. ACM, Oct. 2020, pp. 2357–2360. DOI: 10.1145/3340531.3412103.

**The Efficiency of Pre-training with Objective Masking in Pseudo Labeling for Semi-Supervised Text Classification**

Arezoo Hatefi, Xuan-Son Vu, Monowar Bhuyan, and Frank Drewes

*Submitted to the Northern European Journal of Language Technology (NEJLT), 2023.*



# The Efficiency of Pre-training with Objective Masking in Pseudo Labeling for Semi-Supervised Text Classification\*

Arezoo Hatefi, Xuan-Son Vu, Monowar Bhuyan, Frank Drewes

*Department of Computing Science, Umeå University, Umeå, Sweden*

*arezoo@cs.umu.se, sonvx@cs.umu.se, monowar@cs.umu.se, drewes@cs.umu.se*

**Abstract:** We extend and study a semi-supervised model for text classification proposed earlier by Hatefi et al. for classification tasks in which document classes are described by a small number of gold-labeled examples, while the majority of training examples is unlabeled. The model leverages the teacher-student architecture of Meta Pseudo Labels in which a “teacher” generates labels for originally unlabeled training data to train the “student” and updates its own model iteratively based on the performance of the student on the gold-labeled portion of the data. We extend the original model of Hatefi et al. by an unsupervised pre-training phase based on objective masking, and conduct in-depth performance evaluations of the original model, our extension, and various independent baselines. Experiments are performed using three different datasets in two different languages (English and Swedish).

## 1 Introduction

Automatic topic classification of news articles is of great practical and commercial interest because of the huge number of news articles produced around the globe every day. Hatefi et al. [9] have proposed a semi-supervised model Cformer for this task. One application area described at some length in that article is contextual advertising, also called cookieless advertising, which places ads in online news media based on the content of the news being viewed rather than on personal information about the viewer.

Applications such as this one often need to pre-determine the topics of interest. For example, a company running an advertisement campaign usually wants its ads to be seen in certain contexts and – often more importantly – not to be seen in others. For this, one would like to tag each article by a topic of interest.

---

\*The paper has been submitted to the *Northern European Journal of Language Technology (NEJLT)*, and has been re-typeset to match the thesis style.



To create such a topic model, one would ideally want to train it on labeled data. However, since the topics of interest may frequently change and cannot be expected to be taken from a preconceived global set, the assumption of having sufficiently much labeled data for training is unrealistic. The best one may hope for is to be given a training set that consists of (a) a comparatively large set  $D_u$  of unlabeled articles of the general type to be classified and (b) a much smaller set  $D_g$  of gold-labeled examples for each of the topics in question.

Given the effectiveness of neural methods in natural language processing, a widespread approach to grouping documents into topics is to define a similarity measure via the distance between document embedding vectors and then use a general purpose clustering method such as  $K$ -Means to group documents into distinct classes [28, 7]. However, as pointed out by Aggarwal, Hinneburg, and Keim [1] the high dimensionality of these vector spaces has a negative impact on the accuracy of such an approach. Therefore, more robust methods such as MixText [5] and UDA [25] have been proposed in the literature. A third approach, recently proposed by Hatefi et al. [9], is called Cformer. It is a semi-supervised approach that makes use of a small set  $D_g$  of gold-labeled documents (such as news articles provided as typical examples of the clusters of interest) and a larger set  $D_u$  of unlabeled documents. Cformer employs a student-teacher architecture originally proposed by Pham et al. [19] for computer vision: two BERT models, the *teacher* and the *student*, are trained in an iterated fashion. The teacher predicts pseudo-labels for documents in  $D_u$ , thus turning it into a pseudo-labeled dataset  $D_p$ . The dataset  $D_p$  is then used for supervised training of the student. In the next step, the teacher’s ability to predict pseudo-labels is improved, using the performance of the student on  $D_g$  as its objective function. Eventually, when the iterative process has converged,  $D_g$  is used to fine-tune the student, yielding the final model. An advantage of Cformer is that the student can be replaced by one based on a smaller model such as DistilBERT to reduce the model size. This variant of Cformer is called Distil-Cformer. The empirical results of Hatefi et al. [9] indicated that both Cformer and Distil-Cformer yield results on par with or better than MixText and UDA.

The major contributions of this paper are

- the model CformerM that extends Cformer by including an unsupervised pre-training phase based on objective masking,
- a comparison of Cformer and CformerM with each other as well as with MixText [5], UDA [25], FLiText [13], PGPL [26], and BERT [6] on AG News and Yahoo! Answers, Medical Abstracts, and a Swedish real-world dataset of news articles, and
- an investigation of the effectiveness of the objective masking in CformerM, its impact on zero-shot classification, and its effect on the reliability and interpretability of the model.

The purpose of introducing masking into the learning process is to fine-tune the basic BERT models underlying the teacher and the student to improve

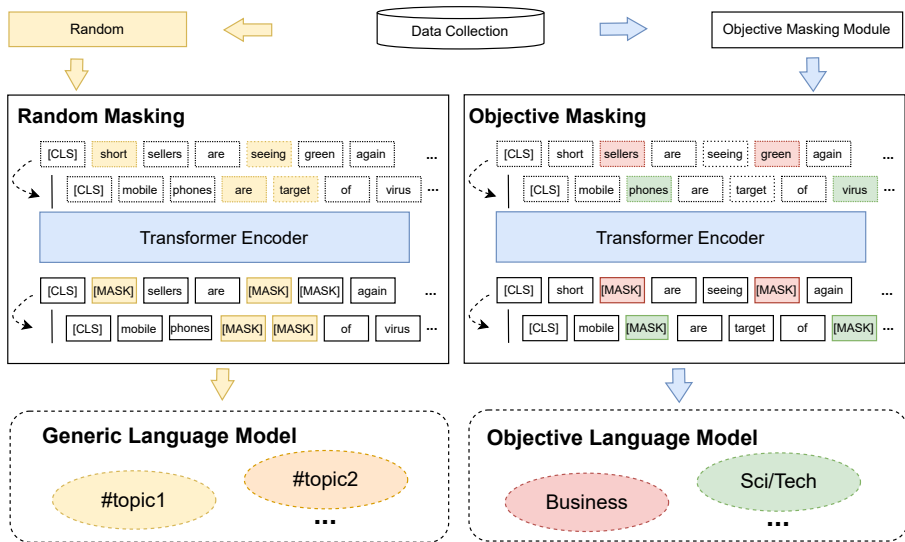


Figure 1: A high-level overview of *Random Masking* in comparison to *Objective Masking*. While the former is used for general purpose language models, the latter is preferable for increasing the sensitivity of a language model to topical information.

their ability to recognize topical information in  $D$  (see Figure 1). For this, we first create an independent unsupervised topic model for the dataset using Latent Dirichlet Allocation (LDA, Blei, Ng, and Jordan [3]). This enables us to choose words carrying topical information. These words are used in objective masking to pre-train the BERT models which will afterwards be trained in the semi-supervised fashion of Cformer.

As we shall see, the model that uses objective masking, called CformerM, outperforms Cformer and other SoTA baselines over two public benchmark datasets (in English) and one private dataset (in Swedish).<sup>1</sup> This confirms the effectiveness of objective masking in adjusting the language models (LMs) to the classification task. However, the extent of the improvement depends on the characteristics of the dataset and the number of labeled examples.

*Organization.* The rest of the paper is organized as follows. An introduction to the basic notions used in this research and a discussion of related work is provided in Section 2. A description of the architecture of Cformer and how it is turned into CformerM can be found in Section 3. The experimental setup and a comprehensive analysis of results are presented in Section 4. Finally, Section 5 summarizes our main conclusions.

<sup>1</sup>The private dataset is available from the owner upon request; see Section 5. This ensures that our results can be reproduced.

## 2 Terminology and Related Work

We briefly discuss some known concepts and methods that are used in this work.

### 2.1 Unsupervised Topic Modeling

The model CformerM introduced in this paper requires an unsupervised topic model of the dataset. To create such a model, we use Latent Dirichlet Allocation (LDA) by Blei, Ng, and Jordan [3]. LDA is a statistical topic modeling algorithm that considers topics as distributions over vocabulary words, and documents as probabilistic mixtures of these topics and attempts to infer these distributions using statistical information. To model the distributions, LDA uses Dirichlet distributions and to infer their parameters, it uses a generative process whereby documents are created from words. Given a corpus  $D$  and a number  $K$  that determines the number of latent topics to look for in  $D$ , LDA attempts to assign multinomial distributions  $\theta_d \sim \text{Dir}(\alpha)$  to each document  $d$ , and  $\phi_k \sim \text{Dir}(\beta)$  to each topic  $k$  in such a way that the probability of  $D$  is maximized if we imagine to generate each document  $d$  word by word.<sup>2</sup> To generate document  $d$ , for each position in the document, LDA first chooses a topic  $k$  from  $\theta_d$  and then samples a word from  $\phi_k$ . Blei, Ng, and Jordan [3] utilize variational Bayes to estimate the optimal parameters of  $\theta_d$  and  $\Phi_k$ .

### 2.2 Semi-supervised Classification

Semi-supervised learning is an emerging research direction attempting to find ways to deal with a lack of labeled samples by relying only on a very small gold labeled subset  $D_g$  of the dataset. Many of the recent approaches in semi-supervised learning use consistency training on a large amount of unlabeled data [11, 24]. These methods regularize model predictions to be invariant to small levels of noise. Xie et al. [25] investigate the role of noise injection in consistency training and proposed Unsupervised Data Augmentation (UDA) to substitute the traditional noise injection with high quality data augmentation (e.e., back translation for textual data).

Pseudo labeling and its extension to meta pseudo labeling are other examples of semi-supervised learning approaches. Pseudo labeling [12] uses two networks, called *teacher* and *student*. The teacher is trained on the gold-labeled portion  $D_g$  of the dataset to predict labels, so-called pseudo labels, for the unlabeled portion  $D_u$  of the dataset. This turns  $D_u$  into the pseudo labeled dataset  $D_p$ .  $D_g \cup D_p$  is then used to train the student in a supervised manner. A drawback of this approach is that it lacks a mechanism for correcting inaccurate pseudo

---

<sup>2</sup>Here,  $\alpha$  and  $\beta$  are the parameters of the Dirichlet prior distributions. They reflect a-priori beliefs on the document-topic distribution and topic-word distribution, respectively.

labels. To solve this problem, Pham et al. [19] invented the meta pseudo label approach which trains both models in an iterative fashion: when the student has been trained with pseudo-labeled data, its performance on  $D_g$  is used as an objective function to improve the ability of the teacher to create helpful pseudo labels.

Chen, Yang, and Yang [5] introduce a new text augmentation method called TMix that takes in two texts and interpolates them in their corresponding semantic hidden space. The idea behind TMix is to enforce a regularization on the model to make it behave linearly over the training data. Furthermore, the paper proposes a new semi-supervised learning method for text classification based on TMix called MixText: a text encoder (BERT) with TMix augmentation with a linear classifier on top. In each training iteration, it first predicts labels for  $D_u$  using the current model and then continues to train the model with  $D_p$  using TMix.

Liu et al. [13] introduce another model, FLiText, that uses a two-stage approach, where an inspirer network based on a language model is first trained using both labeled and unlabeled data. Subsequently, this network is distilled into a smaller model. In the second stage, FLiText uses output-based distillation, which relies on the output of the inspirer, and feature-based distillation, which uses the layer weights of the inspirer to guide the training of the target network while maintaining the parameters of the inspirer network.

Yang et al. [26] introduce prototype-guided pseudo-labeling (PGPL) for semi-supervised text classification. To mitigate bias caused by imbalanced datasets, they track the number of samples used from each class in the training history. For each class, they select the  $k$  nearest samples to the corresponding class prototype for the subsequent training iteration to ensure a balanced training process. Additionally, they employ prototypes for prototype-anchored contrasting, pushing samples toward their respective class prototypes and away from others.

## 2.3 Masking

Masking is a technique that can be used in training a language model (LM). It was originally applied to transformer architectures like BERT in the form of masked language modeling (MLM), to make them learn lexical and syntactic patterns from unlabeled text data. Recently, new masking tasks have been proposed to embed downstream task-related information into general pre-trained language models. Joshi et al. [10] propose SpanBERT that masks random contiguous spans of text instead of individual tokens to better represent and predict spans of text.

In this paper, we investigate whether pre-training based on masking can improve Cformer. For this, we first use LDA to find words in the dataset that carry topical information (independently of the specific topics directing the

later classification). We then mask these words in a pre-training phase using the whole-word-masking approach to make the language model more sensitive to topic information in the dataset. In contrast to the work of Gu et al. [8], our pre-training works in a completely unsupervised manner (using LDA).

## 2.4 Topic Coherence

Our method uses measures of topic coherence to find out which words to mask. Topic coherence measures take the  $N$  top words of a topic, compute confirmation values for individual words or subsets of words, and sum those confirmation values up. Röder, Both, and Hinneburg [20] proposed a unifying framework consisting of four parts to represent coherence measures. According to this framework, given a sequence of words  $w_1, w_2, \dots, w_N$  characterizing a topic (ordered by importance), a topic coherence measure considers pairs  $(W_{\text{target}}, W_{\text{supp}})$  of subsets of  $W = \{w_1, \dots, w_N\}$ . For each of the considered pairs, a confirmation value  $\kappa(W_{\text{target}}, W_{\text{supp}})$  is computed based on word probabilities. This value is intended to reflect how well  $W_{\text{supp}}$  supports  $W_{\text{target}}$ . These values are then accumulated into a single value representing the overall topic coherence. Topic coherence measures differ in (a) which pairs  $(W_{\text{target}}, W_{\text{supp}})$  are considered, (b) how, given certain word probabilities,  $\kappa(W_{\text{target}}, W_{\text{supp}})$  is defined, (c) how word probabilities are calculated, and (d) how the values are accumulated.

For CformerM, we use the two coherence measures  $C_{\text{UMass}}$  and  $C_v$  by Mimno et al. [16] and Röder, Both, and Hinneburg [20], respectively. They are briefly described below, using the framework of [20].

$C_{\text{UMass}}$  is given as follows.

- The pairs considered are all  $(W_{\text{target}}, W_{\text{supp}}) = (\{w_i\}, \{w_j\})$  (simplified to  $(w_i, w_j)$  below) with  $1 \leq j < i \leq N$ .
- The probability of  $w_i$  is defined to be the fraction of documents in which  $w_i$  occurs.
- For every pair  $(w_i, w_j)$ , the confirmation value is defined to be the logarithm of the conditional probability of  $w_i$  given  $w_j$ , slightly adjusted by a small term  $\epsilon$  to avoid taking the logarithm of zero:

$$\kappa(w_i, w_j) = \log \frac{P(w_i, w_j) + \epsilon}{P(w_j)}.$$

- The overall topic coherence is the average of all  $\kappa(w_i, w_j)$ ,  $1 \leq i < j \leq N$ .

$C_v$  is a more complex measure given as follows:

- The considered pairs are all  $(W_{\text{target}}, W_{\text{supp}}) = (\{w_i\}, W)$ , for  $1 \leq i \leq N$ ,
- The probability of  $w_i$  is defined to be the fraction of *virtual* documents in which  $w_i$  occurs. For this, a sliding window technique is used: viewing

every original document as a string of words, every substring of length  $\lambda$  is a virtual document. (We use the same  $\lambda$  as Röder et al., namely  $\lambda = 110$ .)

- An indirect measure is used to capture semantic relations: first, a direct measure  $\kappa_0$  is used to map every word  $w_i$  to a context vector  $v_i = (v_{i1}, \dots, v_{iN})$  by setting  $v_{ij} = \kappa_0(w_i, w_j)$ .<sup>3</sup> Moreover,  $v_W = \sum_{i=1}^N v_i$ . Now, the confirmation measure used by  $C_v$  is  $\kappa(w_i, W) = \text{sim}_{\text{cos}}(v_i, v_W)$ , where  $\text{sim}_{\text{cos}}$  denotes cosine similarity.
- Again, the overall topic coherence is the average of all  $\kappa(w_i, W)$ ,  $1 \leq i < j \leq N$ .

Since the indirect confirmation of  $C_v$  captures even semantic relations that may not materialize as direct confirmation,  $C_v$  is considered closer to human perception of coherence; cf. [20]). The downside of  $C_v$  is a much larger running time resulting from the consideration of a large number of virtual documents and the computation of the context vectors. However, we note also that  $C_v$  requires only  $\Omega(N)$  memory cells during segmentation, whereas  $C_{\text{UMass}}$  requires  $\Omega(N^2)$ . Therefore, even though  $C_{\text{UMass}}$  has a shorter running time, it may reach the limit of available memory if the number  $N$  of words describing each topic is large.

### 3 Cformer and CformerM

This section describes both Cformer, as proposed by Hatefi et al. [9], and its extension to CformerM, proposed in the current paper.

Cformer uses the iterative teacher-student architecture described in the introduction, which leverages pseudo labels created by the teacher to teach the student, improves the teacher by observing the resulting performance of the student, and iterates the process.

#### 3.1 Cformer

A schematic overview of Cformer can be seen in Figure 2. It shows the teacher  $T$  on the left and the student  $S$  on the right. The teacher is trained with the Un-supervised Data Augmentation (UDA) objective [25] and feedback consisting of the performance of the student on  $D_g$ . Thus, the UDA objective consists of supervised loss on  $D_g$  and consistency loss between  $D_u$  and an augmented version  $D_a$  of  $D_u$ . The dataset  $D_a$  can be built by applying a suitable text augmentation technique such as word substitution. In our implementation, we replace words with similar substitutes based on contextual word embeddings with probability 0.9.

---

<sup>3</sup>The precise confirmation measure  $\kappa_0$  used by  $C_v$  is not so important for the present discussion .

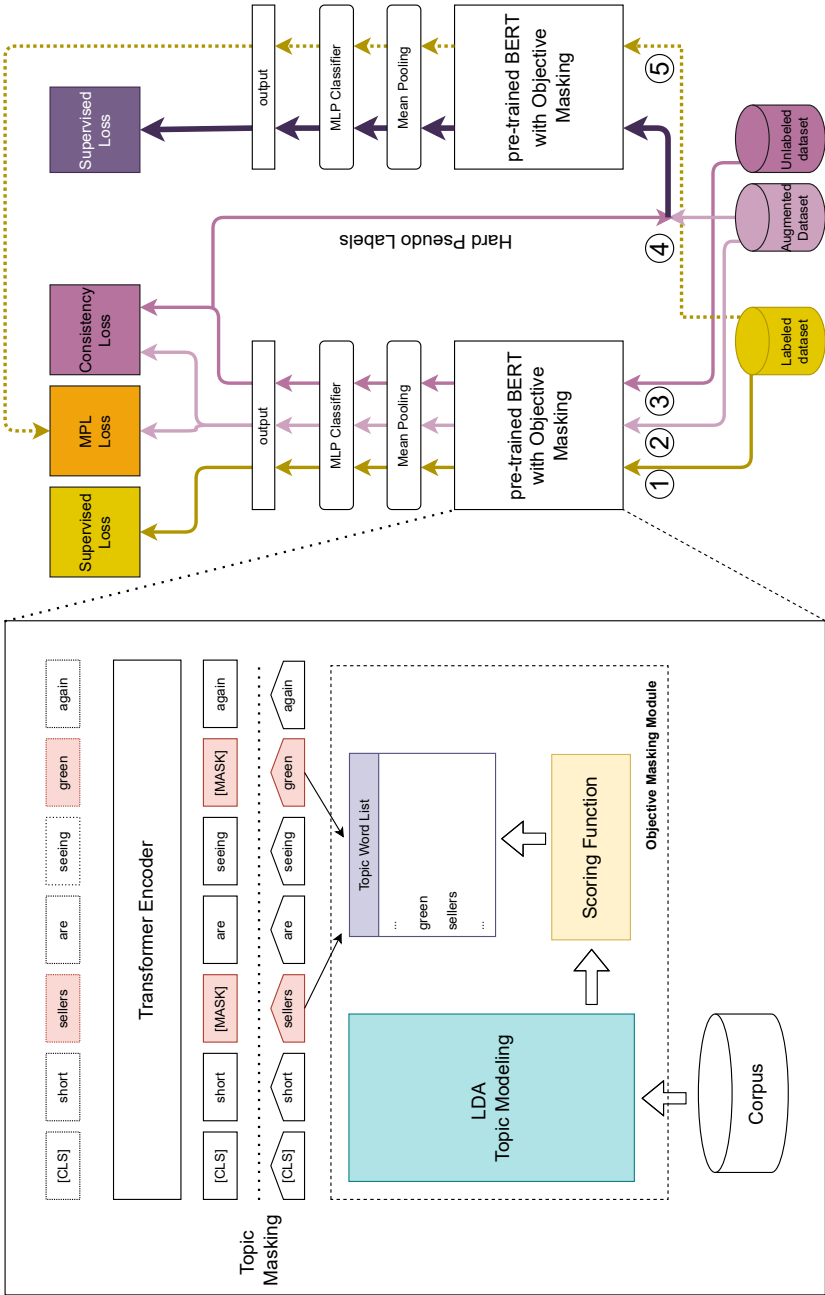


Figure 2: CformerM architecture: BERT encoders are pre-trained on the dataset via Objective Masking.

The training data  $D_p$  for the student is obtained from  $D_a$  by labeling it with labels computed by the teacher, so-called pseudo labels. Thus, in a slight deviation from the earlier, somewhat simplified description,  $D_p$  is based on  $D_a$  rather than  $D_u$  for the purpose of making the student invariant to noise by regularization. The student is then trained with supervised loss on  $D_p$ . As Figure 2 shows, both the student and the teacher consist of an encoder that maps documents to their distributed vector representations (using a transformer and a mean pooling module that computes the average of the transformer outputs in different positions) followed by a classifier.

To summarize, we have the following sets of training data and their derivatives:

- $D_g$  is the gold-labeled portion of the original training data. Below, we will use the notations  $x_g$  and  $\ell(x_g)$  to denote an arbitrary element of  $D_g$  and its label, respectively.
- $D_u$  is the unlabeled portion of the original training data. Elements of  $D_u$  will be denoted by  $D_u$ .
- $D_a$  is the augmented version of  $D_u$ , obtained using UDA. The element of  $D_a$  that is the augmented version of  $D_u$  will be denoted by  $x_a$ .
- $D_p$  is  $D_a$  enhanced by pseudo labels provided by the teacher. The element of  $D_p$  obtained from  $x_a \in D_a$  will be denoted by  $x_p$  and the corresponding pseudo label by  $\ell(x_p)$ .

In each training step, a batch  $D'_g \subseteq D_g$  of labeled data (track ① in Figure 2), and a batch  $D'_u \subseteq D_u$  of unlabeled data (track ③) and its augmented version  $D'_a \subseteq D_a$  (track ②) are fed into the teacher model. The cross entropy loss on the gold-labeled batch  $D'_g$  is computed as the mean cross entropy loss between labels  $\ell(x_g)$ , for  $x_g \in D'_g$ , and teacher predictions  $T(x_g; \theta_T)$ :

$$\text{Loss}_T(D'_g) = \sum_{x_g \in D'_g} \frac{\text{CrossEntropy}(\ell(x_g), T(x_g; \theta_T))}{|D'_g|}.$$

Unsupervised consistency loss is computed using  $D'_u$  and  $D'_a$ . The consistency loss constrains the model predictions to be less sensitive to input noise by demanding that the model assign the same labels to augmented samples  $x_a \in D'_a$  as to the predictions of the teacher for  $D'_u$ . To encourage the model to predict confident low-entropy labels for unlabeled data, we use a sharpening function applied to the soft predictions  $T(D_u; \theta_T)$  of the teacher for  $\ell(D_u)$ . For this, we apply the same sharpening function as Chen, Yang, and Yang [5] to the soft predictions  $T(D_u; \theta_T)$  of the teacher. Thus, given a temperature hyperparameter  $t$ , we let  $\ell_{\text{sharpen}}(D_u) = \text{sharpen}(T(D_u; \theta_T), t)$ , where  $\text{sharpen}(T(\ell, t)) = \frac{\sqrt[t]{\ell}}{\|\sqrt[t]{\ell}\|}$  for a soft label  $\ell$ . Here,  $\|v\|$  is the  $l_1$ -norm of a vector  $v$  and  $\sqrt[t]{\cdot}$  is applied componentwise to a vector. Thus, without any adjustments the unsupervised



loss of the teacher would be

$$\text{Loss}_T(D'_u) = \sum_{D_u \in D'_u} \frac{\text{CrossEntropy}(\ell_{\text{sharp}}(D_u), T(x_a; \theta_T))}{|D'_u|} .$$

However, we found it beneficial to omit samples that the current model is not confident about. Therefore, the consistency loss for each batch is computed only on samples whose maximal probability over all clusters is greater than an experimentally determined confidence threshold  $\beta$ . This gives rise to the unsupervised loss of the teacher which we denote by  $\text{Loss}_T(D'_u)$ .

Next, the student model learns from  $D_p$ . For this, the soft labels are turned into hard labels. Formally, for a vector  $\ell$  of soft labels, let  $\text{hard}(\ell) = \text{argmax}_i \ell_i$ . Then  $\text{Loss}_S(D'_p)$  is given by

$$\sum_{x_a \in D'_a} \frac{\text{CrossEntropy}(\text{hard}(T(D_u; \theta_T)), S(x_a; \theta_S))}{|D'_a|} .$$

Providing every  $x_a \in D_a$  with the (pseudo) label  $\text{hard}(T(D_u; \theta_T))$  turns  $D_a$  into  $D_p$  (cross-point ④ in Figure 2) which is fed to the student. The learning objective of the student is to minimize the cross entropy loss between the pseudo labels and its own predictions.

To complete the current iteration of the learning process, the teacher learns from the reward signal of how well the student performs on the labeled batch  $D'_g$  (signified by the dotted line ⑤ from student to teacher in Figure 2). Following Pham et al. [19], this loss is called *meta pseudo labels (MPL) loss*. Denoting the updated parameters of the student by  $\theta'_S$ , we get:

$$\text{Loss}_T^{\text{MPL}}(D'_g) = \nabla_{\theta_T} \sum_{x_g \in D_g} \frac{\text{CrossEntropy}(\ell(x_g), S(x_g; \theta'_S))}{|D'_g|} .$$

See [19] for more details.

Combining the three losses, we define the overall objective function of the teacher as follows.

$$\text{Loss}_T = \text{Loss}_T(D'_u) + \lambda_u * \text{Loss}_T(D'_u) + \text{Loss}_T^{\text{MPL}}(D'_g) ,$$

where  $\lambda_u$  is the contribution coefficient of the consistency loss. To prevent overfitting to  $D_g$ , we employ label smoothing [18] when computing the supervised losses  $\text{Loss}_T$  and  $\text{Loss}_S$ .

As the student only learns from pseudo-labeled data generated by the teacher, in a very final step after convergence we fine-tune it on  $D_g$  to improve its accuracy.

## 3.2 Pre-training by Objective Masking

We now describe how we employ masking to attune the pre-trained language model to the dataset. For this, we create an unsupervised topic model for it, using LDA. The steps are as follows:

1. We perform LDA on the given dataset to find  $K$  suitable topics. For each of the resulting topics  $T_i$  ( $1 \leq i \leq K$ ), this provides us with a sorted list  $L_i$  of its most indicative words.
2. From the total vocabulary, we now choose a subset  $W$  of  $N$  words by selecting the most relevant words of each topic. For this we reorder  $L_i$  according to a relevance measure (see below). Then if  $L_i = w_0^i, w_1^i, w_2^i, \dots, w_{K-1}^i$ , we let  $W = \bigcup_{i=1}^K \{w_0^i, \dots, w_{j_i}^i\}$  for suitable  $j_1, \dots, j_K$ . (How to choose  $K$  and  $j_1, \dots, j_K$  is discussed below.)
3. Afterwards, in the task-specific pre-training of the transformer model, we mask random occurrences of words on  $W$  from each document in such a way that 15% of the tokens of each document are masked. If a document does not contain sufficiently many words from  $W$  to reach the 15% limit, we mask additional random words.
4. Finally, we use the fine-tuned language model as a basis for the teacher and student of the architecture described in Section 3.1.

Algorithm 1 illustrates the overall algorithm underlying CformerM including the pre-training phase with objective masking.

## 3.3 Choosing the Number of Topics for Topic Modeling

For the LDA topic modeling, we need to provide the LDA algorithm with the number  $K$  of topics of the model to be created. To find a suitable  $K$ , we can use coherence measures such as  $C_{UMass}$  and  $C_v$ . Here, we apply  $C_v$ . Thus, given a dataset, we run the LDA algorithm on it for a range of candidate values for  $K$  and compare the resulting topic models with respect to  $C_v$ . We determine  $K$  from the coherence plot using the well-known heuristics of the “elbow method” (cf. Blashfield, Aldenderfer, and Morey [2]). This rule helps identify the point where the rate of increase in the coherence scores starts to level off, resulting in an “elbow” shape in the plot. In the degenerate case that the first point of the graph is the highest one, we choose that one for our experiments.

## 3.4 Choosing Word Lists for Masking

After choosing the most promising topic model for the dataset, we extract the  $N$  most relevant words for each topic and compile them into a list. The selection of  $N$  is guided by heuristics, and we subsequently assess the quality of each list

---

**Algorithm 1:** CformerM

---

**Input** :  $D_g, D_u$  – dataset (labeled and unlabeled)  
 $f_\theta$  – pre-trained language model (BERT)  
 $W_T$  – MLP head of teacher  
 $W_S$  – MLP head of student  
 $K$  – the number of topics for LDA  
 $MaxIter$  – maximum number of iterations  
 $B$  – training batch size  
 $t$  – temperature parameter for sharpening  
 $\beta$  – confidence threshold for unlabeled dataset  
 $\eta_T, \gamma_T$  – teacher’s learning rates  
 $\eta_S, \gamma_S$  – student’s learning rates

**Output** :  $(\theta_T^*, W_T^*)$  – learned weights of teacher  
 $(\theta_S^*, W_S^*)$  – learned weights of student

- 1  $TM_K \leftarrow$  run LDA on  $D_g \cup D_u$  to find  $K$  topics;
- 2  $TWL \leftarrow$  topic word list from  $TM_K$  ▷ see Section 3.4
- 3  $f_{\theta_T}, f_{\theta_S} \leftarrow$  pre-train  $f_\theta$  on  $D$  with MLM objective;  
using  $TWL$  for masking
  
- 4  $D_a \leftarrow$  do augmentation for  $D_u$ ;
- 5 **for**  $iter = 1$  **to**  $MaxIter$  **do**
- 6      $D'_g, D'_u \leftarrow$  batches of size  $B$  from  $D_g$  and  $D_u$ ;
- 7      $D'_a \leftarrow$  augmented version of  $D'_u$  from  $D_a$ ;
- 8      $L_T \leftarrow W_T(f_{\theta_T}(D'_u))$ ;
- 9     // Training teacher;
- 10    compute  $Loss_T(D'_g)$  ▷ teacher’s supervised loss
- 11     $L_T^{sharp} \leftarrow \frac{\sqrt[t]{L_T}}{\|\sqrt[t]{L_T}\|}$ ;
- 12    compute  $Loss_T(D'_u)$  ▷ teacher’s unsupervised loss
- 13     $L_T^{hard} \leftarrow \operatorname{argmax}_i L_T^i$ ;
- 14    // Training student;
- 15     $L_S \leftarrow W_S(f_{\theta_S}(D'_a))$ ;
- 16     $Loss_S \leftarrow$  cross-entropy-loss( $L_S, L_T^{hard}$ ) ▷ student’s supervised loss
- 17     $\theta_S \leftarrow \theta_S - \eta_S * Loss_S(\theta_S)$  ▷ Update  $\theta_S$
- 18     $W_S \leftarrow W_S - \gamma_S * Loss_S(W_S)$  ▷ Update  $W_S$
- 19    // Improving teacher;
- 20     $L'_S \leftarrow W_S(f_{\theta_S}(D'_g))$ ;
- 21     $Loss_T^{MPL} \leftarrow \nabla_{\theta_T}$  cross-entropy-loss( $L'_S, L_{gold}$ ) ▷ teacher’s MPL loss
- 22     $Loss_T \leftarrow Loss_T(D'_g) + Loss_T(D'_u) + Loss_T^{MPL}$ ;
- 23     $\theta_T \leftarrow \theta_T - \eta_T * Loss_T(\theta_T)$  ▷ Update  $\theta_T$
- 24     $W_T \leftarrow W_T - \gamma_T * Loss_T(W_T)$  ▷ Update  $W_T$
- 25 **end**
- 26 **return**  $(\theta_T^*, W_T^*), (\theta_S^*, W_S^*)$ ;

---

to identify the best one. The list quality is evaluated by computing the average coherence of all topics, considering their respective  $N$  most relevant words. To compute the topic coherence we use the coherence measures explained in Section 2.4.

For a given  $N$ , the actual selection of the  $N$  most relevant words from each topic uses the relevance measure introduced by Sievert and Shirley [23]. Let  $\phi_{kw}$  denote the probability of a word  $w$  to occur in a document of topic  $k \in \{1, \dots, K\}$ , and let  $p_w$  denote the marginal probability of  $w$  in the entire corpus. The *relevance* of word  $w$  to topic  $k$  given a weight parameter  $\lambda$  (where  $0 \leq \lambda \leq 1$ ) is defined as:

$$r(w, k|\lambda) = \lambda \log(\phi_{kw}) + (1 - \lambda) \log\left(\frac{\phi_{kw}}{p_w}\right) .$$

The  $N$  words picked from topic  $k \in \{1, \dots, K\}$  are the  $N$  most relevant words of topic  $k$  according to this measure. When  $\lambda$  is small, words that are highly associated with the topic and not very common in other topics will receive higher relevance scores. This makes the topics more distinct and easily interpretable. However, it might not consider words that are relevant but are more common across multiple topics. Conversely, when  $\lambda$  is higher, words that are more frequent in the topic but also have higher general frequency across topics will receive higher relevance scores. The advantage of using a larger  $\lambda$  is that it can capture more general aspects of the topic, helping to identify commonly occurring terms related to the topic across different documents.

## 4 Experiments and Analysis

Our experiments revolve around answering the following research questions:

- Q<sub>1</sub>:** How should one choose the parameters of the topic word selection method?
- Q<sub>2</sub>:** What is the overall performance of Cformer in comparison to the baselines?
- Q<sub>3</sub>:** How is the overall performance of Cformer and CformerM affected if the BERT model of the student is replaced by DistilBERT, yielding Distil-Cformer and Distil-CformerM, respectively?
- Q<sub>4</sub>:** Does the objective masking used in CformerM indeed improve Cformer?
- Q<sub>5</sub>:** What factors could impact the effectiveness of the proposed objective masking?
- Q<sub>6</sub>:** Are there any benefits to utilizing topic modeling for generating topic word lists as opposed to simpler methods such as TF-IDF?
- Q<sub>7</sub>:** Does the proposed masking approach affect the reliability and interpretability of Cformer?
- Q<sub>8</sub>:** How does the performance of the models vary with different values of hyper-parameters such as the number of GPUs and the batch size?

**Q<sub>9</sub>:** How well does the proposed model perform in a zero-shot setting in comparison to the baselines?

## 4.1 Experimental Setup

### Datasets

We perform experiments with two English text classification benchmark datasets and a private dataset which is in Swedish. The dataset statistics and splits are given in Table 1.

Dataset	Classes	Documents	Average #s	Max #s	Average #w	Max #w	Vocabulary	Unlabeled	Dev	Test
Yahoo! Answers	10	140 000	7.0	158	112.0	2 001	267 610	5 000	5 000	6 000
AG News	4	120 000	1.7	20	36.2	212	94,443	5 000	2 000	1 900
Bonnier News	17	78 757	22.23	382	362.12	2 252	538 655	-	-	-

Table 1: Dataset statistics and dataset split. The numbers of sentences and words are denoted by #s and #w, respectively. The number of unlabeled, dev, and test data items is given in terms of the number of data items per class. For Yahoo! Answers, the reported statistics concerns the subset used in the experiments.

**Yahoo! Answers** For Yahoo! Answers [4], we obtain the text to be classified by concatenating the question (title and content) and the best answer. Since the original training set is very large, we only use a randomly chosen subset of 10% of its documents (i.e., 140 000 documents). To be comparable with our baselines, we randomly sample the same amount of data as in MixText [5] from the training subset for our unlabeled and validation sets, and use the original Yahoo! Answers test set. We use the training subset to create word lists for objective masking.

**AG News** Of AG News [27], we only use the news content (without titles). Again, we randomly sample the same amount of data as in MixText [5] from the original training set for our unlabeled and validation sets and use the original AG News test set. We use the original training set to create word lists for objective masking.

**Bonnier News** The Bonnier News<sup>4</sup> is a private dataset comprised of 127 161 articles in Swedish published on 35 different Bonnier News brands during the period February 2020 to February 2021. Its documents are labeled according to the Category Tree for Swedish Local News that has been developed and used by local newsrooms within Bonnier News<sup>5</sup>. This category tree is based

<sup>4</sup><https://www.bonniernews.se/>; while we cannot make this dataset publicly available, researchers who want to reproduce our results or use it for their own work may contact [datasets@bonniernews.se](mailto:datasets@bonniernews.se) to gain access.

<sup>5</sup>[github.com/mittmedia/swedish-local-news-categories](https://github.com/mittmedia/swedish-local-news-categories)

on the IPTC Media Topics<sup>6</sup>. The dataset includes 545 categories distributed across four hierarchy levels. The dataset is highly imbalanced, with the most frequent category occurring 30 531 times and the least frequent one occurring 102 times. Furthermore, the number of categories articles are labeled with varies greatly. The maximum number of categories used to label an article is 46 and the minimum number is one, with 5.1 categories on average. For our experiments, we only consider top-level labels and use only documents labeled with a unique label. The resulting dataset consists of 78 757 samples in 17 classes. We split these samples into test and training datasets according to a 1:4 ratio and randomly select 20% of the training examples for validation. Since the dataset is imbalanced, instead of choosing an absolute number of examples per class to create the labeled and unlabeled datasets, we split samples of each class in the training data into labeled and unlabeled parts using 1%, 10%, and 30% as labeled documents, respectively. Table 2 displays the classes in the dataset along with the corresponding number of examples in the training, validation, and test sets.

## Baselines and Experimental Settings

To verify the effectiveness of Cformer and CformerM, we compare them with several baselines:

**UDA** [25] uses the consistency loss between unlabeled and augmented data as a training signal to improve classification.

**MixText** [5] augments training samples by interpolating in the hidden space.

**FLiText** [13] applies pseudo-labeling and distillation to a lightweight setting using convolution networks.

**PGPL** [26] proposes prototype-guided pseudo-labeling to avoid bias from imbalanced data and presents prototype-anchored contrasting to make clear boundaries between classes.

**BERT/DistilBERT** consists of a BERT/DistilBERT encoder followed by a two-layer MLP serving as the classifier, similar to the Cformer classification layer. This classifier is exclusively trained with labeled data. Moreover, we have pre-trained versions of the BERT/DistilBERT classifier, utilizing pre-trained BERT models tailored for use in the CformerM classifier.

We implement MixText and UDA using the code<sup>7</sup> provided by Chen, Yang, and Yang [5] with the hyperparameters specified in the original paper and the code repository. Similarly, for FLiText, we run the available code<sup>8</sup> on AG News and Yahoo! Answers, using the hyperparameters reported in the original paper and its code repository. We implement the BERT/DistilBERT baselines ourselves.

---

<sup>6</sup>[iptc.org/standards/media-topics/](https://iptc.org/standards/media-topics/) is a comprehensive standard taxonomy for categorizing news text.

<sup>7</sup><https://github.com/SALT-NLP/MixText>

<sup>8</sup><https://github.com/valuesimplex/FLiText>

<b>Class name</b>	<b>Training</b>	<b>Test</b>	<b>Valid.</b>
Olyckor & katastrofer (Accidents & disasters)	2656	830	664
Brott & straff (Crime & punishment)	6587	2059	1647
Personligt (Personal)	1503	470	376
Vetenskap & teknologi (Science & technology)	235	74	59
Samhälle & välfärd (Society & welfare)	3198	1000	800
Religion & tro (Religion & faith)	185	57	46
Ekonomi, näringsliv & finans (Economy, business & finance)	5222	1632	1305
Politik (Politics)	2423	757	606
Sport (Sports)	16746	5233	4187
Livsstil & fritid (Lifestyle & leisure)	1922	601	480
Miljö (Environment)	911	285	228
Väder (Weather)	454	142	114
Hälsa & sjukvård (Health & medical care)	1886	589	471
Konflikter, krig & terrorism (Conflicts, war & terrorism)	70	21	17
Kultur & nöje (Culture & entertainment)	4829	1509	1207
Skola & utbildning (School & education)	1201	375	300
Arbetsmarknad (Labor market)	376	118	94

Table 2: Classes and sample distribution in Bonnier News across Training, Validation, and Test sections

For all baselines, we report the average results obtained from five different runs, just as we do for different versions of the Cformer model. It is worth noting that all baseline models are trained using the same amount of labeled, unlabeled, and validation data as used in our Cformer experiments. Additionally, we apply the same text augmentation approach whenever augmented data is needed for the baseline models.

Our experiments evaluate two versions of Cformer and their CformerM coun-

terparts:

**Cformer:** The student and the teacher models both employ the BERT language model.

**Distil-Cformer:** The teacher is as in Cformer but the student uses DistilBERT as its language model.

**CformerM and Distil-CformerM:** These are the CformerM versions of the two previous models.

For the English datasets, we use *bert-base-uncased* and *distilbert-base-uncased* as the BERT and DistilBERT language models, respectively. For Swedish, we use the pre-trained BERT model available in the hugging face repository<sup>9</sup>, namely *KB/bert-base-swedish-cased* as the BERT language model and *distilbert-base-multilingual-cased* as the DistilBERT language model. The Swedish models are referred to as Cformer (SE-SE) and Distil-Cformer (SE-multi) to indicate the types of language model used by teachers and students, respectively. To have a fair comparison between Cformer and Distil-Cformer, we also train a Cformer called Cformer (SE-multi) with the Swedish BERT as the teacher’s encoder and the *bert-base-multilingual-cased* model for the student’s encoder. The teacher and student models in the English Cformer have 109.58 million parameters each, and the student in the English Distil-Cformer has 66.46 million parameters. In the Swedish version of Cformer, the teacher model has 124.79 million parameters and the student model has 124.79 and 177.95 million parameters when we use *KB/bert-base-swedish-cased* and *bert-base-multilingual-cased*, respectively, as its language model. The student in the Swedish Distil-Cformer has 134.83 million parameters.

We employ average pooling over the output of the encoder to aggregate word embeddings into document embeddings, and a two-layer MLP with a 128 hidden size and hyperbolic tangent as its activation function (the same as in Mix-Text) to predict the labels. For the input of the model, documents are truncated to their first 256 tokens. To generate augmented data from unlabeled data, we use the library *nlpaug*<sup>10</sup>. We substitute text words based on contextual word embeddings with a probability of 0.9. All models are trained with the AdamW optimizer [14]. We train the models for 7 000 steps (including 50 warm-up steps) and evaluate them every 500 steps. To avoid overfitting, we use early stopping with delta 0.005 and patience 4. We set the learning rate of the transformer and classifier components in both teacher and student to 1e-5 and 1e-3, respectively. After training both the teacher and the student models, we fine-tune the student on the labeled data set using the AdamW optimizer with a fixed learning rate of 5e-6 and a batch size of 32, running for 10 epochs. The temperature  $T$  for sharpening is set to 0.5 for Yahoo! Answers and Bonnier News and to 0.3 for AG News. The confidence threshold  $\beta$  is set to 0.9 and the label smoothing parameter is 0.15 for all datasets. For the contribution

---

<sup>9</sup><https://huggingface.co/KB/bert-base-swedish-cased>

<sup>10</sup><https://github.com/makcedward/nlpaug>



Dataset	Number of GPUs	Labeled Batch Size per GPU	Unlabeled Batch Size per GPU	Occupied Memory per GPU
AG News	3	4	8	15.4GB
Yahoo! Answers	2	8	16	23.6GB
Bonnier News	2	4	8	15.4GB

Table 3: The number of GPUs and local batch sizes across three datasets.

coefficient of unsupervised loss in the teacher loss function  $\lambda_u$ , we start from 0 and increase it linearly for 6 000 steps until it reaches its highest value of 1.

We use the PyTorch Distributed package for distributed GPU training on 3 V100 GPUs with 32GB memory. However, it is not mandatory to train Cformer on such large memory GPUs. With small batch sizes, the model can be trained using regular GPUs. Table 3 shows the number of GPUs and local batch sizes<sup>11</sup> we used for training the models across three datasets.

## 4.2 Choosing Suitable Parameters for the Selection of Topic Words ( $Q_1$ )

This section reports on our experiments with the topic word list selection method to determine the appropriate values for its parameters across different datasets.

**Choosing the Number of Topics using LDA** To create the LDA topic model as described in Section 3.3, we use the Gensim library<sup>12</sup>. Before feeding the documents into the model, a preprocessing step is performed, involving the elimination of stop words, removal of some of the most frequent words in the dataset, and retention of only nouns, adjectives, and verbs. For Bonnier News, we also perform lemmatization using the spaCy library<sup>13</sup>. Swedish has considerably more inflections than English. For example, the definite article in Swedish is mostly expressed by a suffix on the noun, and agreement rules stipulate that adjectives are inflected depending on the gender and number of the nouns they refer to, for instance: *en fin bil* (a beautiful car), *ett fint hus* (a beautiful house), and *finna bilar* (beautiful cars). Hence, performing lemmatization during preprocessing is justified for Swedish.

Now, our goal is to determine the number of topics that maximizes the coherence score of the resulting LDA model. To determine this number, we used the following reasoning. As we work in a semi-supervised setting, we know a lower bound on the number of topics in each dataset, namely the number of distinct labels occurring in the labeled portion of the dataset. If this number is  $m$ , we

<sup>11</sup>The local batch size is the batch size per GPU.

<sup>12</sup><https://radimrehurek.com/gensim/>

<sup>13</sup><https://spacy.io/models/sv>

compute coherence scores of  $k$  LDA models, each dividing the dataset into  $im$  topics for  $i = 1, \dots, k$ . The number  $k$  should be sufficiently large to ensure that the highest coherence score is likely to be included. For the three datasets considered in this paper, we use  $k = 19$  (Yahoo! Answers),  $k = 12$  (AG News), and  $k = 32$ . We then choose  $i_{\max}k$  as the number of topics for the LDA model to be used, where  $i_{\max}$  is the value of  $i$  resulting in the highest coherence score. Figure 3 shows the resulting coherence diagrams for the three datasets. For Yahoo! Answers, we actually chose to set  $m = 5$  rather than  $m = 10$  in order to confirm that smaller numbers of topics than the actual number would not result in higher coherence scores. Figure 3 shows the coherence diagrams for the three datasets. As one may expect, the result would have been  $i_{\max} = 1$  in all three cases if we had chosen  $m = 10$  in the case of Yahoo! Answers. Based on the curves shown in Figure 3, we choose 10 topics for Yahoo! Answers, 4 for AG News, and 17 for Bonnier News as presumably good numbers of topics for our LDA models.

These experimental results indicate that it may usually be safe to skip the coherence calculations, instead simply choosing the number of topics that coincides with the number  $m$  of distinct labels found in the labeled portion of the data. We thus recommend to do this unless there is reason to suspect that the labeled portion of the data does not cover all of the classes.

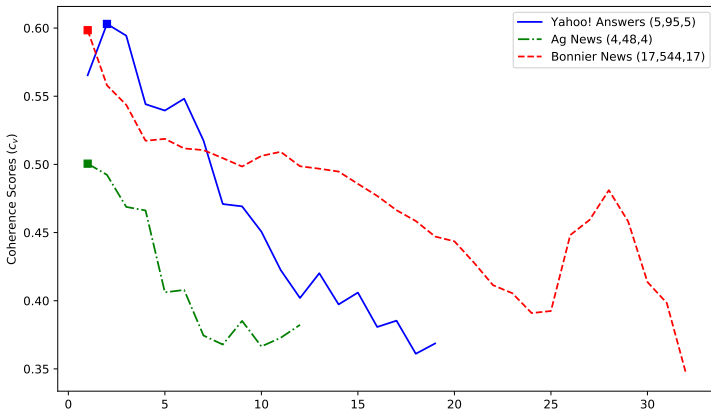


Figure 3: The coherence diagram over a range of values of the number of topics for (a) Yahoo! Answers (b) AG News, (c) Bonnier News. Note that values on the  $y$ -axis range between 0.35 and 0.6. For each dataset, the range of values explored (resulting from the respective choice of  $m$  and  $k$ ) is denoted within parentheses in the legend. For instance, “Yahoo! Answers (5, 95, 5)” means that the number of topics ranges from 5 to 95 with a step size of 5, in this case,  $m = 5$  and  $k = 19$ .

**Relevance-Based Word Lists** For each dataset, we choose relevance-based word lists for masking as explained in Section 3.4. Thus, we create several candidate lists with varying values of  $N$ , comprising the most informative words from the vocabulary. Table 4 presents the coherence scores ( $C_v$  and  $C_{UMass}$ ) of these lists for different datasets. For each dataset, we create two sets of lists, one corresponding to  $\lambda = 0.7$  and the other to a lower  $\lambda$ . From each set, we select one list. The choice of the lower  $\lambda$  value was made based on dataset examination, aided by topic model visualization tools<sup>14</sup>.

A larger value of  $C_v$  and a higher absolute value of  $C_{UMass}$  indicate stronger coherence. To determine the appropriate value of  $N$ , we identify the point that marks the end of the rapid growth of topic coherence. The rationale behind this approach is to select a point after which the marginal increase in coherence score does not justify further enlargement of  $N$ . If we were to pick a larger  $N$  beyond this point, less significant words would be added to the list, diluting the valuable information and leading to a reduction in the objective masking capacity.

To investigate the effectiveness of relevance-based word lists for objective masking and explore the relationship between coherence scores and classifier performance, we pre-train the BERT model on Yahoo! Answers employing topic word lists corresponding to  $\lambda = 0.2$  and different values of  $N$  for objective masking and then compute the accuracy of the resulting BERT classifier for three semi-supervised cases (10, 200, 2 500 labeled example per cluster). Table 5 illustrates the results of these experiments. From the table, we see that the BERT classifier performs the best when  $N = 1000$ . In addition, the results suggest that when we increase  $N$ , the model accuracy starts increasing, reaches a peak, and then starts dropping again. This seems to be a reasonable behavior: by increasing  $N$ , we first add informative words to the list (considering that in each topic, words are sorted with respect to relevance), but eventually, less significant words are also included, thus diluting the useful information.

The results of CformerM and Distil-CformerM using relevance-based word lists for objective masking for Yahoo! Answers and AG News are available in Table 6. Also, the results of CformerM using relevance-based word lists for objective masking for Bonnier News are reported in Table 7.

### 4.3 Result Analysis (Q<sub>2</sub> & Q<sub>3</sub> & Q<sub>4</sub>)

Table 6 and Table 7 present the performance of different versions of Cformer and CformerM in comparison to each other and against baseline models on three datasets.

**Overall performance of Cformer** Overall, Cformer performs better than current SoTA models across all considered datasets. Cformer consistently ex-

<sup>14</sup><https://pympi.org/project/pyLDavis/>

Dataset	$\lambda$	$N$	$C_v$	$C_{UMass}$
Yahoo! Answers	0.2	500	0.68	-13.34
	<b>0.2</b>	<b>1000</b>	<b>0.79</b>	<b>-15.75</b>
	0.2	1500	0.80	-15.55
	0.2	2000	0.81	-15.46
	0.7	500	0.54	-9.11
	0.7	1000	0.65	-11.95
	<b>0.7</b>	<b>1500</b>	<b>0.72</b>	<b>-13.48</b>
	0.7	2000	0.77	-14.50
	0.7	2500	0.80	-15.56
	0.7	3000	0.83	-16.23
AG News	0.2	500	0.61	-13.78
	0.2	700	0.69	-15.31
	<b>0.2</b>	<b>900</b>	<b>0.74</b>	<b>-16.31</b>
	0.2	1100	0.77	-16.94
	0.2	1300	0.80	-17.44
	0.7	500	0.47	-10.03
	0.7	1000	0.63	-13.30
	0.7	1500	0.72	-15.32
	<b>0.7</b>	<b>2000</b>	<b>0.78</b>	<b>-16.58</b>
	0.7	2500	0.82	-17.43
0.7	3000	0.85	-18.00	
Bonnier News	0.1	500	0.79	-16.84
	<b>0.1</b>	<b>1000</b>	<b>0.83</b>	<b>-17.19</b>
	0.1	1500	0.83	-16.96
	0.1	2000	0.83	-16.63
	0.7	500	0.54	-9.70
	0.7	1000	0.61	-11.58
	<b>0.7</b>	<b>1500</b>	<b>0.68</b>	<b>-12.95</b>
	0.7	2000	0.73	-13.83
0.7	2500	0.76	-14.52	

Table 4: Coherence scores ( $C_v$  and  $C_{UMass}$ ) of different candidate lists corresponding to different numbers of words per topic ( $N$ ) for AG News, Yahoo! Answers, and Bonnier News.  $\lambda$  is the parameter of the relevance score used for choosing the  $N$  most relevant words of the topics. Selected values are emphasized in boldface letters.

hibits significant performance improvements over UDA, ranging from 0.4% to 2.9% across all experiments conducted on AG News and Yahoo! Answers. A noteworthy aspect is that the teacher in Cformer is trained with the UDA

Dataset	$N$	Labeled examples per class		
		10	20	2500
Yahoo! Answers	500	62.4	71.4	74.3
	<b>1000</b>	<b>62.9</b>	<b>71.5</b>	<b>74.4</b>
	1500	62.6	71.5	74.4
	2000	62.4	71.4	74.3

Table 5: Performance of the BERT classifier on Yahoo! Answers using relevance-based word lists with a  $\lambda$  value of 0.2 and varying values of  $N$  for pre-training BERT via objective masking.

objective. This difference in performance strongly suggests that the knowledge distillation from student to teacher within the teacher-student architecture does indeed have a significant effect. Moreover, across all experiments and datasets, Cformer demonstrates superior accuracy compared to both Mix-Text and BERT/DistilBERT. It outperforms FLiText by a significant margin as well, which may be thanks to the larger size of the student model compared to FLiText’s target network which is designed to be lightweight. Last but not least, Cformer outperforms PGPL on Yahoo! Answers and AG News in all cases except the 10-shot case of Yahoo! Answers. We note here that there is no code available for the PGPL baseline. Hence, we use the results reported in the original research paper to avoid re-implementation bias. However, their setup is slightly different than the setup of our experiments as they, e.g., use back-translation for data augmentation.

**Performance of Cformer versus Distil-Cformer** To assess the usefulness of Cformer for limited environments we developed Distil-Cformer and evaluated its performance. Overall, Distil-Cformer performs well in comparison to Cformer even though its student model is considerably smaller. In particular, the performance gap between Distil-Cformer and Cformer on AG News and Yahoo! Answers is less than 0.4% in most cases.

For a fair comparison between Cformer and Distil-Cformer on Bonnier News, one should only compare Distil-Cformer (SE-multi) with Cformer (SE-multi), using a multilingual language model for the student in both cases. However, among all of the available options the unilingual Cformer (SE-SE) is clearly the most reasonable one considering its size and performance.

**The effect of objective masking on classification performance** The results in Tables 6 and 7 indicate that pre-trained BERT with objective masking consistently outperforms the BERT classifier without pre-training. In particular, in the 10-shot case, pre-trained BERT shows a significant improvement

Model	Labeled examples per class			Model	Labeled examples per class		
	10	200	2500		10	200	2500
BERT	60.2	69.6	73.5	BERT	81.9	88.6	91.5
BERT (random)	61.6	71.3	74.2	BERT (random)	83.8	89.0	91.7
BERT (relevance-0.2)	62.9	71.5	74.4	BERT (relevance-0.2)	84.5	89.2	91.9
BERT (relevance-0.7)	62.7	71.5	74.3	BERT (relevance-0.7)	84.3	89.1	91.9
DistilBERT	60.4	69.9	73.2	DistilBERT	83.4	88.4	91.1
DistilBERT (random)	61.9	71.0	73.7	DistilBERT (random)	84.3	89.0	91.5
DistilBERT (relevance-0.2)	63.3	71.5	73.9	DistilBERT (relevance-0.2)	84.6	89.3	91.6
DistilBERT (relevance-0.7)	63.1	71.4	73.9	DistilBERT (relevance-0.7)	84.4	89.2	91.6
UDA <sup>♣</sup> (2019)	61.7	69.7	73.5	UDA <sup>♣</sup> (2019)	86.3	89.0	91.5
MixText <sup>♣</sup> (2020)	64.1	70.6	73.7	MixText <sup>♣</sup> (2020)	86.9	88.9	91.3
FLiText <sup>♣</sup> (2021)	45.9	65.5	68.5	FLiText <sup>♣</sup> (2021)	77.9	87.6	89.1
PGPL (2023)	<b>67.4</b>	70.7	—	PGPL (2023)	87.8	89.2	—
Cformer	64.6	71.9	74.7	Cformer	88.1	90.0	91.9
CformerM (random)	65.1	72.7	75.0	CformerM (random)	88.0	90.1	<b>92.2</b>
CformerM (relevance-0.2)	66.3	<b>72.9</b>	<b>75.1</b>	CformerM (relevance-0.2)	88.4	90.1	<b>92.2</b>
CformerM (relevance-0.7)	66.1	72.8	75.0	CformerM (relevance-0.7)	<b>88.5</b>	<b>90.2</b>	<b>92.2</b>
Distil-Cformer	64.2	71.7	74.5	Distil-Cformer	87.2	90.0	91.8
Distil-CformerM (random)	64.5	72.5	74.7	Distil-CformerM (random)	87.5	89.9	92.0
Distil-CformerM (relevance-0.2)	65.3	72.8	75.0	Distil-CformerM (relevance-0.2)	88.2	90.1	92.1
Distil-CformerM (relevance-0.7)	65.0	72.7	74.9	Distil-CformerM (relevance-0.7)	88.3	90.1	<b>92.2</b>

Table 6: Comparison of the test accuracy of Cformer and CformerM with the baselines on Yahoo! Answers and AG News. The results are the average accuracy of 5 different runs with different random seeds. <sup>♣</sup> means “run by us”.

Dataset	Model	Proportion of labeled examples per class		
		0.01	0.1	0.3
Bonnie News	BERT-SE	81.7	85.4	86.7
	BERT-SE (random)	82.1	85.6	87.0
	BERT-SE (relevance-0.1)	82.2	85.8	87.2
	BERT-SE (relevance-0.7)	82.0	85.7	87.1
	MixText <sup>♣</sup> (2020)	82.5	85.7	87.0
	Cformer (SE-SE)	83.3	86.5	87.5
Cformer (SE-multi)	81.8	84.8	85.8	
CformerM (SE-SE) (random)	<b>83.8</b>	86.6	87.6	
CformerM (SE-SE) (relevance-0.1)	<b>83.8</b>	<b>86.7</b>	<b>87.7</b>	
CformerM (SE-SE) (relevance-0.7)	83.6	86.6	87.6	
Distil-Cformer (SE-multi)	81.4	84.0	84.8	

Table 7: Comparison of the test accuracy of Cformer and CformerM on Bonnie News with baselines. The results are the average accuracy of 5 different runs with different random seeds. Since the purpose of Distil-Cformer is to have a model for limited environments and Distil-Cformer (SE-multi) is larger than Cformer (SE-SE) (and still cannot outperform it), Distil-CformerM (SE-multi) is not selected for further experiments. <sup>♣</sup> means “run by us”.

of 2.7% for Yahoo! Answers, 2.6% for AG News, and 0.5% for Bonnier News in terms of accuracy. Additionally, when compared to pre-trained BERT with random masking, the objective masking approach yields a performance improvement of 0.2% to 1.3% across all datasets. This observation highlights that the superiority of pre-trained BERT with objective masking is not solely attributed to domain adaptation but also to the effectiveness of the topic-based masking approach. Looking at Table 8, which shows some examples of how the two masking strategies choose words for masking, this seems intuitively reasonable: the words chosen according to the LDA model are obviously semantically more important.

Next, we compare Cformer and CformerM. The performance of CformerM is superior to that of Cformer on all datasets in all cases. Notably, in the 10-shot case, CformerM exhibits an absolute increase in accuracy of 1.7% for Yahoo! Answers, 0.4% for AG News, and 0.5% for Bonnier News when compared to Cformer. CformerM with objective masking also outperforms CformerM with random masking. In most cases the advantage is significant, the exception being Bonnier News, on which the difference is small. Similar findings were observed for pre-trained DistilBERT with objective masking and Distil-CformerM, where objective masking gave improved results. However, the impact of objective masking varies depending on the dataset and the complexity of the classification task.

Tables 6 and 7 show that objective masking is particularly advantageous when there is a scarcity of supervised information. We trained BERT/DistilBERT classifiers using labeled data only, in contrast to Cformer, which incorporates both labeled and pseudo-labeled data. As a result, we observe more substantial improvements with objective masking for the BERT/DistilBERT classifiers compared to their Cformer/Distil-Cformer counterparts. In three distinct scenarios, the most significant enhancement is observed for the 10-labeled case in both BERT and Cformer. Additionally, as the amount of labeled data increases, the distinction between objective masking and random masking becomes less pronounced for both BERT and Cformer. We conjecture that this can be attributed to the increasing influence of fine-tuning during model training.

The optimal choice of the relevance parameter  $\lambda$  (see Section 3.4) for making topic word lists depends on the characteristics of the dataset. In complex datasets with numerous topics and overlapping word distributions between many of them, a smaller  $\lambda$  value proves more beneficial. On the other hand, for simpler classification tasks, such as those with fewer classes to be recognized and well-separated classes without overlaps, a larger  $\lambda$  performs better. As evidenced by the results in Tables 6 and 7 for Yahoo! Answers and Bonnier News, lambda values of 0.2 and 0.1 are more effective than 0.7, while for AG News, a lambda value of 0.7 yields better results. The visualization of the topic models for these datasets via pyLDavis<sup>15</sup> shows that Yahoo! Answers and Bonnier News both contain several overlapping topics meaning there are a

---

<sup>15</sup><https://pypi.org/project/pyLDavis/>

Dataset	Sentence (tokenized)	Masking policy	Selected words
Yahoo! Answers	'[CLS]', 'do', 'u', 'think', 'that', 'golf', 'is', 'the', 'most', 'boring', 'high', 'paid', 'sport', 'to', 'watch', 'on', 'tv', '?', 'did', 'you', 'ever', 'watch', 'curling', '?', '[SEP]'	threshold	'tv', 'high', 'boring', 'watch'
		relevance	'tv', 'high', 'boring', 'watch'
		random	'on', 'you', 'that', 'paid'
Yahoo! Answers	'[CLS]', 'what', 'does', 'e', '=', 'mc', '##2', 'mean', '?', 'it', 'is', 'an', 'equation', 'by', 'albert', 'einstein', 'showing', 'that', 'energy', 'and', 'mass', 'are', 'interchange', '##able', '.', 'e', 'is', 'energy', 'm', 'is', 'mass', 'c', 'is', 'the', 'speed', 'of', 'light', 'thus', 'energy', 'equals', 'the', 'amount', 'of', 'mass', 'multiplied', 'by', 'the', 'speed', 'of', 'light', 'squared', '.', '[SEP]'	threshold	'einstein', 'squared', 'speed', 'mass', 'speed', 'albert', 'mass', 'multiplied'
		relevance	'einstein', 'squared', 'speed', 'mass', 'speed', 'albert', 'mass', 'multiplied'
		random	'the', 'of', 'amount', 'c', 'e', 'is', 'is', 'does'
AG News	'[CLS]', 'wages', 'rose', 'faster', 'than', 'expected', 'in', 'the', 'june', '.', 'august', 'period', 'but', 'analysts', 'say', 'the', 'increases', 'are', 'still', 'not', 'high', 'enough', 'to', 'cause', 'inflation', 'worries', 'at', 'the', 'bank', 'of', 'england', '.', '[SEP]'	threshold	'bank', 'increases', 'inflation', 'august', 'england'
		relevance	'bank', 'increases', 'inflation', 'england', 'worries'
		random	'wages', 'high', 'analysts', 'than', 'bank'
AG News	'[CLS]', 'italian', 'anti', '-', 'mafia', 'magistrates', 'ordered', 'the', 'arrest', 'of', '65', 'people', 'as', 'part', 'of', 'a', 'massive', 'police', 'sw', '##oop', 'in', 'naples', 'early', 'today', 'in', 'a', 'bid', 'to', 'staunch', 'the', 'blood', '##lett', '##ing', 'in', 'a', 'turf', 'war', 'which', 'has', 'killed', 'more', 'than', '120', 'people', '.', 'interior', 'minister', 'giuseppe', 'pisa', '##nu', 'said', '.', '[SEP]'	threshold	'people', 'bid', 'part', 'magistrates', 'mafia', 'massive', 'police', 'war'
		relevance	'people', 'bid', 'part', 'massive', 'police', 'war', 'said', 'italian'
		random	'said', 'which', '65', 'a', '120', 'today', 'naples', 'people'

Table 8: Examples of how different masking policies choose words for masking in pre-training



lot of common words between these topics. Thus, these datasets favor a low  $\lambda$  value which aligns with our previous explanation. In the case of BERT, smaller values of  $\lambda$  consistently prove to be superior. We conjecture that with limited labeled data, a smaller  $\lambda$  helps the model distinguish between different classes by learning the topic-specific words.

Moreover, the quality of data used for pre-training significantly influences the effectiveness of objective masking. For instance, objective masking demonstrates greater efficacy on Yahoo! Answers compared to AG News. Specifically, for Yahoo! Answers, the accuracy of CformerM improved by 1.7% compared to the accuracy of Cformer in the case of 10 labeled examples per class, whereas for AG News, the improvement was 0.4%. Based on some additional experiments that we performed, we conclude that this variation can most likely be attributed to the fact that AG News consists of short texts and is smaller overall, comprising only 3.6M words in total, whereas Yahoo! Answers consists of longer texts with 12.2M words in total. Consequently, AG News offers considerably less context for BERT during pre-training than Yahoo! Answers does. We tested this hypothesis by running experiments on the text of AG News documents only, stripping away the titles. As expected, the performance dropped. Table 9 presents the results of these experiments.

<b>Pre-training data</b>	<b>CformerM (random)</b>	<b>CformerM (relev.-0.2)</b>	<b>CformerM (relev.-0.7)</b>
Text	87.5	87.8	87.9
Text+title	88.0	88.4	88.5

Table 9: Comparison of the accuracy of different CformerM versions with encoders pre-trained under two different settings on AG News for the 10-labeled case. We either use only the text or the concatenation of the text and the document title. The average accuracy of five runs with distinct random seeds is reported.

As it can be seen in Table 2, Bonnier News exhibits a significant class imbalance, with a majority of its documents belonging to the Sports category. This imbalance poses a considerable challenge for LDA to generate topics that accurately align with the actual categories. Table 10 presents the topics obtained from LDA for Bonnier News. As depicted in the table, certain topics lack coherence and cannot be accurately matched with a specific class in the dataset. Additionally, more than one topic is associated with the ‘Sport’ category, while no matches were found for the topics ‘Personligt’, ‘Vetenskap & teknologi’, ‘Religion & tro’, ‘Miljö’, ‘Konflikter, krig & terrorism’, and ‘Arbetsmarknad’. As a consequence, the topic word lists generated by LDA do not adequately capture all the diverse topic words in the dataset, resulting in limited effectiveness of objective masking compared to random masking. Moreover, as noted in the

Swedish BERT paper [15], the training data for Swedish BERT heavily leans towards newspaper text, leading to a similarity between the data distribution of Swedish BERT training data and Bonnier News. This similarity, in turn, restricts the effectiveness of pre-training.

#### 4.4 The Impact of Objective Masking on Domain-Specific Tasks (Q<sub>5</sub>)

When the dataset deviates significantly from the BERT training data and incorporates domain-specific information, such as medical documents, the distinction between the effect of objective masking and random masking on the classification performance is more pronounced than if the classification task relies mostly on general language understanding. In the latter case, BERT’s pre-trained knowledge from a large general corpus may be sufficient to handle the task.

To gain deeper insights into this aspect, we compare the impact of objective masking and random masking on the classification performance of the BERT and CformerM classifiers on the Medical Abstracts dataset<sup>16</sup> [22]. Table 11 shows the class distribution within this dataset.

We create an LDA topic model for the dataset with 5 topics and make two relevance-based topic word lists: one with  $\lambda = 0.2$ , comprising 500 words for each of the 5 topics, and another with  $\lambda = 0.7$  consisting of 700 words per topic. For the experiments, we allocated 0.1 of the training data as the validation set. In three distinct settings, we divided the remaining data into labeled and unlabeled datasets, using proportions of 0.01, 0.1, and 0.3 for the labeled dataset, while the rest was assigned to the unlabeled dataset. Additionally, we applied the same augmentation method used for other datasets to generate an augmented version of the dataset. The results of these experiments are displayed in Table 12.

Comparing the results in Table 12 with those in Table 6, we observe that objective masking outperforms random masking by a larger margin for the Medical Abstracts dataset in comparison to the other datasets, particularly in the BERT setting. Specifically, in the case with the minimal labeled data, the BERT (relevance) and CformerM (relevance) models demonstrate superior performance over BERT (random) and CformerM (random), achieving 3.7% and 1.9% increase in accuracy respectively. In contrast, the corresponding accuracy improvements on Yahoo! Answers and AG News are (1.3%, 1.2%) and (0.7%, 0.5%), respectively.

We note also that the topics in Medical Abstracts are fairly well separated and  $\lambda = 0.7$  is preferable in all cases except the 0.01 labeled case of the BERT classifier. In this case, the supervised data is extremely limited, so using more

---

<sup>16</sup><https://www.kaggle.com/datasets/chaitanyakck/medical-text>

Topic words	Best match
kund, butik, företag, bolag, konkurs, företagsnamn, produkt, köpare, fabrik, säte	Ekonomi, näringsliv & finans
trafikverk, trafik, hastighet, cyklist, cykelväg, järna, tågtrafik, resenär, järnväg, cykelbana	Samhälle & välfärd
patient, minskat, regering, region, parti, procent, politik, vård, politisk, införa	Politik
vatten, utställning, snö, sjö, sol, vädr, träd, regn, väder, vind	Väder
sång, klubb, spelare, träning, trupp, träna, kontrakt, sportchef, stars, nyförvärv	Sport
faktafel, hofors, flygplats, orientering, arbetsplatsolycka, ockelbo, flygplatse, amnå, euro, lot, auktion, aktieägg	-
bolhäs, söderhamn, edsby, carlström, hudik, aida, edsbyn, vänersborg, falbygd, johannesson	-
vaccin, ericsson, vaccinerat, dos, habo, vaccination, vaccinering, indycar, rosenqvist, wolley	Hälsa & sjukvård
elev, förskola, skola, lokaler, rektor, bygge, undervisning, högskola, lärare, grundskola	Skola & utbildning
herr, bortalag, tabell, ikk, leksand, brynäs, köping, innebandy, målgörare, idf	Sport
polis, räddningstjänst, larm, larma, brande, brand, ambulans, olycka, pressalesperson, förar	Olyckor & katastrofer
kvinnor, åtala, tingsrätt, döma, fängelse, åklagar, sexuell, brott, våld, våldtäkt	Brott & straff
liv, familj, bok, mamma, pappa, läsare, äta, son, språk, roman	Livsstil & fritid
tävling, lopp, final, tävlar, set, tävlingarna, medalj, brons, åkare, sprint	Sport
domare, gif, vsk, boll, degerfors, hörna, brage, kjäll, skutskar, kvarnsed	Sport
musik, scen, band, föreställning, konsert, låt, artist, kvrka, sång, festival	Kultur & nöje
varg, hus, katt, arkitektur, kvadratnet, poststadsrätt, revir, hyresgäst, skyddsjak, hyresrätt	-

Table 10: Top 10 words (based on relevance score with  $\lambda = 0.1$ ) for the topics identified by LDA in Bonnier News and the best matches we found for them from the real classes in the dataset. Some topics lack sufficient coherence to be matched to specific classes in the dataset.

Class name	Training	Test	Total
Neoplasms	2530	633	3163
Digestive system diseases	1195	299	1494
Nervous system diseases	1540	385	1925
Cardiovascular diseases	2441	610	3051
General pathological conditions	3844	961	4805
Total	11550	2888	14438

Table 11: Class distribution within Medical Abstracts

Model	0.01	0.1	0.3
BERT	49.2	59.0	61.0
BERT (random)	51.0	60.5	61.9
BERT (relevance-0.2)	54.7	61.4	62.1
BERT (relevance-0.7)	53.8	61.8	62.2
Cformer	53.7	60.8	62.8
CformerM (random)	55.2	62.2	63.8
CformerM (relevance-0.2)	56.4	<b>62.6</b>	<b>64.2</b>
CformerM (relevance-0.7)	<b>57.1</b>	<b>62.6</b>	<b>64.2</b>

Table 12: Comparison of test accuracy for different variations of BERT and Cformer pretrained with new domain-specific topic word lists on Medical Abstracts dataset. The average accuracy of five runs with distinct random seeds is reported.

specific words for masking could potentially enable the model to concentrate on the distinct characteristics of each topic and enhance its ability to classify instances belonging to those specific topics.

## 4.5 Comparison of LDA and TF-IDF for Topic Word Extraction ( $Q_6$ )

To compare the LDA-based methods for generating topic word lists with simpler techniques that do not rely on topic models, we present alternative versions of our models that use TF-IDF to identify topic words within the corpus. Specifically, we sort the words based on their average TF-IDF scores across all documents and select the top words. The preprocessing step remains consistent with the LDA-based method: we remove general stop words and a few of the most frequent words in the dataset, keeping only nouns, verbs, and adjectives. After preprocessing, we calculate the average TF-IDF score for all words in the vocabulary and choose the top  $N$  words. The value of  $N$  matches the length

Model		Labeled examples per class			Model		Labeled examples per class		
		10	200	2500			10	200	2500
Yahoo! Answers	BERT (relevance-0.2)	62.9	71.5	74.4	AG News	BERT (relevance-0.2)	84.5	89.2	91.9
	BERT (relevance-0.7)	62.7	71.5	74.3		BERT (relevance-0.7)	84.3	89.1	91.9
	BERT (tf-idf)	62.5	71.5	74.2		BERT (tf-idf)	84.3	89.1	91.7
Yahoo! Answers	CformerM (relevance-0.2)	<b>66.3</b>	<b>72.9</b>	<b>75.1</b>	AG News	CformerM (relevance-0.2)	88.4	90.1	<b>92.2</b>
	CformerM (relevance-0.7)	66.1	72.8	75.0		CformerM (relevance-0.7)	<b>88.5</b>	<b>90.2</b>	<b>92.2</b>
	CformerM (tf-idf)	65.4	72.8	75.0		CformerM (tf-idf)	88.4	<b>90.2</b>	92.1

Table 13: Comparison of LDA and TF-IDF for Yahoo! Answers and AG News

Dataset	Model	Labeled examples per class		
		0.01	0.1	0.3
Bonnier News	BERT-SE (relevance-0.1)	82.2	85.8	87.2
	BERT-SE (relevance-0.7)	82.0	85.7	87.1
	BERT-SE (tf-idf)	82.0	85.6	87.0
	CformerM (SE-SE) (relevance-0.1)	<b>83.8</b>	<b>86.7</b>	<b>87.7</b>
	CformerM (SE-SE) (relevance-0.7)	83.6	86.6	87.6
	CformerM (SE-SE) (tf-idf)	83.5	86.6	87.6

Table 14: Comparison of LDA and TF-IDF for Bonnier News

of the relevance-based list that has been proven to be more effective for each dataset (e.g., the list with  $\lambda = 0.2$  for Yahoo! Answers, the list with  $\lambda = 0.7$  for AG News, and the list with  $\lambda = 0.1$  for Bonnier News). Tables 13 and 14 show the results of these comparisons.

As can be seen in Tables 13 and 14, masking with topic words is slightly less effective if the selection is based on TF-IDF instead of LDA. As could be expected given the relatively good performance of even random masking, the difference is small if there is a sufficient amount of labeled data, but if the labeled data is severely limited the effect is more pronounced. We hypothesize that this superiority can be attributed to the fact that the topic model considers the underlying structure of the dataset, whereas TF-IDF relies on individual documents. Nevertheless, the TF-IDF approach does identify a certain number of true topic words, making it a reasonable compromise when facing resource constraints such as time and computational power. However, it should also be noted that the LDA-based method offers flexibility in choosing between highly topic-specific words and more general ones, catering to the specific needs of the analysis, while the TF-IDF method offers less control over the generated lists.

## 4.6 Effect of Objective Masking on Reliability and Interpretability (Q<sub>7</sub>)

Moon et al. [17] illustrated that fine-tuning a text classifier using masked keyword regularization helps it consider context rather than solely relying on certain keywords, which leads to improved out-of-distribution detection and cross-domain generalization. Inspired by this research, we conducted a qualitative study to examine how our proposed masking strategy impacts the reliability of CformerM. To do so, we performed case studies on Yahoo! Answers and AG News, using CformerM (relevance). We selected samples from the test set of the datasets that were correctly classified by CformerM (relevance) and arranged them in order of their predicted probability of belonging to the correct class. Afterwards, we chose 20 samples classified with lower probabilities and examined cases that were misclassified by both Cformer and CformerM (random) to identify the factors that contribute to the accurate classification by CformerM (relevance). Tables 15 and 16 show some of these examples.

In the upper example of Table 15, “gun” was the most commonly attended word across all models. However, CformerM (relevance) correctly predicted the class “Sports” by considering the context and specifically the word “safe”. In contrast, CformerM (random) predicted “Business” by taking into account the words “gun” and “steel”. Furthermore, the prediction of the class “Politics & Government” by Cformer indicates a disregard for the context. This trend is consistent across models in the second example as well, where CformerM (relevance) accurately associates “digital technology” with the “Computers & Internet” category. Moreover, Table 16 shows examples of AG News where CformerM (relevance) predicts the correct topic for the document by considering not only the keywords but also their contextual information. These observations indicate that pre-training with objective masking on domain-specific datasets teaches the model the contexts in which a given keyword may appear, thus enabling it to account for contextual factors during classification.

It is also worth noting that both CformerM variations exhibit higher interpretability due to their attention to more relevant and informative words, as compared to Cformer.

## 4.7 Effect of Batch-size and Number of GPUs on the Performance (Q<sub>8</sub>)

Since the performance of a model depends on the training data, it varies when the amount of training samples used to train the model is dynamic. In case of a teacher-student model, the dynamic change of the number of training examples affects the information sharing between teacher and student (including the student feedback for meta pseudo-labeling approaches). Therefore, the dependency of the performance on the number of training examples may be suspected to be even stronger in such a model. To study this, we experiment

Class	Model	Text	Prediction
Sports	Cformer	where can i find a che ##ep gun safe made of steel ? i don t think you should use the words cheap and safe in the same sentence it sounds dangerous	Politics & Government
	CformerM (random)	where can i find a che ##ep gun safe made of steel ? i don t think you should use the words cheap and safe in the same sentence it sounds dangerous	Business & Finance
	CformerM (relevance)	where can i find a che ##ep gun safe made of steel ? i don t think you should use the words cheap and safe in the same sentence it sounds dangerous	Sports
Computers & Internet	Cformer	what are the differences between digital technology and information technology	Business & Finance
	CformerM (random)	what are the differences between digital technology and information technology	Science & Mathematics
	CformerM (relevance)	what are the differences between digital technology and information technology	Computers & Internet

Table 15: Visualization of selected samples from Yahoo! Answers. The color saturation indicates the average attention to the word from other words in the sentence.

Class	Model	Text	Prediction
Business	Cformer	how will companies and investors fare if the storm spawn ##s moderate damage	Sci/Tech
	CformerM (random)	how will companies and investors fare if the storm spawn ##s moderate damage	Sci/Tech
	CformerM (relevance)	how will companies and investors fare if the storm spawn ##s moderate damage	Business
Business	Cformer	in recent years hundreds of multinational companies have set up research laboratories in china	Sci/Tech
	CformerM (random)	in recent years hundreds of multinational companies have set up research laboratories in china	Sci/Tech
	CformerM (relevance)	in recent years hundreds of multinational companies have set up research laboratories in china	Business

Table 16: Visualization of selected samples from the AG News. The color saturation indicates the average attention to the word from other words in the sentence.

with the batch size and the number of GPUs being used in two main scenarios, namely 10 labeled samples and 200 labeled samples for the labeled set of AG News. We base our analysis on AG News for two reasons. First, it is a publicly available dataset, meaning that the results of the analysis can be replicated by others. Second, Section 4.3 concluded that CformerM performs better on long-text datasets. As seen in Table 1, AG News consists of shorter documents than the other two datasets. Hence, any differences revealed by comparing the average performance of Cformer and CformerM on AG News can be expected to carry over to other cases.

Tables 17–20 show the results of these experiments. In Tables 17 and 18, the *global* batch size is the product of the local batch size and the number of GPUs used for the experiment. The reported accuracies are averages over multiple experiments with varying numbers of GPUs and local batch sizes, keeping the global batch size constant as shown in the Table. Tables 17 and 18 show a rather distinctive behavior: in the 10-sample experiments, the smallest batch sizes favor Cformer, but at a certain point CformerM starts to perform better. For all 200-sample experiments, CformerM turns out to be at least as good as Cformer for all batch sizes.

In addition, Tables 19 and 20 show the performance of Cformer and CformerM with respect to different numbers of GPUs used for running the experiments. The reported accuracy for each number is the average accuracy of multiple experiments with different local batch sizes. Again, in most cases CformerM outperforms Cformer; in 200-sample experiments, CformerM always performs better than Cformer but in 10-sample experiments, CformerM overcomes Cformer only when 3 GPUs are used.

We also examined the effect of the local batch size on Cformer performance. For this, we ran Cformer on different numbers of GPUs using different numbers of local batch sizes. Figure 4 illustrates the results of these runs. We observe that when the number of training samples grows, the batch size should be modestly increased (but not too much). In the case of AG News, Figure 4 suggests that for Cformer a batch size of 4 gives stable performance for the 10-sample case, while a batch size of 8 gives better performance for the 200-sample case. Moreover, Figure 5 shows the effect of batch size and number of GPUs on CformerM performance in 10-sample case. Similar to Cformer, it benefits from a smaller batch size, and CformerM will remain stable (and perform well) with 3 GPUs.

Overall, based on these experiments, we draw the following conclusions:

- When the size of the labeled training data is sufficient (a few hundred samples), CformerM performs better under any choice of the number of GPUs and batch size.
- With a more limited number of supervised data samples, CformerM prefers larger batch sizes than Cformer. We hypothesize that this is because CformerM has seen the data before in the pre-training and is thus



Global B-Size	Accuracy	
	Cformer	CformerM
4	<b>88.30</b>	88.00
6	<b>88.80</b>	88.40
8	<b>88.80</b>	88.65
12	<b>88.73</b>	88.23
16	<b>88.70</b>	88.40
18	<b>88.70</b>	<b>88.70</b>
24	88.75	<b>88.85</b>
36	88.40	<b>88.60</b>

Table 17: The accuracy of Cformer and CformerM w.r.t. global batch size with 10 labeled samples on AG News

Global B-Size	Accuracy	
	Cformer	CformerM
4	89.70	<b>90.20</b>
6	<b>90.20</b>	<b>90.20</b>
8	90.00	<b>90.15</b>
12	90.20	<b>90.30</b>
16	90.10	<b>90.30</b>
18	90.20	<b>90.40</b>
24	90.20	<b>90.35</b>
36	90.00	<b>90.40</b>

Table 18: The accuracy of Cformer and CformerM w.r.t global batch size with 200 labeled samples on AG News

more stable towards the dataset. Hence, it can be expected to require more coarse-grained information to update itself.

- When the number of training samples grows, a moderate increase in batch size is beneficial for both Cformer and CformerM.

## 4.8 Effect of Topic Word Masking in Zero-Shot Evaluation ( $Q_9$ )

Practical applications of topic modeling may aim at topics that are compound or ultra fine grained, that change over time, etc. One approach that can support such application scenarios is to develop models capable of classifying unseen categories without any training instances, so-called zero-shot classification. The rise of pre-trained language models made the idea of providing task descriptions

GPUs	Accuracy	
	Cformer	CformerM
1	<b>88.65</b>	88.22
2	<b>88.87</b>	88.52
3	88.40	<b>88.70</b>

Table 19: The accuracy of Cformer and CformerM w.r.t. number of GPUs with 10 labeled samples on AG News

GPUs	Accuracy	
	Cformer	CformerM
1	90.03	<b>90.27</b>
2	89.95	<b>90.22</b>
3	90.13	<b>90.35</b>

Table 20: The accuracy of Cformer and CformerM w.r.t. number of GPUs with 200 labeled samples on AG News

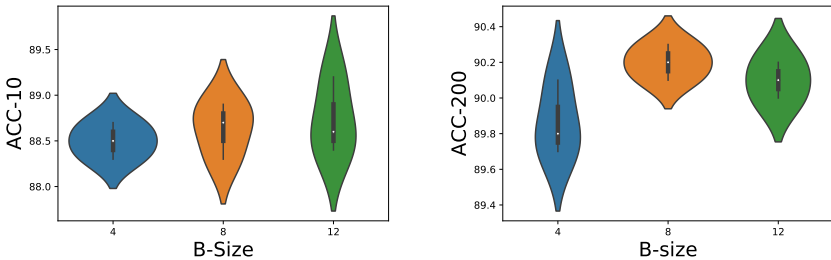


Figure 4: Effect of batch-size per GPU on Cformer performance. A local batch size of 4 gives stable performance for the 10-sample case (left), while a batch size of 8 gives better performance for the 200-sample case (right). This suggests that when more training data is available, a larger batch size yields better performance.

for neural architectures in zero-shot experiments feasible. Therefore, it is a valid question to ask if CformerM performs better in zero-shot classification than Cformer.

To study this question, we

1. split Yahoo! Answers into Yahoo! Answers<sub>(A)</sub> (society, health, computer, business, relationship) and Yahoo! Answers<sub>(B)</sub> (science, education, sports, entertainment, politics),

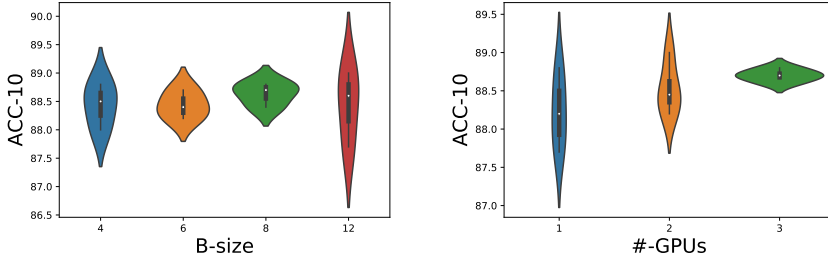


Figure 5: Effect of batch size and number of GPUs on CformerM performance. Similar to Cformer, the 10-sample case benefits from a smaller batch size, and CformerM will remain stable (and perform well) with 3 GPUs.

2. pre-train *bert-base-uncased* and *distilbert-base-uncased* language models on Yahoo! Answers<sub>(A)</sub> using objective masking (see Section 3.2),
3. train Distil-CformerM on Yahoo! Answers<sub>(A)</sub> using the language models of the previous step as the text encoders in the teacher and the student models (we use 200 labeled examples and 5 000 unlabeled examples per class for training),
4. train Distil-Cformer on Yahoo! Answers<sub>(A)</sub>, and
5. compare the performance of the two resulting student models on the Yahoo! Answers<sub>(B)</sub> test set in a zero-shot evaluation setting.

For zero-shot evaluation, we use the Pattern Exploiting Training (PET) approach proposed by Schick and Schütze [21] for semi-supervised text classification. Initially, PET trains several language models with labeled data using different input patterns. The ensemble of these language models is then used to predict pseudo labels for unlabeled data. Finally, a standard classifier is trained based on the pseudo-labeled data. In our experiments, we use language models for prediction without training them. So, we combine two language models of the same type with two different input patterns and use them to classify samples in the test set of Yahoo! Answers<sub>(B)</sub>. Table 21 shows the cloze style patterns we use for the inputs.

In four different experiments we initialize PET language models with

1. *distilbert-base-uncased* (DistilBERT)
2. *distilbert-base-uncased* pre-trained on Yahoo! Answers<sub>(A)</sub> with topic word masking (DistilBERTM)
3. the student model in Distil-Cformer trained on Yahoo! Answers<sub>(A)</sub> (Student<sub>Distil-Cformer</sub>)
4. the student model in Distil-CformerM trained on Yahoo! Answers<sub>(A)</sub> (Student<sub>Distil-CformerM</sub>)

Table 22 shows the accuracy of PET on the Yahoo! Answers<sub>(B)</sub> test set in

these experiments. DistilBERTM and Student<sub>Distil-CformerM</sub> outperform DistilBERT and Student<sub>Distil-Cformer</sub>, respectively, which shows that the proposed pre-training of the language model with objective masking can increase the ability of the language model to recognize examples of classes that have not been seen before. Also, the superiority of the student models over the corresponding original models confirms that a knowledge transfer from the teacher to the student happens.

<b>P1:</b>	$\langle \text{Mask} \rangle$ : Text
<b>P2:</b>	[Category: $\langle \text{Mask} \rangle$ ] Text

Table 21: Cloze style patterns for zero-shot evaluation. The task of the transformer is to replace  $\langle \text{Mask} \rangle$  with a suitable category label.

Language Model	Accuracy
DistilBERT	0.5699 $\pm$ 0.020
DistilBERTM	0.6039 $\pm$ 0.003
Student <sub>Distil-Cformer</sub>	0.5807 $\pm$ 0.050
Student <sub>Distil-CformerM</sub>	<b>0.6310 <math>\pm</math>0.050</b>

Table 22: Performance of PET with 4 different initializations in the zero-shot evaluation setting.

## 5 Conclusions

In this paper, we proposed CformerM, an extension of a semi-supervised text classification approach Cformer by Hatefi et al. [9]. CformerM uses objective masking in an unsupervised pre-training phase to improve Cformer. The idea is to use LDA topic modeling for finding lists of words that are likely to carry topic information. We adopt relevance scores to select topic words from the LDA topic model. By adjusting the parameter  $\lambda$ , we can control the specificity of the chosen words. A lower lambda value will prioritize words that are very specific to the topics, while a higher value will include words that are more frequent and may appear in multiple topics. This flexibility allows us to tailor the selection process based on the specific requirements of the dataset. We studied the performance of Cformer and CformerM via extensive experiments over three public datasets (Yahoo! Answers, AG News, and Medical Abstracts datasets) in English and one private dataset (Bonniere News) of news articles in Swedish. While the latter is not publicly available, interested researchers may request access by sending an e-mail to [datasets@bonniernews.se](mailto:datasets@bonniernews.se) in order to reproduce our results or perform their own experiments.

Our experimental findings demonstrate that CformerM outperforms Cformer, BERT classifiers (both pre-trained and non-pre-trained), and SoTA baselines in most cases over all datasets. However, the impact of objective masking on classification accuracy is more pronounced when the amount of supervised data for classification is limited. Moreover, CformerM outperforms the variant obtained by using random masking instead of objective masking. However, the effectiveness of objective masking compared to random masking is influenced by dataset characteristics, such as document length, deviation from BERT training data, and the amount of data available for pre-training. For instance, the use of objective masking proves to be particularly effective for the Medical Abstracts dataset, which we believe to be caused by the different characteristics of this dataset compared to the training data used to develop BERT. Additionally, objective masking performs better on Yahoo! Answers than on AG News, an effect we attribute to fact that short texts like those of AG News provide less context for BERT during the pre-training phase of CformerM.

Furthermore, our qualitative analysis indicates that pre-training with objective masking can help the language model learn in which contexts – and thus in which topics – certain keywords are likely to appear, enabling it to account for contextual factors when classifying documents. This improves the reliability and interpretability of the model and leads to more accurate classification results.

## References

- [1] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. “On the Surprising Behavior of Distance Metrics in High Dimensional Space”. In: *International conference on database theory*. Springer. 2001, pp. 420–434.
- [2] Roger K. Blashfield, Mark S. Aldenderfer, and Leslie C. Morey. *Cluster Analysis*. Ed. by Paruchuri R. Krishnaiah and Laveen N. Kanal. Vol. 44. Quantitative Applications in the Social Sciences. SAGE University, 1982.
- [3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. “Latent Dirichlet Allocation”. In: *Journal of machine Learning research* 3 (2003), pp. 993–1022. ISSN: 1532-4435.
- [4] Ming-Wei Chang et al. “Importance of Semantic Representation: Dataless Classification”. In: *AAAI*. Vol. 2. 2008, pp. 830–835.
- [5] Jiaao Chen, Zichao Yang, and Diyi Yang. “Mixtext: Linguistically-informed Interpolation of Hidden Space for Semi-supervised Text Classification”. In: *arXiv preprint arXiv:2004.12239* (2020).
- [6] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

- [7] Maarten R. Grootendorst. “BERTopic: Neural topic modeling with a class-based TF-IDF procedure”. In: *ArXiv abs/2203.05794* (2022).
- [8] Yuxian Gu et al. “Train No Evil: Selective Masking for Task-Guided Pre-Training”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 6966–6974. DOI: 10.18653/v1/2020.emnlp-main.566.
- [9] Arezoo Hatefi et al. “Cformer: Semi-Supervised Text Clustering Based on Pseudo Labeling”. In: *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management*. Ed. by Gianluca Demartini et al. ACM, 2021, pp. 3078–3082. DOI: 10.1145/3459637.3482073.
- [10] Mandar Joshi et al. “Spanbert: Improving Pre-training by Representing and Predicting Spans”. In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 64–77.
- [11] Samuli Laine and Timo Aila. “Temporal Ensembling for Semi-supervised Learning”. In: *arXiv preprint arXiv:1610.02242* (2016).
- [12] Dong-Hyun Lee. “Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks”. In: *ICML 2013 Workshop on Challenges in Representation Learning (WREPL)* (2013).
- [13] Chen Liu et al. “FLiText: A Faster and Lighter Semi-Supervised Text Classification with Convolution Networks”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 2481–2491. DOI: 10.18653/v1/2021.emnlp-main.192. URL: <https://aclanthology.org/2021.emnlp-main.192>.
- [14] Ilya Loshchilov and Frank Hutter. “Decoupled Weight Decay Regularization”. In: *arXiv preprint arXiv:1711.05101* (2017).
- [15] Martin Malmsten, Love Börjeson, and Chris Haffenden. “Playing with Words at the National Library of Sweden – Making a Swedish BERT”. In: (2020). arXiv: 2007.01658 [cs.CL].
- [16] David Mimno et al. “Optimizing Semantic Coherence in Topic Models”. In: *Proceedings of the 2011 conference on empirical methods in natural language processing*. 2011, pp. 262–272.
- [17] Seung Jun Moon et al. “MASKER: Masked Keyword Regularization for Reliable Text Classification”. In: *AAAI Conference on Artificial Intelligence*. 2020.
- [18] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. “When does Label Smoothing Help?” In: *arXiv preprint arXiv:1906.02629* (2019).
- [19] Hieu Pham et al. “Meta Pseudo Labels”. In: *arXiv preprint arXiv:2003.10580* (2020).

- [20] Michael Röder, Andreas Both, and Alexander Hinneburg. “Exploring the Space of Topic Coherence Measures”. In: *Proceedings of the eighth ACM international conference on Web search and data mining*. 2015, pp. 399–408.
- [21] Timo Schick and Hinrich Schütze. “Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 255–269.
- [22] Tim Schopf, Daniel Braun, and Florian Matthes. “Evaluating Unsupervised Text Classification: Zero-Shot and Similarity-Based Approaches”. English. In: *Proceedings of the 2022 6th International Conference on Natural Language Processing and Information Retrieval*. 6th International Conference on Natural Language Processing and Information Retrieval, NLPPIR 2022, NLPPIR 2022 ; Conference date: 16-12-2022 Through 18-12-2022. Association for Computing Machinery, Dec. 2022, pp. 6–15. ISBN: 9781450397629. DOI: 10.1145/3582768.3582795.
- [23] Carson Sievert and Kenneth Shirley. “LDAvis: A Method for Visualizing and Interpreting Topics”. In: *Proceedings of the workshop on interactive language learning, visualization, and interfaces*. 2014, pp. 63–70.
- [24] Antti Tarvainen and Harri Valpola. “Mean Teachers are Better Role Models: Weight-averaged Consistency Targets Improve Semi-supervised Deep Learning Results”. In: *arXiv preprint arXiv:1703.01780* (2017).
- [25] Qizhe Xie et al. “Unsupervised Data Augmentation for Consistency Training”. In: *arXiv preprint arXiv:1904.12848* (2019).
- [26] Weiyi Yang et al. “Prototype-Guided Pseudo Labeling for Semi-Supervised Text Classification”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 16369–16382. URL: <https://aclanthology.org/2023.acl-long.904>.
- [27] Xiang Zhang, Junbo Zhao, and Yann LeCun. “Character-level Convolutional Networks for Text Classification”. In: *arXiv preprint arXiv:1509.01626* (2015).
- [28] Zihan Zhang et al. “Is Neural Topic Modelling Better than Clustering? An Empirical Study on Clustering with Contextual Embeddings for Topics”. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 3886–3893. DOI: 10.18653/v1/2022.naacl-main.285. URL: <https://aclanthology.org/2022.naacl-main.285>.

---

**ADCluster: Adaptive Deep Clustering for Unsupervised Learning from Unlabeled Documents**

Arezoo Hatefi, Xuan-Son Vu, Monowar Bhuyan, and Frank Drewes

*In Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP), pp. 68-77, Association for Computational Linguistics, 2023.*





# ADCluster: Adaptive Deep Clustering for Unsupervised Learning from Unlabeled Documents\*

Arezoo Hatefi, Xuan-Son Vu, Monowar Bhuyan, Frank Drewes

*Department of Computing Science, Umeå University, Umeå, Sweden*

*arezoo@cs.umu.se, sonvx@cs.umu.se, monowar@cs.umu.se, drewes@cs.umu.se*

**Abstract:** We introduce ADCluster, a deep document clustering approach based on language models that is trained to adapt to the clustering task. This adaptability is achieved through an iterative process where K-Means clustering is applied to the dataset, followed by iteratively training a deep classifier with generated pseudo-labels – an approach referred to as *inner adaptation*. The model is also able to adapt to changes in the data as new documents are added to the document collection. The latter type of adaptation, *outer adaptation*, is obtained by resuming the inner adaptation when a new chunk of documents has arrived. We explore two outer adaptation strategies, namely accumulative adaptation (training is resumed on the accumulated set of all documents) and non-accumulative adaptation (training is resumed using only the new chunk of data). We show that ADCluster outperforms established document clustering techniques on medium and long-text documents by a large margin. Additionally, our approach outperforms well-established baseline methods under both the accumulative and non-accumulative outer adaptation scenarios.

## 1 Introduction

Document clustering is the task of arranging large volumes of unlabeled documents into clusters according to some notion of similarity. A particularly common goal is to discover the most common topics in a given collection of text documents and to assign each document to its corresponding cluster. Given the ever-growing number of documents available online and the fact that manually structuring them is impossible, there are countless applications of document clustering techniques.

---

\*The paper has been published in the *Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP 2023)*, Association for Computational Linguistics, and has been re-typeset to match the thesis style.

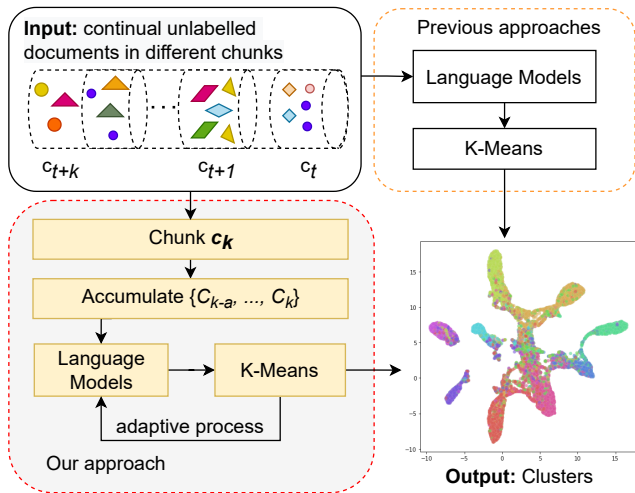


Figure 1: Overview of traditional approaches in comparison to ours in unsupervised text clustering tasks, where chunk data can be accumulated for the adaptive process.

General purpose clustering algorithms not specifically designed to work on text documents can be used for document clustering by creating vector representations of documents using deep neural networks and then clustering those vectors. One way of doing so is to use autoencoders [2, 24] applied to *term frequency – inverse document frequency* document representations (tf-idf, [22]). However, such representations neglect contextual information. Alternatively, one can use contextual representations obtained from pre-trained language models (LMs). Such approaches run a clustering algorithm such as K-Means over the output of the LM [9, 25, 8, 30, 6]. In another line of work, some studies proposed the simultaneous learning of document representations and clustering through a self-learning approach. This involves computing an auxiliary target distribution using the output of the model and minimizing the loss between these distributions [14, 26, 11]. A problem with this approach is the risk of self-confirmation bias, potentially leading to trivial solutions. Moreover, the majority of these proposals rely on autoencoders, with limited exploration of LMs. In this paper, we introduce ADCluster, which uses K-Means as a teacher to train an LM-based classifier in an iterative manner to adapt it to the clustering task. Figure 1 shows the comparison between our approach and previous approaches (which use LMs) in the unsupervised clustering task. We hypothesize that the adaptation process is essential for any real-world application where there is no labeled training data.

In applications that rely on document clustering, the collection of documents is seldom static. For example, consider an online service using web crawlers to find new content of interest for them, or an online advertising service

trying to discover appropriate web pages for ad placement [12]. Given that new content is created every day, their document collections will steadily increase. With time, clustering will become unreliable because of subtle topic shifts or previously unknown terms such as Fridays for Future or King Charles III. Our method facilitates resuming the iterative adaptation of the model to the clustering task from its previous state when a new chunk of documents is to be incorporated.

Thus, we distinguish between inner and outer adaptation. Inner adaptation adjusts the LM to the clustering task at hand by an iterative training process during which the data is considered immutable. Outer adaptation adjusts the model over time to growing sets of documents by resuming the inner adaptation when a significant amount of new data becomes available, either by considering the entire dataset (*accumulative outer adaptation*) or using only the new data (*non-accumulative outer adaptation*). An obvious third possibility is to rebuild the model from scratch or use a scheduled combination of the three possibilities, depending on the practical conditions under which the model is used.

In this paper, we mainly focus on introducing the model and studying its performance under the accumulative and non-accumulative adaptation regimes. Future work will study the dynamic behavior arising when the model adapts to growing document collections as topics evolve.

Apart from introducing the clustering technique itself, and the algorithm used for training, we experiment with three different datasets, each of which we divide into five chunks in order to simulate growing collections of documents. The empirical results show the following:

1. Under each variant of the outer adaptation (training from scratch, accumulative, and non-accumulative adaptation), ADCluster outperforms the baselines.

2. In the absence of significant topic shifts, the three outer adaptation regimes usually result in comparable performance. Hence, one can choose between them as fits the application.

In addition to these main results, we conduct experiments to show that the method is insensitive to the type of language model used (our main experiments use BERT).

## 2 Related Work

Clustering is a much studied unsupervised problem in machine learning and data mining which is central to many data-driven applications. Many strategies for clustering arbitrary sets of data points in an  $n$ -dimensional space have been studied. These include density-based, hierarchical, centroid- and partition-based clustering; see Xu and Tian [27] for an overview. K-Means [17] and HDBSCAN [3] are two of the most popular traditional clustering algorithms.

The progress in deep learning that has been made during the last decade has made it natural to apply deep learning to clustering tasks [31]. An example

of this is seen in DEC [26], which utilizes a stacked autoencoder to acquire document representations from tf-idf vectors. Subsequently, it improves these representations while learning clustering in a self-supervising manner. Hosseini and Varzaneh [13] present a hybrid deep clustering method combining a stacked autoencoder and k-Means to organize Persian texts into clusters.

In recent years, large language models trained for language understanding and generation have achieved impressive results across a wide range of tasks. These LMs produce excellent general-purpose contextual representations that reflect topical information and can thus be used for clustering. Guan et al. [9] generate document representations by pooling the outputs of ELMo [21] pre-trained LM and apply K-Means to these representations after normalizing them. Gupta et al. [10] employ language models for unsupervised model interpretation and syntax induction through deep clustering of text representations. Huang et al. [14] fine-tune the LM simultaneously with masked language modeling and clustering losses.

To our knowledge, no existing research explores deep clustering with LMs for dynamic scenarios involving a growing set of documents. Our method provides a simple yet effective approach to improve cluster assignments by training the LM in an adaptive manner to provide clustering-friendly representations that, over time, can be adapted to a growing set of documents.

### 3 Methodology

We first describe how the *inner* adaptation of the proposed model ADCluster works. Its pseudocode is given in Algorithm 1. It uses a conventional K-Means algorithm and a Deep Neural Network (DNN) classifier. The classifier is adapted iteratively in order to improve the clusterability of the embedding vectors. This is the inner adaptation. The classifier consists of a LM-based text encoder (a pre-trained LM with a mean pooling layer over its last layer) denoted by  $f_\theta$  (where  $\theta$  is the set of parameters) followed by a Multi-Layer Perceptron (MLP) head denoted by  $W$  that maps document representations to cluster assignments. Suppose we have an unlabeled dataset  $D = \{d_n\}_{n=1}^N$  of  $N$  documents. At the beginning of each training epoch, we map each document  $d_n$  to its contextual representation  $f_\theta(d_n)$ . So,  $E = \{f_\theta(d_n)\}_{n=1}^N$  is the set of document contextual representations. Often, it is beneficial to reduce the dimensionality of these representations using a dimension reduction method such as PCA [19] or UMAP [18], resulting in a set  $E'$  of vectors of fewer dimensions. Next, we use K-Means (based on cosine similarity rather than squared Euclidean distance) to cluster  $E'$  into  $K$  distinct clusters. We use these cluster assignments  $\{p_n\}_{n=1}^N$  as *pseudo-labels* to train the classifier. For this, the MLP  $W$  and the encoder  $f_\theta$  are jointly trained to minimize the cross

---

**Algorithm 1:** ADCluster (inner adaptation)

---

**Input** :  $D$ : the set of unlabeled documents  
 $f_\theta$ : LM-based encoder of DNN classifier  
 $W$ : MLP head of DNN classifier  
 $MaxIter$ : the max training iterations  
 $EpochSize$ : iterations per training epoch  
 $b$ : the mini-batch size  
 $\eta, \gamma$ : the training learning rates  
 $DR$ : the dimension reduction method  
 $\tau$ : a threshold for the minimum percentage of changing assignments within two consecutive epochs (convergence threshold)

**Output** :  $(\theta^*, W^*)$ : The optimal weights  
 $C$ : final cluster assignments for  $D$

```
1  $MaxEpoch \leftarrow MaxIter / EpochSize$ ;  
2 for  $epoch = 1$  to  $MaxEpoch$  do  
3    $E \leftarrow$  encode  $D$  with  $f_\theta$ ;  
4    $E' \leftarrow DR(E)$  ▷ Apply DR with condition  
5    $P \leftarrow$  run K-means on  $E'$  using cosine similarity;  
6    $X \leftarrow$  choose  $b * EpochSize$  documents from pseudo-labeled set  $P$  with a  
   uniform sampler;  
7    $W \leftarrow$  initialize  $W$  with Xavier initialization;  
8   for  $iter = 1$  to  $EpochSize$  do  
9      $B_{iter} \leftarrow$  choose a mini-batch from  $X$ ;  
10     $Y_{iter} \leftarrow W(f_\theta(B_{iter}))$ ;  
11     $\hat{Y}_{K\text{-means}} \leftarrow P(B_{iter})$ ;  
12     $l \leftarrow$  cross-entropy-loss ( $Y_{iter}, \hat{Y}_{K\text{-means}}$ );  
13     $\theta \leftarrow \theta - \eta * l(\theta)$  ▷ Update  $\theta$   
14     $W \leftarrow W - \gamma * l(W)$  ▷ Update  $W$   
15  end  
16   $C_{curr} \leftarrow W_{predict}(f_\theta(D))$  ▷ predict cluster assignments for  $D$   
   with DNN classifier  
17   $t \leftarrow$  compute ( $C_{curr}, C_{prev}$ ) ▷ Compute the percentage of changing  
   cluster assignments compared to previous epoch;  
18  if  $t < \tau$  then  
19    | stop the iterative process  
20  end  
21   $C_{prev} \leftarrow C_{curr}$   
22 end  
23 return  $\theta^*, W^*, C$ ;
```

---

entropy loss

$$\frac{\sum_{n=1}^b -\log \frac{\exp(y_{n,p_n})}{\sum_{k=1}^K \exp(y_{n,k})}}{b} \quad (1)$$

where  $y_n$  is the output of the classifier for document  $d_n$  and  $b$  is the mini-batch size. This cost function is minimized using AdamW [16] and backpropagation to compute the gradients. With the goal of preventing the classifier from overfitting to the current pseudo-labels, we employ only a subset of the data in every training epoch and restrict the number of iterations (i.e., *EpochSize* in Algorithm 1).

It is worth mentioning that there is no correspondence between two consecutive cluster assignments. Hence, the final classification layer learned for an assignment becomes irrelevant for the following one and thus needs to be re-initialized from scratch at each epoch. We found that re-initializing the entire MLP head of the classifier rather than the final classifier layer is also beneficial for reducing the risk of overfitting. Since the MLP is a shallow network (having only one hidden layer), it can be trained sufficiently in one epoch.

In addition, we predict cluster assignments for all documents at the end of each epoch using the classifier and stop our procedure when the change in assignments is less than a threshold  $\tau$ , i.e., the algorithm terminates when the number of documents for which the cluster assignment changes falls below  $\tau$ .

Overall, ADCluster alternates between clustering document representations to produce pseudo-labels and updating the parameters of the classifier by predicting these pseudo-labels using Eq. (1). This iterative adaptation of the encoder teaches the LM to generate more clustering-friendly representations. This distinguishes ADCluster from conventional methods, resulting in an improved K-Means clustering in subsequent epochs. The final clusters are obtained using the adapted classifier to predict cluster assignments.

If K-Means assigns almost all documents to a few large clusters,  $\theta$  will only discriminate between them. A trivial parameterization occurs when all clusters except one are singletons, and therefore the classifier predicts the same output for all inputs [4]. To overcome this problem, we train the classifier on uniformly sampled documents from the pseudo-labeled classes. The result is the same as weighting the contribution of a document to the loss function by the inverse of the size of the cluster to which it belongs.

Let us now briefly explain the *outer* adaptation of ADCluster. Imagine a data stream where new data arrives sequentially in chunks  $C_t$ , where  $t$  denotes the time step. In the *accumulative* scenario, we resume the inner adaptation of ADCluster at time  $t$  using  $C_0 \cup \dots \cup C_t$  as training data when a new chunk  $C_t$  arrives. In contrast, the *non-accumulative* approach resumes inner adaptation solely with the latest chunk  $C_t$ .

Table 1: Datasets and statistics. *Silhouette Coefficient* refers to the Silhouette score of Rousseeuw [23] which measures how similar a document is to its own cluster compared to other clusters, the best and worst values being 1 and -1, respectively. We compute the mean Silhouette Coefficient of all samples of the datasets using their true labels. As our LM for creating document representations, we use a BERT language model.

Dataset	Yahoo!5	Ag News	Fake News
#-Documents	38 812	40 000	480
Avg # sents	25.12	1.45	6.05
Avg # word (in doc)	578.26	36.09	141.20
Avg Silhouette Coefficient	0.01234	0.03736	0.04356

## 4 Experiments

### 4.1 Datasets

We employ the following three datasets whose statistics are summarized in Table 1:

**Yahoo!5** is a subset of Yahoo! Answers [29]. The dataset comprises 10 classes, each document consisting of a question, a title, and the best answer to the question. We obtain the text to be clustered by concatenating these parts. To obtain a long-text dataset we only choose samples of over 500 tokens. The resulting dataset includes 38 812 documents.

**Ag News** [29] consists of 4 classes: World, Sports, Business, and Sci/Tech news. The number of training and testing samples for each class is 30 000 and 1 900, respectively. We choose 40 000 documents at random from the training set. To have a very short-text dataset, we only consider the news text and ignore the titles.

**Fake News** [20] comprises 480 medium-length news articles belonging to six different domains. While half of the articles are real and the other half are fake news, we do not make use of this distinction but use only the six topics of the dataset as labels.

Following the approach of prior studies [14, 26, 11], we form unlabelled documents by removing all labels for the training set, using the labels only to evaluate unsupervised performance.

### 4.2 Baselines

We use the following baselines for comparisons:

**Traditional clustering algorithms** We compare our model with K-Means and HDBSCAN. For HDBSCAN, we use the soft (or fuzzy) imple-



mentation<sup>1</sup> of the algorithm that predicts probability vectors for all dataset samples; no samples are considered noise. These vectors show the membership probability for each cluster, so we assign the sample to the cluster for which the highest probability has been determined. Instead of using pure BERT vectors, we apply normalization on them prior to performing dimension reduction and clustering. Before running HDBSCAN on the datasets, we perform dimension reduction using UMAP<sup>2</sup>. For each dataset, we test several values for parameters of HDBSCAN and UMAP and report the highest accuracy we get. On Yahoo! Answers, we perform PCA dimension reduction ( $n\_components = 0.8$ ; preserving at least 80% of variance) before K-Means.

**DEC-tfidf** we compare our model with that of Xie, Girshick, and Farhadi [26], using the available PyTorch implementation from <https://github.com/vlukiyanov/pt-dec>. We slightly adjust the parameters reported in the paper to our datasets and present the highest value obtained.

**DEC-BERT** To have a more fair comparison between ADCluster and DEC [26], we replace the stacked autoencoder part of DEC with a BERT language model followed by a mean pooling layer to encode documents and train it with the same objective function as in DEC.

**UFT** We compare our model with the model presented in Huang et al. [14]. We refer to this baseline as UFT. We obtained the source code from the authors of the paper and applied it to our datasets.

**ADCluster-noIter** is a non-iterative version of ADCluster. We run K-Means only once using contextual representations of documents from BERT and train the neural classifier with the generated pseudo-labels for some iterations.

**Centroid-ADCluster** Since in ADCluster there is no correspondence between two consecutive cluster assignments, the final classification layer learned for an assignment becomes irrelevant for the following one and thus needs to be re-initialized from scratch at each epoch. We do this to prevent the model from overfitting to the noisy pseudo-labels. For verification, we implemented another version of ADCluster in which we, instead of learning a classification layer predicting the cluster assignments, perform explicit comparisons between features and centroids.

### 4.3 Evaluation Metric

We adopt a standard unsupervised evaluation metric that is widely used in deep clustering studies to compare our proposed method to other algorithms. For all the algorithms, the number of clusters is set to the number of ground-truth categories of each dataset, and we evaluate the clustering performance

<sup>1</sup>[https://hdbscan.readthedocs.io/en/latest/soft\\_clustering.html](https://hdbscan.readthedocs.io/en/latest/soft_clustering.html)

<sup>2</sup><https://umap-learn.readthedocs.io/en/latest/>

using the unsupervised clustering accuracy (ACC):

$$ACC = \max_m \frac{\sum_{n=1}^N 1\{l_n = m(c_n)\}}{N}$$

where  $N$  is the total number of documents,  $l_n$  is the ground-truth label of document  $d_n$ ,  $c_n$  is the cluster assignment that is predicted by the clustering algorithm for  $d_n$ , and  $m$  maps cluster assignments to labels, ranging over all possible one-to-one mappings. This metric seeks the best possible alignment between the ground-truth label and the cluster assignments generated by an unsupervised clustering algorithm. The Hungarian algorithm, presented in the work of Xu, Liu, and Gong [28], offers a means to efficiently calculate the most effective mapping function within the context of a linear assignment problem.

## 4.4 Experimental Setup

We implemented ADCluster using the PyTorch framework, utilizing bert-base-uncased LM of Hugging Face<sup>3</sup>. Documents are truncated to their first 256 tokens. To generate document embeddings, we employ average pooling over the output of the language model. For label prediction, we employ a two-layer MLP with a single hidden layer. The hidden layer size is set to 128 for Yahoo!5 and Fake News and 768 for Ag News. The hyperbolic tangent function is used as the activation function for the MLP.

We set the mini-batch size to 4 and the learning rate of the LM and MLP head to  $10^{-6}$  and  $10^{-4}$  correspondingly. We also use a cosine scheduler for the learning rate of the LM. We train ADCluster for at most 10 000 iterations and reassign the clustering labels by applying K-Means on document representations every 200 iteration (which we call an *epoch*). The threshold for stopping training when cluster assignments do not significantly change anymore is set to 1% of the documents. The model is trained using the AdamW optimizer with  $\alpha$  and  $\beta$  equal to 0.999. We use the first 200 iterations as warm-up steps for the LM. To initialize the centroids of K-Means we use the K-Means++ seeding strategy proposed by Arthur and Vassilvitskii [1] and to initialize weights of MLP head in each epoch we use Xavier initialization [7]. We train ADCluster-noIter and Centroid-ADCluster under the same settings. The only difference for Centroid-ADCluster is that the size of the hidden layer of the MLP head is 768 for all datasets and the weights of the last layer ( $768 \cdot K$ , where  $K$  is the number of classes in the dataset) are initialized with the centroids of the K-Means which are constant during training. For the other baselines, we test several sets of values for their hyperparameters and report the best results.

---

<sup>3</sup><https://huggingface.co/bert-base-uncased>

Table 2: Overall performances of ADCluster in comparison to baselines. ♥ indicates short-text datasets.

Method		Yahoo!5	Ag News♥	Fake News
Classic Clustering	Kmeans (BERT)	44.64	81.6	73.96
	HDBSCAN (BERT)	58.8	<b>83.68</b>	72.71
DEC [26]*	tf-idf	50.23	68.93	45.41
	BERT	46.43	78.32	75.83
UFT [14]*		46.94	65.46	66.67
ADCluster (ours)	Centroid-ADCluster	60.64	80.93	76.67
	ADCluster-Final	<b>67.94</b>	83.44	<b>77.50</b>

\* The result is produced by us following the original paper

Table 3: Performance analysis of ADCluster across varied dataset sizes compared to baselines. Note that, because of the unsupervised setting, there is no expectation of monotonic increases in performance.

Dataset	Method	10%	50%	80%	100%
Ag News	K-Means	82.4	81.39	81.41	81.6
	DEC-BERT	79.3	78.22	78.4	78.32
	ADCluster	<b>84.08</b>	<b>82.56</b>	<b>84.3</b>	<b>83.44</b>
Yahoo!5	K-Means	53.23	53.5	59.95	52.17
	DEC-BERT	45.74	46.44	46.56	46.43
	ADCluster	<b>66.3</b>	<b>66.03</b>	<b>67.38</b>	<b>67.94</b>
Fake News	K-Means	64.58	77.08	77.34	73.96
	DEC-BERT	<b>68.75</b>	79.58	77.60	75.83
	ADCluster	64.58	<b>83.75</b>	<b>79.95</b>	<b>77.50</b>

## 5 Results and Discussions

### 5.1 Overall Performance

Generally, ADCluster achieves better performances than most of the baseline methods across multiple datasets (see Table 2). Compared to traditional clustering algorithms, ADCluster outperforms K-Means from 1.84% (Ag News) up to 23.3% (Yahoo!5), indicating that the iterative learning process (inner adaptation) of our model is effective. We can also note that HDBSCAN achieves better performance than K-Means in most cases but outperforms ADCluster only in the case of Ag News. In Table 1, we see that Ag News consists of very short texts, its average number of sentences per document being 1.45 and the average number of words being 36.09. It does not seem to provide enough context for BERT to make distinctive representations, thus limiting the efficacy of our model on this particular dataset. However, in Section 5.5 we will see that

by replacing BERT with more advanced LMs the performance of our model on this dataset improves. For Yahoo!5 and Fake News, HDBSCAN gains better performance than most of the other methods except ADCluster. In fact, for these datasets, ADCluster displays better performance than all baselines. This holds even in the case of Fake News, which consists of a very limited number of documents (i.e., 480 documents).

The comparison with DEC-based models yields the following observations. Firstly, ADCluster outperforms DEC-tfidf, which we attribute to its use of BERT contextual representations (whereas tf-idf representations only consider text as a bags of words and neglect their semantic relations). Secondly, even though DEC-BERT has similar access to the contextual information of the language model, its performance is still lower than that of our model. The same applies to the UFT baseline. The reason could be that these models are trained in a self-learning fashion and may thus suffer from self-confirmation. Our model avoids this by using K-Means as an external teacher for our neural classifier. It also uses a uniform sampling technique for batch creation, mitigating biases stemming from imbalanced clusters.

## 5.2 Dynamic Performance Analysis of ADCluster Across Varied Dataset Sizes

In this experiment, we examine the performance of ADCluster in comparison to baselines as the dataset size gradually increases. The outcomes of this experiment are presented in Table 3, illustrating the results as the document size expands from 10% to 100%. In general, ADCluster consistently maintains stable performance throughout these experiments and surpasses baseline models for all datasets, with the exception of the 10% case for Fake News.

## 5.3 Illustration of Learned Representations by ADCluster

In order to investigate how ADCluster develops clustering-friendly representations through internal adaptation, we visualize the evolution of clusters during the training process using the Yahoo!5 dataset. Figure 2 shows how ADCluster clusters the documents during different epochs with ground-truth classes represented by different colors. The figure clearly demonstrates that at the very beginning, the structure is random. Along with the adaptation process, documents are arranged into more distinct groups, which is signified by both color separation and spatial characteristics. This trend is further confirmed by the continuous enhancement in clustering performance observed in each successive epoch.

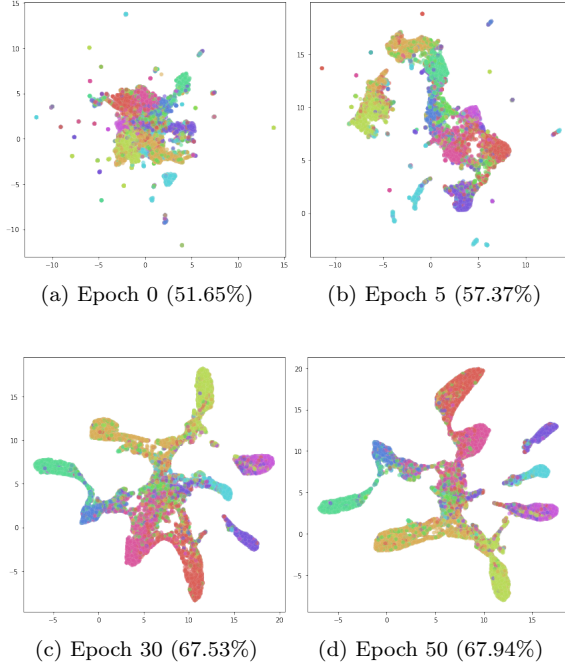


Figure 2: Illustration of clustered contextual representations according to ADCluster for Yahoo! Answer during inner adaptation. Colors indicate ground-truth classes. We have used UMAP to map 768-dimensional representations to a 2D feature space for illustration.

## 5.4 The Model Behavior on Data Streams

**Notation.** Hereafter, if not otherwise specified, we use  $A_c$  to abbreviate *Accumulation*. We randomly split each unlabelled data collection into 5 chunks and denote them by  $C_1$  (1–20%),  $C_2$  (21–40%),  $C_3$  (41–60%),  $C_4$  (61–80%),  $C_5$  (81–100%).

We now analyze the outer adaptation behavior of ADCluster. In this experiment, we assume the number of the clusters to be constant over time, only receiving new samples. We compare our model with three baselines:

**Word2vec+KM** We generate document representations as the average of the Word2vec embeddings of all words in the document and use K-Means to cluster these representations.

**BERT+KM** We create document representations by taking the average of the output of the last BERT layer for non-pad tokens and use K-Means to cluster these representations.

**ADCluster-scratch** This baseline is the same as ADCluster except that

instead of performing outer adaptation, we train the model from scratch (accumulatively on the whole dataset or non-accumulatively on the last chunk only, respectively). Thus, we remove the outer adaptation and the model only benefits from the inner adaptation.

Tables 4–6 show the results of our experiments.

As our main take-aways from these experiments, we note that ADCluster outperforms the Word2vec+KM and BERT+KM baselines in all cases in both the *Ac* and *non-Ac* settings. The superior accuracy of ADCluster on chunk  $C_1$  can be attributed to the inner adaptation which the baseline models lack. However, interestingly the outer adaptation results in superior performances in most cases on chunks  $C_2$ – $C_5$  even compared to ADCluster-scratch, which is remarkable and shows the effectiveness of outer adaptation.

Table 4: Comparing the outer adaptation performance of ADCluster with baselines on Yahoo!5.

Method	Ac	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$
Word2vec+KM	Yes	52.09	41.86	47.08	44.94	49.02
BERT+KM	Yes	46.28	53.84	53.67	55.24	53.70
ADCluster-scratch	Yes	<b>67.33</b>	66.44	64.06	64.51	62.06
ADCluster	Yes	<b>67.33</b>	<b>67.99</b>	<b>68.07</b>	<b>67.8</b>	<b>67.48</b>
Word2vec+KM	No	52.09	42.51	45.72	49.79	50.22
BERT+KM	No	46.28	57.02	52.00	54.86	55.04
ADCluster-scratch	No	<b>67.33</b>	67.11	65.19	61.79	65.50
ADCluster	No	<b>67.33</b>	<b>68.07</b>	<b>68.24</b>	<b>67.61</b>	<b>67.98</b>

Table 5: Comparing the outer adaptation performance of ADCluster with baselines on Ag News.

Method	Ac	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$
Word2vec+KM	Yes	80.65	79.98	80.55	80.87	80.83
BERT+KM	Yes	81.66	81.42	81.50	81.51	81.52
ADCluster-scratch	Yes	<b>84.07</b>	84.56	<b>84.09</b>	<b>83.07</b>	81.76
ADCluster	Yes	<b>84.07</b>	<b>84.81</b>	82.56	83.05	<b>84.03</b>
Word2vec+KM	No	80.65	79.59	81.49	80.80	80.85
BERT+KM	No	81.66	81.43	81.20	81.82	81.05
ADCluster-scratch	No	<b>84.07</b>	83.74	81.95	<b>83.87</b>	82.51
ADCluster	No	<b>84.07</b>	<b>84.01</b>	<b>84.25</b>	83.6	<b>83.44</b>

Table 6: Comparing the outer adaptation performance of ADCluster with baselines on Fake News.

Method	Ac	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$
Word2vec+KM	Yes	67.71	79.69	78.47	71.35	74.58
BERT+KM	Yes	57.29	77.60	77.08	77.34	77.29
ADCluster-scratch	Yes	<b>69.79</b>	82.81	<b>84.37</b>	79.69	79.58
ADCluster	Yes	<b>69.79</b>	<b>83.33</b>	83.68	<b>81.25</b>	<b>80.62</b>
Word2vec+KM	No	67.71	80.21	62.50	54.17	57.29
BERT+KM	No	57.29	77.08	58.33	53.12	51.04
ADCluster-scratch	No	<b>69.79</b>	82.29	67.71	57.29	59.37
ADCluster	No	<b>69.79</b>	<b>86.46</b>	<b>79.17</b>	<b>61.46</b>	<b>73.96</b>

## 5.5 Ablation study

In this ablation study, we design two settings to study the effectiveness of each ADCluster component. First, we replace the default BERT language model with recent models such as RoBERTa, SBERT, and BART. Second, we test various settings: (1) removing outer adaptation, (2) using a random sampler instead of a uniform sampler, and (3) Using UMAP for dimension reduction (instead of PCA for the Yahoo!5, and instead of not using dimension reduction for Ag News and Fake News). Figure 3 clearly shows that recent advanced language models yield better performance on all of the datasets. Table 7 summarizes the performance of ADCluster in the second setting. Across all experiments, the final model of ADCluster shows better performance than

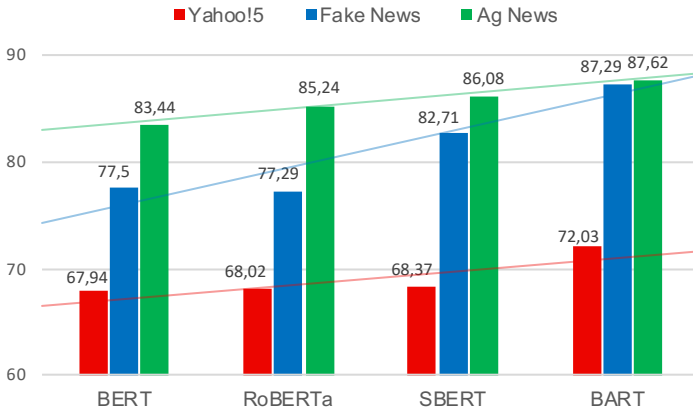


Figure 3: Ablation study w.r.t. different language models being used for the inner adaptation of ADCluster.

Table 7: Ablation study to evaluate the impact of different components of ADCluster to the final performance.

Ablation setting	Yahoo!5	Ag News	Fake News
Non iterative	53.89	82.88	73.96
UMAP	64.74	58.33	66.25
Random sampler	65.78	79.2	76.04

these variants.

## 6 Conclusion and Future Work

We have introduced ADCluster, a neural document clustering model that iterates between a contextual language model and K-Means. K-Means is applied to contextualized document representations created by a BERT language model in order to obtain pseudo-labels. The weights of the language model are then iteratively adapted to improve the prediction of cluster assignments using discriminative loss. Not only does this *inner adaptation* result in superior clustering performance, it also enables us to resume training when the dataset grows (outer adaptation), as is often the case in real-world applications. Our empirical results show that for medium to long-text documents, ADCluster consistently outperforms conventional clustering models by a considerable margin with respect to the unsupervised accuracy measure.

Future work will have to study the inner and outer adaptation in more detail. For instance, one interesting direction could be a “soft adaptation”, which continuously measures how much weight the outer adaptation shall place on earlier and later chunks. So far, we only presented two extreme cases, i.e., accumulation or non-accumulation.

Moreover, text data is often accompanied by additional modalities such as images, audio, and video. Such multimodal data has the potential to help the model understand the semantics of documents and assign them to the right cluster [5, 15]. Multimodality can also open the door to new real-world downstream applications. Therefore, we are interested in extending our model to multimodal data clustering in the future.

## References

- [1] David Arthur and Sergei Vassilvitskii. “K-Means++: The Advantages of Careful Seeding”. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. ISBN: 9780898716245. URL: <https://dl.acm.org/doi/10.5555/1283383.1283494>.



- [2] Dana H. Ballard. “Modular Learning in Neural Networks”. In: *Proceedings of the Sixth National Conference on Artificial Intelligence*. Vol. 1. AAAI’87. Seattle, Washington: AAAI Press, 1987, pp. 279–284. ISBN: 0934613427. URL: <https://dl.acm.org/doi/10.5555/1863696.1863746>.
- [3] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. “Density-based clustering based on hierarchical density estimates”. In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer Berlin Heidelberg, 2013, pp. 160–172. URL: [https://link.springer.com/content/pdf/10.1007/978-3-642-37456-2\\_14.pdf](https://link.springer.com/content/pdf/10.1007/978-3-642-37456-2_14.pdf).
- [4] Mathilde Caron et al. “Deep clustering for unsupervised learning of visual features”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 132–149. URL: [https://doi.org/10.1007/978-3-030-01264-9\\_9](https://doi.org/10.1007/978-3-030-01264-9_9).
- [5] Brian Chen et al. “Multimodal clustering networks for self-supervised learning from unlabeled videos”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 8012–8021. URL: <https://doi.org/10.48550/arXiv.2104.12671>.
- [6] Anton Eklund and Mona Forsman. “Topic Modeling by Clustering Language Model Embeddings: Human Validation on an Industry Dataset”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Abu Dhabi, UAE: Association for Computational Linguistics, Dec. 2022, pp. 635–643. DOI: 10.18653/v1/2022.emnlp-industry.65. URL: <https://aclanthology.org/2022.emnlp-industry.65>.
- [7] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings*. 2010, pp. 249–256. URL: <https://api.semanticscholar.org/CorpusID:5575601>.
- [8] Maarten Grootendorst. “BERTopic: Neural topic modeling with a class-based TF-IDF procedure”. In: *Computing Research Repository arXiv:2203.05794* (2022). URL: <https://doi.org/10.48550/arXiv.2203.05794>.
- [9] Renchu Guan et al. “Deep Feature-Based Text Clustering and its Explanation”. In: *IEEE Transactions on Knowledge and Data Engineering* 34.8 (2022), pp. 3669–3680. DOI: 10.1109/TKDE.2020.3028943.
- [10] Vikram Gupta et al. “Deep clustering of text representations for supervision-free probing of syntax”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 10. 2022, pp. 10720–10728. URL: <https://doi.org/10.1609/aaai.v36i10.21317>.

- [11] Amir Hadifar et al. “A Self-Training Approach for Short Text Clustering”. In: *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 194–199. DOI: 10.18653/v1/W19-4322. URL: <https://aclanthology.org/W19-4322>.
- [12] Arezoo Hatefi et al. “Cformer: Semi-Supervised Text Clustering Based on Pseudo Labeling”. In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. Virtual Event, Queensland, Australia: Association for Computing Machinery, 2021, pp. 3078–3082. URL: <https://doi.org/10.1145/3459637.3482073>.
- [13] Soodeh Hosseini and Zahra Asghari Varzaneh. “Deep Text Clustering Using Stacked AutoEncoder”. In: *Multimedia Tools Appl.* 81.8 (2022), pp. 10861–10881. ISSN: 1380-7501. DOI: 10.1007/s11042-022-12155-0. URL: <https://doi.org/10.1007/s11042-022-12155-0>.
- [14] Shaohan Huang et al. “Unsupervised Fine-tuning for Text Clustering”. In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, 2020, pp. 5530–5534. DOI: 10.18653/v1/2020.coling-main.482. URL: <https://aclanthology.org/2020.coling-main.482>.
- [15] Yangbangyan Jiang et al. “DM2C: Deep Mixed-Modal Clustering”. In: *Neural Information Processing Systems*. 2019. URL: <https://api.semanticscholar.org/CorpusID:202777596>.
- [16] Ilya Loshchilov and Frank Hutter. “Decoupled Weight Decay Regularization”. In: *International Conference on Learning Representations*. 2019. URL: <https://api.semanticscholar.org/CorpusID:53592270>.
- [17] James MacQueen et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. 1967, pp. 281–297. URL: <https://api.semanticscholar.org/CorpusID:6278891>.
- [18] Leland McInnes, John Healy, and James Melville. “Ummap: Uniform manifold approximation and projection for dimension reduction”. In: *Computing Research Repository* arXiv:1802.03426 (2020). Version 3. URL: <https://doi.org/10.48550/arXiv.1802.03426>.
- [19] Karl Pearson. “LIII. On lines and planes of closest fit to systems of points in space”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: 10.1080/14786440109462720. URL: <https://doi.org/10.1080/14786440109462720>.

- [20] Verónica Pérez-Rosas et al. “Automatic Detection of Fake News”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, 2018, pp. 3391–3401. URL: <https://aclanthology.org/C18-1287>.
- [21] Matthew E. Peters et al. “Deep Contextualized Word Representations”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*. Association for Computational Linguistics, 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: <https://aclanthology.org/N18-1202>.
- [22] Anand Rajaraman and Jeffrey David Ullman. “Data Mining”. In: *Mining of Massive Datasets*. Cambridge University Press, 2011, pp. 1–17. DOI: 10.1017/CB09781139058452.002. URL: <https://doi.org/10.1017/CB09781139058452.002>.
- [23] Peter J. Rousseeuw. “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. ISSN: 0377-0427. DOI: [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7). URL: <https://www.sciencedirect.com/science/article/pii/0377042787901257>.
- [24] Jürgen Schmidhuber. “Deep learning in neural networks: An overview”. In: *Neural Networks* 61 (2015), pp. 85–117. ISSN: 0893-6080. DOI: <https://doi.org/10.1016/j.neunet.2014.09.003>. URL: <https://www.sciencedirect.com/science/article/pii/S0893608014002135>.
- [25] Alvin Subakti, Hendri Murfi, and Nora Hariadi. “The performance of BERT as data representation of text clustering”. In: *Journal of Big Data* (2022). DOI: 10.1186/s40537-022-00564-9. URL: <https://doi.org/10.1186/s40537-022-00564-9>.
- [26] Junyuan Xie, Ross Girshick, and Ali Farhadi. “Unsupervised Deep Embedding for Clustering Analysis”. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning*. Vol. 48. New York, NY, USA, 2016, pp. 478–487. URL: <https://dl.acm.org/doi/10.5555/3045390.3045442>.
- [27] Dongkuan Xu and Yingjie Tian. “A Comprehensive Survey of Clustering Algorithms”. In: *Annals of Data Science* 2 (2015), pp. 165–193. URL: <https://doi.org/10.1007/s40745-015-0040-1>.
- [28] Wei Xu, Xin Liu, and Yihong Gong. “Document clustering based on non-negative matrix factorization”. In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. 2003, pp. 267–273. URL: <https://doi.org/10.1145/860435.860485>.

- [29] Xiang Zhang, Junbo Zhao, and Yann LeCun. “Character-level Convolutional Networks for Text Classification”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf).
- [30] Zihan Zhang et al. “Is Neural Topic Modelling Better than Clustering? An Empirical Study on Clustering with Contextual Embeddings for Topics”. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 3886–3893. URL: <https://aclanthology.org/2022.naacl-main.285>.
- [31] Sheng Zhou et al. “A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions”. In: *Computing Research Repository* arXiv:2206.07579 (2022). URL: <https://doi.org/10.48550/arXiv.2206.07579>.

**PromptStream: Self-Supervised News Story Discovery Using Topic-Aware Article Representations**

Arezoo Hatefi, Anton Eklund, and Mona Forsman

*Accepted to Appear in the Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING), 2024.*



# PromptStream: Self-Supervised News Story Discovery Using Topic-Aware Article Representations\*

Arezoo Hatefi<sup>†</sup>, Anton Eklund<sup>†,\*</sup>, Mona Forsman\*

<sup>†</sup>*Department of Computing Science, Umeå University, Umeå, Sweden*

<sup>\*</sup>*Aeterna Labs, Umeå, Sweden*

*arezooh@cs.umu.se, anton eklund@cs.umu.se, mona@adlede.com*

**Abstract:** Considering the importance of identification and tracking of news stories within the constant stream of news content, this paper introduces PromptStream, a novel approach to the task of unsupervised news story discovery. The key to achieving coherence and completeness in story identification throughout the stream lies in embedding as much topic-related information from the articles as possible. PromptStream constructs these article embeddings using cloze-based prompting. These representations continually adjust to the evolving context of the news stream through self-supervised learning, employing a contrastive loss and a memory of the most confident article-story assignments from the most recent days. Extensive experiments with real news datasets highlight the notable performance of our model, establishing a new state of the art. Additionally, we delve into selected news stories to reveal how the model’s structuring of the article stream aligns with story progression.

**Key words:** news story discovery, online clustering, data stream, contrastive learning, cloze-style prompting, article embedding

## 1 Introduction

In the abundance of news being generated daily, online news story discovery streamlines individual news consumption and is invaluable for news summarization, recommendation systems, and other services reliant on structured news content understanding. The concept of recognizing and tracking topics within a stream was initially introduced in the Topic Detection and Tracking (TDT) task [2]. This task revolves around techniques for the automated structuring of

---

\*The paper has been accepted for publishing in the *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, and has been re-typeset to match the thesis style.

textual data streams into coherent topic groupings. In the context of a news stream, these topics essentially represent news stories. Individual news articles report on real-world events, and a subset of articles within the news stream that concern the same event constitutes a news story. We present a model that utilizes cloze-based prompting and self-supervised contrastive learning techniques to tackle this task.

Early efforts in news story discovery relied on sparse document representations such as keywords, and TF-IDF vectors [13, 22]. However, as dense document representations encompassing richer semantic information started to emerge, researchers began exploring their potential in news story discovery. Staykovski et al. [22] compared TF-IDF and Doc2Vec representations for news story discovery and concluded that sparse representations are better for this task. Recently, Saravanakumar et al. [18] demonstrated that integrating contextual BERT representations alongside TF-IDF representations could enhance task performance. This improvement could be achieved through fine-tuning BERT on event similarity using a triplet network architecture [10] and providing external entity knowledge.

Alignment and uniformity [24] represent fundamental characteristics inherent to any embedding space. In the context of news story discovery, alignment pertains to the proximity of articles related to the same story within the embedding space, while uniformity assesses how uniformly random articles are distributed throughout that space. One reason why document representations from pre-trained language models (PLMs) like BERT, without fine-tuning, are less effective for news event discovery is their lack of uniformity. This uniformity issue makes it challenging to differentiate between two articles that share the same theme but concern distinct events.

In recent years, contrastive learning has demonstrated its remarkable effectiveness in numerous language processing and computer vision tasks. This effectiveness stems primarily from its ability to enhance the alignment and uniformity of embedding spaces, as indicated by Wang and Isola [24]. A notable example of this success in news story discovery is the work conducted by Yoon et al. [26]. They used contrastive learning for training story-indicative document representations from sentence representations in a continual learning setting over the news stream and showed that these representations are superior to sparse alternatives.

Recently, prompting has emerged as a groundbreaking technique that has significantly enhanced the performance of natural language processing tasks that require deep understanding. Prompt-based methods align with the Masked Language Modeling (MLM) pretraining task of language models. In MLM, a portion of the input tokens are masked and the model is trained to predict those tokens. Similarly, in cloze-style prompting, a template like "A  $\textit{jmask}_i$  event" is integrated into the input, and the prediction can be obtained by decoding the output embedding associated with  $\textit{jmask}_i$ . Thus, effectively leveraging the large-scale knowledge of PLMs, ultimately resulting in the generation of more



informative representations.

Table 1: Visualization of selected samples from the News14 dataset. The color saturation indicates the attention the tokens receive from the  $jmask_z$  token in the prompt. These examples are an indication that prompting results in topic-tailored representations for the articles by attending to the most important tokens in the text such as events and named entities.

---

A	Chinese	doctor	has	admitted	in	court	that	she	stole	babies	from	the
hospital	where	she	worked	and	sold	them	to	human	traffickers	state	media	
and	a	court	said	Zhang	Sh	ux	ia	,	a	locally	respected	and
soon	-	to										
-	ret	ire	obst	etric	ian	stood	trial	on	Monday	in	Sha	an
xi	Province	's										
F	up	ing	County	according	to	online	postings	from	the	court	Zhang	told
parents	their	newborn	s	had	congen	ital	problems	and	persuaded	them	to	
sign	and	give	the	babies	up	,	"	the	court	postings	said	The
case	exposed											
the	operations	of	a	baby	trafficking							

---

Pass	engers	and	crew	aboard	a	Russian	ship	trapped	for	eight	days	in
ice	off	Antarctica	planned	to	ring	in	the	New	Year	with	dinner	drinks
and	song	as	they	waited	for	a	break	in	a	bl	izzard	to
allow	a	Chinese	helicopter	to	rescue	them	But	they	can	't	party	too
hard	because	the	rescue	could	come	at	any	minute	The	Ak	adem	ik
Sh	ok	als	ki	y	trapped	since	December	24	about	100	n	autical
miles	east	of	a	French	Antarctic	station	Dum	ont	D	Ur	ville	and
about	1	500	n	autical	miles	south	of	Tasmania	welcomes	the	New	Year
at	1100	GMT	two	hours	ahead	of	sydney					

---

In this paper, we present a pioneering approach utilizing cloze-based prompting to enhance article representations with topic-related information, tailoring them to the specific needs of news story discovery in a dynamic news stream. Table 1 presents two instances that illustrate how cloze-based prompts select the most topic-related words from the text to generate the article representation. These representations undergo continuous fine-tuning via cluster-level contrastive learning, making use of a memory bank of confident article-story assignments for self-supervision, to remain relevant within the latest context. The primary objective of confidence-aware memory replay is to effectively mitigate concerns regarding data scarcity and ensure the provision of robust supervision for contrastive learning.

The main contributions of this work are:

1. To the best of our knowledge, our approach is the first in its utilization of cloze-based prompting to enhance article representations for news story discovery.
2. We continuously fine-tune article representations via contrastive learning that makes use of a memory bank of confident article-story assignments for

self-supervision.

3. We make an extensive experimental comparison of PromptStream with SOTA methods for unsupervised online news story discovery. We use three real news datasets for these evaluations and establish a new state of the art.
4. Additionally, we make a deeper exploration of some stories to reveal connections between natural story progression and how PromptStream structures the article stream.

## 2 Related Work

### 2.1 News Story Discovery

Laban and Hearst [13] create a keyword-based graph of articles within a window spanning over  $N$  days by connecting articles that share more keywords than a specified threshold. The system then identifies local topic clusters within overlapping windows using the Louvain community detection algorithm [6]. For long-term stories, it combines topics from non-overlapping windows with a similarity above a given threshold. Staykovski et al. [22] enhance this method by using TF-IDF vectors rather than keywords.

In contrast to the above batch-clustering approach, Miranda et al. [16] employ an online clustering approach, where streaming news articles are compared against existing topic clusters to find the best match or to create a new cluster. Their method computes the similarities between an article and a cluster according to multiple sparse document representations (such as TF-IDF vectors for title, body, and concatenation of title and body) and then aggregates them using a Rank-SVM model. The decision to merge a document with a cluster or create a new cluster is again taken by an SVM classifier. Both SVM models are trained using a supervised training set. Moreover, it uses article timestamps to avoid merging recent documents with older clusters.

Saravanakumar et al. [18] follow an approach similar to that of Miranda et al. [16], but attune BERT embeddings for news event recognition by fine-tuning and adding external entity knowledge. Both Miranda et al. [16] and Saravanakumar et al. [18] exploit external knowledge and labeled datasets, which renders them less practical for real-world applications where supervised data is scarce.

In a recent study, Yoon et al. [26] employ a hierarchical architecture to construct article representations from sentence representations derived from pre-trained sentence encoders. Sentence representations are aggregated into article representations through a one-layer transformer. These representations are then compared with the existing cluster representations within the current window to either identify the best match or establish a new cluster. Notably, the article representations are continuously refined in a self-supervised fashion, with a focus on the most confident assignments within the current window.

## 2.2 Prompt-Based Prediction

Prompt-based prediction [7, 19, 8] approaches NLP downstream tasks as masked language modeling problems. In this approach, a language model initially generates an output based on a predefined prompt utilizing a task-specific template that subsequently is mapped to the output space of the downstream task. This methodology allows for cost-effective knowledge extraction from pre-trained language models and maximizes the utilization of pre-trained corpora. It proves to be an ideal approach for tasks like keyword identification and topic detection since it does not rely on external tools or corpora, in contrast to several of the approaches mentioned in Section 2.1. Examples of successful uses of prompting in natural language processing tasks are: [28, 27] for zero-shot and few-shot event detection, [12] for sentence embedding, and [20, 4] for named entity recognition.

## 3 Preliminaries

An article  $d$  is a sequence  $[w_1, w_2, \dots, w_{|d|}]$  of words. A news story  $s$  is a set of articles,  $s = \{d_1, d_2, \dots, d_{|s|}\}$ , all related to the same event. The objective of online story discovery is to incrementally assign each new article  $d$  in an unbounded news article stream  $\mathbb{D} = [d_1, d_2, \dots]$  to an existing story or create a new cluster if  $d$  does not match any existing one. This process is unsupervised. To account for the publication time of news articles and prevent the assignment of articles to outdated, no longer relevant stories, we employ the concept of a sliding window  $\mathbb{W} \subseteq \mathbb{D}$ . This approach is commonly used for mining data streams [13, 21]. The window and sliding size determine the time span of interest for ongoing stories and the frequency of updates, respectively. For example, a sliding window of 3 days, sliding by one day, addresses the articles published within the last 3 days, with daily updates.

For simplicity, we assume that each article is associated with a single story, and a story is considered alive if at least one article within the window  $\mathbb{W}$  is part of that story. The set of alive stories within the window  $\mathbb{W}$  is denoted by  $\mathbb{S}_{\mathbb{W}}$ .

## 4 The New Model: PromptStream

PromptStream is an online story discovery model that generates topic-aware representations by employing a cloze-based prompting technique for articles. The model architecture is illustrated in Figure 1 and the procedure is described in Algorithm 1. In summary, new articles within a sliding window are assigned to relevant stories, and the prompt-based encoder is updated every  $N$  days in a self-supervised manner, utilizing a memory of confident article-story assignments. Detailed explanations are provided in the subsequent sections.

---

**Algorithm 1:** PromptStream pseudocode

---

**Data:**  $\mathbb{D}$ : a news article stream  
*prompt\_enc*: prompting-based PLM  
*mean\_enc*: mean pooling-based PLM  
*update\_freq*: updating frequency of *prompt\_enc*  
*memory*: confident article-story assignments  
 $\theta$ : article-story similarity threshold  
 $\delta$ : confidence threshold

**Result:** A set  $\mathbb{S}$  of stories in stream  $\mathbb{D}$

- 1 *prompt\_enc*  $\leftarrow$  fine-tune with data of initial *update\_freq* days in a cold start  $\triangleright$  Section 4.3
- 2  $\mathbb{S} \leftarrow \emptyset$
- 3 *memory*  $\leftarrow \emptyset$
- 4 *counter*  $\leftarrow 0$
- 5 **for** every sliding window  $\mathbb{W}$  in  $\mathbb{D}$  **do**
- 6      $\mathbb{S}_{\mathbb{W}} \leftarrow$  existing stories in  $\mathbb{W}$
- 7     **for** every new article  $d \in \mathbb{W}$  **do**
- 8          $R_d^{mean} \leftarrow mean\_enc(d_i)$   $\triangleright$  Section 4.1
- 9          $R_d^{prompt} \leftarrow prompt\_enc(d_i)$   $\triangleright$  Section 4.2
- 10          $R_d \leftarrow R_d^{prompt} + R_d^{mean}$
- 11         **if**  $\max(\{sim_{d,s_j} | s_j \in \mathbb{S}_{\mathbb{W}}\}) > \theta$  **then**
- 12             Assign article  $d$  to corresponding  $s_j$
- 13             **if**  $sim_{d,s_j} > \delta$  **then**
- 14                 *memory*  $\leftarrow memory \cup (d, s_j)$
- 15             **end**
- 16         **else**
- 17              $s_{new} \leftarrow$  make a new story with  $d$
- 18              $\mathbb{S}_{\mathbb{W}} \leftarrow \mathbb{S}_{\mathbb{W}} \cup s_{new}$
- 19             *memory*  $\leftarrow memory \cup (d, s_{new})$
- 20         **end**
- 21     **end**
- 22     *counter*  $\leftarrow counter + 1$
- 23     **if**  $mod(counter, update\_freq) == 0$  **then**
- 24          $\mathbb{S}_{mem} \leftarrow$  existing stories in *memory*
- 25         **for** epoch in epochs **do**
- 26             **for** iter in iters **do**
- 27                  $\mathbb{B} \leftarrow$  a batch from  $\mathbb{S}_{mem}$  with uniform sampling
- 28                  $\mathcal{L}_{cts} \leftarrow$  contrastive loss for  $\mathbb{B}$
- 29                 *prompt\_enc*  $\leftarrow$  update with  $\mathcal{L}_{cts}$   $\triangleright$  Section 4.3
- 30             **end**
- 31         **end**
- 32         *memory*  $\leftarrow \emptyset$
- 33     **end**
- 34      $\mathbb{S} \leftarrow \mathbb{S} \cup \mathbb{S}_{\mathbb{W}}$
- 35 **end**
- 36 **return**  $\mathbb{S}$

---

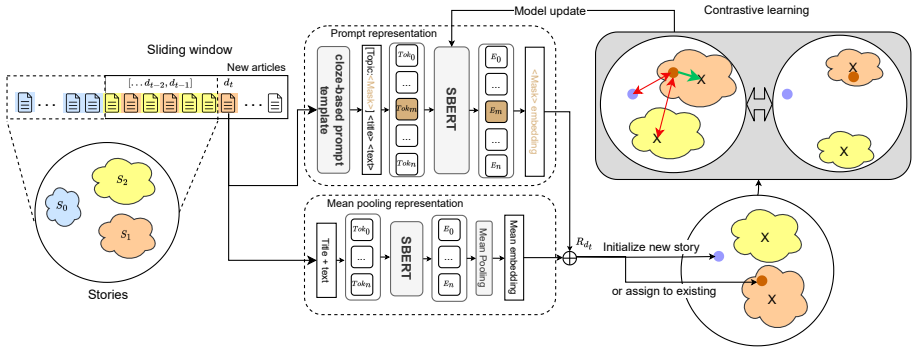


Figure 1: Architectural overview of PromptStream.

## 4.1 Topic-Aware Article Representation

The representation  $R_d$  of an article  $d$  is the sum of two distinct representations: prompt-based representation and the output of the mean pooling over the last layer of the PLM:

$$R_d = R_d^{prompt} + R_d^{mean}$$

**Prompt-Based representation ( $R_d^{prompt}$ )** In a news article, not every word carries equal significance in identifying the described events. Some words, particularly named entities, contain a wealth of crucial information. By employing suitable and task-specific prompting templates, we can create topic-aware representations that focus more on the critical aspects of the document.

To achieve this, we utilize a cloze-style prompt to extract topic-aware representations from the text. This process involves transforming the article into a cloze-based prompt using the template

`[ topic : <mask> ] <title> <body>`

where `<title>` and `<body>` represent the title and body of the news article, respectively. However, unlike the case of text classification and question-answering tasks, we do not use the label tokens predicted by the PLM classification head, but we use the output of the PLM’s last layer for the `<mask>` token as the topic-aware article representation.

**Mean-Pooling Representation ( $R_d^{mean}$ )** Cloze-based prompting focuses on specific tokens or entities within the text, making it well-suited for capturing topic-specific information. Mean pooling, on the other hand, provides a broader and more general representation of the entire document. By combining these two representations, we are effectively leveraging both the fine-grained, contextually rich information obtained from cloze-based prompting and the more holistic

and global context captured by mean pooling. This combination results in a more balanced and informative view of the document that is better suited for clustering tasks. Hence,

$$R_d^{mean} = \frac{1}{n} \sum_{i=1}^n h_i ,$$

where  $h_i$  is the embedding of token  $i$  from the last layer of a frozen PLM.

## 4.2 Online Story Assignment

**Dynamic Story Representation** The story representation  $R_{s_i}$  of story  $i$  is computed as the average of the representations  $R_{d_j}$  of the articles comprising the story:

$$R_{s_i} = \frac{1}{|s_i|} \sum_{d_j \in s_i} R_{d_j} .$$

This representation is updated each time a new article is allocated to this story.

**Article-Story Similarity** To determine which story a new document  $d_i$  in sliding window  $\mathbb{W}$  belongs to, we evaluate the similarity of document  $d_i$  with any story  $s_j \in \mathbb{S}_{\mathbb{W}}$  by using the *cosine similarity* metric as follows:

$$sim(d_i, s_j) = \cos(R_{d_i}, R_{s_j})$$

If the highest similarity between  $d_i$  and the stories in the window exceeds a predefined threshold  $\theta$ , we assign  $d_i$  to the story  $s_j$  that resulted in the highest similarity and update the representation of that story accordingly. Otherwise, we establish a new cluster with document  $d_i$  and set the cluster’s representation to  $R_{d_i}$ . Following Yoon et al. [26], we set the default value of threshold  $\theta$ , which defines the granularity of the stories, to 0.5.

## 4.3 Self-Supervised Continual Learning

We update the prompting-based encoder every  $N$  days using cluster-level contrastive learning. This loss function encourages articles to be moved closer to the center of their respective clusters while simultaneously being pushed away from other cluster centers. Updating the encoder daily with data from the same day can lead to fluctuating distributions, potentially undermining the encoder’s consistency. In addition, contrastive learning benefits from an abundance of negative examples, making it more effective to accumulate data over several days and then update the model with this aggregated dataset. Therefore, we integrate the *memory replay* concept from continual learning. This helps prevent catastrophic forgetting and ensures that the encoder remains temporally consistent as the article stream evolves.

**Confidence-Aware Memory Replay** We establish a memory bank containing data from the most recent  $N$  days, which serves as the data source for contrastive learning. In this context, we quantify the confidence of articles by their similarity with the centers of their respective stories. Only samples with confidence exceeding a predefined threshold  $\delta$  are included in the memory bank.

**Uniform Sampling** Given the varying sizes of different stories, creating training batches with a random sampler from the memory bank can potentially lead to a trivial solution. This occurs when the vast majority of articles are consistently assigned to just a few stories, causing the encoder to become biased toward those clusters and predict them for all subsequent articles. A strategy to address this issue is to sample articles using a uniform distribution across the clusters. This is equivalent to weighting the contribution of an input to the loss function by the inverse of the size of its assigned cluster. Therefore, for training the prompt-based encoder, we construct batches using a uniform sampler from the memory bank.

**Contrastive Loss** Given a batch  $\mathbb{B}$  of positive article-story pairs  $(d, s) \in \mathbb{B}$  the following contrastive loss function is utilized for fine-tuning prompting-based encoder:

$$L_{cts} = - \sum_{(d,s) \in \mathbb{B}} \log \left( \frac{e^{\cos(R_d, R_s)/\tau}}{\sum_{s' \in \mathbb{S}_w} e^{\cos(R_d, R_{s'})/\tau}} \right)$$

Here,  $\tau$  is the temperature parameter. This loss function encourages articles to be moved closer to the center of their respective clusters while simultaneously being pushed away from other cluster centers. This enhances the uniformity and alignment of the embedding space for prompt-based representations. In the initial  $N$  days where  $R_d^{prompt}$  has not yet been fine-tuned, our model relies exclusively on  $R_d^{mean}$  for embedding articles.

## 5 Experiments

We evaluate the performance of PromptStream on three labeled news datasets in Section 6.1 with common extrinsic clustering evaluation metrics. An ablation study is performed to investigate the impact of different components of the model in Section 6.2. Finally, we make a qualitative analysis to investigate the performance of the model beyond the metrics in Section 6.3.

### 5.1 Datasets

We conduct experiments on three labeled datasets that were constructed by Yoon et al. [26] from real news datasets:

**NEWS14:** This dataset consists of 16,136 articles categorized into 788 unique stories from the year 2014, sourced from the dataset introduced in [16].

**WCEP18:** This dataset was created by curating 828 news events published in 2018. It comprises 59,073 articles and has been sourced from the WCEP dataset [9].

**WCEP19:** This dataset was assembled by selecting 519 events from the year 2019, gathered from the WCEP dataset [9]. It encompasses a total of 37,637 articles.

## 5.2 Baselines

We compared PromptStream with five state-of-the-art algorithms that can be used for unsupervised and online story discovery: ConStream [1], NewsLens [13], BatClus [16], DenSps [22], and SCStory [26]. ConStream is a widely recognized streaming document clustering algorithm frequently used for story discovery. It relies on keyword-count statistics and employs incremental clustering through micro-clusters. The other three algorithms were discussed in Section 2.1. We adopted the same naming convention for these baseline methods as Yoon et al. [26]. Following Yoon et al. [26], in the case of BatClus, we employed an unsupervised setting to ensure a fair comparison. For a better assessment, we also compared our model with advanced variants of the existing algorithms that incorporate PLMs.

## 5.3 Evaluation Metrics

The evaluation metrics used were the score by Bagga and Baldwin [5], denoted by B<sup>3</sup>-F1 here, the Adjusted Rand Index (ARI) [11], and the Adjusted Mutual Information (AMI) Vinh, Epps, and Bailey [23]. B<sup>3</sup>-F1 focuses on the precision and recall of individual data points within clusters, rather than pairwise comparisons, and is considered one of the best metrics to evaluate text clustering algorithms [3]. Staykovski et al. [22] provide an extensive explanation of how it is computed. ARI is a symmetric measure that provides an overall assessment of clustering quality, considering both pairwise agreements and disagreements. AMI also favors clusterings where data points that are similar are placed into the same cluster, but it is less sensitive to clusterings that divide ground truth classes into multiple clusters. ARI and AMI are both adjusted for chance agreement. All three metrics have a maximum score of 1.

Following [26], Table 2 reports the average scores for each metric across each sliding window over the data streams. Table 3 presents these metrics over the complete data streams, an approach we also employ in our analysis detailed in Section 6.2.



	NEWS14			WCEP18			WCEP19		
	B <sup>3</sup> -F1	AMI	ARI	B <sup>3</sup> -F1	AMI	ARI	B <sup>3</sup> -F1	AMI	ARI
ConStream†	0.314	0.128	0.069	0.408	0.444	0.222	0.400	0.497	0.292
NewsLeans†	0.481	0.309	0.077	0.527	0.490	0.117	0.554	0.529	0.141
BatClus†	0.706	0.726	0.572	0.694	0.786	0.571	0.698	0.791	0.574
DenSps†	0.669	0.602	0.358	0.697	0.759	0.487	0.701	0.765	0.487
ConStream+SBERT†	0.434	0.413	0.276	0.701	0.784	0.657	0.704	0.795	0.667
NewsLeans+SBERT†	0.749	0.718	0.564	0.767	0.823	0.631	0.784	0.887	0.664
BatClus+SBERT†	0.764	0.785	0.648	0.751	0.835	0.656	0.759	0.837	0.657
DenSps+SBERT†	0.750	0.720	0.567	0.754	0.824	0.624	0.762	0.830	0.660
SCSTory+SBERT	0.895	<b>0.873</b>	<b>0.837</b>	0.867	0.876	0.809	0.873	0.89	0.83
PromptStream	<b>0.915</b>	0.845	0.835	<b>0.913</b>	<b>0.885</b>	<b>0.863</b>	<b>0.919</b>	<b>0.904</b>	<b>0.887</b>

Table 2: The average B<sup>3</sup>-F1, AMI, and ARI over **each sliding window** in the article streams. For PromptStream and SCSTory, the scores are the average of five different runs with different random seeds. Scores marked with † are included from Yoon et al. [26] to self-contain this paper.

	NEWS14			WCEP18			WCEP19		
	B <sup>3</sup> -F1	AMI	ARI	B <sup>3</sup> -F1	AMI	ARI	B <sup>3</sup> -F1	AMI	ARI
SCSTory+SBERT	0.806	0.862	0.294	0.799	0.904	0.628	0.820	0.917	0.718
PromptStream	<b>0.843</b>	<b>0.898</b>	<b>0.610</b>	<b>0.825</b>	<b>0.916</b>	<b>0.644</b>	<b>0.852</b>	<b>0.931</b>	<b>0.766</b>

Table 3: B<sup>3</sup>-F1, ARI, and AMI over the **entire** article streams. For PromptStream and SCSTory, the results are the average scores of five different runs with different random seeds.

## 5.4 Experiment Settings

We implemented our model in PyTorch [17] with Transformer Library [25] and chose sentence-transformers/all-roberta-large-v1<sup>1</sup> as the PLM for both prompting-based and mean pooling based encoders. This is a roberta-large [14] model well-suited for tasks such as clustering and semantic search. For training the prompting-based encoder, we used the AdamW [15] optimizer with a batch size of 64, and a learning rate of 5e-6. We set the max sequence length for the tokenizer to 128. This choice may be advantageous because, in news articles, the most informative content is typically found in the title and the introductory section of the text. The  $\theta$  and  $\delta$  thresholds were both set to 0.5. The window and sliding sizes were 3 and 1, respectively, in both our model and our runs for SCSTory. The temperature for contrastive loss for all datasets was 0.2. We updated the prompting-based encoder every 10 days. Regarding the parameters for SCSTory, with the exception of the window size, which we adjusted to 3, we maintained the default values as reported in the paper.

<sup>1</sup><https://huggingface.co/sentence-transformers/all-roberta-large-v1>

## 6 Results and Discussions

Here we present the performance evaluation, the ablation study, and the qualitative cluster analysis.

### 6.1 Overall Performance

Tables 2 and 3 provide a comparison between the baseline models and PromptStream for the online story discovery task. As shown in Table 2, PromptStream exhibits superior performance compared to SCStory achieving a 3.7% higher average B<sup>3</sup>-F1 score over sliding windows of the entire data stream across all datasets. Furthermore, with respect to other metrics, it outperforms SCStory in most cases. When assessing metrics across the entire data stream, as presented in Table 3, it becomes evident that PromptStream surpasses SCStory in terms of B<sup>3</sup>-F1, AMI, and ARI by an average of 3.1%, 2%, and 12.7%, respectively, across all three datasets.

Both PromptStream and SCStory perform superior to the other baselines, which we attribute to the fact that they both use attention to compute representations that emphasize the relevant parts of each article.

### 6.2 Ablation Study

	NEWS14	WCEP18	WCEP19
PromptStream (default)	0.843	0.825	0.853
w/o prompt-based rep.	0.813	0.767	0.793
w/o mean rep.	0.831	0.814	0.843
w/o uniform sampler	0.842	0.807	0.842
Updating prompting-based encoder			
No update	0.564	0.493	0.525
Only with first 10 days	0.833	0.818	0.842
Every 5 days	0.836	0.822	0.846
Every 15 days	0.843	0.824	0.854
Prompting templates			
(This news is about: ;mask <sub>i</sub> ) [title] [body]	0.845	0.823	0.851
;mask <sub>i</sub> [title] [body]	0.843	0.822	0.851
Keywords: ;mask <sub>i</sub> \n [title] \n [body]	0.845	0.823	0.854
[title] \n [body] \n Keywords: ;mask <sub>i</sub>	0.844	0.817	0.855

Table 4: B<sup>3</sup>-F1 results from PromptStream ablation study with various configurations. Since the variation of the results for different seeds is very low, we report the results only for one run.

Table 4 presents the results of the ablation study. In most cases, the scores closely resemble those of the default model. However, two notable outliers are worth mentioning. Firstly, when the prompting-based representation is removed, there is a substantial drop in the B<sup>3</sup>-F1 score. Secondly, the omission of updates to the prompting-based encoder leads to a dramatic reduction in

this score. These results strongly suggest that the main technical contributions of our proposal, the fine-tuned prompting-based representation, and the self-supervised continual learning, indeed play a central role in achieving superior performance.

The results indicate that continual training results in representations that outperform those trained only for the initial 10 days. Additionally, as previously discussed in Section 4.3, the effectiveness of contrastive loss is highly dependent on the number of negative examples. In our model, these negative examples correspond to the centers of clusters other than the cluster an article belongs to. Therefore, it is advantageous to maintain a larger memory bank with more clusters for contrastive learning. Reducing the update frequency, which effectively retains data for more days in the memory bank, increases the likelihood of having more clusters in the memory bank. As seen in Table 4, updating the prompt-based encoder less frequently, e.g. every 15 days instead of every 5 days, increases the performance.

A surprising finding was the relatively minor impact of the prompting templates on the final scores. It appears that merely having a prompt-based representation in place is sufficient to improve performance. Notably, the scores achieved by the prompt-based representation on its own are only slightly lower than those of the default model.

Furthermore, the results reveal that mean-pooling representations and the uniform sampler significantly contribute to the overall performance of the model.

In Figure 2, we compare the performance of PromptStream and SCStory with regard to  $B^3$ -F1 over the entire data stream while varying the window sizes. As the figure illustrates, our model consistently outperforms SCStory and demonstrates greater stability. However, it is noteworthy that the performance of both models tends to decline as the window size increases. This makes sense because the window size signifies the time period of our interest in the stories, and if it becomes excessively large, the models may merge events with similar themes rather than detecting fine-grained events.

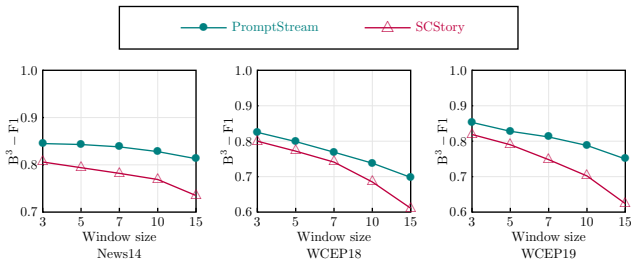


Figure 2: Comparison of PromptStream performance with that of SCStory for different window sizes, using the  $B^3$ -F1 score for the entire stream.

Table 5: Five stories from WCEP18 that PromptStream divided into multiple smaller stories, resulting in low performance on the evaluation. The story\_id, n\_articles, and n\_pred\_labels are the story’s gold label, the story size, and the number of clusters generated by PromptStream respectively. The theme is a general description of the articles in the story which all could be argued to contain multiple sub-stories.

story_id	n_articles	n_pred_labels	Theme
64606	54	28	Opinion about Donald Trump and his expressions <b>Keywords:</b> <i>Trump, president, Haiti, mocks, Modi</i>
67432	72	22	General reporting related to Calgary and British Colombia <b>Keywords:</b> <i>Calgary, family, accident, Stampeders (sports team), wildfires</i>
64773	51	20	American foreign policy and international politics <b>Keywords:</b> <i>Trump, election, Venezuela, Jerusalem, U.S.</i>
66490	68	18	Financial markets related to health and tech <b>Keywords:</b> <i>global, market, treatment, Vodafone, U.S.</i>
65030	66	17	U.S. foreign policy dominated by trade agreements and North Korea <b>Keywords:</b> <i>Trump, north, Korea, tariffs, NAFTA</i>

### 6.3 Qualitative Analysis

To investigate the quality of the clustering into stories we made a basic qualitative analysis of the WCEP18 dataset. The size distribution of the gold labels is rather balanced with the largest story containing 82 articles. In contrast, the largest cluster found by PromptStream comprised 603 articles, suggesting that improvements can be made to the model. The cluster contains news about North and South Korean progress on peace and denuclearization spanning over a month in April–May 2018. This could rightfully be considered a coherent story from a worldwide perspective but one could also argue that the gold labels ( $n = 15$ ) contain individual stories within this larger event.

To further the analysis, we took the five stories that PromptStream had split into the largest number of sub-stories. These have the most negative impact on the model performance and are therefore interesting to investigate for model improvement. In Table 5 we see a summary of the stories and the number of predicted labels by PromptStream. It is fair to say that the clusters in the table do not describe coherent stories, even though the gold labels suggest so. E.g. the story with id 67432 contains general reporting on news in Calgary and British Colombia concerning e.g. local politics, accidents, and sports events. PromptStream has made a finer division of this story into such themes. The same is evident when analyzing the articles in the story with id 64606 centered around various opinions about Trump as a president from different perspectives. Its sub-stories revolve around Trump mocking the accent of the president of India, and Haitians protesting Trump’s derogatory comments, which are individually captured by the model. This indicates that better datasets are needed to test the capacity of extracting granular stories.

For this study, we conclude that PromptStream is performing adequately even though the scores were reduced due to limitations of the gold labeling.

## 7 Conclusion

We introduced PromptStream as a novel approach to unsupervised online story discovery. PromptStream combines a cloze-based prompt representation with mean pooling representation from SBERT to embed articles, ensuring a balance between the article’s topic-specific information and a more general representation of the entire document. These representations are continuously updated throughout the stream with contrastive learning using a memory of the recent confident article-story assignments. This process refines the prompt-based representations and aligns them with the latest context within the news stream. In the evaluation of three labeled datasets, our model demonstrated performance improvements over the previous state of the art. Further, the subsequent ablation study highlighted the efficacy of prompt-based representation and continual training.

## Data and Code Availability

The datasets are publicly available. The code for PromptStream is available at <https://github.com/Aha6988/PromptStream>.

## Acknowledgments

We would like to extend our sincere gratitude to *Frank Drewes*. His insightful comments and feedback greatly aided us in improving the quality and clarity of our work.

## References

- [1] Charu Aggarwal and Philip Yu. “On clustering massive text and categorical data streams”. In: *Knowledge and Information Systems* 24 (Aug. 2010), pp. 171–196. DOI: 10.1007/s10115-009-0241-z.
- [2] J. Allan et al. “Topic Detection and Tracking Pilot Study: Final Report”. In: *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. 007. Lansdowne, VA, USA, Feb. 1998, pp. 194–218.
- [3] Enrique Amigó et al. “Amigó E, Gonzalo J, Artiles J et alA comparison of extrinsic clustering evaluation metrics based on formal constraints. Inform Retrieval 12:461-486”. In: *Information Retrieval* 12 (Oct. 2009), pp. 461–486. DOI: 10.1007/s10791-008-9066-8.
- [4] Dhananjay Ashok and Zachary Chase Lipton. “PromptNER: Prompting For Named Entity Recognition”. In: *ArXiv* abs/2305.15444 (2023). URL: <https://api.semanticscholar.org/CorpusID:258887456>.

- [5] Amit Bagga and Breck Baldwin. “Entity-Based Cross-Document Coreferencing Using the Vector Space Model”. In: *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*. Montreal, Quebec, Canada: Association for Computational Linguistics, Aug. 1998, pp. 79–85. DOI: 10.3115/980845.980859. URL: <https://aclanthology.org/P98-1012>.
- [6] Vincent D Blondel et al. “Fast unfolding of communities in large networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (Oct. 2008), P10008. DOI: 10.1088/1742-5468/2008/10/P10008. URL: <https://dx.doi.org/10.1088/1742-5468/2008/10/P10008>.
- [7] Tom Brown et al. “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/1457c0d6bfbcb4967418bfb8ac142f64a-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfbcb4967418bfb8ac142f64a-Paper.pdf).
- [8] Tianyu Gao, Adam Fisch, and Danqi Chen. “Making Pre-trained Language Models Better Few-shot Learners”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 3816–3830. DOI: 10.18653/v1/2021.acl-long.295. URL: <https://aclanthology.org/2021.acl-long.295>.
- [9] Demian Gholipour Ghalandari et al. “A Large-Scale Multi-Document Summarization Dataset from the Wikipedia Current Events Portal”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 1302–1308. DOI: 10.18653/v1/2020.acl-main.120. URL: <https://aclanthology.org/2020.acl-main.120>.
- [10] Elad Hoffer and Nir Ailon. “Deep metric learning using triplet network”. In: *Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*. Springer. 2015, pp. 84–92.
- [11] Lawrence Hubert and Phipps Arabie. “Comparing partitions”. In: *Journal of Classification* 2 (1985), pp. 193–218. ISSN: 01764268. DOI: 10.1007/BF01908075.
- [12] Ting Jiang et al. “PromptBERT: Improving BERT Sentence Embeddings with Prompts”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 8826–8837. DOI: 10.18653/v1/2022.emnlp-main.603. URL: <https://aclanthology.org/2022.emnlp-main.603>.

- [13] Philippe Laban and Marti Hearst. “newsLens: building and visualizing long-ranging news stories”. In: *Proceedings of the Events and Stories in the News Workshop*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 1–9. DOI: 10.18653/v1/W17-2701. URL: <https://aclanthology.org/W17-2701>.
- [14] Yinhan Liu et al. “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692* (2019).
- [15] Ilya Loshchilov and Frank Hutter. “Decoupled Weight Decay Regularization”. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: <https://openreview.net/forum?id=Bkg6RiCqY7>.
- [16] Sebastiao Miranda et al. “Multilingual Clustering of Streaming News”. In: Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 4535–4544. DOI: 10.18653/v1/D18-1483. URL: <https://aclanthology.org/D18-1483>.
- [17] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.
- [18] Kailash Karthik Saravanakumar et al. “Event-Driven News Stream Clustering using Entity-Aware Contextual Embeddings”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 2330–2340. DOI: 10.18653/v1/2021.eacl-main.198. URL: <https://aclanthology.org/2021.eacl-main.198>.
- [19] Timo Schick and Hinrich Schütze. “Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 255–269. DOI: 10.18653/v1/2021.eacl-main.20. URL: <https://aclanthology.org/2021.eacl-main.20>.
- [20] Yongliang Shen et al. “PromptNER: Prompt Locating and Typing for Named Entity Recognition”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 12492–12507. DOI: 10.18653/v1/2023.acl-long.698. URL: <https://aclanthology.org/2023.acl-long.698>.
- [21] Jonathan A. Silva et al. “Data Stream Clustering: A Survey”. In: *ACM Comput. Surv.* 46.1 (July 2013). ISSN: 0360-0300. DOI: 10.1145/2522968.2522981. URL: <https://doi.org/10.1145/2522968.2522981>.

- [22] Todor Staykovski et al. “Dense vs. Sparse Representations for News Stream Clustering”. In: *Proceedings of Text2Story - 2nd Workshop on Narrative Extraction From Texts, co-located with the 41st European Conference on Information Retrieval, Text2Story@ECIR 2019, Cologne, Germany, April 14th, 2019*. Ed. by Alípio Mário Jorge et al. Vol. 2342. CEUR Workshop Proceedings. CEUR-WS.org, 2019, pp. 47–52. URL: <https://ceur-ws.org/Vol-2342/paper6.pdf>.
- [23] Nguyen Xuan Vinh, Julien Epps, and James Bailey. “Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance”. In: *Journal of Machine Learning Research* 11.95 (2010), pp. 2837–2854. URL: <http://jmlr.org/papers/v11/vinh10a.html>.
- [24] Tongzhou Wang and Phillip Isola. “Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere”. In: *Proceedings of the 37th International Conference on Machine Learning. ICML’20*. JMLR.org, 2020.
- [25] Thomas Wolf et al. “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: <https://aclanthology.org/2020.emnlp-demos.6>.
- [26] Susik Yoon et al. “SCStory: Self-Supervised and Continual Online Story Discovery”. In: *Proceedings of the ACM Web Conference 2023. WWW ’23*. Austin, TX, USA: Association for Computing Machinery, 2023, pp. 1853–1864. ISBN: 9781450394161. DOI: 10.1145/3543507.3583507. URL: <https://doi.org/10.1145/3543507.3583507>.
- [27] Zhenrui Yue et al. “Zero- and Few-Shot Event Detection via Prompt-Based Meta Learning”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 7928–7943. DOI: 10.18653/v1/2023.acl-long.440. URL: <https://aclanthology.org/2023.acl-long.440>.
- [28] Senhui Zhang et al. “Zero-Shot Event Detection Based on Ordered Contrastive Learning and Prompt-Based Prediction”. In: *Findings of the Association for Computational Linguistics: NAACL 2022*. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 2572–2580. DOI: 10.18653/v1/2022.findings-naacl.196. URL: <https://aclanthology.org/2022.findings-naacl.196>.



---

**METHOD: A Dataset and Baseline for Multimodal  
Discovery of Event-Based News Topics**

Arezoo Hatefi, Johanna Björklund, Xuan-Son Vu, and Frank Drewes

*Submitted to the International Journal of Multimedia Information Retrieval,  
2024.*



# METHOD: A Dataset and Baseline for Multimodal Discovery of Event-Based News Topics

Arezoo Hatefi, Johanna Björklund, Xuan-Son Vu, Frank Drewes

*Department of Computing Science, Umeå University, Umeå, Sweden*

*arezoo@cs.umu.se, johanna@cs.umu.se, sonvx@cs.umu.se, drewes@cs.umu.se*

**Abstract:** We propose event-based topic discovery in a text-image data stream of news articles as an important and challenging problem in the larger field of topic discovery. To enable researchers to develop and evaluate methods for this task, we provide METHOD, an annotated dataset of news articles from the New York Times. Each news article consists of text and image data. Finally, we develop a baseline algorithm and analyze its performance on METHOD.

**Key words:** event-based topic, topic discovery, multimodal news, data stream

## 1 Introduction

The number of news articles published daily is vast and continuously growing. For example, Reuters alone averages some 5 000 articles written by their 2 500 associated journalists<sup>1</sup>. While initially designed to inform and entertain individuals, online news has now become a crucial data source for numerous modern information retrieval systems. For instance, automated news monitoring can aid organizations in staying current with industry trends [36], generating market forecasts [23], and facilitating brands in placing their ads in suitable media contexts [51]. Additionally, techniques like document clustering [7] and summarisation [5] assist users in obtaining an overview of the global state of affairs, with topic detection [27] and sentiment analysis [40] adding further depth.

In this paper, we focus on the discovery of fine-grained topics linked to the distinct events such as a particular election, accident, or natural disaster. This is in contrast to classical topic analysis, which clusters articles into thematic categories which can be more or less fine grained but are typically “timeless”, such as natural disasters, culture, sports, or crime. For example, the topic *Hurricanes* could cover all news articles about hurricanes whereas event-based topic discovery would distinguish between the topics *Hurricane Eta* and *Hurricane Iota* in

---

<sup>1</sup><https://www.reutersagency.com/en/about/about-us/>

## Event-based Topics

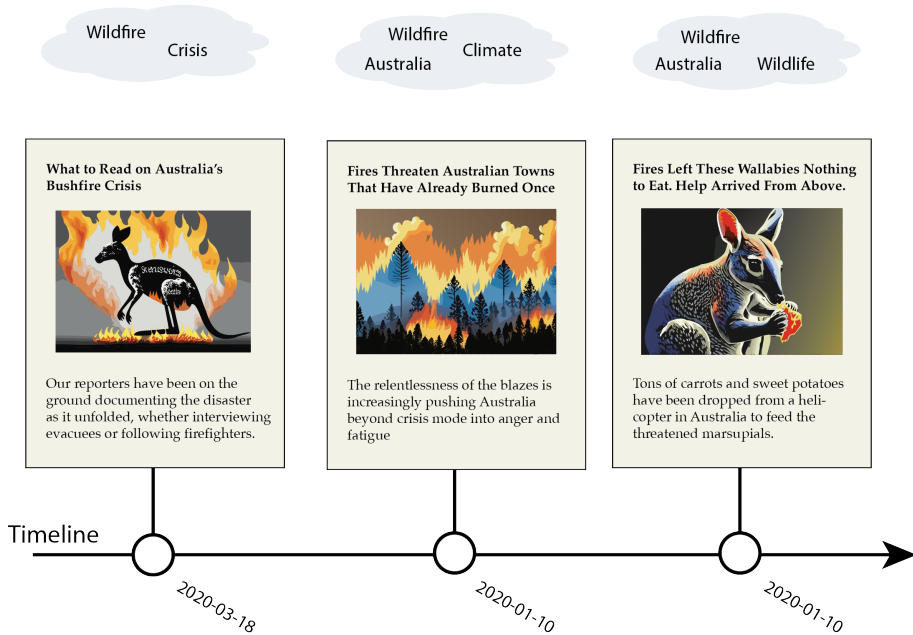


Figure 1: Event-based news topic discovery monitors news streams with the goal to group articles into topics according to their underlying real-world events.

the news of 2020. Figure 1 demonstrates a stream of news articles including coverage of the *Bushfire Crisis in Australia in 2020*. At first glance, it seems that all news articles refer only to the fire crisis. However, they are all referring to a specific event and form an event topic. We believe that this more detailed level of analysis is better suited for automated news monitoring, as it allows us to separate the news flow into individual story lines. However, as we shall see, it is not a trivial processing tasks because also relatively simple events often give rise to a cascade of articles, each addressing a distinct aspect, perspective, or implication of the event itself.

Another consideration is that online news reporting is typically multimodal in nature, in that it combines text, images, video, audio, and different types of tabular and otherwise structured data to convey its message. This arguably makes communication more efficient, as each modality has its own strengths

and limitations, and the combination of modalities can help create a richer and more immersive user experiences. In recent time, social media has grown in importance as a news channel for the general public [15], and the posts shared frequently combine textual and visual elements. The strengths of social media is particularly noticeable during crises, where they help shape the public discourse surrounding events.

In contrast, the research community has traditionally equated news with textual data, which is reflected in many of the analysis methods and datasets currently available. However, we have reached a juncture where this unimodal viewpoint hinders progress by neglecting the information carried by parallel modalities such as audio and images. The value of leveraging a wider set of modalities is demonstrated by recent studies which show the superiority of incorporating both images and text in news classification, as opposed to relying solely on text [43, 18].

It is an open research question to what extent the consideration of images in the analysis of event-based topics is helpful. For example, an image showing protesters may not easily be attributed to a specific event, but together with the text of the article and the information about when it was published, it may very well provide important information. Other images, such as an image of people on a boat who have escaped the wildfires in Australia<sup>2</sup> may even be confounding. To enable researchers to study aspects like these, datasets are needed. In this work, we provide such a dataset, based on articles published by the New York Times in 2020 and 2021. We also adapt a state-of-the-art model for the discovery of event-based topics in the unimodal setting to operate in the text-image setting of news articles with images. Our experiments with this model provide a baseline for further research.

In summary, our main contributions are these:

- We propose event-based topic discovery in a text-image setting as an important and challenging problem in the larger field of topic discovery.
- We provide an annotated dataset for the development and evaluation of algorithms solving this problem.
- We present a baseline algorithm and analyze its performance on the above-mentioned dataset.

## 2 Related Work

This section reviews two categories of prior research relevant to this study: (1) topic detection and tracking, and (2) relevant datasets in news monitoring and clustering tasks.

---

<sup>2</sup>[https://www.nytimes.com/images/2020/01/04/world/04oz-fire-4/merlin\\_166593555\\_156de443-72ec-4068-83c8-308ca3b37f07-jumbo.jpg](https://www.nytimes.com/images/2020/01/04/world/04oz-fire-4/merlin_166593555_156de443-72ec-4068-83c8-308ca3b37f07-jumbo.jpg)

**Topic detection and tracking.** A field of research related to event-based topic discovery is *topic detection and tracking* proposed by Allan [3]. By definition, it studies the problem of segmenting news shows (transcribed or based on audio and perhaps video) into individual segments called stories. A story is a part of the continuous news stream of the show which is centered around the same event.

Topic detection and tracking differs significantly from event-based topic discovery. In the latter, segmentation is not an issue because every news article is an individual story. Instead, the goal is to group the stream of individual news articles into topics at the event level.

The predominant focus in research on event-based topic and story discovery within news streams revolves around the exploration of textual news content, see [20, 26, 37, 50, 9, 16]. Previous work in the area of multimodal news analysis has mostly focused on the thematic classification of news [43, 18, 32] and on fake news detection [46, 53, 52, 48]. The only multimodal work in topic detection and tracking that we are aware of is Li et al. [25], who focus on topic detection and tracking in video news.

**Datasets in news clustering.** Table 1 shows details of popular news datasets [26, 10, 21, 11, 12, 49, 19, 34, 2, 54, 6, 28, 43, 17, 1, 45, 31, 55]. Some datasets are multimodal (e.g., text and image [34, 2], video [45]) while in some cases the data is text only [26, 10, 21, 11, 12, 49, 19]. The data source is diverse, from social media [31, 1] and Twitter [6] to popular news outlets such as Guardian News [12], Yahoo News [49], BBC News [19], Reuters [6], New York Times [43], Breaking News [34], Washington Post [2], Wall Street Journal [17], and The Onion [17]. Most datasets have no timestamps, examples being [21, 11, 43, 17].

Upon reviewing the multimodal datasets, we have identified several significant issues:

- Most of the datasets are not specifically curated for event-based topic discovery from a continuous stream of news articles. Instead, they are designed for other purposes, such as news video analytics [29] or fake news detection [28, 54]. In particular, the collected articles are not necessarily related to specific events and the class labels are not event-based. The datasets that do focus on events in news articles are text-only [26, 10].
- Several datasets lack timestamps and are not structured as continuous streams, hindering their suitability for our research objectives [43].
- Some datasets are sourced from social media platforms like Twitter, resulting in texts quite different in style from traditional news articles [55, 1].
- Some datasets, while based on events [1], have a limited range of classes, which does not align with the extent of topics we aim to explore.
- Additionally, some datasets are not publicly available and require an approval process (e.g., [2]), which poses challenges for benchmarking and

Table 1: Comparison between different news datasets, alongside our METOD dataset. \* denotes that the short dataset name is given by us. + indicates that an approval process is required to gain access to the data.

#	Dataset	Size	Classes	Type	Source	Type	Timestamp
1	20NEWS [21]	20,000	20	text	Newsgroup	real news	no
2	AG NEWS [11]	1,000,000	4	text	AG News	real news	no
3	Guardian News [12]	52,900	4	text	Guardian News	real news	no
4	Yahoo News [49]	160,515	31	text	Yahoo	real news	no
5	BBC News [19]	2,225	5	text	BBC	real news	no
6	Miranda2018* ( <i>English</i> ) [26]	20,959	632	text	RSS Feeds	real news	yes
7	WCEP [10]	2,390,000	10,200	text	Wikipedia Current Events Portal	real news	yes
8	BreakingNews [34]	110000	none	text, image	RSS Feeds	real news	no
9	TREC Washington Post+ [2]	728,626	none	text, image	Washington Post	real news	yes
10	Fauxtography [54]	1,233	2	text, image	Snopes, Reuters	fake news	no
11	Image-verification-corpus [6]	17,806	2	text, image	Twitter	fake news	yes
12	Fakeddit [28]	1,063,106	2,3,6	text, image	Reddit	fake news	no
13	N24News [43]	61,218	24	text, image	New York Times	real news	no
14	NewsBag [17]	215,000	2	text, image	Wall Street Journal & The Onion	real & fake news	none
15	MEED [42]	37,807	66	text, image	none	none	none
16	PHEME [55]	60,000	9	text, image	Twitter	rumours & non-rumours	none
17	CrisisMMD [1]	16,097	7	text, image	Twitter	real news	yes
18	Wu2011* [45]	19,972	22	news video	none	none	yes
19	Qian2018* [31]	13,637	8	text, image	Social media	real news	yes
20	<b>METOD</b> (ours)	902	51	text, image	New York Times	real news	yes

reproducibility.

The existing multimodal news datasets exhibit at least one of these issues. Therefore, our objective is to construct a new dataset, METOD, designed for the multimodal discovery of event-based news topics in a stream of news articles.

### 3 The METOD Dataset

As explained in the introduction, one of the central objectives of this work is to provide a multimodal dataset of news articles, annotated with event topics. We endeavour to include a wide variety of topics and provide statistics regarding the characteristics of each. The purpose of these statistics is to make it easier for researchers to (a) select subsets of the topics according to target properties, thus enabling the study of event-topic discovery algorithms for specialized cases, (b) formulate and test hypotheses regarding the influence of certain characteristics on the performance of algorithms, and (c) explore unanticipated correlations between the performance of algorithms and the characteristics of topics they discover or overlook.

In this section, we describe the dataset and the characteristic properties we expect to be relevant. We should note here that in all instances where we refer

to specific pictures in the dataset, including Tables 2, 5, and 6, we have used AI-generated pictures instead of the real news images due to copyright issues. However, we also provide links to the original news articles to access the original images.

### 3.1 Dataset Collection

The New York Times (NYT) is a well-established newspaper that has been in publication since 1851. It is one of the most influential of its kind in the United States, covering a range of topics such as politics, business, culture, and sports. To support research and development efforts, the NYT provides access to its archive through an Application Programming Interface (API), from which various types of content and data can be retrieved.

We take advantage of the Archive API to curate a multimodal dataset for research in event-based topic discovery. From the NYT articles published in 2020–2021 in the *World* section, we selected 902 event-related articles consisting of both text and an image. Each record describing an article contains the attributes ‘headline’, ‘abstract’, ‘leading paragraph’, ‘date’, ‘section’, ‘subsection’, ‘keywords’, ‘url’, and ‘image url’.

We grouped the 902 selected articles into 51 event-based topics and labeled them accordingly. The resulting dataset is referred to as METOD<sup>3</sup>. Despite their availability in the original data, we opted not to include the attributes section, subsection, and keywords in METOD, as they are specific to the NYT data source.

A sample entry from the METOD dataset is presented in Table 2. As we can see, it consists of an event-level topic name, together with links to the article image, and the published article itself. Then follow three text blocks, namely headline, abstract and leading paragraph. The last entry shows the publication date and time of the article.

Due to the constraints outlined in the NYT Terms of Service, we are unable to publish the full METOD dataset, which includes all the attributes such as abstracts and leading paragraphs. Consequently, we have shared a modified version of the dataset on our GitHub repository<sup>4</sup>, containing only the headlines and URLs of articles, along with event labels. However, within the repository, we provide Python code that facilitates the reconstruction of the complete dataset.

It is worth mentioning that the code also generates and stores the original raw dataset comprising of 10 283 unlabeled records, referred to as the RAW dataset. This is to encourage the expansion of the METOD dataset by tagging additional topics from the RAW dataset.

---

<sup>3</sup>Multimodal Event-based Topic Detection

<sup>4</sup>[https://github.com/Aha6988/METOD\\_dataset](https://github.com/Aha6988/METOD_dataset)



Table 2: A document in the METOD dataset.

<b>Topic</b>
Wildfire in Australia
<b>Image</b>

<b>URL</b>
<a href="https://www.nytimes.com/2020/01/10/world/australia/bushfire.html">https://www.nytimes.com/2020/01/10/world/australia/bushfire.html</a>
<b>Headline</b>
What to Read on Australia's Bushfire Crisis
<b>Abstract</b>
Our reporters have been on the ground documenting the disaster as it unfolded, whether interviewing evacuees or following firefighters.
<b>Leading Paragraph</b>
Australians have started the new year anxious and alarmed with unprecedented bushfires engulfing parts of the country, causing thousands to flee the southeastern coast under blood-red skies.
<b>Publication date</b>
2020-01-10 05:19:58

### 3.2 Characteristics of Event-Based Topics

In contrast to general topics, event-based ones usually have a limited life span. The first article typically appears shortly after the event has occurred. After some time, the reporting stops (e.g., if the situation has been resolved) or fades away together with public interest. Further, topics usually evolve more or less significantly, e.g. from an armed conflict to peace negotiations and retrospective analyses. A more coarse-grained analysis would classify such a set of articles as one topic, especially if the changes are gradual ones, while a more fine-grained analysis would discover several topics. One could also imagine algorithms that

would compute a hierarchy of topics labeled by events and sub-events triggered by them. In METOD, we have tried to find reasonably distinct major topics which may include subtopics.

We now discuss characteristics of topics in METOD that set the problem of discovering event-based topics apart from general topic discovery. Some of these characteristics can to a satisfactory extent be captured by statistics (which we provide along with our dataset), while others are difficult to quantify, but may still be expected to bear significance for the performance of topic discovery algorithms.

### *1. Topic size*

The topic size, i.e., the number of articles a topic consists of, is one of the most basic characteristics. In the METOD dataset, this characteristic varies a lot, and usually relates to how important (often in the sense of devastating) the event is considered to be. The smallest and largest topics consist of 2 and 125 articles, respectively. Note that, as with all the other characteristics below, our analysis refers to the concrete set of articles that have been selected as belonging to the topic, rather than the intuitive topic with its fuzzy border.

### *2. Topic duration*

Naturally, the duration of a topic is defined to be the time that elapses between the publication of the first article in the topic and the last one. We count topic duration in whole days (rounded upwards). In METOD topic duration ranges from 2 to 718 days.

### *3. Article frequency*

Article frequency is a derived attribute: it is the number of articles divided by the topic duration. We expect topics of low frequency to be more difficult to discover in a data stream than topics characterized by a high frequency. This is because temporal proximity of similar articles is a strong indicator that they belong to the same topic. If this indicator is missing in a topic, it becomes harder to discover that the articles do in fact relate to the same event. The article frequency of topics in METOD ranges from 0.008 to 4.5 articles per day.

### *4. Temporal irregularity*

In the study of (infinite) time series, the Hurst Exponent [14] plays an important role. It measures the autocorrelation of the time series and how quickly it diminishes with an increase in the lag between pairs of values. The Hurst Exponent is defined as  $\lim_{n \rightarrow \infty} \frac{\log(R(n))}{\log(S(n))}$ , where  $R(n)$  and  $S(n)$  are the range and the standard deviation of the first  $n$  observations, respectively. While event-based topics can be viewed as (finite) time series, we do not expect their autocorrelation to bear special significance. However, depending on the method used, event-based topic discovery may be more challenging if the temporal distribution of the topic in question is very irregular. We thus define the temporal irregularity of a topic to be the standard deviation of the amount of time in (fractions of) days which passes between each two subsequent articles

belonging to the topic. High irregularity means that articles are unevenly distributed over time, which can make it difficult to determine that they belong to the same topic. METHOD covers topics whose irregularity ranges between 0 (which is necessarily the case for topics of size 2) and 77.

### 5. *Disconnectedness index*

Intuitively, a topic is disconnected if there are significant “holes” in its temporal distribution. This property is related to but not the same as temporal irregularity. Disconnected topics usually occur if public interest in a topic reawakens due to new developments or the discovery of new facts. Such temporal holes in topics are not only a potentially complicating factor for the discovery of event-based topics. They also raise the question whether it is actually appropriate to classify the topic as one. For example, news articles may report a particularly cruel murder case. Months later the police arrests a suspect, and after another few months later the court trial raises awareness a third time. Whether this is one topic based on a single event (the murder of  $X$ ) or three, based on three related but different events (the actual murder, the arrest, and the trial) is a matter of perspective and cannot be answered affirmatively.

To quantify disconnectedness by a number between 0 and 1, we define the *disconnectedness index* of a topic to be

$$\tanh(\log(h/a))$$

where  $h$  and  $a$  are the maximum and the average, respectively, of the amount of time between two successive articles in the topic.<sup>5</sup> Thus, this value is 0 if the articles are perfectly spaced and approaches 1 as the largest hole becomes more pronounced relative to the average time between articles. The disconnectedness index of topics in METHOD ranges between 0 and 0.83.

### 6. *Suddenness*

The term suddenness describes a more loosely defined characteristics of topics, namely how distinct and sudden the defining event is. Natural disasters like earthquakes and tsunamis are prime examples of sudden events which immediately cause a rapid sequence of very similar articles to appear. Such topics should be comparatively easy to detect. Other topics creep into existence by less sharply defined events such as political protests which gain traction over a couple of days or even weeks, or topics the events of which are known beforehand. In METHOD, the assassination of Haitian president Jovenel Moïse is one of many examples of a sudden event. The war over Nagorno-Karabakh is an example of the opposite, as at the beginning it was not even clear that the fighting would escalate to an all-out war. A less violent example is the European Song Contest 2021, which was foreshadowed by articles before the actual event. We rate the suddenness of topics as either low, medium, or high. Naturally, this categorization is somewhat subjective and should thus be used with care.

---

<sup>5</sup>Formally, if the topic comprises  $k + 1$  articles with publication times  $t_0 < \dots < t_k$ , then  $h = \max\{t_i - t_{i-1} \mid 1 \leq i \leq k\}$  and  $a = \frac{1}{k} \sum_{i=1}^k t_i - t_{i-1}$ .

### 7. Specificity

Specificity refers to how clear the borderline between in-topic and out-of-topic articles is. Big topics often appear to have a fuzzy border, an example from METOD being the seizing of control by the Taliban in Afghanistan. Do the preparations by the US government and military to leave the country belong to the topic? How about the subsequent chaotic evacuation efforts and later reports about the worsening situation of women in Afghanistan under Taliban rule? Just like suddenness, specificity is a somewhat subjective quality which we rate as either low, medium, or high in METOD.

### 8. Image informativeness

An aspect that deserves further study, both in general and with respect to particular characteristics, is how informative the article images are. Generic or merely decorative images can be expected to be of little use for topic discovery. However, articles about wildfire events such as those in Australia and Greece in METOD frequently (but not always!) show burning landscapes and people trying to extinguish the flames, which should be helpful. Other examples are events with prominent victims or perpetrators, where images often depict the person in question. Here, especially algorithms including facial recognition could improve the performance of event-based topic discovery. We have not attempted to rate image informativeness in METOD.

Table 3 shows the list of topics in METOD together with their characteristics.

## 4 A Baseline Algorithm

In this section, we describe a baseline algorithm for bimodal discovery of event-based topics.

### 4.1 Basic Outline of Multimodal EventTracker

The basic outline of our algorithm, called Multimodal EventTracker, is depicted in Figure 2. The online algorithm reads a stream of news articles  $d_1, d_2, \dots$  and assigns a topic to each article  $d_i$ . If  $d_i$  fits into an already existing topic, it is put into this topic; otherwise, a new topic  $t_j$  is created. A topic  $t$  is represented as the average of the vector representations  $R_d$  of every article  $d$  with topic  $t$ :

$$R_t = \frac{1}{|t|} \sum_{d \in t} R_d .$$

This representation is updated every time a new article is assigned the topic  $t$ . Building on prior research [20, 38], we use a sliding window, denoted as  $\mathbb{W}$ , that traverses the news stream. A topic  $t$  is considered *active* as long as there is an article  $d$  with this topic in  $\mathbb{W}$ . The window  $\mathbb{W}$  is moved forward by one day

Table 3: Complete table of METOD topics and their characteristics

Topic	Size	Duration	Frequency	Irregularity	Disconnect- edness	Sudden- ness	Speci- ficity
Assassination of Haitian president	36	32	1.12	1.42	0.71	High	High
AstraZeneca vaccine concerns	10	30	0.33	4.55	0.56	High	High
Bow-and-Arrow Killing in Norway	4	5	0.8	0.52	0.13	High	High
Christchurch massacre	8	586	0.01	62.73	0.33	High	Medium
Coronavirus on cruise ship in Singapore	2	2	1	0	0	Medium	High
Coronavirus spread in Africa	3	40	0.07	1.67	0.03	Low	Low
Coronavirus spread in China	96	57	1.68	0.82	0.72	Low	Low
Coronavirus spread in Spain	9	37	0.24	4.53	0.48	Low	Low
Cruise ship quarantine in Japan	12	33	0.36	3.99	0.60	Medium	High
Early elections in Canada	15	39	0.38	4.75	0.60	Medium	Medium
Esther Dingley’s disappearance	2	242	0.00	0	0	High	High
Europe confronted with coronavirus	11	83	0.13	5.23	0.30	Low	Medium
Eurovision Song Contest 2021	5	59	0.08	24.00	0.52	Low	Medium
Explosion in Beirut’s port	29	436	0.06	33.25	0.73	High	Medium
Flight 752	18	358	0.05	77.54	0.83	High	Medium
Flood catastrophe in Europe	12	4	3	0.37	0.46	Medium	Medium
G7 summit 2021	17	41	0.41	8.92	0.82	Low	Low
German elections	21	74	0.28	9.03	0.78	Low	Low
Haiti earthquake	10	9	1.11	1.02	0.42	High	Medium
Hong Kong pro-democracy movement	125	718	0.17	9.69	0.83	Low	Low
Hurricane Eta	4	6	0.66	1.98	0.34	Medium	High
Hurricane Iota	5	4	1.25	0.63	0.26	Medium	High
Impeachment Martín Vizcarra	4	60	0.06	22.71	0.39	Medium	Medium
Iran-US conflict in Iraq	36	8	4.5	0.29	0.69	Low	Low
Kidnapping of missionaries in Haiti	9	61	0.14	10.17	0.54	High	Medium
Mexico City metro accident	6	44	0.13	15.32	0.57	High	High
Migration over Turkey-Greece boarder	8	15	0.53	0.93	0.18	Low	Medium
Myanmar military coup	38	306	0.12	11.78	0.64	Medium	Medium
Nobel Prize awards 2021	12	5	2.4	1.11	0.73	Low	Medium
Origin and expansion of Omicron	27	8	3.37	0.25	0.42	Low	Low
Palestinians escape Israeli prison	4	13	0.30	2.99	0.27	High	High
Paralympics 2020 in Tokyo	6	128	0.04	43.22	0.56	Low	Low
Peter Madsen escapes prison	2	113	0.01	0	0	High	Medium
Photo journalist killed in Afghanistan	2	15	0.13	0	0	High	High
Poisoning of Aleksei Navalny	23	152	0.15	12.59	0.73	High	Low
Poland bans abortion	7	99	0.07	19.03	0.45	Low	High
Quarantine in Canary Islands	3	3	1	0.24	0	Medium	High
Sriwijaya Air Flight 182	4	30	0.13	8.87	0.33	High	High
Storming of the US Capitol	10	152	0.06	37.98	0.69	High	Medium
Sudan military coup	13	29	0.44	2.28	0.38	Medium	High
Suez Canal Blocked	13	114	0.11	23.32	0.74	High	Medium
Taiwan railroad accident	5	108	0.04	38.90	0.49	High	High
Taliban seize control in Afghanistan	55	25	2.2	0.41	0.55	Low	Low
Ugandan 2021 election	9	88	0.10	14.71	0.55	Medium	High
UK confronted with coronavirus	27	379	0.07	14.54	0.49	Low	Medium
Ultramarathon disaster	4	20	0.2	3.75	0.16	High	High
Violence in Gaza	45	10	4.5	0.25	0.54	Low	Medium
War over Nagorno-Karabakh	23	76	0.30	3.19	0.50	Medium	Medium
Wildfires in Australia	29	78	0.37	3.76	0.57	Low	Medium
Wildfires in Greece	11	10	1.1	0.55	0.31	Low	Low
World climate summit in Glasgow	13	15	0.86	1.60	0.58	Medium	Medium

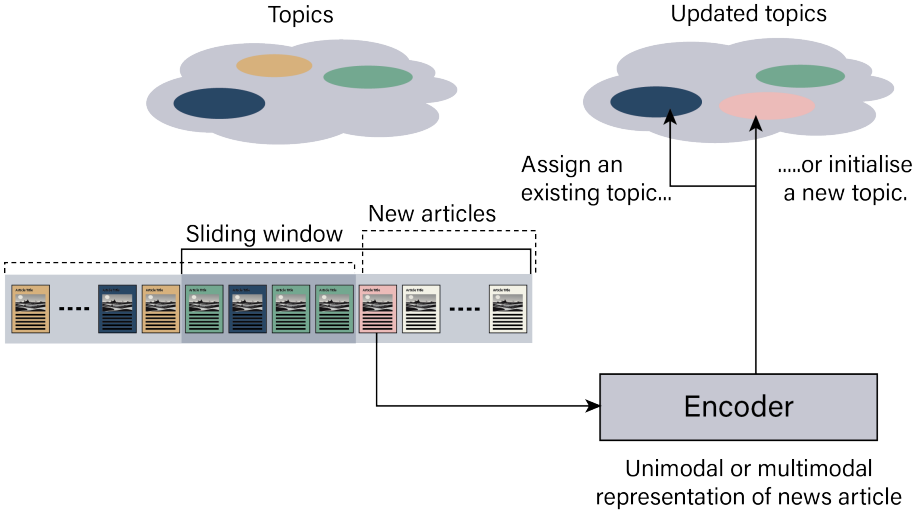


Figure 2: High-level overview of Multimodal EventTracker.

at a time along the news stream. When this happens, all articles published during the last day are processed according to the order of publication as follows: For each new document  $d$ , we evaluate its similarity compared to every active topic  $t$  using the cosine similarity between  $R_d$  and  $R_t$ , denoted by  $\text{sim}(d, t)$ . If the maximum similarity between  $d$  and the active topics exceeds a predefined threshold  $\theta$ , we allocate  $d$  to the topic  $t$  that maximizes  $\text{sim}(d, t)$  and update  $R_t$ . Otherwise, a new active cluster  $t$  is created and its representation is initialized with  $R_d$ . In our experiments, we use a threshold value of  $\theta = 0.5$ .

## 4.2 Multimodal EventTracker

Let us now discuss alternative ways of realizing the multimodal document representation  $R_d$  introduced in Section 5 (i.e., the output of the encoder in Figure 2). Figure 3 illustrates how a representation for the bimodal data can be constructed, by first encoding the two modalities separately, and then fusing the information.

To make this more precise, consider a news article  $d = (txt, img)$  consisting of a text part  $txt$  and an image part  $img$ . The multimodal representation of  $d$  is then given by

$$R_d = R_{txt} + \cos(R_{txt}^{\text{CLIP}}, R_{img}^{\text{CLIP}}) \cdot R_{img} .$$

Here,  $R_{txt}$  and  $R_{txt}^{\text{CLIP}}$  denote the SBERT [35] and CLIP [33] representations of  $txt$ , and  $R_{img}$  and  $R_{img}^{\text{CLIP}}$  denote the ViT [8] and CLIP representations of  $img$ . CLIP is a multimodal model trained on diverse image-text pairs to predict relevant text snippets for given images and vice versa. This integration

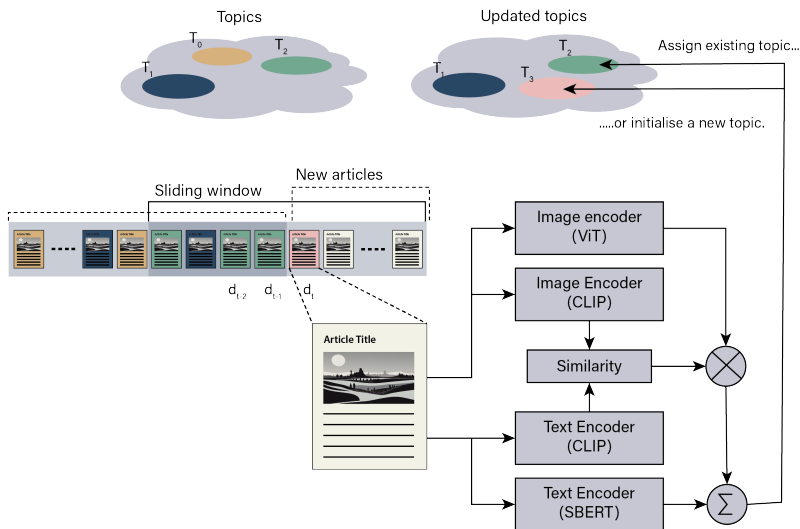


Figure 3: A more detailed view of Multimodal EventTracker.

allows CLIP to embed texts and images into a unified mathematical space for calculating cross-modal correlations. Hence, the cosine similarity between  $R_{txt}^{CLIP}$  and  $R_{img}^{CLIP}$  indicates the degree to which the article text and image are aligned and serves as a criterion to regulate the contribution of the image modality in the overall multimodal representation of the article.

### 4.3 Implementation Details

We implemented our model using PyTorch [30] and the Transformer Library [44]. As text and image encoders we selected sentence-transformers/all-roberta-large-v1<sup>6</sup> and google/vit-large-patch16-224<sup>7</sup>, respectively. Additionally, for the CLIP vision-language model, we used openai/clip-vit-large-patch14<sup>8</sup>. The maximum sequence length for the SBERT tokenizer was set to 256. We defined the threshold  $\theta$  to be 0.5 and specified window and sliding sizes of 7 days and 1 day, respectively.

## 5 Experiments and Analysis

In this section, we explore the impact of incorporating multimodal content (i.e., text and image) as opposed to utilizing text or image individually for

<sup>6</sup><https://huggingface.co/sentence-transformers/all-roberta-large-v1>

<sup>7</sup><https://huggingface.co/google/vit-large-patch16-224>

<sup>8</sup><https://huggingface.co/openai/clip-vit-large-patch14>

two unsupervised tasks: K-Means clustering and event-based topic discovery within a news stream. Following this exploration, we apply the proposed Multimodal EventTracker to the METOD dataset and present analyses based on the characteristics outlined in Section 3.2.

## 5.1 Evaluation Metrics

In this section, we explain the measures we will use in our comparative analysis of different approaches.

The *Accuracy* of the K-Means clustering is determined through the calculation of unsupervised clustering accuracy. This measure is computed as:

$$ACC = \max_m \frac{\sum_{n=1}^N 1\{l_n = m(c_n)\}}{N}$$

Here,  $N$ ,  $l_n$ , and  $c_n$  denote the total number of documents, the ground-truth label of document  $d_n$ , and the cluster predicted by the clustering algorithm for  $d_n$ , respectively. The maximization ranges over all mappings  $m$  of cluster assignments to labels. The Hungarian algorithm can be employed to compute the  $m$  that maximizes the sum [47].

*Precision*, *Recall*, and *F1* scores are derived from pairwise comparisons. Let  $tp$  represent the count of correctly clustered document pairs,  $fp$  the count of incorrectly clustered document pairs, and  $fn$  the count of incorrectly separated document pairs. The reported precision is then  $\frac{tp}{(tp+fp)}$  the recall is  $\frac{tp}{(tp+fn)}$ , and F1 is their harmonic mean.

The *BCubed F1*, denoted as  $B^3-F1$  here, is a metric introduced by Bagga and Baldwin [4] that emphasizes the precision and recall of individual data points within clusters, rather than pairwise comparisons. A detailed explanation of its computation is provided by Staykovski et al. [39].

The *Adjusted Rand Index (ARI)* [13] is a symmetric metric that offers a comprehensive evaluation of clustering quality by considering both pairwise agreements and disagreements. *Adjusted Mutual Information (AMI)* [41] also favors grouping similar data points in the same cluster but is less sensitive to breaking the ground truth clusters into multiple clusters. Both ARI and AMI are adjusted for chance agreement. All metrics have a maximum score of 1.

## 5.2 Clustering with K-Means

We compare various modality fusion approaches for clustering the METOD dataset with K-Means. Subsequently, we select the most suitable approach for further analysis. Not all available modality fusion models, such as attention-based fusion [46, 52], are easily applicable to the unsupervised setting. Therefore, we focus our study on concatenation, summation, and weighted summation



Table 4: Comparison of various modalities and fusion methods in clustering the validation dataset using K-Means. The reported results represent the average across three separate runs with distinct random seeds. The highest and second highest scores in each metric are bolded and underlined, respectively.

Modality	Fusion Model	Accuracy	F1	Precision	Recall	AMI	ARI
text		0.655	0.548	0.863	0.401	0.811	0.533
image		0.172	0.076	0.106	0.059	0.144	0.042
multimodal	text + image	0.640	0.558	0.843	0.417	0.805	0.543
multimodal	concat(text, image)	0.630	0.549	0.822	0.412	0.796	0.533
multimodal	text + CLIP_sim * image	<b>0.669</b>	<b>0.586</b>	<u>0.865</u>	<b>0.443</b>	<u>0.822</u>	<b>0.571</b>
multimodal	concat(text, CLIP_sim * image)	<b>0.669</b>	<u>0.579</u>	<b>0.879</b>	<u>0.432</u>	<b>0.826</b>	<u>0.565</u>
multimodal	CLIP_text + CLIP_image	0.636	0.547	0.789	0.420	0.780	0.531
multimodal	concat(CLIP_text, CLIP_image)	0.569	0.486	0.744	0.362	0.732	0.469

fusion approaches. The detailed results of these experiments can be found in Table 4.

According to the findings, the most effective fusion approach is the weighted sum when leveraging the cosine similarity of CLIP text and image representations as the contribution weight for the image modality in the final multimodal representation. In this approach, when there is a higher alignment between news text and image (reflected in their elevated cosine similarity), the contribution of the image modality in the multimodal representation is more pronounced.

While a news image is more likely to visually depict an event, image encoders concentrate solely on identifying objects in the image and their relationships. Consequently, the features derived from these models may not adequately capture the nuanced contextual information embedded in news images. Table 5 presents captions produced by BLIP-2 [24] for images corresponding to five news articles, along with their respective headlines. The observed misalignment between the generated captions and headlines indicates the limitations of this image encoder when it comes capturing contextual information.

Additionally, there are instances where the image may not be highly relevant to the news topic; for example, it might only feature a person central to the event, which the image encoder interprets merely as a person without recognizing the actual identity of that individual in the real world. Figure 4 depicts the distribution of cosine similarity among CLIP representations of text and image for news articles in the METHOD dataset. Overall, a low correlation is observed between these modalities, with the mean similarity measuring 0.23. Therefore, news text emerges as a more dependable source for news topic detection, emphasizing the need to regulate the contribution of the image modality in the multimodal representation.

Nonetheless, strategically incorporating images alongside text proves advantageous for clustering text+image content. Table 6 illustrates an example where the sole use of a text-based approach fails in clustering, while the multimodal

Table 5: Examples of captions generated by BLIP-2 for news images (those included here are AI-generated placeholders). Headlines and generated captions are written in upright and italic font, respectively.

Image	Headline vs Caption
	<p>New Caledonia Says ‘Non’ to Independence  <i>A woman is standing at a table with a man</i>  <i>Link: <a href="https://www.nytimes.com/2021/12/12/world/asia/new-caledonia-independence-vote.html">https://www.nytimes.com/2021/12/12/world/asia/new-caledonia-independence-vote.html</a></i></p>
	<p>Progress for Saudi Women Is Uneven, Despite Cultural Changes and More Jobs  <i>A woman in a green dress is looking at a mirror</i>  <i>Link: <a href="https://www.nytimes.com/2021/12/09/world/middleeast/saudi-arabia-women-mbs.html">https://www.nytimes.com/2021/12/09/world/middleeast/saudi-arabia-women-mbs.html</a></i></p>
	<p>India’s Top Military General Dies in Helicopter Crash  <i>Two men stand near a pile of debris in the forest</i>  <i>Link: <a href="https://www.nytimes.com/2021/12/08/world/asia/helicopter-crash-india-top-general.html">https://www.nytimes.com/2021/12/08/world/asia/helicopter-crash-india-top-general.html</a></i></p>
	<p>Taliban Allow Polio Vaccine Program to Restart in Afghanistan  <i>A man is giving a child a bottle of water</i>  <i>Link: <a href="https://www.nytimes.com/2021/10/19/world/asia/taliban-polio-vaccines-afghanistan.html">https://www.nytimes.com/2021/10/19/world/asia/taliban-polio-vaccines-afghanistan.html</a></i></p>

approach successfully clusters the data.

### 5.3 Event-based Topic Discovery within a News Stream

We assess unimodal and multimodal news representations within the framework outlined in Section 4.1 for the task of event-based topic discovery. Our comparison of unimodal and multimodal representations for the task unfolds in two scenarios:

1. Running the model on the METOD dataset without any window. In this case, identified topics persist permanently in the topic pool, and each new news article is compared with all existing topics.
2. Running the model on the METOD dataset with a 7-day window.

The outcomes of these experiments are presented in Table 7. The results suggest that employing a multimodal representation of news articles marginally improves

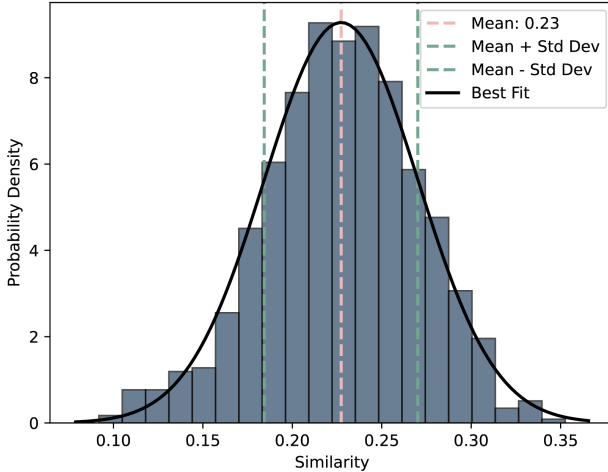


Figure 4: The distribution of cosine similarity between CLIP representations of text and image for news articles in the METOD dataset.





performance across all metrics compared to using unimodal representations.

Furthermore, it is observed that omitting the window from the model leads to elevated values for all criteria. However, for large-scale datasets with numerous topics, comparing every document to an extensive pool of all topics might not be very efficient. In such scenarios, incorporating the window enhances the algorithm’s efficiency and speed. Nonetheless, one could potentially enhance the model by introducing a link prediction component. This component could establish connections between current topics and previous topics beyond the window, enabling the tracking of topics beyond the specified time frame for those interested.

## 5.4 Results

We apply Multimodal EventTracker introduced in Section 4.2 to the METOD dataset, resulting in 186 clusters. Some topics are fragmented into multiple clusters due to reasons such as a time period longer than 7 days (window size) between two consecutive articles or the model’s inability to recognize the articles as a single cohesive cluster. A crucial component of the model is the minimum similarity threshold  $\theta$ . As illustrated in Table 3, the topics in the dataset exhibit varying granularity and specificity. Consequently, selecting a suitable value for  $\theta$  such that it can accommodate the diverse characteristics of topics and ensure the assignment of all members of each topic into a distinct cluster, proves challenging.

Table 6: Articles correctly grouped into two clusters when using both image and text modalities, but incorrectly predicted as a single cluster when relying solely on text. The initial pair of articles discusses a quarantine aboard a cruise ship, while the subsequent two articles cover a quarantine situation at a resort on Tenerife Island. Although the textual content shares similarities as they both address being quarantined during a vacation, the images highlight the disparity in the events.

Image	Headline
	Japan Reports 2 Deaths Among Cruise Ship Passengers Link: <a href="https://www.nytimes.com/2020/02/19/world/asia/china-coronavirus.html">https://www.nytimes.com/2020/02/19/world/asia/china-coronavirus.html</a>
	Hundreds Released From Diamond Princess Cruise Ship in Japan Link: <a href="https://www.nytimes.com/2020/02/19/world/asia/japan-cruise-ship-coronavirus.html">https://www.nytimes.com/2020/02/19/world/asia/japan-cruise-ship-coronavirus.html</a>
	Spanish Hotel Is Locked Down After Guests Test Positive for Coronavirus Link: <a href="https://www.nytimes.com/2020/02/25/world/europe/spain-coronavirus-hotel-canary.html">https://www.nytimes.com/2020/02/25/world/europe/spain-coronavirus-hotel-canary.html</a>
	At a Locked Down Spanish Resort, Many Questions, Little Information Link: <a href="https://www.nytimes.com/2020/02/27/world/europe/tenerife-coronavirus-lockdown.html">https://www.nytimes.com/2020/02/27/world/europe/tenerife-coronavirus-lockdown.html</a>

For instance, the topic *UK confronted with coronavirus* encompasses various articles related to coronavirus, spanning from the virus’s spread to extended waits for hospital treatment or virus testing, and so forth. Due to its relatively low specificity and a duration of 379 days, this topic has been subdivided into 16 clusters. Conversely, the topic *Sudan military coup* demonstrates high specificity with a duration of 29 days. Consequently, the model successfully assigns all articles pertaining to this topic into a single cluster.

Table 8 displays dispersion, precision, recall, and F1 scores for each topic in the dataset. Dispersion represents the count of clusters that include at least one member of the corresponding topic, possibly alongside articles from other topics. To compute precision, recall, and F1 based on pairwise comparisons, we reformulate the problem as a binary clustering, where all topic articles serve as positive samples, and articles from other topics act as negative samples.

Table 7: Comparison of various modalities in clustering the METOD dataset as an stream with a window size of 7 days. Boldface font emphasizes highest scores.

	Modality	Fusion Model	F1	B <sup>3</sup> -F1	AMI	ARI
no window	text		0.816	0.804	0.840	0.806
	image		0.063	0.189	0.183	0.048
	multimodal	text + CLIP_sim * image	<b>0.839</b>	<b>0.813</b>	<b>0.851</b>	<b>0.831</b>
window size 7	text		0.652	0.732	0.784	0.639
	image		0.049	0.180	0.193	0.046
	multimodal	text + CLIP_sim * image	<b>0.656</b>	<b>0.736</b>	<b>0.790</b>	<b>0.643</b>

Table 9 shows the correlation between the dataset characteristics and the performance measures. The strongest correlation is observed between duration and dispersion. This is attributed to the sliding window technique, where a topic with a longer duration tends to be fragmented into more clusters. As the frequency of articles in a topic increases, there is a higher probability that they share similar content and revolve around a very specific event. Consequently, the model finds it easier to identify them collectively as a single topic. This results in a positive correlation between frequency and performance measures, and conversely, a negative correlation between frequency and dispersion. Unsurprisingly, irregular temporal spacing between the publication dates of articles tends to increase precision but reduces recall. Also, increased disconnectedness tends to increase the number of predicted clusters a topic is spread out over. One may assume that this behavior is, at least in part, caused by the sliding window technique: the more disconnected a topic is, the higher the probability that a previous phase of articles in that topic will have fallen out of the window before the next article in the topic is encountered. Furthermore, when a topic exhibits greater specificity, there is a higher likelihood that the model recognizes it as a single topic, resulting in fewer assigned clusters. This is reflected in the negative correlation observed between dispersion and specificity in the table.

## 6 Discussion

Leveraging multimodal content, such as pairing text with images, is a common strategy in news reporting to enhance communication effectiveness, capture attention, and evoke emotional responses. However, the intricate relationship between news article text and images poses a significant challenge, leaving ample room for advancements in the use of news article images for improving event-based topic discovery.

Typically, news article texts encompass a wealth of information, including

Table 8: Precision, recall, and F1 scores for topics in the METOD dataset obtained from the clustering results of the Multimodal EventTracker. The column labeled Dispersion reports the number of computed clusters which contain at least one member of the topic.

Topic	Size	Dispersion	Precision	Recall	F1
Assassination of Haitian president	36	2	1.0	0.94	0.97
AstraZeneca vaccine concerns	10	2	1.0	0.64	0.78
Bow-and-Arrow Killing in Norway	4	1	1.0	1.0	1.0
Christchurch massacre	8	7	1.0	0.04	0.07
Coronavirus on cruise ship in Singapore	2	1	1.0	1.0	1.0
Coronavirus spread in Africa	3	1	0.01	1.0	0.02
Coronavirus spread in China	96	3	0.84	0.96	0.89
Coronavirus spread in Spain	9	2	0.08	0.61	0.14
Cruise ship quarantine in Japan	12	3	0.25	0.68	0.37
Early elections in Canada	15	4	0.82	0.63	0.71
Esther Dingley’s disappearance	2	2	0.0	0.0	0.0
Europe confront with coronavirus	11	2	0.12	0.67	0.21
Eurovision Song Contest 2021	5	3	1.0	0.3	0.46
Explosion in Beirut’s port	29	7	1.0	0.57	0.73
Flight 752	18	4	1.0	0.69	0.81
Flood catastrophe in Europe	12	2	1.0	0.7	0.82
G7 summit 2021	17	4	1.0	0.67	0.8
German elections	21	4	0.88	0.48	0.62
Haiti earthquake	10	1	1.0	1.0	1.0
Hong Kong pro-democracy movement	125	32	0.86	0.14	0.24
Hurricane Eta	4	2	1.0	0.5	0.67
Hurricane Iota	5	1	1.0	1.0	1.0
Impeachment Martín Vizcarra	4	2	1.0	0.33	0.5
Iran-US conflict in Iraq	36	3	1.0	0.89	0.94
Kidnapping of missionaries in Haiti	9	3	1.0	0.44	0.62
Mexico City metro accident	6	3	1.0	0.27	0.42
Migration over Turkey-Greece boarder	8	2	1.0	0.75	0.86
Myanmar military coup	38	12	0.53	0.28	0.37
Nobel Prize awards 2021	12	4	1.0	0.36	0.53
Origin and expansion of Omicron	27	6	1.0	0.36	0.53
Palestinians escape Israeli prison	4	1	1.0	1.0	1.0
Paralympics 2020 in Tokyo	6	3	1.0	0.4	0.57
Peter Madsen escapes prison	2	2	0.0	0.0	0.0
Photo journalist killed in Afghanistan	2	2	0.0	0.0	0.0
Poisoning of Aleksei Navalny	23	4	1.0	0.55	0.71
Poland bans abortion	7	3	1.0	0.33	0.5
Quarantine in Canary Islands	3	1	0.09	1.0	0.17
Sriwijaya Air Flight 182	4	2	1.0	0.5	0.67
Storming of the US Capitol	10	4	1.0	0.47	0.64
Sudan military coup	13	1	1.0	1.0	1.0
Suez Canal Blocked	13	4	1.0	0.58	0.73
Taiwan railroad accident	5	2	1.0	0.6	0.75
Taliban seize control in Afghanistan	55	4	1.0	0.6	0.75
UK confronted with coronavirus	27	16	0.15	0.05	0.08
Ugandan 2021 election	9	4	1.0	0.25	0.4
Ultramarathon disaster	4	1	1.0	1.0	1.0
Violence in Gaza	45	1	1.0	1.0	1.0
War over Nagorno-Karabakh	23	4	1.0	0.41	0.58
Wildfires in Australia	29	8	1.0	0.52	0.68
Wildfires in Greece	11	1	1.0	1.0	1.0
World climate summit in Glasgow	13	3	1.0	0.59	0.74

Table 9: Correlation between the characteristics of topics in the METOD dataset and the evaluation metrics.

Characteristic	Dispersion	Precision	Recall	F1
Size	0.65	0.16	-0.01	0.09
Duration	0.69	0.02	-0.48	-0.35
Frequency	-0.13	0.23	0.36	0.38
Irregularity	0.15	0.23	-0.30	-0.11
Disconnectedness	0.40	0.45	-0.09	0.24
Suddenness	-0.16	0.05	-0.08	0.07
Specificity	-0.24	-0.08	0.03	0.04

details about the timing, content, location, and individuals involved in the events they report. In contrast, the role of accompanying images in news articles is diverse. Images may serve as decorative elements, offer additional details, or, in some instances, become potential sources of misinformation [22]. For example, consider a news article focused on an action taken by Trump on a specific issue, where the accompanying image exclusively features Trump.

Navigating this complexity demands effective methods that evaluate cross-modal consistency in real-world news articles, ensuring that images significantly contribute to understanding and accurately detecting the underlying topic of the article. In our research, we conducted this assessment by comparing the CLIP embeddings of the text and image of news articles. However, this approach serves as a basic baseline, and there is a need for more advanced and efficacious methodologies. An advanced model could encompass different techniques such as face recognition, scene classification, and linking entities from text and image simultaneously, creating a comprehensive contextual representation of the image. Furthermore, it should possess the capability to determine when each modality provides richer information and should be prioritized.

In an experiment, we assessed the combination of images with different text sections of news articles, and the results are detailed in Table 10. The findings suggest that, among the headline, abstract, and leading paragraph of the article, event-level topic discovery is most easily accomplished using the headline and are most challenging when using the abstract. Thus, the headline emerges as the most informative, and the abstract as the least informative section for topic discovery. Consequently, the combination of text with images proves to be more advantageous, especially when using the abstract as the text source, given its relatively lower information content about the topic.

Table 10: Evaluating the integration of images with various text sections of news articles for event-based topic discovery.

	Section	F1	B3-F1	AMI	ARI
text only	headline	0.559	0.662	0.731	0.546
	abstract	0.415	0.494	0.566	0.400
	paragraph	0.462	0.564	0.636	0.448
	all	0.652	0.732	0.784	0.639
text+image	headline	0.569	0.676	0.741	0.556
	abstract	0.485	0.559	0.617	0.469
	paragraph	0.500	0.606	0.670	0.485
	all	<b>0.656</b>	<b>0.736</b>	<b>0.790</b>	<b>0.643</b>

## 7 Conclusion

In our research, we explored the correlation between the text and images of news articles for the purpose of event-based topic discovery. To facilitate this investigation, we curated a text-image dataset named METOD using the NYT API and annotated it with event-level topics. Additionally, we introduced the Multimodal EventTracker for automatically annotating the dataset with topic labels, serving as a straightforward baseline for this task. The annotated dataset has been made publicly available, and we envision this as an initial stride towards encouraging further research in this particular research domain.

### Use of Generative AI

All “photos” included in this article have been generated with Adobe Illustrator as stand-ins for the original, copy-righted, photos. We have used ChatGPT to copy-edit the text for grammar and clarity.

## References

- [1] Firoj Alam, Ferda Offi, and Muhammad Imran. “Crisismmd: Multimodal twitter datasets from natural disasters”. In: *Proceedings of the international AAAI conference on web and social media*. Vol. 12. 1. 2018.
- [2] Alexander et al. *Webis at TREC 2018: Common Core Track*. <https://github.com/irgroup/datasets/blob/master/WAPost/README.md>. 2018.



- [3] James Allan. “Introduction to Topic Detection and Tracking”. In: *Topic Detection and Tracking: Event-based Information Organization*. Boston, MA: Springer US, 2002, pp. 1–16. ISBN: 978-1-4615-0933-2. DOI: 10.1007/978-1-4615-0933-2\_1. URL: [https://doi.org/10.1007/978-1-4615-0933-2\\_1](https://doi.org/10.1007/978-1-4615-0933-2_1).
- [4] Amit Bagga and Breck Baldwin. “Entity-Based Cross-Document Coreferencing Using the Vector Space Model”. In: *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*. Montreal, Quebec, Canada: Association for Computational Linguistics, Aug. 1998, pp. 79–85. DOI: 10.3115/980845.980859. URL: <https://aclanthology.org/P98-1012>.
- [5] Regina Barzilay and Kathleen R McKeown. “Sentence fusion for multidocument news summarization”. In: *Computational Linguistics* 31.3 (2005), pp. 297–328.
- [6] Christina Boididou et al. “Detection and visualization of misleading content on Twitter”. In: *International Journal of Multimedia Information Retrieval* 7.1 (2018), pp. 71–86.
- [7] Christos Bouras and Vassilis Tsogkas. “A clustering technique for news articles using WordNet”. In: *Knowledge-Based Systems* 36 (2012), pp. 115–128.
- [8] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *Computing Research Repository* arXiv:2010.11929 (2020). URL: <https://doi.org/10.48550/arXiv.2010.11929>.
- [9] Wentao Fan et al. “Clustering-Based Online News Topic Detection and Tracking Through Hierarchical Bayesian Nonparametric Models”. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’21. <conf-loc>, <city>Virtual Event</city>, <country>Canada</country>, </conf-loc>: Association for Computing Machinery, 2021, pp. 2126–2130. ISBN: 9781450380379. DOI: 10.1145/3404835.3462982. URL: <https://doi.org/10.1145/3404835.3462982>.
- [10] Demian Gholipour Ghalandari et al. “A Large-Scale Multi-Document Summarization Dataset from the Wikipedia Current Events Portal”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 1302–1308. DOI: 10.18653/v1/2020.acl-main.120. URL: <https://aclanthology.org/2020.acl-main.120>.
- [11] Antonio Gulli. *AG’s corpus of news articles*. [http://groups.di.unipi.it/~gulli/AG\\_corpus\\_of\\_news\\_articles.html](http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html). 2005.
- [12] Sameed Hayat. *Guardian news dataset*. <https://www.kaggle.com/datasets/sameedhayat/guardian-news-dataset>. 2018.

- [13] Lawrence Hubert and Phipps Arabie. “Comparing partitions”. In: *Journal of Classification* 2 (1985), pp. 193–218. ISSN: 01764268. DOI: 10.1007/BF01908075.
- [14] Harold Edwin Hurst. “Long-term storage capacity of reservoirs”. In: *Transactions of the American society of civil engineers* 116.1 (1951), pp. 770–799.
- [15] Ipsos. *Media and News Survey 2023*. Available at <https://europa.eu/eurobarometer/surveys/detail/3153>. EB-ID: FL012EP | Fieldwork: 18/10/2023 – 24/10/2023 | Conducted by Ipsos European Public Affairs. 2023.
- [16] Hang Jiand et al. “Topic Detection and Tracking with Time-Aware Document Embeddings”. In: *Computing Research Repository* arXiv:2112.06166 (2021). URL: <https://doi.org/10.48550/arXiv.2112.06166>.
- [17] Sarthak Jindal et al. “Newsbag: A multimodal benchmark dataset for fake news detection”. In: *CEUR Workshop Proc.* Vol. 2560. 2020, pp. 138–145.
- [18] Namgyu Jung et al. “News Category Classification via Multimodal Fusion Method”. In: *Proceedings of the 2023 International Conference on Research in Adaptive and Convergent Systems*. RACS '23. Gdansk, Poland: Association for Computing Machinery, 2023. ISBN: 9798400702280. DOI: 10.1145/3599957.3606237. URL: <https://doi.org/10.1145/3599957.3606237>.
- [19] Kaggle. *Bbc news dataset*. <https://www.kaggle.com/c/learn-ai-bbc>. 2018.
- [20] Philippe Laban and Marti Hearst. “newsLens: building and visualizing long-ranging news stories”. In: *Proceedings of the Events and Stories in the News Workshop*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 1–9. DOI: 10.18653/v1/W17-2701. URL: <https://aclanthology.org/W17-2701>.
- [21] Ken Lang. “Newsweeder: Learning to filter netnews”. In: *Machine learning proceedings 1995*. Elsevier, 1995, pp. 331–339.
- [22] Duc-Trong Le et al. “ReINTEL: A Multimodal Data Challenge for Responsible Information Identification on Social Network Sites”. In: *Proceedings of the 7th International Workshop on Vietnamese Language and Speech Processing*. Hanoi, Vietnam: Association for Computational Linguistics, Dec. 2020, pp. 84–91. URL: <https://aclanthology.org/2020.vlsp-1.16>.
- [23] Jian Li et al. “Forecasting oil price trends with sentiment of online news articles”. In: *Procedia Computer Science* 91 (2016), pp. 1081–1087.
- [24] Junnan Li et al. “BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models”. In: *Computing Research Repository* arxiv:2301.12597 (2023). URL: <https://doi.org/10.48550/arXiv.2301.12597>.

- [25] Weixin Li et al. “Joint Image-Text News Topic Detection and Tracking by Multimodal Topic And-Or Graph”. In: *IEEE Transactions on Multimedia* 19.2 (2017), pp. 367–381. DOI: 10.1109/TMM.2016.2616279. URL: <https://doi.org/10.1109/TMM.2016.2616279>.
- [26] Sebastiao Miranda et al. “Multilingual Clustering of Streaming News”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 4535–4544. DOI: 10.18653/v1/D18-1483. URL: <https://aclanthology.org/D18-1483>.
- [27] Masaki Mori, Takao Miura, and Isamu Shioya. “Topic detection and tracking for news web pages”. In: *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI’06)*. IEEE, 2006, pp. 338–342.
- [28] Kai Nakamura, Sharon Levy, and William Yang Wang. “r/fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection”. In: *arXiv preprint arXiv:1911.03854* (2019).
- [29] Shi-Yong Neo et al. “The use of topic evolution to help users browse and find answers in news video corpus”. In: *Proceedings of the 15th ACM international conference on Multimedia*. 2007, pp. 198–207.
- [30] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Article 721 (12 pages). Red Hook, NY, USA: Curran Associates Inc., 2019.
- [31] Shengsheng Qian, Tianzhu Zhang, and Changsheng Xu. “Online multimodal multiexpert learning for social event tracking”. In: *IEEE Transactions on Multimedia* 20.10 (2018), pp. 2733–2748.
- [32] Shengsheng Qian et al. “Open-World Social Event Classification”. In: *Proceedings of the ACM Web Conference 2023*. WWW ’23. Austin, TX, USA: Association for Computing Machinery, 2023, pp. 1562–1571. ISBN: 9781450394161. DOI: 10.1145/3543507.3583291. URL: <https://doi.org/10.1145/3543507.3583291>.
- [33] Alec Radford et al. “Learning Transferable Visual Models From Natural Language Supervision”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 8748–8763. URL: <https://proceedings.mlr.press/v139/radford21a.html>.
- [34] Arnau Ramisa Ayats et al. “The breakingNews dataset”. In: *Proceedings 6th Workshop on Vision and Language (VL)*. 2017, pp. 38–39.

- [35] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. DOI: 10.18653/v1/D19-1410. URL: <https://aclanthology.org/D19-1410>.
- [36] María del Pilar Salas-Zárate et al. “Sentiment analysis and trend detection in Twitter”. In: *Technologies and Innovation: Second International Conference, CITI 2016, Guayaquil, Ecuador, November 23-25, 2016, Proceedings 2*. Springer. 2016, pp. 63–76.
- [37] Kailash Karthik Saravanakumar et al. “Event-Driven News Stream Clustering using Entity-Aware Contextual Embeddings”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 2330–2340. DOI: 10.18653/v1/2021.eacl-main.198. URL: <https://aclanthology.org/2021.eacl-main.198>.
- [38] Jonathan A. Silva et al. “Data Stream Clustering: A Survey”. In: *ACM Comput. Surv.* 46.1 (July 2013). ISSN: 0360-0300. DOI: 10.1145/2522968.2522981. URL: <https://doi.org/10.1145/2522968.2522981>.
- [39] Todor Staykovski et al. “Dense vs. Sparse Representations for News Stream Clustering”. In: *Proceedings of Text2Story - 2nd Workshop on Narrative Extraction From Texts, co-located with the 41st European Conference on Information Retrieval, Text2Story@ECIR 2019, Cologne, Germany, April 14th, 2019*. Ed. by Alípio Mário Jorge et al. Vol. 2342. CEUR Workshop Proceedings. CEUR-WS.org, 2019, pp. 47–52. URL: <https://ceur-ws.org/Vol-2342/paper6.pdf>.
- [40] Soonh Taj, Baby Bakhtawer Shaikh, and Areej Fatemah Meghji. “Sentiment analysis of news articles: A lexicon based approach”. In: *2019 2nd international conference on computing, mathematics and engineering technologies (iCoMET)*. IEEE. 2019, pp. 1–5.
- [41] Nguyen Xuan Vinh, Julien Epps, and James Bailey. “Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance”. In: *Journal of Machine Learning Research* 11.95 (2010), pp. 2837–2854. URL: <http://jmlr.org/papers/v11/vinh10a.html>.
- [42] Shuo Wang et al. “Meed: A multimodal event extraction dataset”. In: *Knowledge Graph and Semantic Computing: Knowledge Graph Empowers New Infrastructure Construction: 6th China Conference, CCKS 2021, Guangzhou, China, November 4-7, 2021, Proceedings 6*. Springer. 2021, pp. 288–294.

- [43] Zhen Wang et al. “N24News: A New Dataset for Multimodal News Classification”. In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, June 2022, pp. 6768–6775. URL: <https://aclanthology.org/2022.lrec-1.729>.
- [44] Thomas Wolf et al. “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: <https://aclanthology.org/2020.emnlp-demos.6>.
- [45] Xiao Wu et al. “Mining event structures from web videos”. In: *IEEE MultiMedia* 18.1 (2011), pp. 38–51.
- [46] Yang Wu et al. “Multimodal fusion with co-attention networks for fake news detection”. In: *Findings of the association for computational linguistics: ACL-IJCNLP 2021*. 2021, pp. 2560–2569.
- [47] Wei Xu, Xin Liu, and Yihong Gong. “Document clustering based on non-negative matrix factorization”. In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. SIGIR ’03. Toronto, Canada: Association for Computing Machinery, 2003, pp. 267–273. ISBN: 1581136463. DOI: 10.1145/860435.860485. URL: <https://doi.org/10.1145/860435.860485>.
- [48] Sin-han Yang et al. “Entity-Aware Dual Co-Attention Network for Fake News Detection”. In: *Findings of the Association for Computational Linguistics: EACL 2023*. Ed. by Andreas Vlachos and Isabelle Augenstein. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 106–113. DOI: 10.18653/v1/2023.findings-eacl.7. URL: <https://aclanthology.org/2023.findings-eacl.7>.
- [49] Ze Yang et al. “Read, Attend and Comment: A Deep Architecture for Automatic News Comment Generation”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5076–5088. DOI: 10.18653/v1/D19-1512. URL: <https://www.aclweb.org/anthology/D19-1512>.
- [50] Susik Yoon et al. “SCStory: Self-Supervised and Continual Online Story Discovery”. In: *Proceedings of the ACM Web Conference 2023*. WWW ’23. Austin, TX, USA: Association for Computing Machinery, 2023, pp. 1853–1864. ISBN: 9781450394161. DOI: 10.1145/3543507.3583507. URL: <https://doi.org/10.1145/3543507.3583507>.
- [51] Kaifu Zhang and Zsolt Katona. “Contextual advertising”. In: *Marketing Science* 31.6 (2012), pp. 980–994.

- [52] Yangming Zhou et al. “Multi-Modal Fake News Detection on Social Media via Multi-Grained Information Fusion”. In: *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval*. ICMR '23. Thessaloniki, Greece: Association for Computing Machinery, 2023, pp. 343–352. ISBN: 9798400701788. DOI: 10.1145/3591106.3592271. URL: <https://doi.org/10.1145/3591106.3592271>.
- [53] Yangming Zhou et al. “Multimodal Fake News Detection via CLIP-Guided Learning”. In: *Computing Research Repository* arXiv:2205.14304 (2022). URL: <https://doi.org/10.48550/arXiv.2205.14304>.
- [54] Dimitrina Zlatkova, Preslav Nakov, and Ivan Koychev. “Fact-checking meets fauxtography: Verifying claims about images”. In: *arXiv preprint arXiv:1908.11722* (2019).
- [55] Arkaitz Zubiaga et al. “PHEME dataset of rumours and non-rumours”. In: (2016).