공학박사학위논문

Learning Label Relationships Using Deep Neural Networks for Hierarchical Text Classification

계층적 문서 분류를 위한 심층 신경망 기반 라벨 간 관계 학습

2023 년 8 월

서울대학교 대학원

산업공학과

방 진 현

Learning Label Relationships Using Deep Neural Networks for Hierarchical Text Classification

계층적 문서 분류를 위한 심층 신경망 기반 라벨 간 관계 학습

지도교수 박 종 헌

이 논문을 공학박사 학위논문으로 제출함

2023 년 5 월

서울대학교 대학원

산업공학과

방 진 현

방진현의 공학박사 학위논문을 인준함

2023 년 5 월

위 원 장 조 성 준 (인)

부위원장 이 경 식 (인)

위 원 박 종 헌 (인)

위 원 이 재 욱 (인)

위 원 박 종 혁 (인)

Abstract

Learning Label Relationships Using Deep Neural Networks for Hierarchical Text Classification

Jinhyun Bang
Department of Industrial Engineering
The Graduate School
Seoul National University

Hierarchical text classification has been receiving increasing attention due to its vast range of applications in real-world natural language processing tasks. With the recent advances in deep learning, deep learning-based approaches achieved state-of-the-art hierarchical text classification performance. While existing approaches focus on exploiting the label hierarchy or modeling implicit label relationships, only a few studies integrated these two concepts. This thesis proposes a graph attention capsule network for hierarchical text classification (GACaps-HTC), a deep learning-based approach designed to capture both the explicit hierarchy and latent label relationships. A graph attention network is employed in the proposed approach for fusing information on the label hierarchy into a textual representation, while a capsule network is employed to understand the latent label relationships and infer classification probabilities. The proposed approach is optimized using a loss term designed to

address the innate label imbalance issue of the task and post-processed using various methods specified for hierarchical text classification. Results of the experiments conducted on two benchmark datasets demonstrate that the proposed approach outperformed previous state-of-the-art approaches and ablation studies show that each component in the GACaps-HTC played a part in enhancing the performance.

Furthermore, this thesis proposes a semantic-aware dynamic routing algorithm, a new dynamic routing algorithm that initializes and updates a capsule network's coupling coefficients using semantic representations of labels. As a coupling coefficient of a pair of capsules indicates how similar their information is, the coefficient is initialized from the similarity of semantic representations corresponding to the capsules' labels. Experiment results show that the proposed algorithm outperformed other methods that inject semantic information of labels and GACaps-HTC with semantic-aware dynamic routing algorithm reached faster convergence compared to GACaps-HTC with conventional dynamic routing algorithm.

Finally, this thesis investigates another use case of GACaps-HTC by employing the model for aspect category sentiment analysis that can be formulated as hierarchical text classification. Experiments were conducted on four sentiment analysis datasets, and the results show that the proposed approach performs well on not only the semantic analysis of a document but also sentiment analysis.

Keywords: Hierarchical text classification, graph neural network, capsule network, deep learning, natural language processing

Student Number: 2017-28575

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

Natural language processing (NLP) is a subfield of computer science and linguistics concerned with the computational process of understanding human languages[7, 8]. NLP technologies play a crucial part in various real-world industries, including manufacturing[9, 10], finance[11, 12, 13], healthcare[14, 15], and the legal industry[16, 17], as they can analyze huge volumes of textual data. Due to the field's practical importance, NLP has been one of the most actively researched fields with the advent of deep learning, and deep learning-based approaches have shown state-of-the-art performances in a variety of NLP tasks such as sentiment analysis[18, 19, 20], machine translation[21, 22], and text summarization[23, 24]. Text classification, the task of automatically assigning a set of labels to a given text document, is another task that has benefited from employing deep learning[19, 25] and is a vital task due to its wide range of applications in a number of other NLP tasks including information retrieval[26, 27], sentiment analysis[28, 29], and question answering[30].

This thesis focuses on hierarchical text classification (HTC), a subtask of text classification where labels form a hierarchical structure. The HTC has been receiving increasing attention from NLP researchers as hierarchical structures can be found

in real-world textual data of various domains, including e-commerce products[31], news articles[32, 33], patents[34, 35], and scientific articles[1]. Figure 1.1 shows an example of a scientific article and corresponding labels in a label hierarchy, where the goal of the task is to assign labels, which are depicted as filled circles, in the hierarchy to the input document.

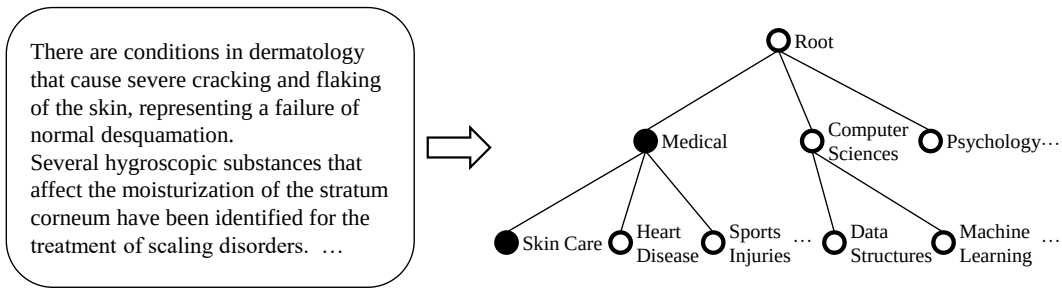


Figure 1.1: Example of input text document and label hierarchy in hierarchical text classification. The example is sampled from the WOS-46985 dataset[1], a dataset of scientific articles, where the document is the abstract by Perez *et al.*[2].

The importance of this task grows due to the accelerating accumulation of documents and the diversification of document topics (labels). Figure 1.2 depicts how fast new academic papers are submitted in arxiv.org[1], an open-access archive for academic articles. As the number of labels increases, a (hierarchical) structure is required to understand the relationships between the labels. Furthermore, more labels mean harder classification problems, which require a classification model based on an understanding of the relationship between labels.

While early works on HTC[36, 37, 38, 39, 40, 41] discard information on a given label hierarchy, Dumais and Chen[42] and Moyano *et al.*[43] have shown that effectively exploiting the hierarchy is the key to achieving good HTC performance. To this end, several methods have been adopted in deep learning-based HTC approaches to

[1]https://arxiv.org/

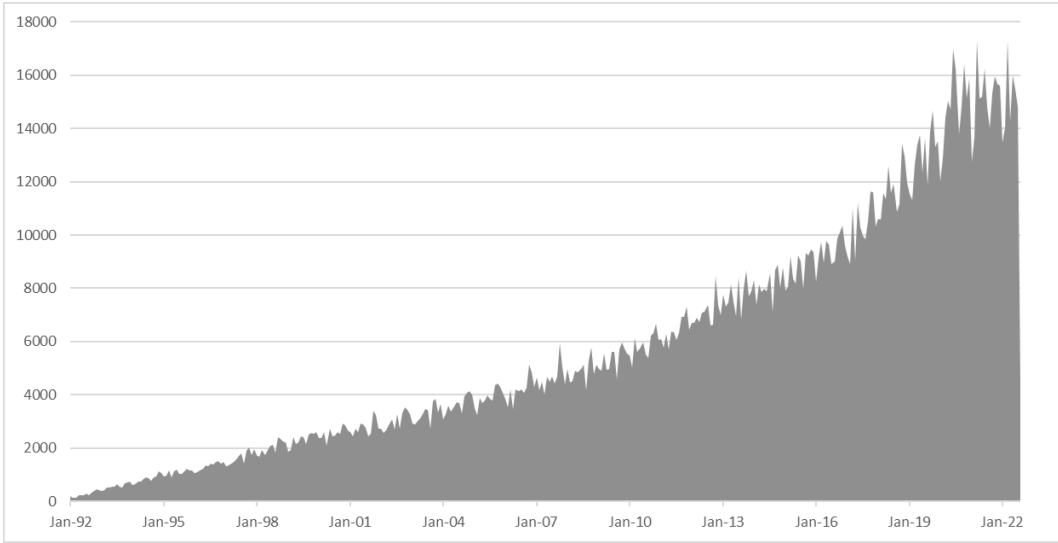


Figure 1.2: Number of monthly submissions in arxiv.org from January 1992 to August 2022.

make use of the hierarchy. The most common method used to capture the hierarchy is employing a graph neural network (GNN)[44, 45, 46, 47, 48, 49], which is a neural network designed to process data expressed as graphs[50]. Other approaches incorporate the label hierarchy by designing task-specific learning objectives[51, 52, 53] or employing meta-learning[54] and reinforcement learning[55]. While these approaches have achieved state-of-the-art HTC performance thanks to their capability to make use of the hierarchy, they do not analyze relationships outside the hierarchy, thus failing to understand the relationship between labels accurately.

On the other hand, text classification approaches that derive implicit label relationships from data have also been proposed. For example, Chatterjee *et al.*[56] and Chen *et al.*[57] employ hyperbolic neural networks, which are neural networks that operate on a hyperbolic space, as this space is known to be effective when expressing hierarchical structures[58, 59]. Other approaches infer the latent rela-

tionships between labels from textual descriptions or summaries of each label[60] or graphs constructed using label co-occurrence statistics[61]. However, these approaches lack the capability to utilize a given label hierarchy and show relatively poor HTC accuracy. Unfortunately, integrating methods that extract implicit label relationships and incorporate label hierarchy has been relatively less investigated despite the shortcomings of previous approaches.

In this thesis, a hierarchical text classification approach that exploits label hierarchy while capturing latent label relationships beyond the hierarchy is proposed. A novel architecture composed of three subnetworks, each for extracting textual representations, analyzing the label hierarchy, and learning the latent relationships between labels, is proposed. The proposed approach is trained using task-specific loss functions and post-processed with various methods designed to enhance hierarchical text classification performance. Through extensive experiments on widely-used benchmark datasets, this thesis shows that the proposed approach outperformed previous approaches. Furthermore, a novel algorithm is proposed to accelerate the proposed approach using textual descriptions of labels. Finally, this thesis employs the proposed approach on sentiment analysis tasks to demonstrate possible use cases of the approach.

1.2 Research Contribution

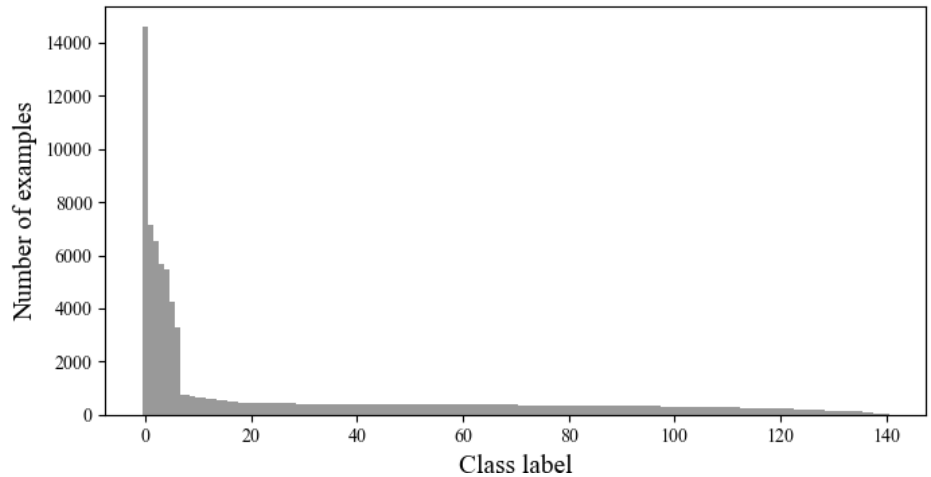
As described in Section 1.1, this thesis tackles hierarchical text classification by making use of structural information extracted from label hierarchy and capturing implicit relationships between labels. Detailed contributions of this work are presented in the following subsections.

1.2.1 Graph Attention Capsule Network

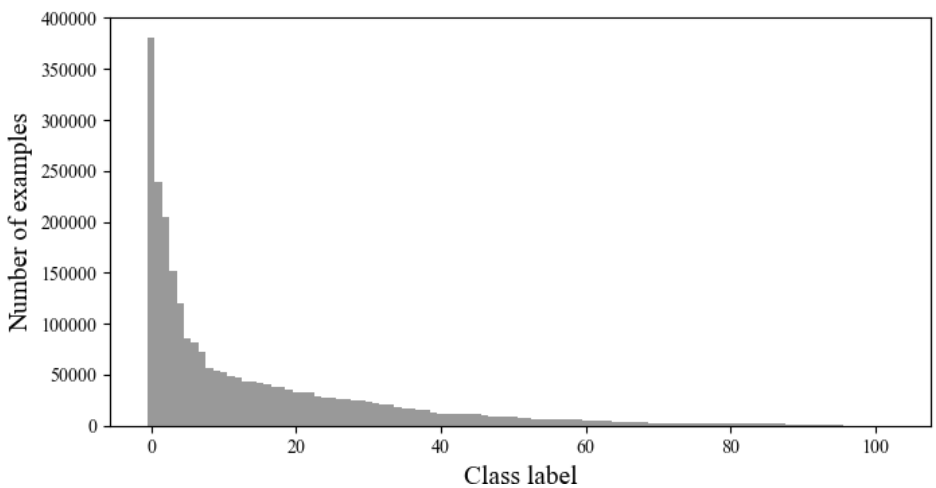
The first contribution of this work is proposing a novel neural network classifier named graph attention capsule network. As the name suggests, the proposed classifier comprises a graph neural network and a capsule network[62], each for handling explicit label relationships provided as a label hierarchy and implicit label relationships, respectively. While previous studies on capsule networks (CapsNets) focused on capturing relationships between implicit entities (objects, attributes, or structures) and class labels, the CapsNet in the proposed classifier is designed to learn relationships between class labels.

Furthermore, this work recognizes the innate class imbalance issue that lies in the task and trains the proposed network to mitigate this imbalance. The number of text documents assigned with each label of HTC datasets is illustrated in Figure 1.3 to demonstrate that such an issue exists in the task. Also, several cases where classification results do not agree with the given label hierarchy are defined, and a loss term and post-processing methods are proposed to avoid such contradictions.

The proposed approach was trained and evaluated using the WOS-46985 dataset[1] and the RCV1 dataset[32], which are two of the most commonly used HTC datasets. Experiment results showed that the graph attention capsule network outperformed



(a) The WOS-46985 dataset[1]



(b) The RCV1 dataset[32]

Figure 1.3: Number of examples in each label of hierarchical text classification datasets.

the baselines and that the network can capture interpretable latent label relationships. Ablation studies showed that each component proposed or employed in this work contributed towards this enhanced performance.

1.2.2 Semantic-Aware Dynamic Routing Algorithm

This thesis also proposes a semantic-aware dynamic routing algorithm for incorporating label semantics into the classifier discussed in Subsection 1.2.1. A dynamic routing algorithm[63, 64] is an iterative algorithm that captures latent relationships in a CapsNet. The proposed algorithm injects label semantic information into representations passed onto a CapsNet, allowing the network to identify implicit relationships with the help of the semantic information. Furthermore, a new initialization method based on semantic similarities is proposed and employed in this algorithm.

GACaps-HTC with the semantic-aware dynamic routing algorithm was trained and evaluated using the WOS-46985 dataset and the RCV1 dataset. This model was compared with GACaps-HTC with the dynamic routing algorithm proposed by Zhao *et al.*[64], which is the model described in Subsection 1.2.1, and variations of GACaps-HTC exploiting label semantic representations. Experiment results showed that injecting label semantic information using the proposed dynamic routing algorithm outperformed other variations with label semantic representations and that utilizing the proposed algorithm could accelerate the convergence of a classifier while maintaining (or slightly enhancing) its performance.

1.2.3 Employing Graph Attention Capsule Network on Aspect Category Sentiment Analysis

This thesis employs the model on aspect category sentiment analysis, a subtask of sentiment analysis where the goal is not only to extract the sentiment polarities in a given document but also their subjects, to investigate practical use cases for the proposed model. Aspect category sentiment analysis is transformed into a hierarchical text classification task by constructing a hierarchy of aspect categories and

attaching sentiment polarities as leaf nodes in the hierarchy.

GACaps-HTC was trained and evaluated using the SemEval2015 datasets[6] and the SemEval2016 datasets[65]. Specifically, the Laptop2015 and Restaurant2015 datasets from SemEval2015 datasets and the Laptop2016 and Restaurant2016 datasets from SemEval2016 datasets were used. Experiment results showed that the proposed model could achieve competitive or better performance compared to previous work, indicating that GACaps-HTC can achieve good HTC performance and can be employed for other practical applications involving HTC.

1.2.4 Summary of the Contributions

In short, main contributions of this work are as follows:

(a) A novel approach for hierarchical text classification using a GNN and a CapsNet is proposed to exploit label hierarchy and capture label relationships.

(b) A dynamic routing algorithm that incorporates label semantic information in a CapsNet is newly proposed to aid a CapsNet-based classifier to better understand relationships between labels.

(c) The effectiveness of the proposed approaches was evaluated using widely-used datasets, and the quantitative results showed that the proposed approaches outperformed previous approaches. The qualitative results showed that the approaches are interpretable and that they capture intuitive label relationships.

(d) The effectiveness of the proposed approach was evaluated on aspect category sentiment analysis to investigate other practical use cases of the approach.

1.3 Thesis Outline

The rest of the thesis is organized as follows: In Chapter 2, a literature review on related topics, including HTC, GNN, CapsNet, and aspect category sentiment analysis, is conducted. The proposed HTC approach and its experiment results are explained in Chapter 3. In Chapter 4, a semantic-aware dynamic routing algorithm proposed to complement GACaps-HTC is introduced along with related experiment results. Chapter 5 explains how aspect category sentiment analysis is transformed into a hierarchical text classification problem and demonstrates the effectiveness of GACaps-HTC in identifying sentiments in a given document. Finally, in Chapter 6, concluding remarks and future work are presented.

Chapter 2

Literature Review

2.1 Hierarchical Text Classification

Silla and Freitas[66] groups HTC approaches into flat, local, and global approaches, as illustrated in Figure 2.1, based on how the hierarchical structure of the labels is explored. In Figure 2.1, a dotted box represents a group of labels, illustrated as circles, that a classifier is responsible for in each approach. Flat approaches discard information on the hierarchy and transform the task into a simple text classification. A set of flat approaches ignores a subset of labels to convert the task into a single-label classification[36, 67, 68]. For example, Fürnkranz *et al.*[36] assigns only leaf labels to a text document while ignoring nonleaf labels in the hierarchy. Other flat approaches treat HTC simply like multi-label text classification[38, 56, 69]. Such approaches provide a simple solution for HTC, but Dumais and Chen[42] reveals that flat approaches achieve suboptimal performance as they do not utilize the information on label relationships, which is crucial for the task[43].

Local approaches place multiple classifiers, where each classifier is responsible for a partial hierarchy. The first method to implement a local approach is to assign one binary classifier per label. Fagni and Sebastiani[70] proposes an approach with label-wise binary classifiers and a negative sampling method that selects each classifier's

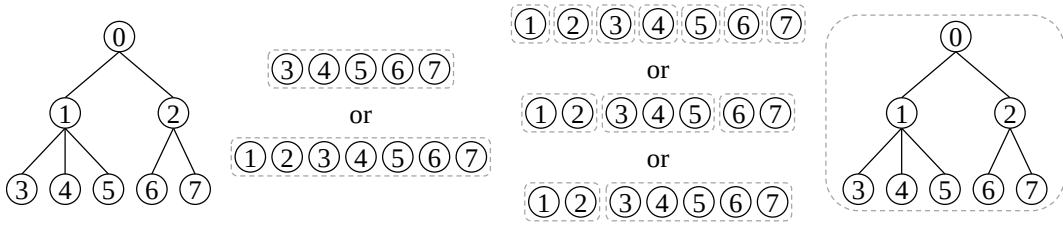


Figure 2.1: Comparison between flat, local, and global HTC approaches.

training examples for faster training. Banerjee *et al.*[71] presents an approach that initializes a binary classifier for each label as that of its parent label under the assumption that dependencies between a label and its parent can be modeled using transfer learning.

Another set of local approaches uses a classifier for each parent label, where the classifier categorizes a text document into one of the child labels. Dumais and Chen[42] employs a support vector machine (SVM)[72] classifier for each parent label to classify web contents in a top-down fashion. Krendzelak and Jakab[73] proposes an approach that places a convolutional neural network (CNN)[74] classifier for each parent label in the hierarchy of news article labels and demonstrated that their approach outperformed flat approaches and previous local approaches using SVMs.

Finally, a local approach can be implemented by employing a classifier for each hierarchy level. The local approach proposed by Shimura *et al.*[75] defines level-wise classifiers that share a group of parameters while other parameters are fine-tuned for each level. The approach presented by Wehrmann *et al.*[76] trains local classifiers for different hierarchy levels and a global classifier that performs multi-label classification for all labels in the hierarchy. The results obtained from the

local classifiers and the global classifier are then aggregated to produce the final prediction. Huang *et al.*[77] also makes use of level-wise classifiers and a global classifier and employed the attention mechanism[5] to achieve better classification accuracy. These local approaches have been shown to outperform flat approaches, but they are known to be not scalable to the size of the label hierarchy[55]. Furthermore, they are vulnerable to error propagation[78], as a classification error in one classifier can lead to a prediction result that is entirely wrong.

In contrast, global approaches make use of a single multi-label classifier that exploits the label hierarchy. One method to achieve this is to train the classifier using a learning objective designed to take advantage of the hierarchy. Recursive regularization[51] encourages the parameters responsible for a label and its parent to be similar. This method has been shown to improve the HTC performance of SVM and logistic regression approaches. Peng *et al.*[52] proposes an approach that trains a neural network classifier using recursive regularization and demonstrated that this regularization led to enhanced performance. The loss term proposed by Yu *et al.*[53] penalizes contradictions where a document is assigned with a label but not with its parent and demonstrated the effectiveness of this loss term when training a neural network.

A global approach can also incorporate the information on the label hierarchy by employing a GNN, a class of deep learning models designed to analyze data described by graphs[50]. The detailed introduction and literature review on GNN are presented in Subsection 2.2. Zhou *et al.*[45] proposes an approach that analyzes the top-down and bottom-up paths in the hierarchy separately and obtains label-specific textual representations with a GNN. Chen *et al.*[46] defines a neural network that maps

a label representation, which a GNN extracts, and a textual representation onto a joint representation space. The network is trained to minimize a loss term based on distances between a document's textual representation and the representations of the coarse-grained, fine-grained, and irrelevant labels. The approach proposed by Deng *et al.*[47] trains a classifier to maximize the mutual information between a textual representation and a label representation obtained by a GNN so that the textual representation contains crucial information for HTC. Wang *et al.*[49] uses a GNN to generate label representations and performed contrastive learning to encourage textual and label representations to be closer while pushing textual representations away from each other. In addition, Xu *et al.*[79] designs a GNN-based approach that processes a joint graph of words and labels to learn hierarchy-aware word representations for HTC.

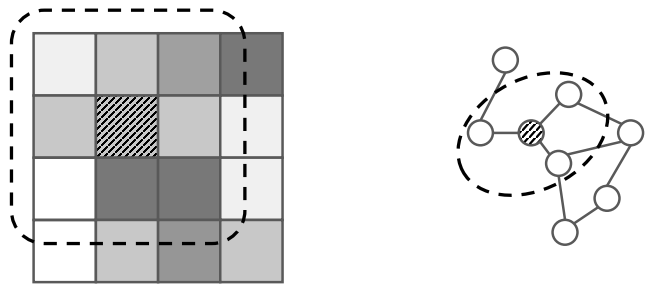
Other methods have also been proposed to leverage information on the hierarchy in global approaches. Yu *et al.*[53] proposes post-processing methods that ensure a document is classified as a child label only if it is classified as its parent. Mao *et al.*[55] transforms HTC into the task of traversing through the label hierarchy and proposed a reinforcement learning approach that learns a label assignment policy. Meta-learning has also been employed in HTC to search for the optimal learning rate and classification threshold for each label[80] or enhance classification performance on few-shot labels in the hierarchy[54].

2.2 Graph Neural Network

A GNN extracts representations on the nodes[81, 82], edges[83, 84], or entire graph[85, 86] from data represented by graphs. Early work on GNNs, such as Scarselli *et al.*[50], Gallicchio and Micheli[87], and Li *et al.*[88], obtains a node representation by recurrently aggregating information propagated from neighboring nodes until the representation converges. These approaches are referred to as recurrent graph neural networks[89]. However, the process of iterative propagation and aggregation is computationally expensive[90], calling for the need to develop more efficient techniques to analyze graphs.

To overcome this challenge, Bruna *et al.*[91] proposes a graph convolutional network (GCN) that utilizes filters with shared weights for propagating the features of neighboring nodes motivated by the success of CNNs in terms of performance and efficiency. The similarity between a CNN and a GCN is briefly illustrated in Figure 2.2, where dotted outlines denote groups of pixels or nodes that participate in obtaining representations corresponding to the pixel or the node filled with diagonal lines. Furthermore, Zhou *et al.*[45] introduces a hierarchy-GCN for generating node representations given a set of nodes that form a hierarchical structure. A hierarchy-GCN obtains a node representation by adding a top-down, bottom-up, and loop representation, each propagated from its parent node, child nodes, and itself, respectively, and therefore can take parent-child relationships into account rather than simple connectivity. This network has been shown to outperform other GNNs when adopted for HTC.

On the other hand, inspired by the attention mechanism[5], which enables a neural network to focus on the relevant information in a representation, Veličković *et*



Convolutional Neural Network Graph Convolutional Network

Figure 2.2: Illustrations of a two-dimensional convolutional neural network and a graph convolutional network.

al.[92] introduces a graph attention network, which employs the attention mechanism on a GNN. A GAT concatenates representations of a node and its neighbor before transforming the concatenated representations into a scalar weight, indicating how much information is propagated from the neighbor. It has been demonstrated that a GAT has better expressiveness than a GCN with similar time complexity due to its capability of assigning a different level of importance to each neighbor.

2.3 Capsule Network

A CapsNet is a neural network composed of groups, or capsules, of neurons, each corresponding to an object, attribute, or structure[62], initially proposed for image classification. The norm of a capsule's activation vector is interpreted as the probability that the corresponding object or attribute exists in an image, while the direction of the vector represents the properties of the corresponding entity. Two types of capsules are defined in a CapsNet: primary capsules and digit capsules. A primary capsule in a CapsNet is a group of neurons that captures a latent object and propagates its information to digit capsules via the dynamic routing algorithm[63]. The algorithm first defines a coupling coefficient for each pair of a primary capsule and a digit capsule. This coefficient determines how much information is propagated from the primary capsule to the digit capsule. In each iteration, the algorithm updates each coefficient using the activation vectors of the corresponding primary and digit capsules. Then, the algorithm calculates the activation vectors of digit capsules from those of primary capsules and the updated coefficients before proceeding to the next iteration. A digit capsule corresponds to a class label, and the norm of its activation vector, obtained by the dynamic routing algorithm, is used as the class probability.

Several methods have been proposed to enhance the training efficiency and performance of CapsNets. Xiang *et al.*[93] extends the idea of dropout[94], a technique developed to prevent neural networks from overfitting, to CapsNets and proposed capsule dropout. While dropout randomly discards some of a network's neurons while training, capsule dropout randomly removes a portion of primary capsules during training. When employing common dropout in a CapsNet, random elements

are dropped in a primary capsule, and the direction of the capsule is altered, changing the properties of the capsule's corresponding entity, where such perturbations can lead to wrong predictions. Zhao *et al.*[64] proposes an adaptive dynamic routing algorithm that iterates until the routing results converge rather than using a fixed number of iterations. By doing so, the adaptive algorithm guarantees instance-level convergence for each individual example, decreasing the risk of unreliable routing[64]. Gu and Tresp[95] demonstrates that gradients that occur from iterative processes of dynamic routing algorithms have an insignificant influence on backpropagation and proposed detached dynamic routing for better time and memory efficiency.

On the other hand, Jeong *et al.*[96] proposes a capsule pruning algorithm that discards primary capsules with small activation vector norms. These primary capsules correspond to objects or attributes that are unrelated to the given example. Therefore, removing such capsules and retaining only the capsules with relevant information can enhance a CapsNet's generalizability and prevent it from overfitting[96]. Huang and Zhou[97] proposes a dual-attention mechanism CapsNet which employs the attention mechanism to highlight crucial information in primary capsules before propagating the information to digit capsules and demonstrated its effectiveness in image classification.

While CapsNets were initially proposed and researched in the field of computer vision, they have recently been actively studied for NLP and shown to successfully capture underlying structures in a text document[98], and several approaches have adopted them for HTC. The approach proposed by Aly *et al.*[99] obtains activation vectors of primary capsules using a CNN and performed text classification using a dynamic routing algorithm. Peng *et al.*[100] improves this approach by replacing the

CNN with a recurrent CNN[101] to obtain sequence-aware textual representations. Wang *et al.*[102] proposes a hierarchical bidirectional CapsNet, which propagates information through the label hierarchy by alternating between a top-down and a bottom-up fashion. By doing so, both the local relationships between labels and the global hierarchy are effectively captured by the CapsNet and exploited for HTC.

2.4 Exploiting Label Semantics for Classification

Recently, an increasing number of studies have recognized that label semantics contain valuable information for text classification. The first group of such studies focuses on few-shot or zero-shot text classification as exploiting label semantics enables classification via matching a label representation and a text representation in a shared representation space. Chen *et al.*[103] uses a convolutional deep structured semantic model[104], which maps a text representation obtained by max-pooling word representations to a label representation space, for zero-shot speaker intent classification. Puri and Cantanzaro[105] proposes a zero-shot classification-based question answering approach that concatenates a given question and candidate answers (labels) to a given text for a language model to extract token representations in a joint space. Hou *et al.*[106] uses semantic similarity between words for few-shot conditional random field[107]-based slot tagging tasks, which are similar to named entity recognition tasks. Halder *et al.*[108] formulates each few-shot classification task into a group of binary classification tasks and fed a joint sentence of a label and a text into a binary classifier. Luo *et al.*[109] proposes a few-shot learning approach that encourages representations extracted from texts appended with a label name to be more relevant to the corresponding label semantic representation.

Other studies have shown that incorporating label semantics can enhance not only few-shot or zero-shot performance, but also general text classification performance. Zhang *et al.*[110] demonstrates that by concatenating a document and label names and employing an attention mechanism, a classifier can extract word-word, word-label, and label-label correlations through attention. Pappas and Henderson[111] proposes an approach that utilizes label semantic representations as parameters

used to obtain logits in a classifier. Xiao *et al.*[112] proposes a label-specific attention network model that fuses textual representations obtained by self-attention and those obtained from text-label attention. Similarly, The approach proposed by Cai *et al.*[113] obtains representations from text-label attention, where label representations are obtained by feeding label semantics into a GNN.

Research on HTC approaches that employ label semantics has also been actively conducted. Similar to Zhang *et al.*[110], Zhang *et al.*[114] performs HTC by concatenating a document and label names and employing an attention mechanism. Yu *et al.*[115] tackles HTC as a label sequence generation task, where a generated (inferred) label's name is used as an input of a decoder for generating the next label. Chen *et al.*[46] and Wang *et al.*[49] generate label semantic representation using label names and GNNs to match a textual representation with the closest label semantic representation in a joint text-label representation space.

2.5 Aspect Category Sentiment Analysis

As conventional document-level or sentence-level sentiment analysis aims to identify the sentiment polarity of the entire document or sentence, it is assumed that a single topic appears in the document or sentence and that the entire document or sentence has a consistent sentiment polarity. However, in practice, multiple sentiment polarities corresponding to different topics may appear in a document, or even a sentence, raising the need for identifying more fine-grained sentiments. Aspect category sentiment analysis is a subtask of sentiment analysis widely used in real-world industries[116, 117] where the goal is to perform aspect category (subject of sentiments) detection and sentiment classification on the category simultaneously from a given text document.

There are several types of approaches that can tackle aspect category sentiment analysis. The first type is a pipeline approach[116], where an aspect category detection model and a sentiment polarity classification model are consecutively utilized. While pipeline approaches are straightforward, they suffer from low performance as the errors from aspect category detection limit the aspect category sentiment analysis performance. Furthermore, relationships between aspect category detection and sentiment polarity classification are ignored, where these relationships can play a crucial part in enhancing both tasks[118].

Approaches of the second type are Cartesian product approaches[119] which perform binary classification for all combinations of aspect categories and sentiment polarities. However, such approaches face the risk of assigning multiple sentiment polarities for an aspect category[120]. The third type is known as an add-one-dimension approach, where sentiment polarities ("positive," "neutral," and "negative") are ap-

pended with one extra option, "N/A," indicating that a given text document does not mention the corresponding aspect category[121].

The fourth type is a hierarchical classification approach, which aims to explicitly model the hierarchical relationships between aspect category detection and sentiment polarity classification. Cai *et al.*[120] proposes a hierarchical GCN-based approach where the lower-level GCN first detects aspect categories in a given document by capturing the relationships between categories, and the higher-level GCN predicts the sentiment polarities for each category by analyzing the relationships between sentiments and categories. Note that the hierarchical GCN proposed by Huang *et al.*[77] and that of Cai *et al.*[113] share the same name, but have different architectures.

Finally, there are sequence-to-sequence approaches that utilize a pretrained generative language model to produce sentences representing captured aspect categories and corresponding sentiment polarities. For example, given the sentence "I love this restaurant," a sequence-to-sequence approach returns "The sentiment polarity for the restaurant is positive." Liu *et al.*[122] suggests that these approaches are capable of exploiting how a pretrained language model understood natural language.

Chapter 3

Graph Attention Capsule Network for Hierarchical Text Classification

3.1 Problem Definition

The goal of text classification is to classify a text document into a set of predefined categories known as labels. Let D and L denote the input text document and the number of labels, respectively, and $\mathcal{Y}^D \subseteq \{1, \ldots, L\}$ is the ground-truth set of label indices corresponding to D. A text classification model learns a mapping from D to $\mathcal{Y}^D$ and outputs a classification probability for each label, where the classification probability of the l-th label is denoted as $p_l^D \in [0, 1]$.

This work tackles hierarchical text classification, a subtask of text classification with a label hierarchy. The label hierarchy is denoted as $\mathcal{H}$ and is represented as a set of tuples, where each tuple consists of a label and its child label as follows:

$$\mathcal{H} = \left\{ \left(l, l'\right) | 1 \leq l, l' \leq L, l\text{-th label is the parent of } l'\text{-th label} \right\}. \tag{3.1}$$

3.2 Methods

Inspired by the approaches mentioned in Section 2.1 and Section 2.2 that employ various methods to exploit the label hierarchy and learn label relationships, a novel global approach for HTC is proposed in this section, namely, the graph attention capsule network for hierarchical text classification (GACaps-HTC). The GACaps-HTC is composed of three subnetworks: a textual representation extractor, a hierarchy encoder, and an implicit relationship extractor. The textual representation extractor first extracts a textual representation of a document from a language model pretrained on a massive corpus. Then, a convolutional layer[74] generates a label-specific textual representation for each label in the hierarchy. These representations are fed into the hierarchy encoder comprised of a GNN to incorporate the hierarchy information into the label-specific representations, resulting in hierarchy-aware label-specific representations.

The hierarchy-aware label-specific representations are then passed onto the implicit relationship extractor. This subnetwork captures latent relationships between labels that are not expressed by label hierarchy and infers class probabilities. The implicit relationship extractor first employs the attention mechanism to highlight the relevant information in these representations. Then, a CapsNet uses the dynamic routing algorithm to capture relationships between labels via iterative updates and produce the classification probabilities for assigning a set of labels to the given document. Note that the algorithm was initially proposed to model the relationships between underlying structures or objects and labels rather than those between labels. Briefly, the hierarchy encoder is responsible for understanding the relationships between a label and its neighboring labels (parent and child labels) in the hierarchy,

whereas the implicit relationship extractor captures broader relationships. Various methods developed to enhance CapsNets, including capsule dropout[93], adaptive iteration[64], and capsule pruning[96], are employed in GACaps-HTC. The overall architecture of GACaps-HTC is depicted in Figure 3.1. The details for each subnetwork are described in the following subsections.

3.2.1 Textual Representation Extractor

The ability to extract a high-quality textual representation from a document is critical for a model to understand and classify the document. To this end, a Transformer-based language model[3], illustrated in Figure 3.2, pretrained on a large-scale corpus, is employed as it has shown to be effective in various downstream tasks by capturing long-range contexts[123]. Let $\boldsymbol{X}^D \in \mathbb{R}^{|D| \times d_{LM}}$ denote the output of the language model where $|D|$ is the number of tokens in D and d_{LM} is the output size of the language model per token.

Then, a convolutional layer transforms $\boldsymbol{X}^D$ into the input of the next subnetwork. The r-th row of the convolutional layer output, denoted as $\text{Conv}\left(\boldsymbol{X}^D\right) \in \mathbb{R}^{|D| \times d_{Conv}}$, is obtained as follows:

$$\text{Conv}\left(\boldsymbol{X}^D\right)_{[r,:]} = \text{ReLU}\left(\boldsymbol{W}_{Conv}\boldsymbol{X}^D_{[r-1:r+1,:]} + \boldsymbol{b}_{Conv}\right). \tag{3.2}$$

$\boldsymbol{W}_{Conv}$ and $\boldsymbol{b}_{Conv}$ are the weight and bias parameters in the convolutional layer, respectively, and d_{Conv} denotes the number of output channels in the layer. The rectified linear unit (ReLU)[124] is a piece-wise linear function commonly employed to introduce non-linearity to a neural network.

A document-level textual representation, a vector of length d_{Conv}, is obtained by max-pooling the convolutional layer output along the first dimension. This rep-

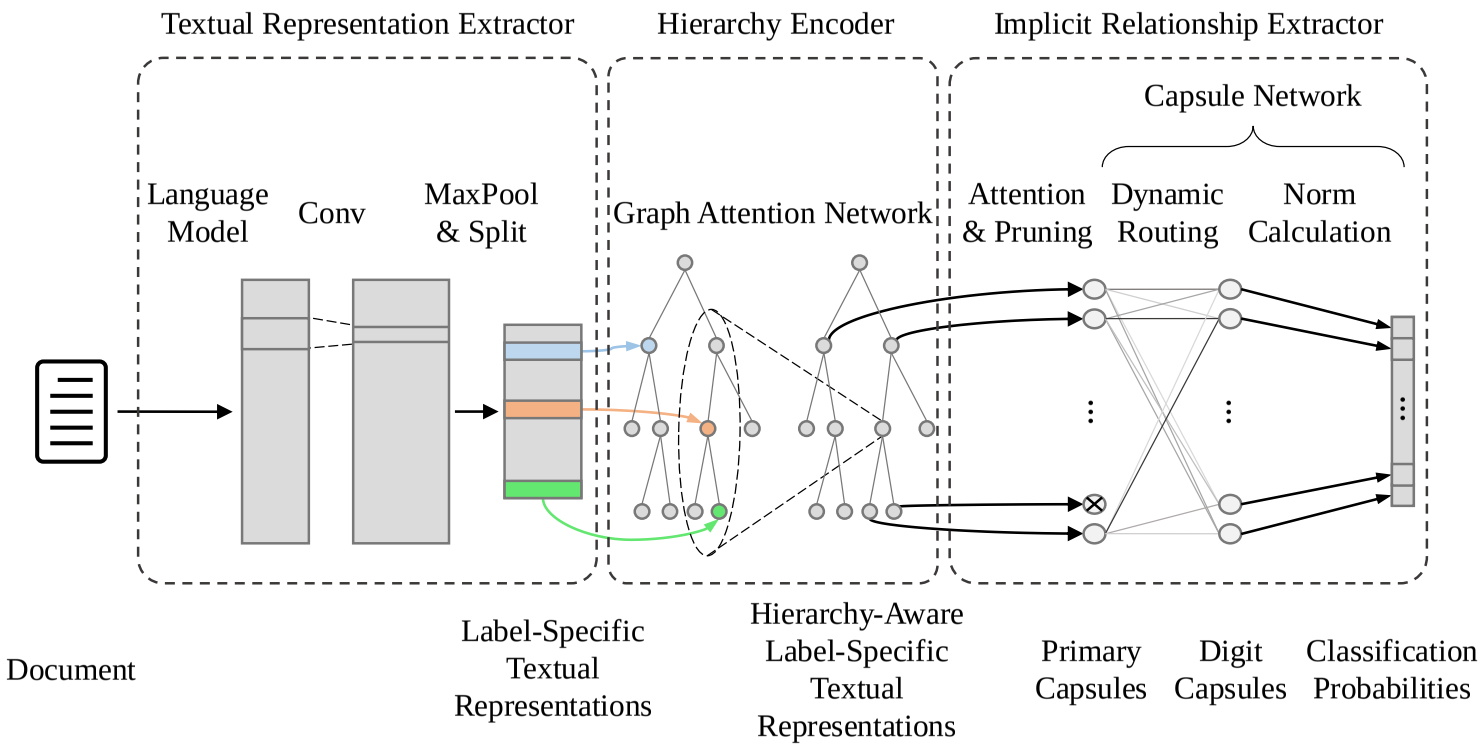


Figure 3.1: Overview of the graph attention capsule network for hierarchical text classification (GACaps-HTC) consisting of a textual representation extractor, hierarchy encoder, and implicit relationship extractor. Conv denotes a convolutional layer.

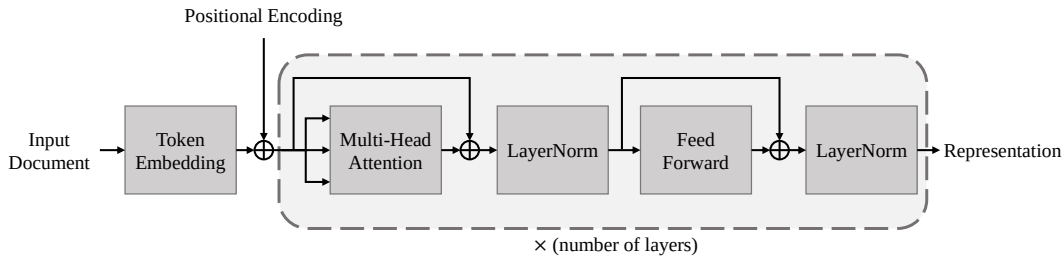


Figure 3.2: Transformer architecture[3]. LayerNorm denotes layer normalization[4] and multi-head attention denotes a process of performing the attention mechanism proposed by Bahdanau *et al.*[5] several times in a parallel fashion.

resentation is split into L vectors of length d_{Conv}/L, and each vector undergoes an affine transformation to generate a label-specific textual representation. The intuition behind extracting label-specific representations is that the model requires the ability to capture distinct pertinent characteristics that are preferred in discriminating each label[61, 125, 126, 127]. As these label-specific representations are passed to the hierarchy encoder, d_{Conv}/L equals the hierarchy encoder's input and output size per label, denoted by d_{HE}. Let $\boldsymbol{z}_l^D \in \mathbb{R}^{d_{HE}}$ denote the label-specific representation corresponding to the l-th label, which is obtained as follows:

$$\boldsymbol{z}_l^D = \boldsymbol{W}_{Aff}\text{MaxPool}\left(\text{Conv}\left(\boldsymbol{X}^D\right)_{[:,(l-1)d_{HE}:ld_{HE}]}\right) + \boldsymbol{b}_{Aff}. \tag{3.3}$$

$\boldsymbol{W}_{Aff} \in \mathbb{R}^{d_{HE} \times d_{HE}}$ and $\boldsymbol{b}_{Aff} \in \mathbb{R}^{d_{HE}}$ are the weight and bias parameters for the affine transformation, and MaxPool denotes the max-pooling operation.

3.2.2 Hierarchy Encoder

Although $\boldsymbol{z}_l^D$ generated by the textual representation extractor contains information on the content of D, it lacks information on the hierarchy. The second subnetwork in GACaps-HTC, namely the hierarchy encoder, embeds the information on the

hierarchy into the representations. In this work, a GAT is selected as the hierarchy encoder due to its ability to understand how relevant each neighboring label is in generating a hierarchy-aware label-specific representation.

Let $\boldsymbol{W}_{HE} \in \mathbb{R}^{1\times 2d_{HE}}$ denote the parameters in the hierarchy encoder. The subnetwork first infers an attention weight $w^D_{ll'} \in \mathbb{R}$ for each pair of label indices $(l, l') \in \mathcal{H}$. This weight indicates how important the l'-th label is for generating the l-th label's label-specific representation and is inferred as follows[92]:

$$w^D_{ll'} = \text{LeakyReLU}\left(\boldsymbol{W}_{HE}\text{Concat}\left(\boldsymbol{z}^D_l, \boldsymbol{z}^D_{l'}\right)\right). \tag{3.4}$$

LeakyReLU[128] is a piece-wise linear function similar to ReLU, and Concat is the operation of concatenating multiple vectors.

These weights are normalized to represent the relative importance of each neighboring label. Let $\mathcal{N}_l$ denote the set of indices corresponding to labels that neighbor the l-th label, including itself, which can be inferred from $\mathcal{H}$ as follows:

$$\mathcal{N}_l = \left\{l' | \left(l, l'\right) \in \mathcal{H}\right\} \cup \left\{l'' | \left(l'', l\right) \in \mathcal{H}\right\} \cup \{l\}. \tag{3.5}$$

The attention weight corresponding to the pair (l, l') is normalized as follows:

$$\tilde{w}^D_{ll'} = \begin{cases} \frac{\exp\left(w^D_{ll'}\right)}{\sum_{i\in\mathcal{N}_l}\exp\left(w^D_{li}\right)} & \text{if } l' \in \mathcal{N}_l \\ 0 & \text{otherwise,} \end{cases} \tag{3.6}$$

where $\tilde{w}^D_{ll'}$ is the normalized attention weight. Normalized weights between non-neighboring labels are set to zero as only the neighboring labels participate in generating the hierarchy-aware representation corresponding to a label.

Finally, the hierarchy encoder propagates the label-specific textual representa-

tions ($\boldsymbol{z}_l^D$) according to the normalized attention weights and returns hierarchy-aware textual representations. The hierarchy-aware representation corresponding to the l-th label is denoted as $\boldsymbol{v}_l^D$ and is obtained as follows:

$$\boldsymbol{v}_l^D = \text{ReLU}\left(\sum_{1 \leq l' \leq L} \tilde{w}_{ll'}^D \boldsymbol{z}_{l'}^D\right). \tag{3.7}$$

The process of obtaining $\boldsymbol{v}_l^D$ is summarized in Figure 3.3, where the process of attaining the hierarchy-aware textual representation corresponding to the second label (highlighted in gray) is illustrated. Different shades of arrowed lines in the rightmost graph denote different levels of attention weights.

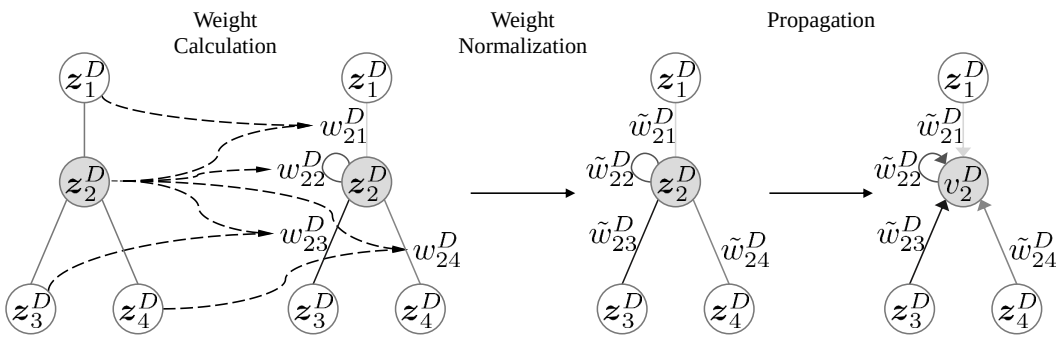


Figure 3.3: Summarization of the hierarchy encoder composed of a graph attention network.

3.2.3 Implicit Relationship Extractor

The implicit relationship extractor in GACaps-HTC predicts the classification probability of each label from the representations generated by the hierarchy encoder. The subnetwork first highlights the crucial information in the representations via the attention mechanism before utilizing them as the primary capsules' activation vectors of the subnetwork's CapsNet. Adopting the attention mechanism in this fashion has been shown to enhance the performance of a CapsNet by emphasizing vital

information while reducing the less important information in primary capsules[97]. Let $\boldsymbol{a}^D \in \mathbb{R}^L$ denote the vector composed of each primary capsule's attention weight and $\boldsymbol{u}_l^D \in \mathbb{R}^{d_{HE}}$ denote the activation vector of the l-th primary capsule in the CapsNet. Using the weight and bias parameters $\boldsymbol{W}_{Attn}^1 \in \mathbb{R}^{L \times Ld_{HE}}$, $\boldsymbol{W}_{Attn}^2 \in \mathbb{R}^{L \times L}$, and $\boldsymbol{b}_{Attn}^1, \boldsymbol{b}_{Attn}^2 \in \mathbb{R}^L$, $\boldsymbol{a}^D$ and $\boldsymbol{u}_l^D$ are obtained as follows[97]:

$$\boldsymbol{a}^D = \tanh\left(\boldsymbol{W}_{Attn}^2 \mathrm{ReLU}\left(\boldsymbol{W}_{Attn}^1 \mathrm{Concat}\left(\boldsymbol{v}_1^D, \ldots, \boldsymbol{v}_L^D\right) + \boldsymbol{b}_{Attn}^1\right) + \boldsymbol{b}_{Attn}^2\right), \tag{3.8}$$

$$\boldsymbol{u}_l^D = \left(1 + \boldsymbol{a}_{[l]}^D\right) \boldsymbol{v}_l^D. \tag{3.9}$$

tanh denotes the hyperbolic tangent operation.

After the activation vectors of the primary capsules are inferred, primary capsules with small activation vector norms are pruned to enhance GACaps-HTC's generalizability and prevent overfitting. These primary capsules are expected to correspond to labels irrelevant to D and contain unimportant information. Let $\rho \in [0, 1)$ denote the hyperparameter indicating the pruning ratio. ρL primary capsules with the smallest activation vector norms are pruned, and their activation vectors are set to zero, while the remaining $(1 - \rho)L$ capsules participate in inferring the classification probabilities.

The CapsNet in the implicit relationship extractor acquires the activation vectors of the digit capsules, each corresponding to a label, where the norm of the activation vector indicates the predicted classification probability of the label. The dynamic routing algorithm, which updates how much information is passed on from a primary capsule to a digit capsule in an iterative fashion, is employed to obtain these activation vectors. The dynamic routing algorithm proposed by Zhao *et al.*[64] is adopted in this work as it has been shown to outperform the dynamic routing

algorithm proposed by Sabour *et al.*[63] on NLP tasks by iterating the algorithm until every example converges[64].

Let $\boldsymbol{o}_l^D \in \mathbb{R}^{d_{Caps}}$ denote the activation vector of the l-th digit capsule corresponding to the l-th label, where d_{Caps} is the size of a digit capsule's activation vector. The detailed dynamic routing algorithm that outputs the activation vectors of the digit capsules from those of the primary capsules is described in Algorithm 1. $c_{ll'}^D \in \mathbb{R}$ and $\tilde{c}_{ll'}^D \in \mathbb{R}$ defined in lines 1 and 6 are the coupling coefficient and normalized coefficient indicating how much information the l-th digit capsule receives from the l'-th primary capsule. The activation vector propagated from the l'-th primary capsule to the l-th digit capsule is referred to as $\boldsymbol{\mu}_{ll'}^D$ and obtained in line 7, where $\boldsymbol{W}_{Caps,l} \in \mathbb{R}^{d_{Caps} \times d_{HE}}$ is the matrix of parameters defined for the l-th digit capsule. The activation vectors of the digit capsules are obtained as the weighted sum of the propagated vectors with the normalized coupling coefficients as weights, as indicated in line 8.

The idea behind the dynamic routing algorithm is that a primary capsule should propagate more information to digit capsules that are similar to itself. Therefore, the algorithm measures the distance, denoted as dist in Algorithm 1, between the activation vectors of the digit capsules and those propagated from the primary capsules in line 9. Then, as shown in line 10, $c_{ll'}^D$ is updated to increase proportionally to $-\text{dist}\left(\boldsymbol{o}_l^D, \boldsymbol{\mu}_{ll'}^D\right)$.

The process of assigning higher coefficients to similar capsules can be interpreted as maximizing the sum of $c_{ll'}^D \left(1 - \text{dist}\left(\boldsymbol{o}_l^D, \boldsymbol{\mu}_{ll'}^D\right)\right)$[64], and the algorithm iterates until this summed value converges, as described in lines 13 and 14 of Algorithm 1. Note that the activation vectors are squashed to have norms between zero and one

Algorithm 1: Dynamic Routing Algorithm[64]

Input: activation vectors of primary capsules $\boldsymbol{u}_l^D$ for $1 \leq l \leq L$
Output: activation vectors of digit capsules $\boldsymbol{o}_l^D$ for $1 \leq l \leq L$

$c_{ll'}^D \leftarrow 0 \quad \forall l, l'$;
prev_Score $\leftarrow -\infty$;
while True **do**
  for $l \leftarrow 1$ **to** L **do**
    for $l' \leftarrow 1$ **to** L **do**
      $\tilde{c}_{ll'}^D \leftarrow \frac{\exp\left(c_{ll'}^D\right)}{\sum_{1 \leq i \leq L} \exp\left(c_{il'}^D\right)}$;
      $\boldsymbol{\mu}_{ll'}^D \leftarrow \boldsymbol{W}_{Caps,l} \boldsymbol{u}_{l'}^D$;
      $\boldsymbol{o}_l^D \leftarrow \sum_{1 \leq l' \leq L} \tilde{c}_{ll'}^D \boldsymbol{\mu}_{ll'}^D$;
      $\text{dist}\left(\boldsymbol{o}_l^D, \boldsymbol{\mu}_{ll'}^D\right) \leftarrow \left\| \frac{\|\boldsymbol{\mu}_{ll'}^D\|}{0.5+\|\boldsymbol{\mu}_{ll'}^D\|^2} \boldsymbol{\mu}_{ll'}^D - \frac{\|\boldsymbol{o}_l^D\|}{0.5+\|\boldsymbol{o}_l^D\|^2} \boldsymbol{o}_l^D \right\|$;
      $c_{ll'}^D \leftarrow c_{ll'}^D + \left(1 - \text{dist}\left(\boldsymbol{o}_l^D, \boldsymbol{\mu}_{ll'}^D\right)\right)$;
    end
  end
  Score $\leftarrow \log\left(\sum_{1 \leq l, l' \leq L} c_{ll'}^D \left(1 - \text{dist}\left(\boldsymbol{o}_l^D, \boldsymbol{\mu}_{ll'}^D\right)\right)\right)$;
  if $|\text{Score} - \text{prev\_Score}| \leq \epsilon$ **then**
    $\boldsymbol{o}_l^D \leftarrow \frac{\|\boldsymbol{o}_l^D\|}{0.5+\|\boldsymbol{o}_l^D\|^2} \boldsymbol{o}_l^D \quad \forall l$;
    return $\boldsymbol{o}_l^D \quad \forall l$;
  end
  prev_Score $\leftarrow$ Score ;
end

in lines 9 and 15. $\epsilon \in \mathbb{R}^+$ is the hyperparameter indicating the threshold determining whether the algorithm has converged. After the dynamic routing algorithm has ended, the classification probability of the l-th label p_l^D is inferred as $p_l^D = \left\|\boldsymbol{o}_l^D\right\|$.

During the training phase, capsule dropout[93] is applied to the CapsNet for enhanced generalizabilty. For a given hyperparameter $\phi \in [0, 1)$ indicating the dropout rate, each primary capsule is removed with the probability of ϕ during the training phase.

3.2.4 Optimization

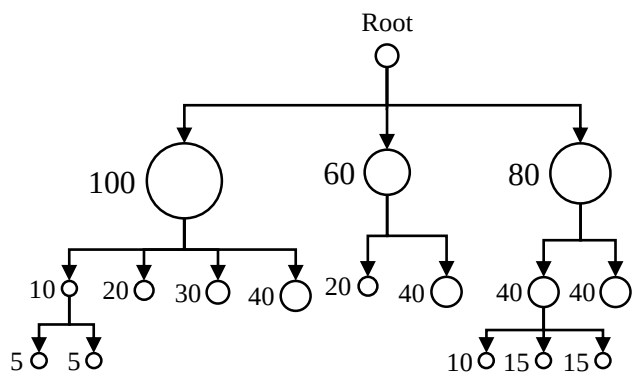


Figure 3.4: An example case of label imbalance naturally occurring in hierarchical text classification.

As a coarse-grained label closer to the root of the hierarchy is more likely to be assigned to more documents than a fine-grained label closer to a leaf, HTC has an innate label imbalance issue[47, 129], as illustrated in Figure 3.4. Circles in Figure 3.4 represent labels, while their sizes and numbers beside them denote the number of data examples corresponding to the labels. Training a classifier using a dataset with such an imbalance can lead to the classifier overfitting to labels with the majority

of training examples while performing poorly on sparse labels[130]. To overcome the imbalance issue, which is one of the key challenges in HTC[55], GACaps-HTC is trained using focal loss[130]. The focal loss relieves a neural network from being overwhelmed by labels with most data examples by reducing the relative loss for easy classification (labels with many examples) and focusing on difficult classification (labels with few examples). The focal loss (FL) is obtained as follows:

$$FL\left(\mathcal{Y}^D, p_1^D, \ldots, p_L^D\right) = -\sum_{l \in \mathcal{Y}^D} \left(1 - p_l^D\right)^\gamma \log\left(p_l^D\right) - \sum_{l' \notin \mathcal{Y}^D} \left(p_{l'}^D\right)^\gamma \log\left(1 - p_{l'}^D\right). \quad (3.10)$$

$\gamma \in \mathbb{R}^+$ denotes the hyperparameter indicating the importance of misclassified examples, likely to be those of labels with few examples. This loss function replaces the binary cross-entropy (BCE) loss, which is the most commonly employed loss for multi-label classification. The BCE loss is calculated as follows:

$$BCE\left(\mathcal{Y}^D, p_1^D, \ldots, p_L^D\right) = -\sum_{l \in \mathcal{Y}^D} \log\left(p_l^D\right) - \sum_{l' \notin \mathcal{Y}^D} \log\left(1 - p_{l'}^D\right). \quad (3.11)$$

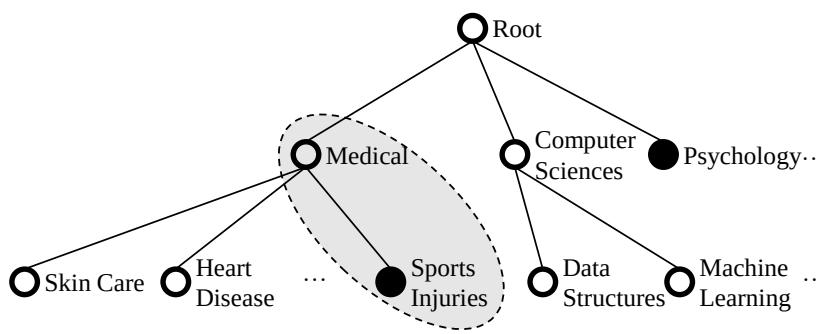


Figure 3.5: An example case of a classification result not coinciding with label hierarchy.

Furthermore, a contradiction penalty term motivated by Yu *et al.*[53] is proposed to encourage the model's classification results to coincide with the label hierarchy.

By minimizing this term, a neural network is trained to assign a label to a document only if the document is assigned the parent label as a document of a child label is trivially also of its parent. An example case of a classification result contradicting the label hierarchy is depicted in Figure 3.5, where filled circles denote a classifier's predictions and the dotted ellipse denotes the part where the result contradicts the hierarchy as a document can be categorized as a child label only if it is classified as the parent label. The proposed contradiction penalty term denoted as *CP* is calculated as follows:

$$CP\left(\mathcal{H}, p_1^D, \ldots, p_L^D\right) = \sum_{1 \leq l \leq L} \max\left(\left| p_l^D - \max_{1 \leq l' \leq L} \left(p_{l'}^D \mathbb{1}_{(l,l') \in \mathcal{H}}\right)\right|, \delta\right). \tag{3.12}$$

$\mathbb{1}_{(l,l') \in \mathcal{H}}$ is the binary indicator that returns 1 if (l, l') is in $\mathcal{H}$, and 0 otherwise. The hyperparameter $\delta \in \mathbb{R}^+$ indicates the maximum value of the allowed difference between two classification probabilities. Training GACaps-HTC without this hyperparameter may lead to the model learning only to output the same class probability for every label.

The final loss term for training GACaps-HTC is a mixture of the loss terms mentioned above, acquired as follows:

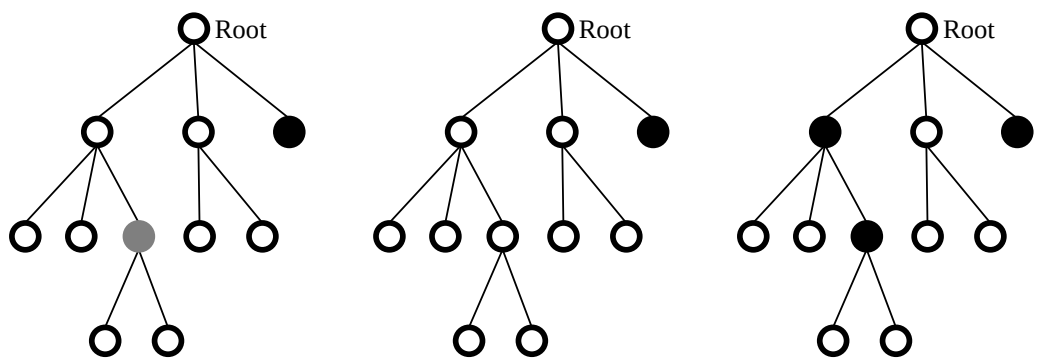
$$Loss\left(\mathcal{Y}^D, \mathcal{H}, p_1^D, \ldots, p_L^D\right) = FL\left(\mathcal{Y}^D, p_1^D, \ldots, p_L^D\right) + \lambda \times CP\left(\mathcal{H}, p_1^D, \ldots, p_L^D\right). \tag{3.13}$$

$\lambda \in \mathbb{R}^+$ is the hyperparameter indicating the weight of the contradiction penalty term.

3.2.5 Post-Processing

While the contradiction penalty term described in Subsection 3.2.4 is used to encourage classification results to coincide with the given hierarchy, the trained classi-

fier can still return results that contradict the hierarchy. Therefore, post-processing methods that add or remove labels are presented in this subsection. Three cases of classification results contradicting the hierarchy are defined in this work. The first case is an isolated label contradiction which occurs when a text document is assigned to a label but not to its parent. The classifier can choose to either do nothing, remove this isolated label, or add the labels that connect the isolated label and the root of the hierarchy. While removing the isolated label or adding the labels between the root and the isolated label guarantees classification results to match the hierarchy, these methods may lead to more false-negatives (lower recall) or false-positives (lower precision), respectively. Figure 3.6 illustrates an example of isolated label contradictions and the described post-processing methods. Filled circles in the figure denote the assigned labels, where the gray circle denotes the label causing the contradiction.



(a) Isolate Label Contradiction (b) Removing Isolated Labels (c) Connecting Isolated Labels

Figure 3.6: Illustration of an isolated label contradiction and two possible post-processing methods.

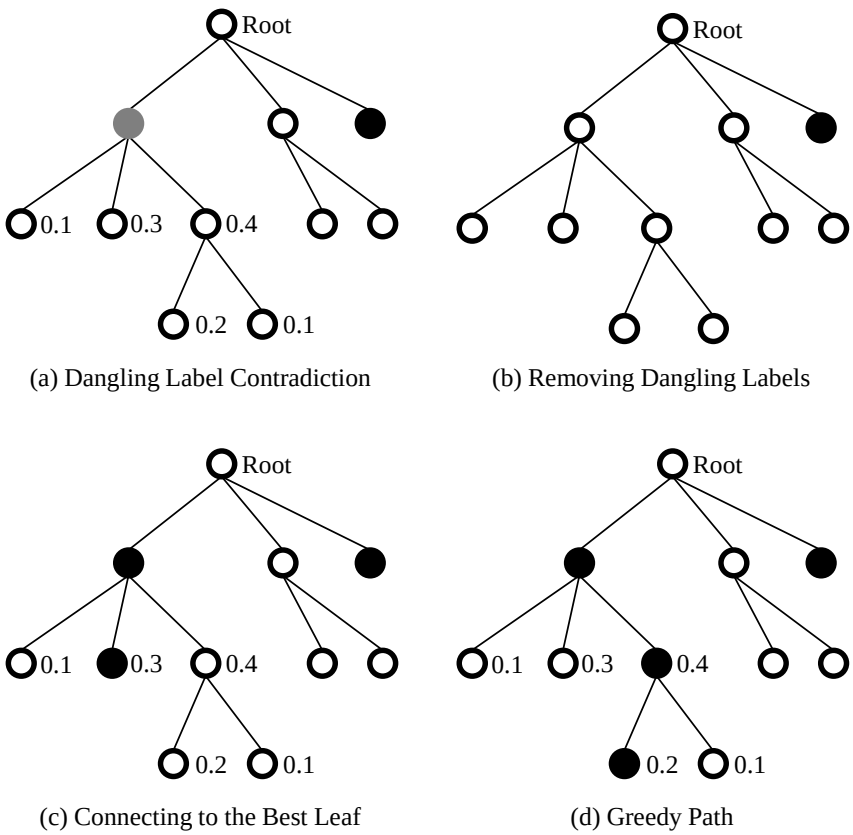
The second case of contradicting results is named a dangling label contradiction, where a text document is assigned to a label but to none of its child labels. Note that

this case may or may not contradict a hierarchy, as some datasets do not require mandatory leaf classifications, where classifying a document as at least one leaf label is necessary. However, in both datasets used in this thesis, the WOS-46985 dataset and the RCV1 dataset, each document is required to be assigned a leaf label. In this case, the classifier can choose to either do nothing or remove labels that connect the root and the dangling label. The classifier can also choose to find a descendent leaf label with the highest class probability and add labels that connect the dangling label and this descendent or to find a greedy path (choosing the child label with the highest probability) to a leaf label. Note that while some datasets meet the mandatory-leaf assumption, where the label paths are always required to end at leaf labels, others do not. Illustrations of dangling label contradiction and post-processing methods are depicted in Figure 3.7. Filled circles denote the assigned labels, where the gray circle denotes the label causing the contradiction. Numbers on the right side of labels denote corresponding class probabilities.

Finally, the third case, or an empty result contradiction, is when a document is classified as none of the labels. This case of contradiction can occur due to labels being removed by the aforementioned post-processing methods or simply because the class probability of every label is below the predefined threshold (most commonly 0.5). When an empty result contradiction happens, the classifier can choose to find a leaf label with the highest class probability and add labels that connect the root and this leaf label or find a greedy path from the root to a leaf label. These post-processing methods are equivalent to the methods shown in Figure 3.7 (c) and (d), respectively.

After GACaps-HTC is trained, validation performance derived from each combi-

nation of post-processing methods for these cases of contradictions is obtained. The combination with the best validation performance is then used for testing.



(a) Dangling Label Contradiction

(b) Removing Dangling Labels

(c) Connecting to the Best Leaf

(d) Greedy Path

Figure 3.7: Illustration of a dangling label contradiction and three possible post-processing methods.

3.3 Experiments

3.3.1 Experiment Settings

Datasets

The effectiveness of the proposed approach was validated on two benchmark HTC datasets widely employed to evaluate HTC approaches[45, 46, 47, 48, 49, 55, 56, 71, 75, 102, 114, 115]. The first benchmark dataset was the WOS-46985 dataset[1], which contains abstracts of 46,985 published papers available in Web Of Science[1]. There are 141 domain labels, including seven top-level domain labels (biochemistry, civil engineering, computer science, electrical engineering, mechanical engineering, medical science, and psychology) and 134 subdomain labels. 37,588 examples in the dataset were used for training, while 9,397 examples were used for testing.

The second dataset was the RCV1 dataset[2][32], comprising 804,414 newswire stories published by Reuters[3] from August 1996 to August 1997. In this dataset, 103 topic labels are defined, composing a hierarchy of four levels, including four top-level topic labels (corporate & industrial, economics, government & social, and markets). The RCV1 dataset was split into a training set with 23,149 examples and a testing set comprising 781,265 examples following the original work that published the dataset[32].

Metrics

The performance of GACaps-HTC and other approaches was measured and compared using two metrics widely employed to evaluate multi-label classification (including HTC) results: the micro-F1 score and macro-F1 score[131]. The micro-F1

[1]https://www.webofscience.com/
[2]https://trec.nist.gov/data/reuters/reuters.html
[3]https://www.reuters.com/

score is a conventional metric used to evaluate classification results by assigning equal weight to every example. In contrast, the macro-F1 score assigns equal weight to every label. Therefore, an approach returning biased results that prefer labels with many examples may achieve a high micro-F1 score while recording a low macro-F1 score.

Let TP_l, FP_l, FN_l denote the number of true-positive, false-positive, and false-negative examples of the l-th label. Micro-precision, micro-recall, and micro-F1 scores, denoted as $Micro\text{-}P$, $Micro\text{-}R$, and $Micro\text{-}F1$, are calculated as follows:

$$\begin{aligned} Micro\text{-}P &= \frac{\sum_{1 \le l \le L} TP_l}{\sum_{1 \le l \le L} TP_l + FP_l} \\ Micro\text{-}R &= \frac{\sum_{1 \le l \le L} TP_l}{\sum_{1 \le l \le L} TP_l + FN_l} \\ Micro\text{-}F1 &= \frac{2 \times Micro\text{-}P \times Micro\text{-}R}{Micro\text{-}P + Micro\text{-}R} \end{aligned} \tag{3.14}$$

Precision and recall scores corresponding to the l-th label are denoted as $Class\text{-}P_l$ and $Class\text{-}R_l$, respectively, and the macro-F1 score, denoted as $Macro\text{-}F1$, is calculated as follows:

$$\begin{aligned} Class\text{-}P_l &= \frac{TP_l}{TP_l + FP_l} \\ Class\text{-}R_l &= \frac{TP_l}{TP_l + FN_l} \\ Macro\text{-}F1 &= \frac{1}{L} \sum_{1 \le l \le L} \frac{2 \times Class\text{-}P_l \times Class\text{-}R_l}{Class\text{-}P_l + Class\text{-}R_l} \end{aligned} \tag{3.15}$$

Baselines

The GACaps-HTC was compared with the following baseline approaches:

- CapsNet-based flat approach:

  - Zhao *et al.*[98]: an approach using a N-gram CNN and two CapsNets

- Other flat approaches

  - Lai *et al.*[101]: approach with bidirectional recurrent neural network (RNN)[132] and a CNN
  - Chen *et al.*[39]: a hierarchical long short-term memory (LSTM)[133]-based approach
  - Yang *et al.*[40]: an approach based on a hierarchical attention network which utilizes word-level attention and sentence-level attention
  - Zhou *et al.*[41]: an attention-based bidirectional LSTM-based approach
  - Liu *et al.*[69]: a CNN-based approach designed for extreme multi-label text classification
  - Chatterjee *et al.*[56]: a CNN-based approach that maps label embeddings onto a hyperbolic space

- Mixed local and global approaches

  - Wehrmann *et al.*[76]: an approach that defines a global classifier and hierarchy level-wise local classifiers
  - Huang *et al.*[77]: an approach using a global classifier and attention-based level-wise local classifiers

- Other local approaches

  - Shimura *et al.*[75]: a CNN-based approach using transfer learning[134]

  - Banerjee *et al.*[71]: a gated recurrent unit (GRU)[135]-based approach using transfer learning and attention mechanism

- GNN-based global approaches
  - Zhou *et al.*[45]: a hierarchy-GCN based approach that extracts label-specific representations
  - Deng *et al.*[47]: a hierarchy-GCN based approach that aims to maximize mutual information between documents and labels

- CapsNet-based global approaches
  - Aly *et al.*[99]: a CapsNet-based approach exploiting hierarchy for parameter initialization
  - Peng *et al.*[100]: a CapsNet-based approach using a CNN, RNN, and word-level GNN for textual representation extraction

- Other global approach
  - Mao *et al.*[55]: a reinforcement learning-based approach that learns hierarchy traversing policy

Implementation Details

For the RCV1 dataset, the pretrained bidirectional encoder representations from Transformers (BERT)[136] was employed as the language model in the textual representation extractor. The BERT is a language model trained on a massive English corpus that has demonstrated success when adopted in downstream NLP

tasks[137, 138]. For the WOS-46985 dataset, BERT pretrained on scientific documents, namely SciBERT[139], was utilized as the language model since the domains used for pretraining the language model and those of the dataset align.

Textual representations with $d_{LM} = 768$ were obtained from the language models mentioned above, and representations of $d_{Conv} = L \times 100$, $d_{HE} = 100$, and $d_{Caps} = 32$ were extracted by GACaps-HTC. The number of labels L was 141 for the WOS-46985 dataset and 103 for the RCV1 dataset. The capsule pruning ratio was set to $\rho = 0.05$. For the capsule dropout rate and the hyperparameter in the contradiction penalty term, $\phi = 0.15$ and $\delta = 0.01$ were selected, respectively, for both datasets, while for the weight of the contradiction penalty term, $\lambda = 0.0005$ and $\lambda = 0.001$ were used for the WOS-46985 dataset and the RCV1 dataset, respectively. Each hyperparameter mentioned above was selected by a coarse hyperparameter search. The convergence threshold of the dynamic routing algorithm and the hyperparameter in the focal loss were set to $\epsilon = 0.05$ and $\gamma = 2$, following Zhao *et al.*[64] and Lin *et al.*[130], respectively.

The GACaps-HTC was trained using an Adam optimizer[140] with mini-batches of size 32. An initial learning rate of 0.0001 was used for the WOS-46985 dataset while a learning rate of 0.00005 was used to train the model using the RCV1 dataset. The learning rate was decayed by a factor of 0.1 if suboptimal validation micro-F1 scores were obtained for five consecutive epochs. We stopped the training of GACaps-HTC after the learning rate was decayed for the fourth time.

Approach	Micro-F1	Macro-F1
Lai *et al.*[101][a]	0.688	0.478
Chen *et al.*[39][a]	0.738	0.543
Yang *et al.*[40][a]	0.750	0.557
Zhou *et al.*[41][a]	0.744	0.551
Liu *et al.*[69][a]	0.706	0.503
Zhao *et al.*[98][a]	0.788	0.632
Aly *et al.*[99][a]	0.769	0.614
Huang *et al.*[77][a]	0.807	0.699
Peng *et al.*[100][a]	0.846	0.723
Zhou *et al.*[45]	0.858	0.803
Deng *et al.*[47]	0.856	0.801
Ours (GACaps-HTC)	**0.876**	**0.829**

The best results are highlighted in **bold.**

[a] F1 scores reported by Wang *et al.*[102]

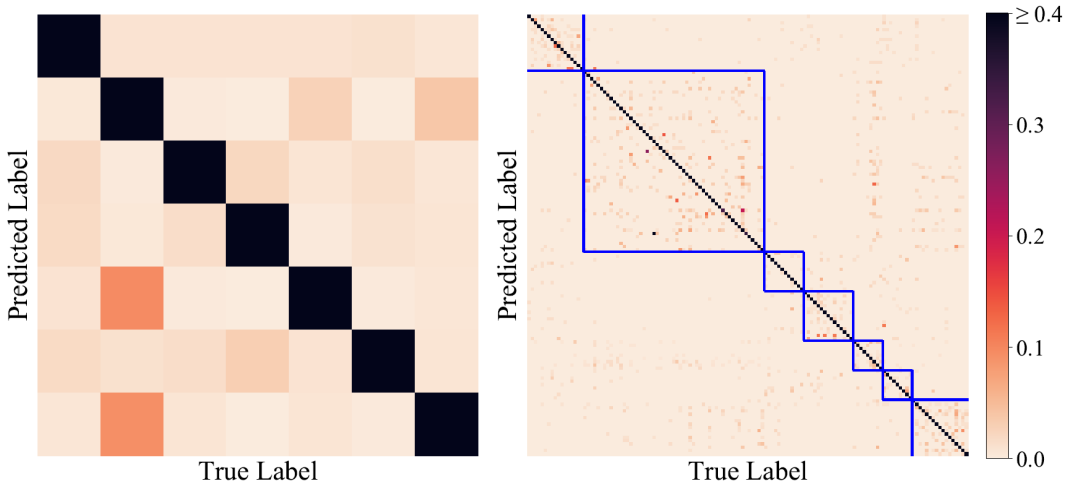
Table 3.1: Experiment results on the WOS-46985 dataset.

3.3.2 Results

Performance on the WOS-46985 Dataset

The experimental results on the WOS-46985 dataset are listed in Table 3.1. The GACaps-HTC achieved the best micro-F1 and macro-F1 scores with a 2.1% and a 3.2% gain compared to the second-best approach, Zhou *et al.*[45], respectively. For further analysis of the classification results, confusion matrices that compared the ground-truth labels and predicted labels with the highest classification probabilities by GACaps-HTC are depicted in Figure 3.8. Note that all cells with values higher than 0.4 were filled with the same color to enhance the contrast in cells with low values. Dark diagonal cells in each confusion matrix are true-positive cases where the proposed approach correctly classified the documents. In Figure 3.8 (a), top-level domain labels in the WOS-46985 dataset are compared. Although the proposed ap-

proach successfully classified top-level domain labels in most cases, it categorized a portion of documents of the second label (medical sciences) as the fifth (biochemistry) or seventh label (psychology). This error is presumed to be due to the numerous similarities between their child labels. For example, documents on immune system-related illnesses, including lymphoma and acquired immune deficiency syndrome, and mental health are labeled as medical sciences, whereas documents on immunology and depression are categorized as biochemistry and psychology, respectively.



(a) Level 1 (b) Level 2

Figure 3.8: Level-wise confusion matrix on the WOS-46985 dataset.

In Figure 3.8 (b), labels in the second hierarchy level in the WOS-46985 dataset are compared. Cells surrounded by a blue square denote the confusion between labels that share a parent label and, therefore, are semantically similar. These cells are mostly light-colored as the proposed approach successfully distinguished one label from another label with a similar meaning.

Performance on the RCV1 Dataset

Approach	Micro-F1	Macro-F1
Lai *et al.*[101][b]	0.686	0.293
Chen *et al.*[39][b]	0.673	0.310
Yang *et al.*[40][b]	0.696	0.327
Zhou *et al.*[41][a]	0.670	0.315
Liu *et al.*[69][b]	0.695	0.301
Shimura *et al.*[75]	0.803	0.514
Wehrmann *et al.*[76][c]	0.808	0.546
Zhao *et al.*[98][b]	0.739	0.399
Aly *et al.*[99][a]	0.710	0.339
Banerjee *et al.*[71]	0.805	0.585
Huang *et al.*[77][a]	0.833	0.601
Mao *et al.*[55]	0.833	0.601
Peng *et al.*[100]	0.778	0.513
Zhou *et al.*[45]	0.840	0.634
Chatterjee[56]	0.793	0.473
Deng *et al.*[47]	0.835	0.627
Ours (GACaps-HTC)	**0.868**	**0.698**

The best results are highlighted in **bold.**

[a] F1 scores reported by Wang *et al.*[102]

[b] F1 scores reported by Peng *et al.*[100]

[c] F1 scores reported by Mao *et al.*[55]

Table 3.2: Experiment results on the RCV1 dataset.

The experimental results on the RCV1 dataset are listed in Table 3.2. The GACaps-HTC achieved the best micro-F1 and macro-F1 scores with a 3.3% gain and a 10.1% gain compared to the second-best approach, Zhou *et al.*[45], respectively. This indicates that the proposed approach could improve the overall classification performance while successfully categorizing sparse (fine-grained) labels.

Such improvements were due to the proposed approach's ability to understand the label hierarchy using the GAT and the latent label relationships with the Cap-

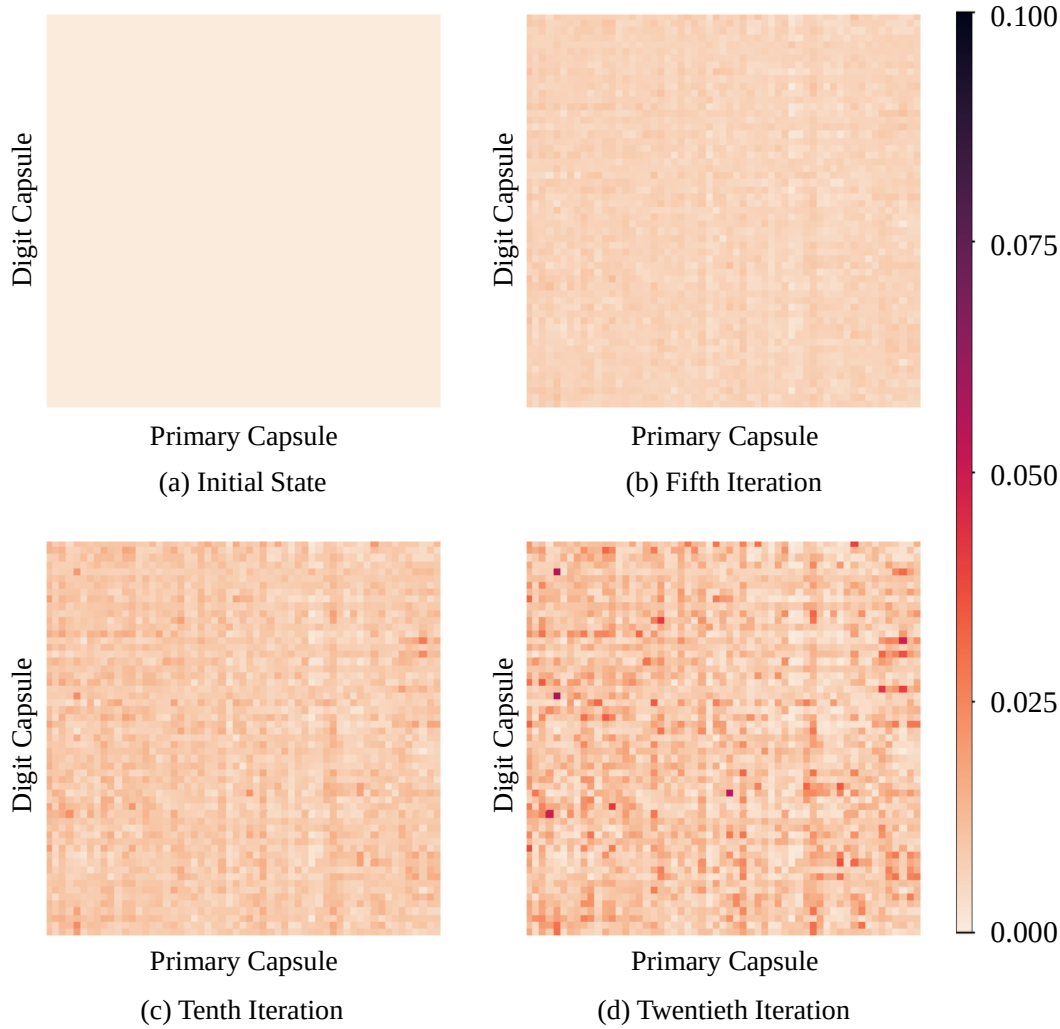


Figure 3.9: Visualization of normalized coupling coefficients in the capsule network on the RCV1 dataset.

sNet. The latent relationships between the labels on the second level of the RCV1 dataset's label hierarchy learned by the CapsNet are illustrated in Figure 3.9. Note that only the labels on the second level of the label hierarchy are depicted, and therefore there were no explicit relationships provided by the label hierarchy between the illustrated labels. As shown in Figure 3.9, the dynamic routing algorithm distinguished salient relationships more clearly as the number of iterations increased. Some examples of the relationships learned by the CapsNet were as follows: The CapsNet captured a correlation between the economic performance label and the economic output and capacity label, which is intuitive as economic output and capacity impact economic performance directly. Furthermore, the CapsNet derived that the commodity market label is correlated with the corporate contracts and orders label, corporate management label, and the corporate production label, where these correlations make sense as a fluctuation in the commodity market impacts corporate contracts, management, and production. The network also inferred that the economic inflation label and the corporate-related regulation label are correlated, where the fastest way to stabilize inflation is by attracting foreign and internal investments by relaxing corporate regulations[141]. Intuitive relationships like these indicate that GACaps-HTC successfully understood the latent relationships between labels. Also, such interpretability of the implicit label relationships extracted by the model can enhance a user's confidence in the model[142].

3.3.3 Ablation Studies

This subsection demonstrates the results of ablation studies performed with the WOS-46985 dataset.

Ablation Studies on Capsule Pruning and Attention

The effectiveness of the capsule pruning and attention mechanism adopted in the CapsNet is presented in Table 3.3. The capsule pruning and attention mechanism led to increases in the micro-F1 and the macro-F1 scores. These results imply that enhanced generalizability[96] and representation power[97] obtained by employing the capsule pruning and attention mechanism, respectively, enabled GACaps-HTC to achieve high F1 scores. On average, approaches with pruning achieved a 0.4% and a 0.6% increase in the micro-F1 and the macro-F1 scores, respectively, compared to those without pruning. Employing the attention mechanism in the CapsNet led to a 0.9% gain in the micro-F1 score and a 1.6% gain in the macro-F1 score, respectively, on average.

Pruning	Attention	Micro-F1	Macro-F1
-	-	0.865	0.811
✓	-	0.871	0.816
-	✓	0.875	0.824
✓	✓	**0.876**	**0.829**

The best results are highlighted in **bold.**

Table 3.3: Ablation study results regarding capsule pruning and attention on the WOS-46985 dataset.

Ablation Studies on Loss Terms

The F1 scores recorded by training GACaps-HTC with various loss terms are listed in Table 3.4, where the results in the first and the third rows were achieved by replacing the focal loss with the BCE loss. Training the network using the focal loss increased both the micro-F1 and macro-F1 scores, indicating that letting the network

focus on labels harder to categorize than others led to improved performance. On the other hand, employing the contradiction penalty term increased the macro-F1 score while maintaining the same level of the micro-F1 score. This indicates that the information on the label hierarchy provided by this additional loss term played a critical role in categorizing fine-grained labels with few examples.

FL	*CP*	Micro-F1	Macro-F1
-	-	0.873	0.819
✓	-	0.875	0.819
-	✓	0.873	0.824
✓	✓	**0.876**	**0.829**

The best results are highlighted in **bold.**

Table 3.4: Ablation study results regarding loss terms on the WOS-46985 dataset.

Ablation Studies on Hierarchy Encoder

Table 3.5 lists the results obtained by replacing the GAT in the hierarchy encoder of GACaps-HTC with other GNNs. Although employing any GNN led to improved performance in the micro-F1 and macro-F1 scores compared to the approach without a GNN, the proposed approach with GAT outperformed other approaches. In addition, employing the GAT was efficient as the number of parameters in the GAT ($\sim$10k) was similar to that of the GCN ($\sim$10k), whereas the hierarchy-GCN, which achieved the second-best performance, required more parameters ($\sim$29k).

Ablation Studies on Implicit Relationship Extractor

The effectiveness of the CapsNet in the implicit relationship extractor of GACaps-HTC is displayed in Table 3.6. The results recorded in the first row were obtained by replacing the CapsNet with a fully-connected layer, abbreviated as FC in Table

Graph Neural Network	Micro-F1	Macro-F1
None	0.864	0.800
GCN[81]	0.868	0.812
Hierarchy-GCN[45]	0.873	0.820
GAT[92]	**0.876**	**0.829**

The best results are highlighted in **bold.**

Table 3.5: Ablation study results regarding graph neural networks on the WOS-46985 dataset.

3.6, which returned the classification probability of each label using every label-specific representation from the hierarchy encoder. The F1 scores in the second row were achieved by a neural network that employed a convolutional layer instead of the CapsNet. Finally, the third row records the performance obtained by utilizing a fully-connected layer for each label that returned a probability from the corresponding label-specific representation. The results show that the proposed approach with the CapsNet was able to achieve the best performance due to its ability to capture latent relationships between labels via the dynamic routing algorithm.

Table 3.6: Ablation study results regarding capsule network on the WOS-46985 dataset.

Employed Subnetwork	Micro-F1	Macro-F1
FC	0.869	0.814
Convolutional	0.869	0.827
FC per label	0.871	0.819
CapsNet	**0.876**	**0.829**

The best results are highlighted in **bold.**

To further illustrate how the GAT and the CapsNet of the proposed approach enhanced the classification performance, additional ablation experiments were con-

Table 3.7: Ablation study results regarding the graph attention network and capsule network on the RCV1 dataset.

Approach	Metric	Level 1	Level 2	Level 3	Level 4	Overall
GACaps-HTC	Micro-F1	0.939	0.821	0.851	0.780	0.868
	Macro-F1	0.929	0.708	0.657	0.780	0.698
Without GAT	Micro-F1	0.853 (-9.240%)	0.743 (-9.492%)	0.769 (-9.624%)	0.779 (-0.146%)	0.788 (-9.192%)
	Macro-F1	0.829 (-10.737%)	0.620 (-12.414%)	0.563 (-14.356%)	0.779 (-0.146%)	0.606 (-13.222%)
Without CapsNet	Micro-F1	0.923 (-1.719%)	0.811 (-1.133%)	0.841 (-1.197%)	0.779 (-0.146%)	0.858 (-1.151%)
	Macro-F1	0.911 (-1.956%)	0.687 (-2.954%)	0.648 (-1.393%)	0.779 (-0.146%)	0.680 (-2.579%)

ducted on the RCV1 dataset. In these experiments, F1 scores obtained from labels in each hierarchy level were measured to compare the roles of the GAT and the CapsNet, where a level of a label denotes the distance from the root to the label in the label hierarchy. Results are presented in Table 3.7. The GACaps-HTC without a GAT (second row of the Approach column) is a local approach that did not utilize the label hierarchy while modeling implicit label relationships using a CapsNet. The GACaps-HTC without a CapsNet (third row of the Approach column) exploited the label hierarchy while ignoring implicit label relationships as the CapsNet of the proposed approach was replaced with a fully-connected layer. Gains and losses in F1 scores compared to GACaps-HTC are presented in parentheses.

While GACaps-HTC without either the GAT or the CapsNet achieved degraded performance, removing the GAT led to steeper decreases in F1 scores, implying that

explicit relationships between labels presented as a label hierarchy played a more significant role in HTC than implicit label relationships. The difference between the roles of the hierarchy encoder and the implicit relationship extractor can be deduced from further level-wise analysis. Table 3.7 shows that information on the label hierarchy extracted by the GAT provided more help in classifying labels further from the root than in classifying those closer to the root. Note that only one label was present in the fourth level; therefore, the results on the fourth level do not provide much insight. On the other hand, no such level-wise tendencies could be deduced from comparing GACaps-HTC and GACaps-HTC without a CapsNet, as the CapsNet in the proposed approach models information on implicit relationships between labels that may be unrelated to the label hierarchy.

Ablation Studies on Capsule Dropout

Finally, the effect of capsule dropout is shown in Figure 3.10 and Table 3.8. As depicted in Figure 3.10, higher dropout rates led to slower learning as more primary capsules were dropped and less information was utilized in the dynamic routing algorithm. However, the enhanced robustness from the capsule dropout led to improved performance, as shown in Table 3.8. The results also show that high dropout rates could cause suboptimal classification performance due to the CapsNet losing too much information.

Table 3.8: Results obtained with different capsule dropout rates on the WOS-46985 dataset.

Dropout Rate (ϕ)	Micro-F1	Macro-F1
0	0.875	0.819
0.15	**0.876**	**0.829**
0.30	0.872	0.819

The best results are highlighted in **bold.**

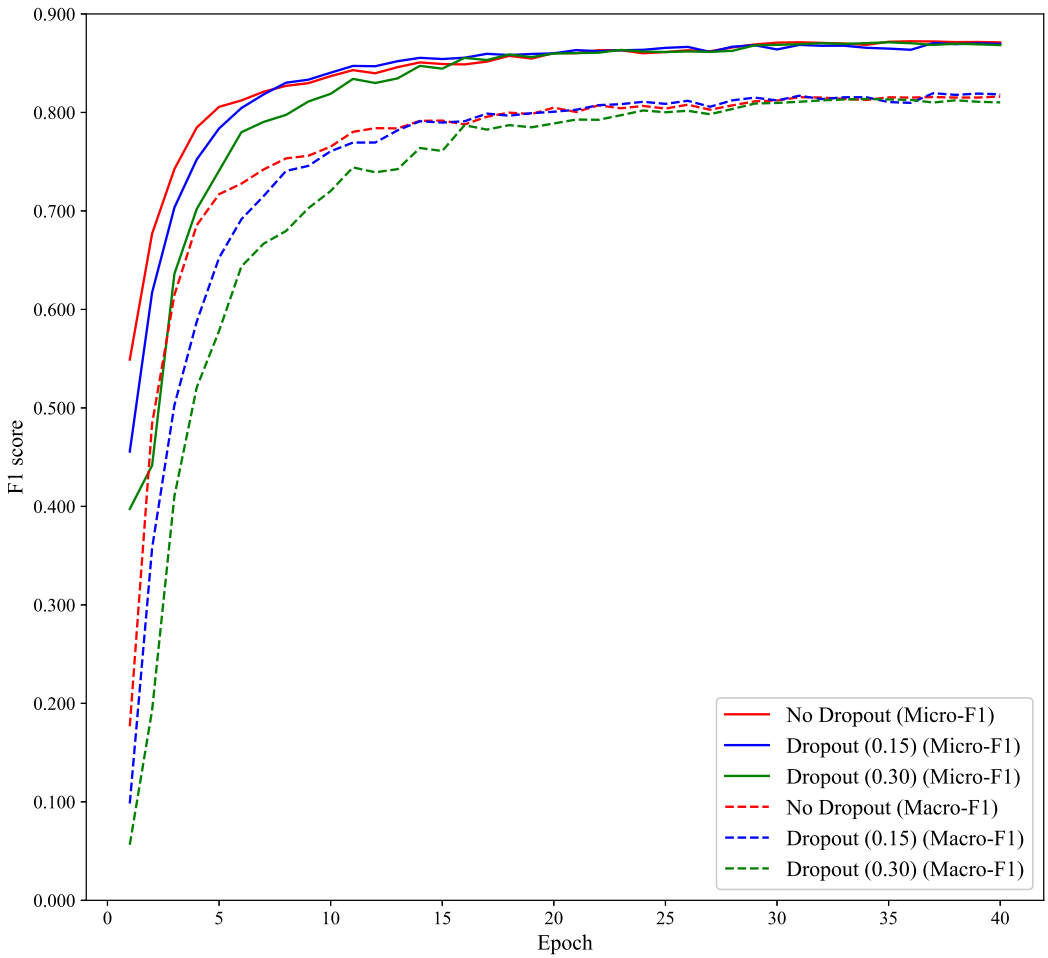


Figure 3.10: Validation F1 score plot obtained by training GACaps-HTC with different capsule dropout rates on the WOS-46985 dataset.

Chapter 4

Incorporating Label Semantics for Hierarchical Text Classification

4.1 Problem Definition

As introduced in Section 2.4, label semantics have been shown to contain valuable information for classification, and several approaches attempted to incorporate this information in HTC[46, 49, 102, 114, 115]. Motivated by these approaches, the approach described in Chapter 3, GACaps-HTC, is introduced with label semantics in this section. Like the previous chapter, this chapter tackles hierarchical text classification. Therefore, notations of an input text document D, the number of labels L, and a corresponding set of ground-truth labels $\mathcal{Y}^D$ follow that of Section 3.1. Notations of classification probabilities p_l^D corresponding to the l-th label and the label hierarchy $\mathcal{H}$ also follow the notations from Section 3.1.

Representations of label semantics are extracted from textual descriptions of labels. A textual description of a label may be a sentence or a paragraph summarizing the meaning of the label or even a simple phrase like the label's name. In the following sections, Δ_l denotes the textual description of the l-th label.

4.2 Methods

A semantic-aware dynamic routing algorithm is proposed to incorporate label semantics in the CapsNet of the implicit relationship extractor. The proposed algorithm incorporates semantic information on labels into the model in the following two ways: First, each activation vector propagated from a primary capsule to a digit capsule is added with a vector obtained from semantic representations of labels corresponding to the capsules. This added vector is named semantic bias as it acts as a bias term for the propagated activation vector. Second, an initial coupling coefficient between two labels is set to the similarity of the labels' semantic representations. Since coupling coefficients represent the similarity between two capsules' activation vectors and semantics[63, 64], such an initialization can accelerate the training.

Other various attempts had been made to introduce semantic information in the textual representation extractor, hierarchy encoder, or attention mechanism of the implicit relationship extractor, only to result in suboptimal performance. Details on these attempts and their performance are discussed in Subsection 4.3.2 and Subsection 4.3.3, respectively.

4.2.1 Semantic Bias

The label semantics are injected into the dynamic routing algorithm of the implicit relationship extractor's CapsNet to allow the subnetwork to derive latent relationships between labels from not only textual representations but also their semantics. Before introducing label semantics into the CapsNet, a semantic representation of each label needs to be extracted. Let $\boldsymbol{s}_l \in \mathbb{R}^{d_{LM}}$ denote the task-agnostic semantic representation of the l-th label, obtained from encoding Δ_l using a pretrained lan-

guage model that generates vector representations of size d_{LM}. A matrix of weight parameters $\boldsymbol{W}_{Sem} \in \mathbb{R}^{d_{Sem} \times d_{LM}}$ is defined to generate task-specific label semantic representations. A task-specific semantic representation of the l-th label is denoted as $\boldsymbol{s}'_l \in \mathbb{R}^{d_{Sem}}$, and is obtained as follows:

$$\boldsymbol{s}'_l = \boldsymbol{W}_{Sem} \boldsymbol{s}_l. \tag{4.1}$$

The dynamic routing algorithm described in Algorithm 1 obtains the activation vector propagated from the l'-th primary capsule to the l-th digit capsule, denoted as $\boldsymbol{\mu}^D_{ll'}$, as follows:

$$\boldsymbol{\mu}^D_{ll'} = \boldsymbol{W}_{Caps,l} \boldsymbol{u}^D_{l'}, \tag{4.2}$$

from the activation vector of the l'-th primary capsule $\boldsymbol{u}^D_{l'}$ and a parameter matrix corresponding to the l-th digit capsule $\boldsymbol{W}_{Caps,l}$. Aforementioned semantic representations are introduced in the dynamic routing algorithm as additive biases in these propagated activation vectors. The semantic bias corresponding to the l'-th primary capsule and the l-th digit capsule is denoted as $\boldsymbol{\varsigma}_{ll'}$, and is obtained as follows:

$$\boldsymbol{\varsigma}_{ll'} = \boldsymbol{W}_{Bias} s'_l + \boldsymbol{W}'_{Bias} s'_{l'} + \mathrm{ReLU}(\boldsymbol{W}''_{Bias} |s'_l - s'_{l'}|). \tag{4.3}$$

$\boldsymbol{W}_{Bias}, \boldsymbol{W}'_{Bias}, \boldsymbol{W}''_{Bias} \in \mathbb{R}^{d_{Caps} \times d_{Sem}}$ are the parameter matrices used for transforming the semantic representations to a semantic bias. Note that the last term, $\mathrm{ReLU}(\boldsymbol{W}''_{Bias} |s'_l - s'_{l'}|)$, incorporates the semantic relationship between two labels in the semantic bias.

To effectively moderate the impact of injecting label semantics, a gating mechanism[143] is utilized in the proposed semantic-aware dynamic routing. This mechanism dynamically controls the flow of multiple channels of information to the resulting feature

representation. The gating mechanism is most popularly adopted in RNNs to adaptively model the flow of the information propagated from the past to the current state, resulting in gated recurrent neural networks[144], including LSTMs[133] and GRUs[135].

In this work, a gating mechanism similar to that of Li *et al.*[145] is employed. This mechanism obtains the valve of each element, which is the ratio of additional information (label semantics) propagated to the original information (textual representation), from both sources of information. Let $\boldsymbol{W}_{Gate,l} \in \mathbb{R}^{d_{Caps} \times d_{HE}}$ denote the matrix of parameters used to obtain the valve vector corresponding to the l-th digit capsule. Also, $\boldsymbol{W}'_{Gate}, \boldsymbol{W}''_{Gate} \in \mathbb{R}^{d_{Caps} \times d_{Sem}}$ are the matrices of parameters used for obtaining valve vectors from label semantic representations. The valve vector corresponding to the l'-th primary capsule and the l-th digit capsule is denoted as $\boldsymbol{\sigma}^D_{ll'}$ and is obtained as follows:

$$\boldsymbol{\sigma}^D_{ll'} = \text{Sigmoid}\left(\text{LayerNorm}\left(\boldsymbol{W}_{Gate,l}\boldsymbol{u}^D_{l'} + \boldsymbol{W}'_{Gate}s'_l + \boldsymbol{W}''_{Gate}s'_{l'}\right)\right). \tag{4.4}$$

Layer normalization[4], abbreviated as LayerNorm, computes the normalization statistics of valve vectors and performs normalization to stable classification.

Equation 4.2 in the dynamic routing algorithm is replaced with the following equation:

$$\boldsymbol{\mu}^D_{ll'} = \boldsymbol{W}_{Caps,l}\boldsymbol{u}^D_{l'} + \boldsymbol{\sigma}^D_{ll'}\boldsymbol{\varsigma}_{ll'}. \tag{4.5}$$

This process of obtaining a semantic bias and a valve vector is depicted in Figure 4.1.

4.2.2 Coupling Coefficient Initialization

According to Sabour *et al.*[63] and Zhao *et al.*[64], a coupling coefficient corresponding to a pair of a primary capsule and a digit capsule is a measurement of agreement (similarity) between their activation vectors. Given two semantically similar labels, their corresponding capsules should capture similar characteristics, resulting in similar activation vectors with a high degree of agreement. In order to leverage this assumption, a label semantic-based coupling coefficient initialization method is proposed.

While the dynamic routing algorithm described in Algorithm 1 and that proposed by Sabour *et al.*[63] initialize every coupling coefficient to zero, the proposed method initializes each coefficient to the similarity of the corresponding labels. In this work, dot products of task-agnostic label semantic representations ($\boldsymbol{s}_l$) are used for initializing coupling coefficients. A coupling coefficient $c^D_{ll'}$ defined for the l-th digit capsule and the l'-th primary capsule is initialized as follows:

$$c^D_{ll'} = (\boldsymbol{s}_l + \boldsymbol{b}_{Sem,l}) \cdot (\boldsymbol{s}_{l'} + \boldsymbol{b}_{Sem,l'})/d_{LM}. \tag{4.6}$$

$\boldsymbol{b}_{Sem,l} \in \mathbb{R}^{d_{Lm}}$ is a vector of trainable parameters assigned to the semantic representation of the l-th label. These vectors allow initial coefficients to be trainable via backpropagation. Ramasinghe *et al.*[146] demonstrates that utilizing trainable initial coupling coefficients leads to faster routing convergence and better classification performance due to the fact that the attributes captured by primary capsules and those corresponding to digit capsules are dependent on each other[146, 147]. $\boldsymbol{b}_{Sem,l}$ is initialized following LeCun initialization[148], which is known to help faster convergence when trained by backpropagation. Note that the dot product is divided by

d_{LM} to scale down the influence of label semantics and prevent the dynamic routing algorithm from converging solely dependent on the label semantics.

4.2.3 Semantic-Aware Dynamic Routing Algorithm

Introducing semantic biases and the proposed coupling coefficient initialization method in a dynamic routing algorithm leads to the proposed semantic-aware dynamic routing algorithm described in Algorithm 2. While the proposed algorithm is similar to the algorithm proposed by Zhao *et al.*[64], note that line 1 of Algorithm 1 is replaced with Equation 4.6 in line 1 of Algorithm 2 for semantic-based coefficient initialization. The process of obtaining semantic biases and valve vectors is presented in lines 2 to 4 in Algorithm 2. Finally, line 7 of Algorithm 1, which calculated activation vectors propagated from primary capsules to digit capsules, is replaced with Equation 4.5, as shown in line 10 of Algorithm 2.

Algorithm 2: Semantic-Aware Dynamic Routing Algorithm

Inputs: activation vectors of primary capsules $\boldsymbol{u}_l^D$ for $1 \leq l \leq L$
task-agnostic label semantic representations $\boldsymbol{s}_l$ for $1 \leq l \leq L$

Output: activation vectors of digit capsules $\boldsymbol{o}_l^D$ for $1 \leq l \leq L$

$c_{ll'}^D \leftarrow (\boldsymbol{s}_l + \boldsymbol{b}_{Sem,l}) \cdot (\boldsymbol{s}_{l'} + \boldsymbol{b}_{Sem,l'})/\sqrt{d_{LM}} \quad \forall l, l'$;

$\boldsymbol{s}'_l \leftarrow \boldsymbol{W}_{Sem}\boldsymbol{s}_l \quad \forall l$;

$\boldsymbol{\varsigma}_{ll'} \leftarrow \boldsymbol{W}_{Bias}\boldsymbol{s}'_l + \boldsymbol{W}'_{Bias}\boldsymbol{s}'_{l'} + \mathrm{ReLU}(\boldsymbol{W}''_{Bias}|\boldsymbol{s}'_l - \boldsymbol{s}'_{l'}|) \quad \forall l, l'$;

$\boldsymbol{\sigma}_{ll'}^D \leftarrow \mathrm{Sigmoid}\left(\mathrm{LayerNorm}\left(\boldsymbol{W}_{Gate,l}\boldsymbol{u}_{l'}^D + \boldsymbol{W}'_{Gate}\boldsymbol{s}'_l + \boldsymbol{W}''_{Gate}\boldsymbol{s}'_{l'}\right)\right) \quad \forall l, l'$;

prev_Score $\leftarrow -\infty$;

while True **do**

  for $l \leftarrow 1$ **to** L **do**

    for $l' \leftarrow 1$ **to** L **do**

      $\tilde{c}_{ll'}^D \leftarrow \frac{\exp\left(c_{ll'}^D\right)}{\sum_{1 \leq i \leq L} \exp\left(c_{il'}^D\right)}$;

      $\boldsymbol{\mu}_{ll'}^D \leftarrow \boldsymbol{W}_{Caps,l}\boldsymbol{u}_{l'}^D + \boldsymbol{\sigma}_{ll'}^D \boldsymbol{\varsigma}_{ll'}$;

      $\boldsymbol{o}_l^D \leftarrow \sum_{1 \leq l' \leq L} \tilde{c}_{ll'}^D \boldsymbol{\mu}_{ll'}^D$;

      $\mathrm{dist}\left(\boldsymbol{o}_l^D, \boldsymbol{\mu}_{ll'}^D\right) \leftarrow \left\| \frac{\|\boldsymbol{\mu}_{ll'}^D\|}{0.5 + \|\boldsymbol{\mu}_{ll'}^D\|^2} \boldsymbol{\mu}_{ll'}^D - \frac{\|\boldsymbol{o}_l^D\|}{0.5 + \|\boldsymbol{o}_l^D\|^2} \boldsymbol{o}_l^D \right\|$;

      $c_{ll'}^D \leftarrow c_{ll'}^D + \left(1 - \mathrm{dist}\left(\boldsymbol{o}_l^D, \boldsymbol{\mu}_{ll'}^D\right)\right)$;

    end

  end

  Score $\leftarrow \log\left(\sum_{1 \leq l, l' \leq L} c_{ll'}^D \left(1 - \mathrm{dist}\left(\boldsymbol{o}_l^D, \boldsymbol{\mu}_{ll'}^D\right)\right)\right)$;

  if $|\text{Score} - \text{prev\_Score}| \leq \epsilon$ **then**

    $\boldsymbol{o}_l^D \leftarrow \frac{\|\boldsymbol{o}_l^D\|}{0.5 + \|\boldsymbol{o}_l^D\|^2} \boldsymbol{o}_l^D \quad \forall l$;

    return $\boldsymbol{o}_l^D \quad \forall l$;

  end

  prev_Score $\leftarrow$ Score ;

end

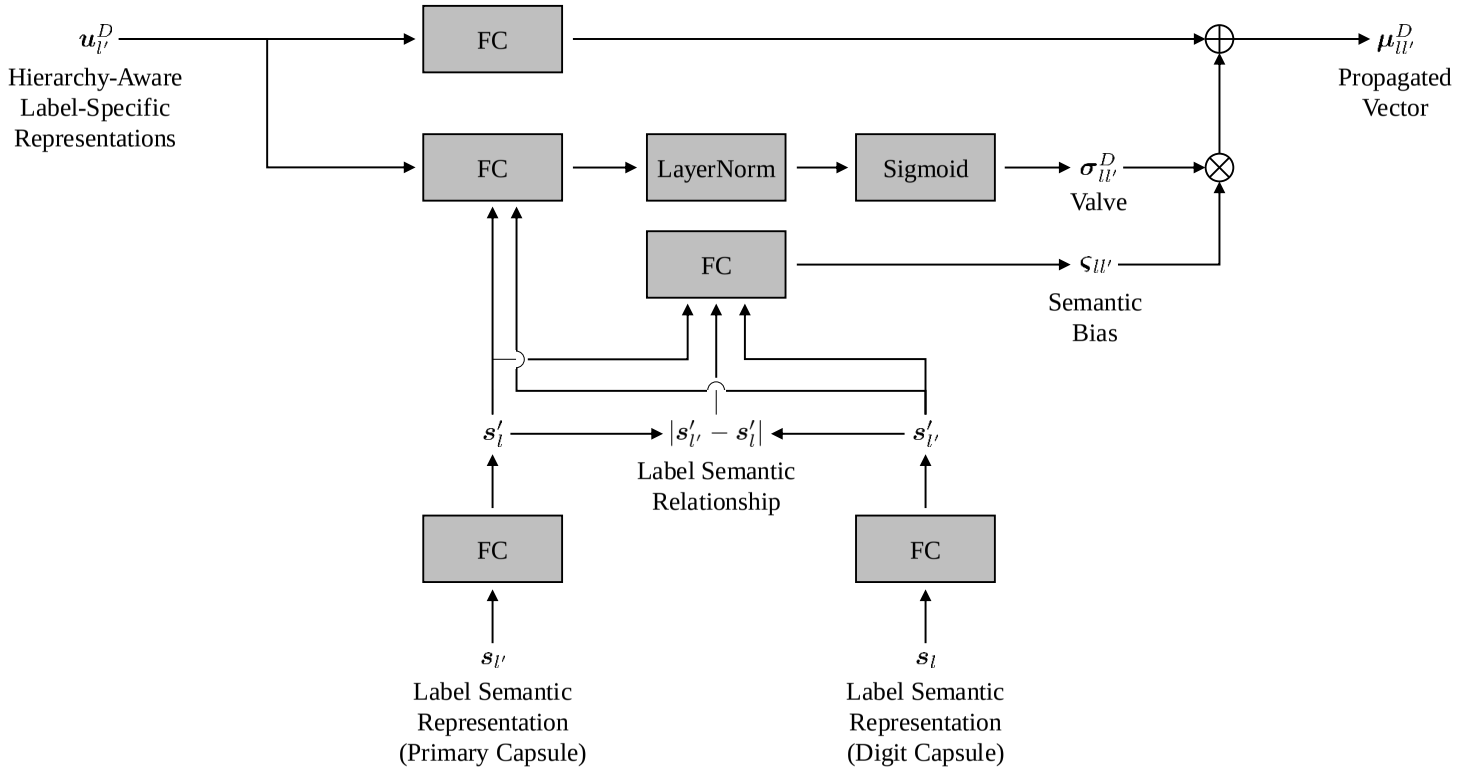


Figure 4.1: Illustration of the process of incorporating semantic information into propagated activation vector of dynamic routing algorithm. LayerNorm and FC stand for layer normalization and a fully-connected layer, respectively.

4.3 Experiments

4.3.1 Experiment Settings

For a fair comparison between the performance presented in Subsection 3.3.2 and the performance of GACaps-HTC with semantic-aware dynamic routing algorithm, the proposed approach was trained and evaluated using the WOS-46985 dataset and the RCV1 dataset described in Subsection 3.3.1 Also, the micro-F1 score and macro-F1 score were used as evaluation metrics for the following experiments. Implementation details of GACaps-HTC remained the same as described in Subsection 3.3.1.

When training the proposed approach using the WOS-46985 dataset, task-agnostic label semantic representations ($\boldsymbol{s}_l$) were extracted using SciBERT. As label names were fed as phrases, the approach was trained and evaluated using semantic representations obtained from other language models that had been shown to be effective in generating phrase-level or document-level representations, but the approach trained with representations from SciBERT achieved the best performance. These language models included PhraseBERT[149], which is BERT specialized in generating phrase-level embeddings, SPECTER[150] (short for scientific paper embeddings using citation-informed Transformers) and ASPIRE[151] (short for aspectual scientific paper relations), which generate document-level embeddings on scientific documents.

As for the RCV1 dataset, label semantic representations were extracted using a pretrained BERT. This language model was compared with other language models, including PhraseBERT, text-to-text transfer Transformer (T5)[152], and MPNet[153] (short for masked and permuted pretraining network), which have

shown state-of-the-art performance when employed for various downstream NLP tasks. Task-specific label semantic representations of $d_{Sem} = 200$ and $d_{Sem} = 250$ were used for the WOS-46985 dataset and the RCV1 dataset, respectively.

4.3.2 Compared Approaches

In this subsection, several approaches that incorporate label semantics into various subnetworks of GACaps-HTC are described. Note that semantic representations ($\boldsymbol{s}'_l$) are obtained in the same way as described in Subsection 4.2.1. An overview of the approaches compared with the proposed approach with a semantic-aware dynamic routing algorithm is illustrated in Figure 4.2, where abbreviations in the figure are described in this subsection. Note that the following approaches use the dynamic routing algorithm proposed by Zhao *et al.*[64] described in Algorithm 1.

Introducing Label Semantics in Textual Representation Extractor

The first approach, abbreviated as in-TRE, modifies the textual representation extractor of GACaps-HTC and incorporates semantic information into the hierarchy-aware textual representations ($\boldsymbol{z}_l^D$). By doing so, semantic information on labels is expected to help the model understand the label hierarchy in the hierarchy encoder. Furthermore, as the semantic information is embedded in the hierarchy encoder's input, it is propagated by the GAT and is also embedded in the output of the hierarchy encoder. Therefore, the semantic information may also be able to aid the model in capturing latent relationships between labels.

In this approach, the semantic representations are used as an additive bias in the textual representation extractor. Equation 3.3 is modified as follows:

$$\boldsymbol{z}_l^D = \boldsymbol{W}_{Aff}\text{MaxPool}\left(\left(\text{Conv}\left(\boldsymbol{X}^D\right)_{[:,(l-1)d_{HE}:ld_{HE}]}\right) + \boldsymbol{s}'_l\right). \tag{4.7}$$

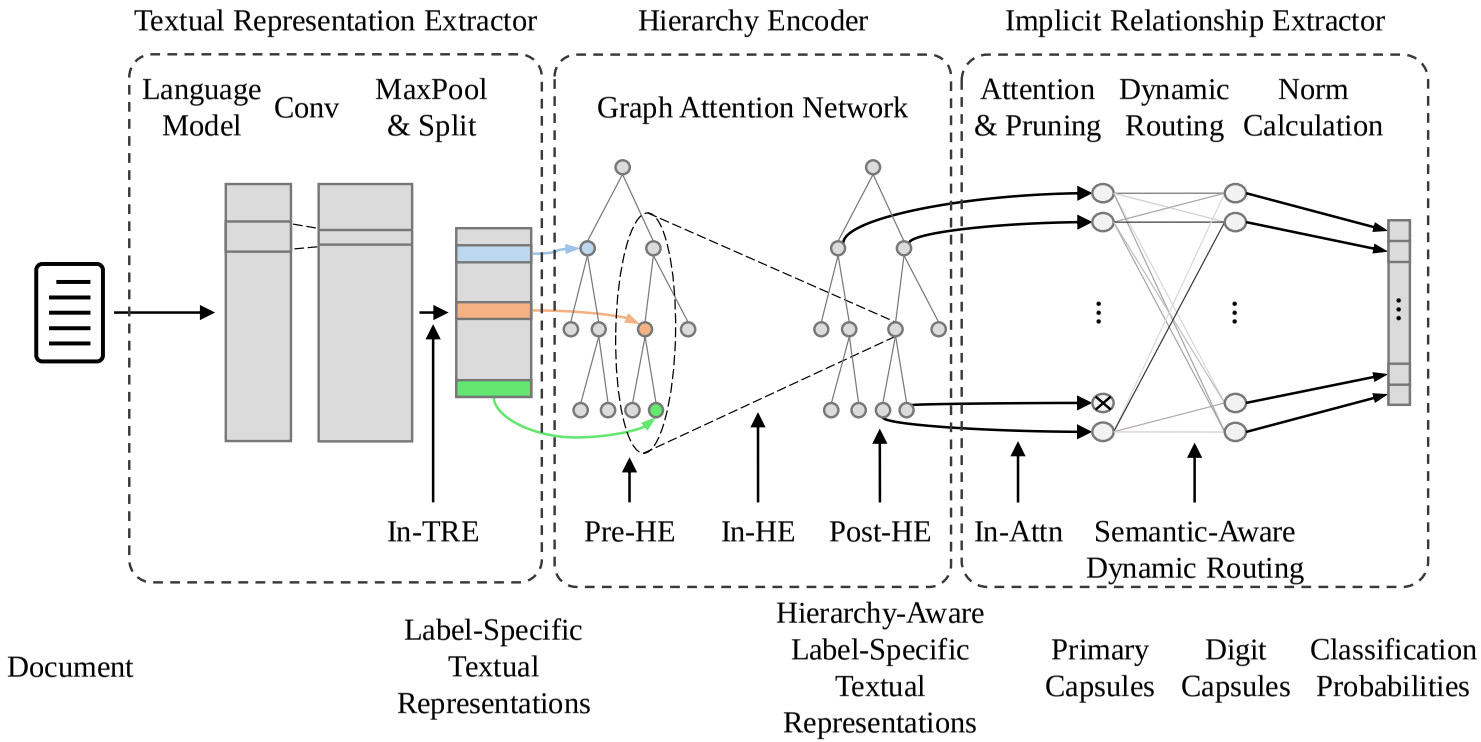


Figure 4.2: Overview of the compared approaches that incorporate label semantics into a graph attention capsule network for hierarchical text classification.

In this approach, $d_{Sem} = d_{HE}$ to match the sizes of added vectors.

Introducing Label Semantics before Hierarchy Encoder

The second approach, abbreviated as pre-HE, is similar to the in-TRE approach as they both merge semantic information with textual representations before the representations are passed to the hierarchy encoder. The difference between the first and the second approach is that while the first approach utilized semantic representations as additive biases, the second approach concatenates the semantic representations with the textual representations. Therefore, compared to the in-TRE approach, the pre-HE approach utilizes merged representations of a document and label semantics that are disentangled. In this approach, Equation 3.3 is replaced with the following equation:

$$\boldsymbol{z}_l^D = \boldsymbol{W}_{Aff}\text{Concat}\left(\text{MaxPool}\left(\text{Conv}\left(\boldsymbol{X}^D\right)_{[:,(l-1)d_{HE}:ld_{HE}]}\right), \boldsymbol{s}'_l\right). \tag{4.8}$$

In this approach, the size of the weight parameters' matrix, $\boldsymbol{W}_{Aff}$, is changed from $d_{HE} \times d_{HE}$ to $d_{HE} \times (d_{HE} + d_{Sem})$.

Introducing Label Semantics in Hierarchy Encoder

This approach, abbreviated as in-HE, injects label semantic information directly into the hierarchy encoder's weight calculation process described in Equation 3.4. Therefore, the hierarchy encoder can infer the importance of different neighboring labels when generating a hierarchy-aware label representation based on their semantic relationships (similarities). However, as the propagated representations themselves do not contain information on label semantics, this information is not provided to the implicit relationship extractor to help understand how labels are implicitly corre-

lated.

Equation 3.4 is replaced with the following equation:

$$w_{ll'}^{D} = \text{LeakyReLU}\left(\boldsymbol{W}_{HE}\text{Concat}\left(\boldsymbol{z}_{l}^{D}, \boldsymbol{z}_{l'}^{D}, \boldsymbol{s}_{l}', \boldsymbol{s}_{l'}'\right)\right). \tag{4.9}$$

The weight parameter matrix $\boldsymbol{W}_{HE} \in \mathbb{R}^{1\times 2(d_{HE}+d_{Sem})}$ is used in this approach. Note that only the weight calculation of the hierarchy encoder is changed while the same weight normalization and propagation processes are used as shown in Equation 3.6 and Equation 3.7, respectively.

Introducing Label Semantics after Hierarchy Encoder

The fourth approach is abbreviated as post-HE, and it feeds semantic representations after hierarchy-aware label-specific representations are generated. Therefore the attention mechanism and the CapsNet in the implicit relationship extractor can take label semantics into account for inferring the relevance of different elements in the representations and capturing latent relationships. Similar to the first approach (in-TRE), semantic representations play the role of additive bias terms in this approach.

Equation 3.7 is modified as follows in the post-HE approach:

$$\boldsymbol{v}_{l}^{D} = \text{ReLU}\left(\sum_{1\leq l'\leq L} \tilde{w}_{ll'}^{D}\boldsymbol{z}_{l'}^{D} + \boldsymbol{s}_{l}'\right). \tag{4.10}$$

d_{Sem} is equal to d_{HE} to match the sizes of added vectors.

Introducing Label Semantics in Attention Mechanism

For this approach, which is abbreviated as in-Attn, attention weights of primary capsules in the implicit relationship extractor are obtained not only from textual

representations but also from label semantic representations. Therefore, the label relationship extractor can infer the importance of each primary capsule using the correlations between the textual and label semantic representations. However, similar to the third approach, the in-HE approach, the output representations of the attention mechanism do not contain information on label semantics. Therefore, the CapsNet of the implicit relationship extractor cannot make use of the semantic information to infer latent relationships via a dynamic routing algorithm.

Equation 3.8 is replaced with the following equation:

$$\boldsymbol{a}^D = \tanh\left(\boldsymbol{W}^2_{Attn}\text{ReLU}\left(\boldsymbol{W}^1_{Attn}\text{Concat}\left(\boldsymbol{v}^D_1, \ldots, \boldsymbol{v}^D_L, \boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_L\right) + \boldsymbol{b}^1_{Attn}\right) + \boldsymbol{b}^2_{Attn}\right). \tag{4.11}$$

While the shapes of $\boldsymbol{W}^2_{Attn}$, $\boldsymbol{b}^1_{Attn}$, and $\boldsymbol{b}^2_{Attn}$ remain the same as Equation 3.8, $\boldsymbol{W}^2_{Attn} \in \mathbb{R}^{L \times L(d_{HE}+d_{Sem})}$ is used in this approach.

4.3.3 Results

Performance on the WOS-46985 Dataset

Approach	Micro-F1	Macro-F1
GACaps-HTC	0.876	0.829
GACaps-HTC + in-TRE semantics	0.874	0.823
GACaps-HTC + pre-HE semantics	0.869	0.807
GACaps-HTC + in-HE semantics	0.873	0.816
GACaps-HTC + post-HE semantics	0.876	0.821
GACaps-HTC + in-Attn semantics	0.864	0.806
GACaps-HTC + semantic-aware dynamic routing	**0.878**	**0.831**

The best results are highlighted in **bold.**

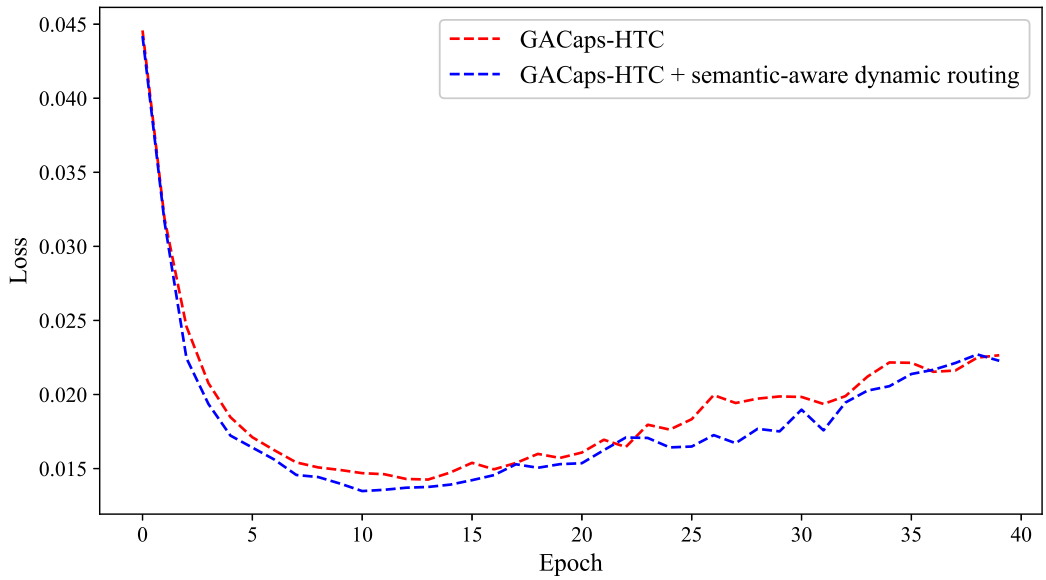
Table 4.1: Experiment results on the WOS-46985 dataset.

The experimental results on the WOS-46985 dataset are listed in Table 4.1.

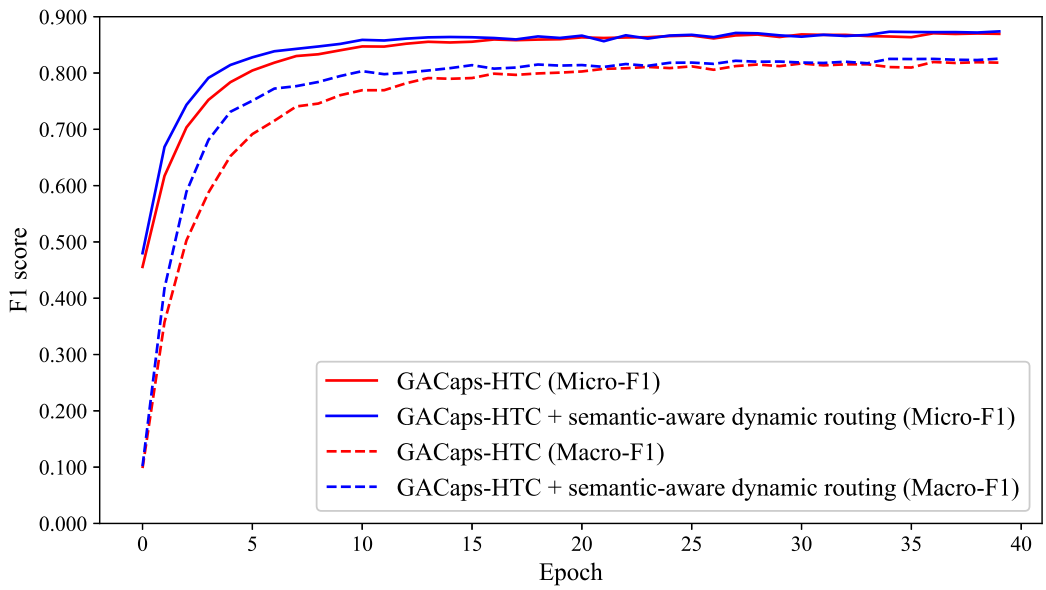
The results demonstrate that the performance of GACaps-HTC was enhanced by augmenting the semantic information of labels. Also, they show that the proposed method, semantic-aware dynamic routing, was the most effective method to inject the semantic information into the classifier.

Note that the in-TRE approach fed textual representations embedded with label semantic information into the hierarchy encoder, and therefore the semantic information took part in inferring attention weights in the hierarchy encoder. Thus, Table 4.1 shows that the approaches that made use of the semantic information when inferring the relative importance of neighboring labels (in-TRE, pre-HE, and in-HE) achieved lower F1 scores than GACaps-HTC without semantics. The same can be said for the approaches that used semantic information when obtaining attention weights in the implicit relationship extractor. These observations are believed to be due to the classifier overfitting to display static attention (where the model learns to highlight features independent to the input text[154]) conditioned on label occurrence statistics and semantics only.

While the performance enhanced by employing semantic-aware dynamic routing may seem subtle, the proposed dynamic routing algorithm accelerated the classifier's training, as shown in Figure 4.3. Both the validation loss plot (Figure 4.3 (a)) and the validation F1 score plot (Figure 4.3 (b) show that employing semantic-aware dynamic routing led to faster convergence. Training time until convergence was reduced by approximately 30% when trained under the same environment (single GTX 1080 Ti).



(a) Loss curve



(b) F1 score curve

Figure 4.3: Validation loss and F1 score plots obtained by training GACaps-HTC on the WOS-46985 dataset with and without augmenting label semantics.

Approach	Micro-F1	Macro-F1
GACaps-HTC	0.868	**0.698**
GACaps-HTC + in-TRE semantics	0.870	0.695
GACaps-HTC + pre-HE semantics	0.871	0.694
GACaps-HTC + in-HE semantics	0.863	0.693
GACaps-HTC + post-HE semantics	0.867	0.694
GACaps-HTC + in-Attn semantics	0.864	0.694
GACaps-HTC + semantic-aware dynamic routing	**0.872**	0.694

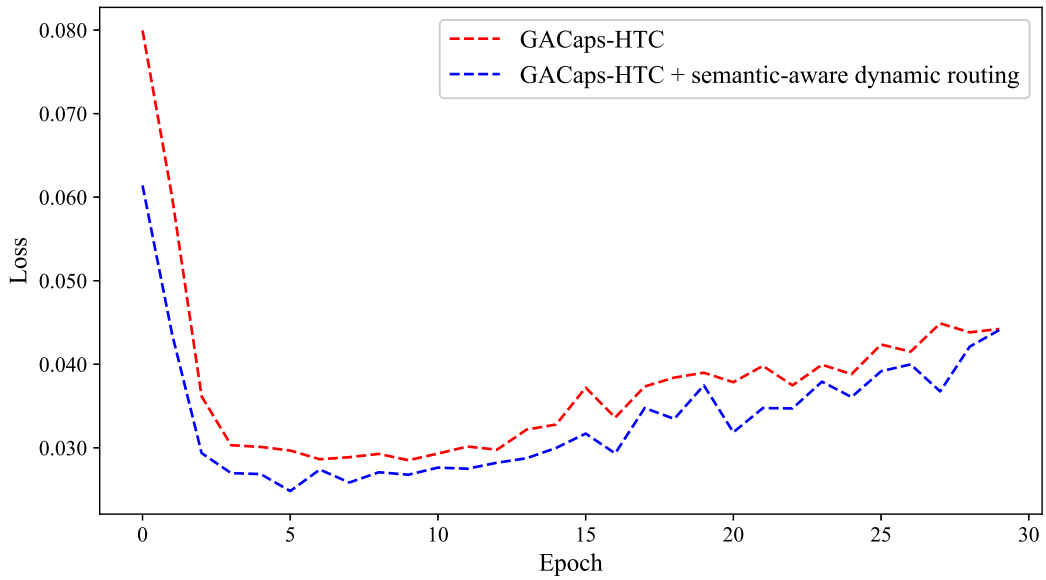
The best results are highlighted in **bold.**

Table 4.2: Experiment results on the RCV1 dataset.

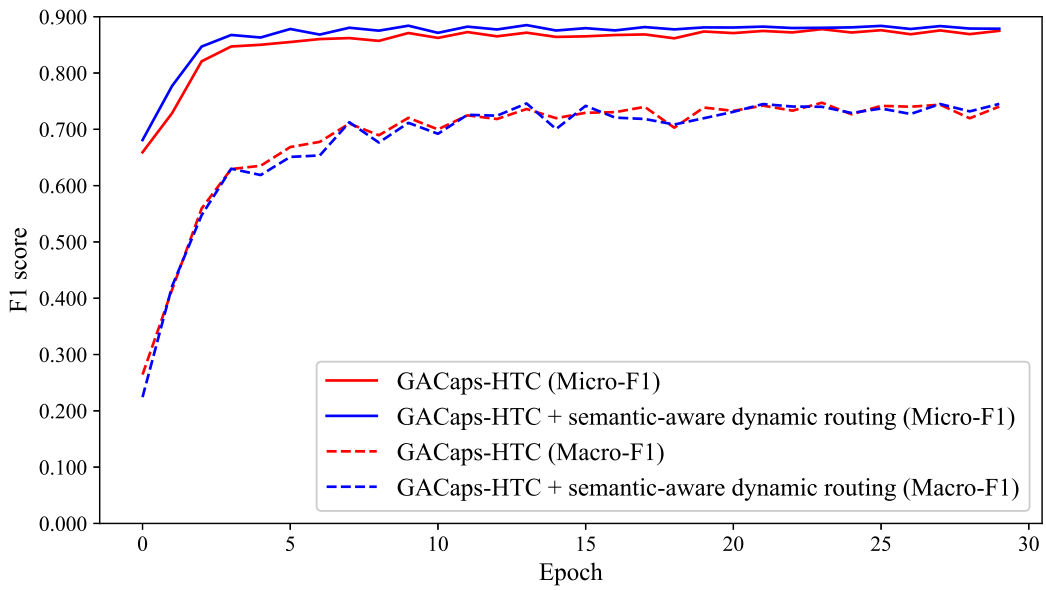
Performance on the RCV1 Dataset

The experimental results on the RCV1 dataset are shown in Table 4.2. Employing semantic-aware dynamic routing led to a slightly improved micro-F1 score and decreased macro-F1 score. Also, while injecting the semantic information of labels in the textual representation extractor, hierarchy encoder, or the attention mechanism led to degraded performance for the experiments performed on the WOS-46985 dataset, in-TRE and pre-HE approaches were able to achieve a similar level of performance compared to the approach with the semantic-aware dynamic routing algorithm.

The effectiveness of the proposed dynamic routing algorithm in enabling the model to achieve faster convergence is depicted in Figure 4.4. The validation loss plot in Figure 4.4 (a) and the micro-F1 plot in Figure 4.4 (b) show that employing semantic-aware dynamic routing led to faster convergence. Furthermore, the classifier with the semantic-aware dynamic routing algorithm took approximately 63%, 52%, and 70% less time to converge compared to GACaps-HTC without label semantics, in-TRE approach, and the pre-HE approaches, which are the approaches



(a) Loss curve



(b) F1 score curve

Figure 4.4: Validation loss and F1 score plots obtained by training GACaps-HTC on the RCV1 dataset with and without augmenting label semantics.

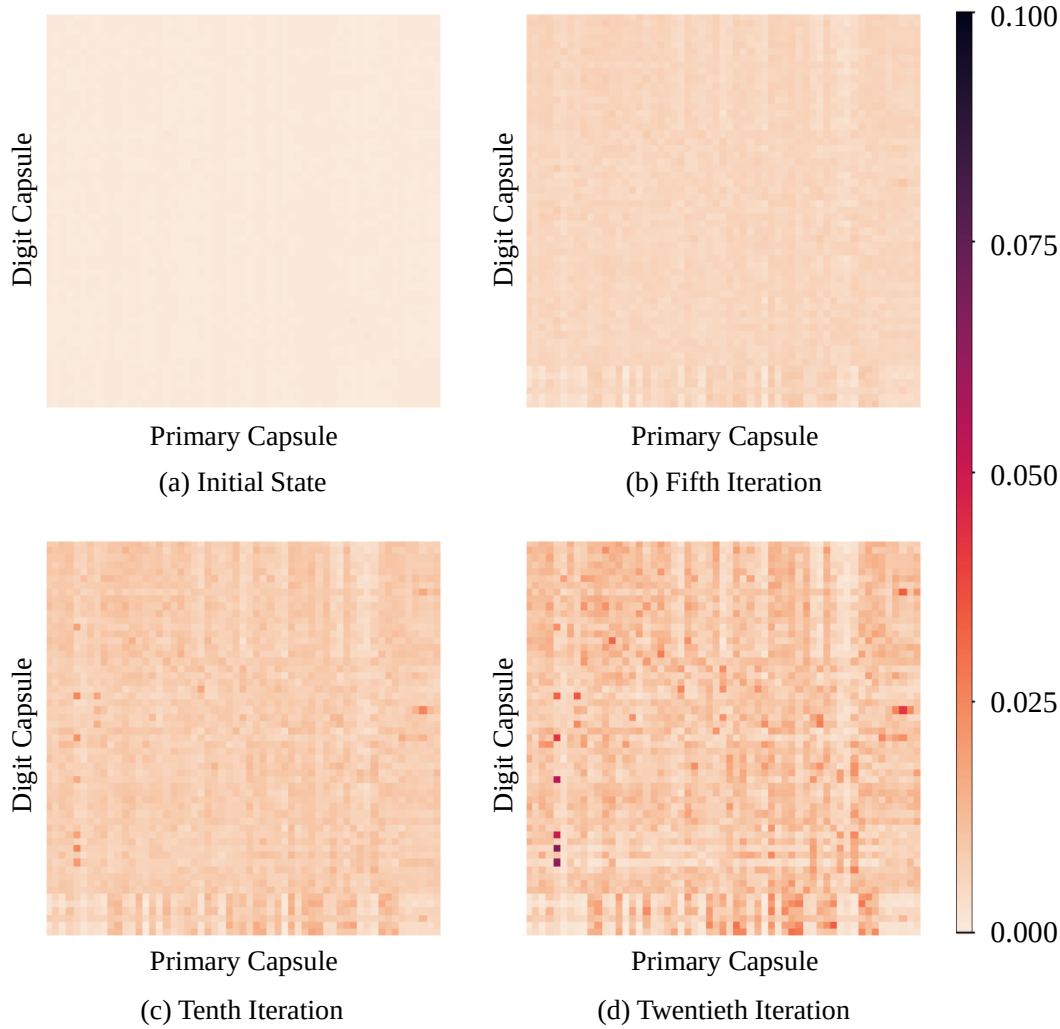


Figure 4.5: Visualization of normalized coupling coefficients in the capsule network on the RCV1 dataset when trained using semantic-aware dynamic routing algorithm.

that achieved a similar level of performance, respectively.

Latent relationships between the RCV1 dataset's labels captured by the CapsNet are illustrated in Figure 4.5 as heatmaps. Like Figure 3.9, only the labels on the second level of the hierarchy are depicted. Similar to GACaps-HTC using the dynamic routing algorithm by Zhao *et al.*[64], the dynamic routing algorithm identified important relationships as the number of iterations grew. Several new intuitive relationships could be derived when adopting the semantic-aware dynamic routing algorithm. For example, the corporate strategy/plans label, corporate performance label, and corporate management label had strong connections with the monetary/economic label. Also, the monetary/economic label, science/technology label, and corporate-related markets/marketing label had high correlations with the corporate performance label.

4.3.4 Ablation Studies

The following ablation studies were performed on the WOS-46985 dataset.

Ablation Studies on Semantic Bias and Gating Mechanism

The effectiveness of semantic bias and the gating mechanism used to inject semantic information into the primary capsules of the implicit relationship extractor's CapsNet is presented in Table 4.3. Simply adding semantic bias with textual representation led to decreased performance due to label semantics overwhelming textual information, leading to the model overfitting to classify a document using semantics rather than from the document's contents. This overfitting issue was elevated via the gating mechanism as the model can learn the relative importance of the semantic

information compared to the textual information.

Semantic Bias	Gating Mechanism	Micro-F1	Macro-F1
-	-	0.875	0.826
✓	-	0.873	0.823
✓	✓	**0.878**	**0.831**

The best results are highlighted in **bold.**

Table 4.3: Ablation study results regarding semantic bias and gating mechanism on the WOS-46985 dataset.

Figure 4.6 depicts the validation loss and F1 score plots illustrated to determine whether the semantic bias or the gating mechanism accelerated the classifier's convergence. Red, blue and green lines, each denoted as "Without semantic bias," "Without gating mechanism," and "With gating mechanism," corresponds to the first, second, and third row in Table 4.3, respectively. Figure 4.6 shows that while employing the semantic bias helped the model to converge faster, the gating mechanism did not affect how fast GACaps-HTC converged.

Ablation Studies on Coupling Coefficient Initialization

The F1 scores recorded by training GACaps-HTC using different coupling coefficient initialization methods are shown in Table 4.4. The first column represents whether an initial coupling coefficient between two labels was initialized from their semantic similarity or from zero. The second column denotes whether the initial coupling coefficients were trainable or not. Using semantic-based initialization had shown to lead to improved F1 scores while making the coefficients trainable was effective only when the coefficients were initialized from semantic similarities.

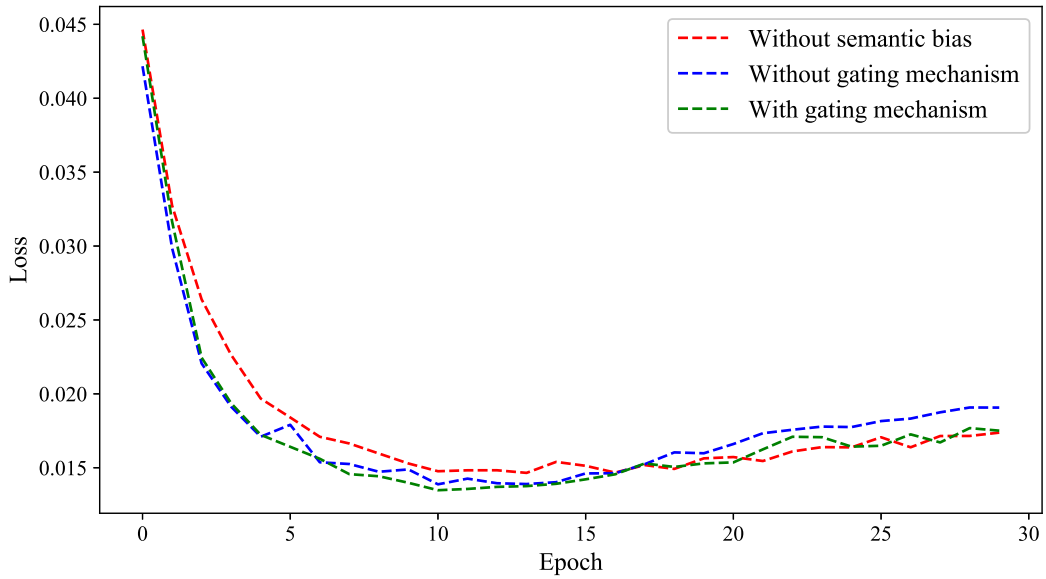
Figure 4.7 illustrates the validation loss and F1 score plots for analyzing whether

Semantic-Based Initialization	Trainable Coefficients	Micro-F1	Macro-F1
-	-	0.874	0.824
✓	-	0.875	0.827
-	✓	0.873	0.822
✓	✓	**0.878**	**0.831**

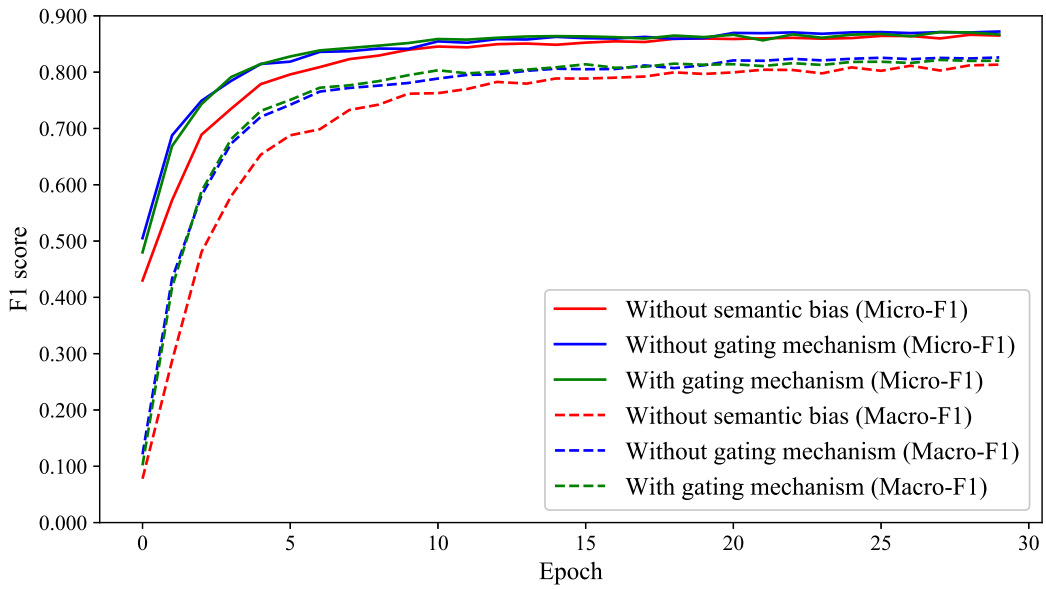
The best results are highlighted in **bold.**

Table 4.4: Ablation study results regarding coupling coefficient initialization and training on the WOS-46985 dataset.

the coupling coefficient initialization strategies affected how fast GACaps-HTC converges. Semantic-based coupling coefficient initialization and coupling coefficient optimization had both shown to lead to similar curves and, therefore, a similar level of convergence speed.

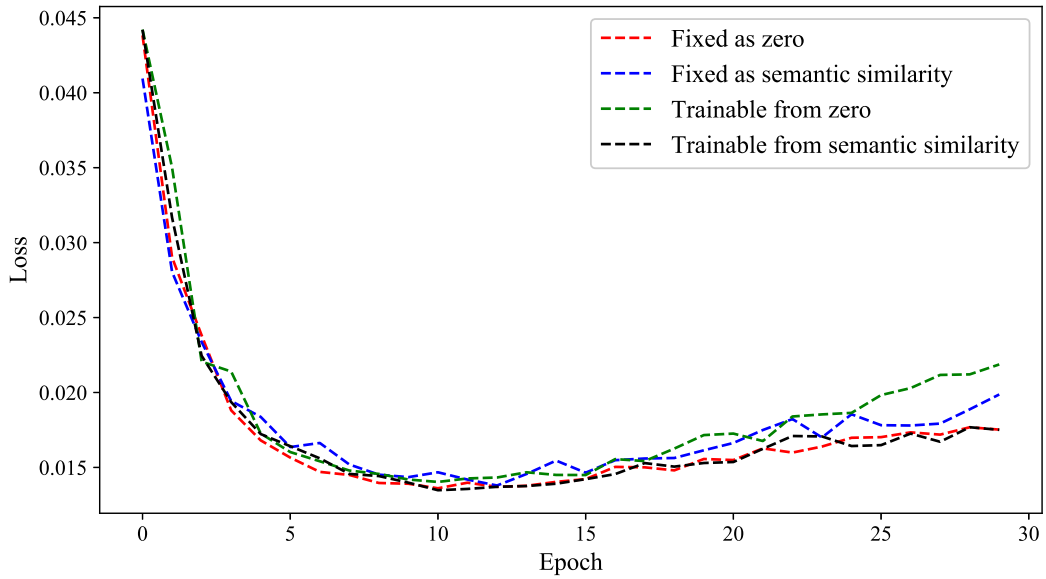


(a) Loss

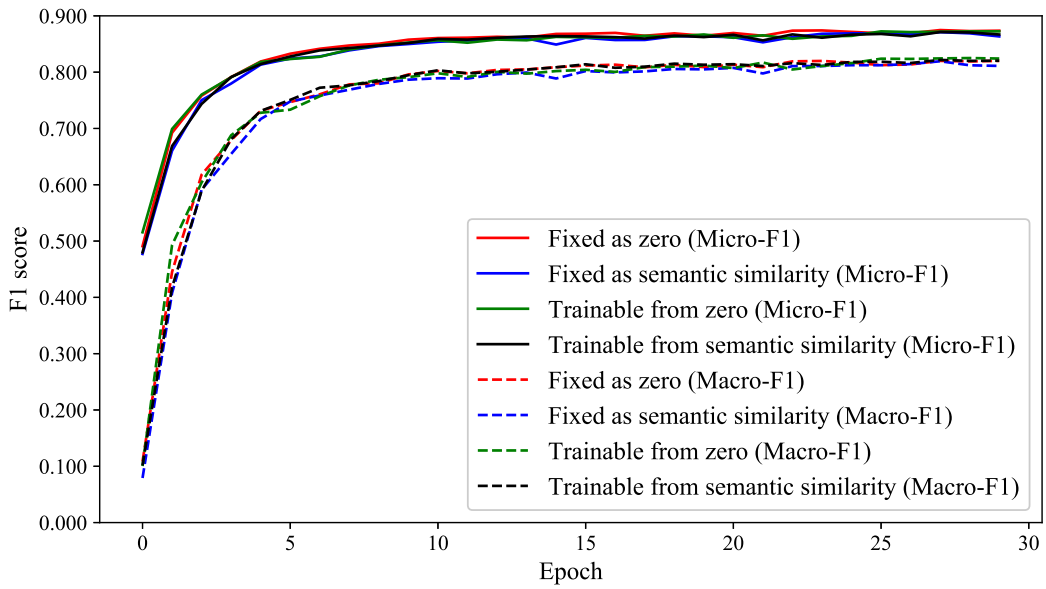


(b) F1 Score

Figure 4.6: Validation loss and F1 score plots obtained by training GACaps-HTC on the WOS-46985 dataset for ablation studies regarding semantic bias and gating mechanism.



(a) Loss



(b) F1 Score

Figure 4.7: Validation loss and F1 score plots obtained by training GACaps-HTC on the WOS-46985 dataset for ablation studies regarding coupling coefficient initialization and training.

Chapter 5

Aspect Category Sentiment Analysis Using Graph Attention Capsule Network

5.1 Problem Definition

The goal of aspect category sentiment analysis is to identify a set of predefined entities that appear in a given text document and classify a sentiment polarity (positive, neutral, and negative) for each entity. Let D denote the text document as defined in Sections 3.1 and 4.1, and $\mathcal{E}$ denote the predefined set of entities, also known as aspect categories. The ground-truth set of aspect category-sentiment pairs is denoted as $\mathcal{S}^D$ and represented as follows:

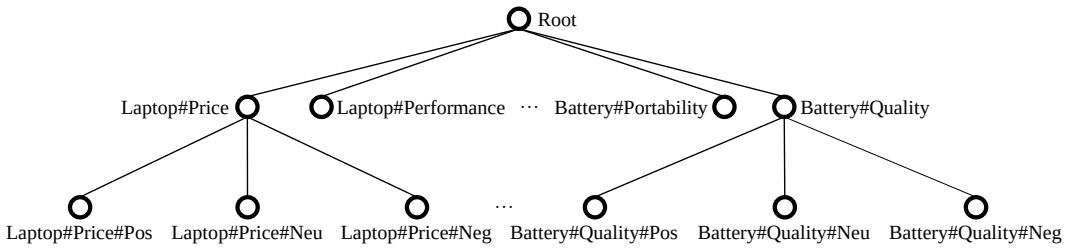
$$\mathcal{S}^{\mathcal{D}} \subset \{(e, \chi) \,|e \in \mathcal{E} \text{ and } \chi \in \{\text{positive}, \text{neutral}, \text{negative}\}\}\,. \tag{5.1}$$

An aspect category sentiment analysis model learns a mapping from D to $\mathcal{S}^D$ and outputs a classification probability for each pair of an aspect category and a sentiment polarity.

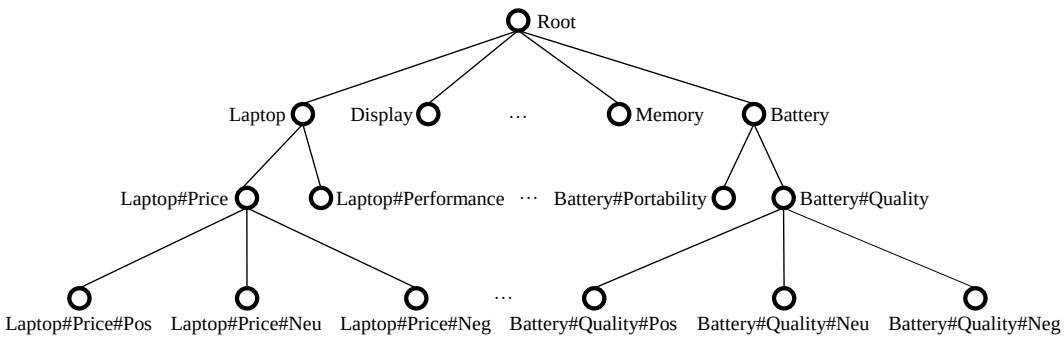
5.2 Methods

In this chapter, GACaps-HTC is applied and evaluated on aspect category sentiment analysis to investigate other practical use cases of the proposed method. This section describes how an aspect category sentiment analysis problem can be transformed into an HTC problem to employ GACaps-HTC. While previous aspect category sentiment analysis methods, including Cartesian product methods and add-one-dimension methods, ignore the hierarchical property of the task, this thesis takes a hierarchical classification approach similar to Cai *et al.*[120]. However, while Cai *et al.*[120] makes use of a hierarchy with only two levels (each level consisting of aspect category labels and sentiment polarity labels), this thesis acknowledges the fact that aspect categories can form a hierarchical structure of their own. For example, in the SemEval2015 and SemEval2016 datasets, each aspect category label consists of an entity type label and an entity attribute label. As an entity attribute label corresponding to a text document depends on the entity type, there is a two-level hierarchical structure of labels. The following example from the Laptop2015 dataset of the SemEval2015 datasets stated on the official website of the SemEval2015 datasets[155] clearly demonstrates this hierarchical property: "It is extremely portable and easily connects to WIFI at the library and elsewhere." While the entire sentence is about a laptop (entity type label), it mentions both its portability and connectivity (entity attribute labels). Therefore, resulting aspect category labels are tuples of entity type labels and entity attribute labels as follows: (laptop, portability) and (laptop, connectivity).

Based on the aforementioned observation, a hierarchical structure of at least two levels is constructed from each aspect category sentiment analysis dataset. Topics, or



(a) Hierarchy with two levels (aspect category level and sentiment level)



(b) Hierarchy with three levels (entity type level, aspect category level, and sentiment level)

Figure 5.1: Hierarchical structure of labels derived from the Laptop2015 dataset[6].

aspect categories, of a given text document are inferred from an HTC model's classification results corresponding to intermediate nodes in the hierarchy. The sentiment polarity corresponding to the inferred aspect category is deduced from the HTC model's classification results on the leaf nodes connected to the intermediate nodes. Figure 5.1 illustrates example hierarchical structures of labels formulated from the Laptop2015 dataset, where Pos, Neu, and Neg are abbreviations for positive, neutral, and negative sentiments, repectively. Figure 5.1 (a) is the hierarchy with two levels comprising of aspect category labels and sentiment labels, respectively, similar to Cai *et al.*[120]. Figure 5.1 (b) is the hierarchy where a hierarchical structure can be deduced from aspect categories only, leading to the resulting label hierarchy having more than two levels. While the hierarchy illustrated in Figure 5.1 (b) has

more labels to classify using a HTC model compared to that in Figure 5.1 (a), it can explicitly express relationships between aspect categories.

The label set of an HTC problem corresponding to an aspect category sentiment analysis problem comprises entity type labels, aspect category labels represented as pairs of an entity type label and an entity attribute label (which correspond to aspect categories), and sentiment polarities corresponding to aspect categories. Therefore, adding the number of entity type labels, the number of aspect category labels ($|\mathcal{E}|$), and the number of sentiment polarity labels ($3|\mathcal{E}|$) results in the number of labels (L). The label hierarchy $\mathcal{H}$ is derived as the union of two hierarchies: the hierarchy of entity type labels and aspect category labels and the hierarchy of aspect category labels and sentiment polarity labels. As defined in Sections 3.1 and 4.1, the hierarchy is represented as a set comprising tuples of parent and child labels. Inferred set of leaf labels obtained from an HTC model's output classification probabilities is the inferred set of aspect category-sentiment pairs and is compared with $\mathcal{S}^D$ for evaluating the sentiment analysis performance.

5.3 Experiments

5.3.1 Experiment Settings

Datasets

The effectiveness of GACaps-HTC in aspect category sentiment analysis was evaluated on two datasets included in the SemEval2015 datasets and two datasets included in the SemEval2016 datasets. Note that versions and (training and testing) splits of the following datasets followed those of Cai *et al.*[120], published in their official implementation repository[1]. The first benchmark dataset was the Laptop2015 dataset from the SemEval2015 datasets, consisting of 2,041 reviews on laptops. There were 22 entity type labels, including display, motherboard, memory, and battery, and nine possible entity attribute labels for each entity type label, including general, price, and quality. 80 distinct aspect categories (combinations of entity type labels and entity attribute labels) existed in the dataset, resulting in 102 intermediate nodes in the label hierarchy and 240 leaf nodes indicating sentiment polarities corresponding to the aspect categories. 1,397 examples in the dataset were used for training while 644 examples were used for testing.

The second dataset, also a part of the SemEval2015 datasets, was the Restaurant2015 dataset comprising 1,674 restaurant review documents. There were six entity type labels (ambiance, drinks, food, location, restaurant, and service) and five possible entity attribute labels (general, price, quality, style, and miscellaneous) for each entity type label, resulting in 13 aspect categories, 19 labels corresponding to intermediate nodes of the label hierarchy, and 39 leaf nodes corresponding to sentiment polarities. The Restaurant2015 dataset was split into a training set of 1,102

[1] https://github.com/NUSTM/ACSA-HGCN

examples and a testing set of 572 examples.

The third dataset was the Laptop2016 dataset, included in the SemEval2016 datasets, which has the same set of entity type labels, entity attribute labels, and sentiment polarity labels as the Laptop2015 dataset. 2,609 reviews on laptops were in the dataset, where 2,037 examples were used for training and 572 examples were used for testing. Finally, the last dataset was the Restaurant2016 dataset in the SemEval2016 datasets that shares the same entity type labels, entity attribute labels, and sentiment polarity labels as the Restaurant2015 dataset. It was comprised of 2,260 examples, which were split into 1,680 training examples and 580 testing examples.

Metrics

The performance on aspect category sentiment analysis was measured and compared using micro-F1 scores (described in Subsection 3.3.1) following Cai *et al.*[120]. Note that while F1 scores obtained to evaluate HTC performance in Chapter 3 and Chapter 4 involved every label in the hierarchy, aspect category sentiment analysis performance was compared using F1 scores from only the leaf labels as they correspond to tuples of aspect categories and sentiment polarities. However, micro-F1 scores obtained from every label in the hierarchy were used for model selection, as doing so led to better test performance. Micro-precision and micro-recall used to calculate F1 scores are presented in the following subsection for more insights into the experiment results.

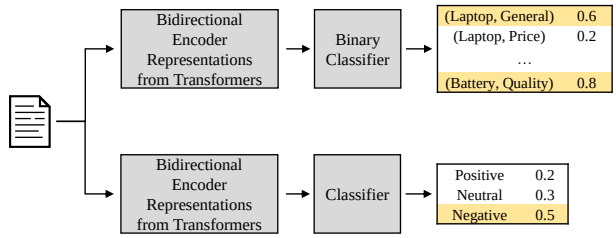
Baselines

The GACaps-HTC was compared with the following baseline aspect category sentiment analysis approaches. The first approach was a pipeline approach that utilizes two BERT models for aspect category detection and sentiment polarity classification. While this approach separates the task into two easier tasks, it fails to exploit the relationships between two partial tasks. Furthermore, it could not capture conflicting sentiments towards different aspect categories that lie in a single document.

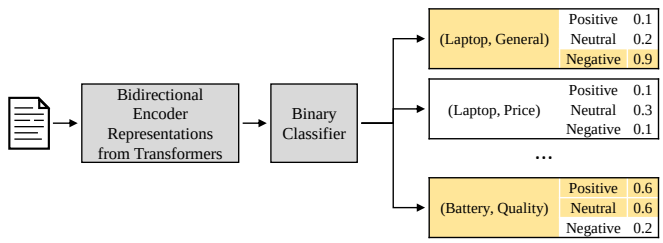
The second approach was a Cartesian product approach that performs a binary classification for each tuple of an aspect category (represented as a tuple of an entity type and an entity attribute) and a sentiment polarity. This approach also used BERT as a document encoder and obtained classification probabilities from document representations from BERT. The third approach was an add-one-dimension approach using BERT as a document encoder that jointly infers whether an aspect category appears in a document and its corresponding sentiment polarity.

The fourth and the last baseline approaches were hierarchical classification approaches. The fourth approach was a hierarchical Transformer approach which used the attention mechanism in a Transformer to model the relationships between aspect categories and the relationships between an aspect category and a sentiment polarity. Finally, the last baseline approach was a hierarchical GCN[120] approach that made use of two GCNs, each to capture the correlations between a pair of aspect categories and to model the relationships between a pair of an aspect category and its corresponding sentiment polarities. Figure 5.2 depicts the architectures of the baseline approaches, where highlighted labels denote inferred labels. Note that plain classifiers in Figure 5.2 are single-label classifiers that return the label with

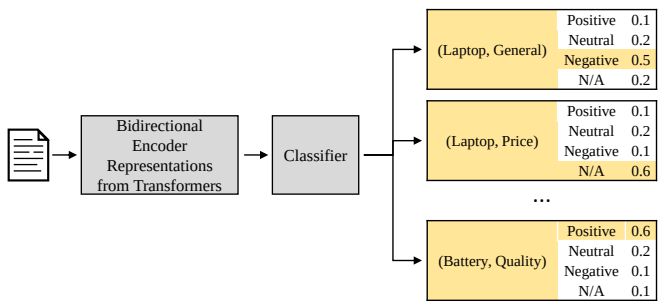
the highest classification probability, and binary classifiers are multi-label classifiers that return a set of labels with classification probabilities higher than a predefined threshold.



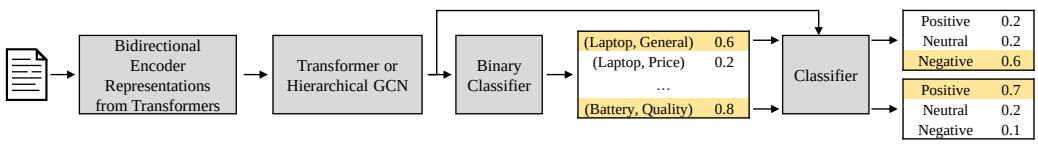
(a) Pipeline Approach



(b) Cartesian Product Approach



(c) Add-One-Dimension Approach



(d) Hierarchical Transformer Approach and Hierarchical GCN Approach

Figure 5.2: Illustrations of baseline approaches for aspect category sentiment analysis.

Implementation Details

For all four datasets, BERT was employed as the language model in the textual representation extractor. While dropout with a drop probability of 0.5 was applied to textual representations in experiments described in Chapter 3 and Chapter 4, the drop probability was set to 0.05 (the Laptop2015, Restaurant2015, and Laptop2016 datasets) or 0.1 (the Restaurant2016 dataset) for aspect category sentiment analysis as documents were relatively shorter and salient words indicating aspect categories or sentiment polarities seldom appear repeatedly.

Textual representations with $d_{LM} = 768$ were obtained from BERT, and representations of $d_{Conv} = L \times 100$, $d_{HE} = 100$, and $d_{Caps} = 32$ were extracted by GACaps-HTC as described in Subsection 3.3.1. The number of labels L was 342 for the Laptop2015 and Laptop2016 datasets and 58 for the Restaurant2015 and Restaurant2016 datasets. The capsule pruning ratio ρ, capsule dropout rate ϕ, contradiction penalty hyperparameter δ, dynamic routing convergence threshold ϵ, and focal loss hyperparameter γ were set to the same values as described in Subsection 3.3.1. The weight of the contradiction penalty term was set to $\lambda = 0.001$.

The GACaps-HTC was trained using mini-batches of size 16. For the first two epochs, a learning rate of 0.0001 was used, BERT was frozen (not updated by gradient descent), and after those two epochs, an initial learning rate of 0.00005 was used to train the entire model, including the BERT. An Adam optimizer was used for the Laptop2015, Restaurant2015, and Laptop2016 datasets, while an Adam optimizer with decoupled weight decay regularization[156], otherwise known as an AdamW optimizer, was used for the Restaurant2016 dataset. As described in Subsection 3.3.1, the learning rate was decayed by a factor of 0.1 when five consecutive epochs

recorded suboptimal validation micro-F1 scores obtained for the entire label hierarchy until the number of consecutive epochs with suboptimal scores reached 20, and training was stopped.

5.3.2 Results

Approach	Laptop2015	Restaurant2015	Laptop2016	Restaurant2016
Pipeline	0.430	0.494	0.394	0.562
Cartesian product	0.328	0.584	0.395	0.689
Add-one-dimension	0.489	0.617	0.472	0.698
Hierarchical Transformer	0.578	0.647	0.527	0.735
Hierarchical GCN	**0.621**	0.642	0.542	**0.746**
GACaps-HTC (2 levels)	0.574	0.629	**0.549**	0.723
GACaps-HTC (3 levels)	0.611	**0.657**	0.548	0.727

The best results are highlighted in **bold.**

Table 5.1: Overview of the experiment results on aspect category sentiment analysis.

In this subsection, experiment results regarding the pipeline, Cartesian product, add-one-dimension, hierarchical Transformer, and hierarchical GCN approaches are results reported by Cai *et al.*[120]. The overview of the aspect category sentiment analysis experiment results is shown in Table 5.1. Each version of GACaps-HTC was trained and tested three times on each dataset, and the average F1 scores are presented. Utilizing the aspect category hierarchy (annotated as 3 levels in the sixth row) had shown to lead to enhanced performance, except for the case of the Laptop2016 dataset that showed an insignificant difference, as, while it increased the number of labels, it enabled the hierarchy encoder to capture the relationships between aspect categories. For further analysis, micro-precision, micro-recall, and micro-F1 scores obtained from a single run are presented below.

Performance on the Laptop2015 Dataset

Approach	Micro-Precision	Micro-Recall	Micro-F1
Pipeline	0.369	0.516	0.430
Cartesian product	0.731	0.212	0.328
Add-one-dimension	0.642	0.396	0.489
Hierarchical Transformer	0.656	0.520	0.578
Hierarchical GCN	0.719	0.547	0.621
GACaps-HTC	0.693	0.571	**0.626**

The best results are highlighted in **bold.**

Table 5.2: Experiment results on the Laptop2015 dataset.

Experiment results obtained from the Laptop2015 dataset are shown in Table 5.2. The GACaps-HTC had could outperform baseline approaches in micro-F1 scores in this single run while the average micro-F1 score of GACaps-HTC in Table 5.1 was lower than that of hierarchical GCN. The results indicate that while the proposed approach yielded volatile performance, the proposed approach can be adopted for aspect category sentiment analysis with a simple coarse hyperparameter search. While the proposed approach recorded the highest micro-recall compared to other baselines, it ranked third in micro-precision. Such results indicate that GACaps-HTC was able to discover sentiments that other baseline approaches could not find, thanks to its explicit parent-child relationship modeling and implicit relationship extraction.

Performance on the Restaurant2015 Dataset

Experiment results from the Restaurant2015 dataset are listed in Table 5.3. The proposed approach had shown to outperform baseline approaches in micro-F1 scores, which showed that GACaps-HTC was able to perform well for both semantic analysis

Approach	Micro-Precision	Micro-Recall	Micro-F1
Pipeline	0.381	0.700	0.494
Cartesian product	0.720	0.492	0.584
Add-one-dimension	0.688	0.559	0.617
Hierarchical Transformer	0.702	0.600	0.647
Hierarchical GCN	0.719	0.580	0.642
GACaps-HTC	0.677	0.639	**0.657**

The best results are highlighted in **bold.**

Table 5.3: Experiment results on the Restaurant2015 dataset.

(HTC and aspect category detection) and sentiment analysis (sentiment polarity classification) on documents.

Performance on the Laptop2016 Dataset

Approach	Micro-Precision	Micro-Recall	Micro-F1
Pipeline	0.319	0.516	0.394
Cartesian product	0.650	0.274	0.395
Add-one-dimension	0.588	0.395	0.472
Hierarchical Transformer	0.581	0.483	0.527
Hierarchical GCN	0.614	0.484	0.542
GACaps-HTC	0.545	0.547	**0.546**

The best results are highlighted in **bold.**

Table 5.4: Experiment results on the Laptop2016 dataset.

Table 5.4 shows the experiment results acquired from training and evaluating sentiment analysis approaches using the Laptop2016 dataset. While the hyperparameters for the Restaurant2015 dataset and the Laptop2016 dataset were set to the same values as those obtained from a coarse hyperparameter search using the Laptop2015 dataset, GACaps-HTC still outperformed baseline approaches in micro-

F1 scores. This lack of need for a fine-grained (or even any) hyperparameter search shows the practical usability of the proposed approach, as training and evaluating a model repeatedly for a fine-grained hyperparameter search can be costly.

While the proposed approach also recorded the highest micro-recall compared to other baselines similar to the results described in Table 5.2, it ranked fifth in micro-precision. It can be deduced that GACaps-HTC's explicit and implicit label relationship modeling contributed towards discovering sentiments that are relatively harder to find rather than removing sentiments that were incorrectly inferred.

Performance on the Restaurant2016 Dataset

Approach	Micro-Precision	Micro-Recall	Micro-F1
Pipeline	0.436	0.791	0.562
Cartesian product	0.750	0.638	0.689
Add-one-dimension	0.718	0.680	0.698
Hierarchical Transformer	0.737	0.732	0.735
Hierarchical GCN	0.764	0.728	**0.746**
GACaps-HTC	0.738	0.735	0.736

The best results are highlighted in **bold.**

Table 5.5: Experiment results on the Restaurant2016 dataset.

Experiment results on the Restaurant2016 dataset are shown in Table 5.5. Results obtained from the pipeline approach and the Cartesian product approach shown in Tables 5.2, 5.3, 5.4, and 5.5 clearly indicate the tradeoff between precision and recall as the pipeline approach recorded high micro-recall while achieving low micro-precision and the Cartesian product approach obtained high micro-precision with low micro-recall. However, the proposed approach always recorded relatively high (or even the highest) precision and recall, indicating that GACaps-HTC's high recall

or precision is not an outcome of a tradeoff but a sign of a well-performing sentiment analysis model.

While GACaps-HTC outperformed the baselines on other sentiment analysis datasets, it recorded the second-best F1 score on the Restaurant2016 dataset. This performance was obtained from a separate hyperparameter search, unlike the results shown in Tables 5.2, 5.3, and 5.4, as using the same set of hyperparameters led to a lower F1 score (0.702). Such results show that there is room for improvement in GACaps-HTC when it comes to aspect category sentiment analysis.

Chapter 6

Conclusions

6.1 Summary and Contributions

This thesis proposes a deep learning-based HTC approach by acknowledging the importance of not only explicit parent-child relationships between labels but also implicit label relationships that may appear for any pair of labels. The proposed approach, GACaps-HTC, comprises three parts: a textual representation extractor, a hierarchy encoder, and an implicit relationship extractor. The textual representation extractor uses a pretrained language model to generate a rich textual representation for a given text document. The hierarchy encoder models label relationships expressed by the label hierarchy using a GAT, and the implicit relationship extractor uses a CapsNet to model the label relationships that are not fully captured by the hierarchy. The model was trained and evaluated using widely used benchmark HTC datasets. After training GACaps-HTC, various post-processing methods were applied and compared to select the best method for each dataset. The results demonstrated that the proposed approach outperformed the compared baselines.

This thesis also proposes a dynamic routing algorithm that injects the information on what each label means and how semantically close each pair of labels are into a CapsNet. The proposed semantic-aware dynamic routing algorithm initial-

izes the coupling coefficient corresponding to a pair of labels from the similarity of the labels' semantic representations. Furthermore, the algorithm defines an additive bias term from labels' semantic representations and uses a semantic-based gating mechanism that can control how much a label's semantic information affects the dynamic routing algorithm. GACaps-HTC using semantic-aware dynamic routing algorithm was compared with GACaps-HTC using conventional dynamic routing algorithm and other variants of GACaps-HTC that inject the semantic information into various parts of the model. The results showed that GACaps-HTC with the proposed algorithm outperformed the variants of GACaps-HTC and that adopting the algorithm led to much faster convergence.

Finally, this thesis investigates whether GACaps-HTC can be adopted for other use cases. Aspect category sentiment analysis problems are transformed into HTC problems, and GACaps-HTC is employed for these problems. Experiments were conducted on widely used benchmark sentiment analysis datasets, and the results demonstrate that GACaps-HTC showed competitive performance compared to the baseline approaches proposed specifically for aspect category sentiment analysis or outperformed them.

6.2 Limitations and Future Research

Modifying the proposed approach to extract representations of hyperbolic space is the future work of this thesis. Neural networks that use hyperbolic representations, or hyperbolic neural networks[58, 157], have been shown to be effective in generating representations that fully capture hierarchical structures. There are several ways GACaps-HTC can exploit a hyperbolic space, such as employing a hyperbolic language model[158], a hyperbolic GNN[159, 160], or a hyperbolic CapsNet[161]. Employing a hyperbolic space in the aforementioned fashion may lead to enhanced hierarchy-related expressive power and better HTC performance.

While this study investigates the effectiveness of the proposed approach using various post-processing methods, it does not cover how to preprocess input text documents. However, multiple fragments in a document may not be required to infer the labels corresponding to the document, and removing such fragments can lead to faster training and inference. Furthermore, a document may have slang terms and typos that can harm the classification results[162, 163]. Therefore, employing and comparing various text preprocessing methods and developing preprocessing methods specified for HTC is the future work of this thesis.

Finally, while this thesis aims to boost the HTC performance using a newly proposed approach and a new dynamic routing algorithm, it does not employ one of the most widely adopted methods to enhance classification performance, data augmentation. There are various task-agnostic textual data augmentation methods, including synonym replacement[164], data noising[165], and easy data augmentation[166]. Furthermore, there are auto-augmentation methods[167] that search for task-specific and model-specific optimal augmentation policies. As such augmentation methods

have shown to generally enhance a model's text classification performance, they can enable GACaps-HTC to achieve higher HTC performance and better noise robustness by providing the model with diverse input data. Demonstrating the enhanced performance obtained using these augmentation methods and acquiring the optimal augmentation policy via auto-augmentation will be the future development of this work.

Bibliography

[1] K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, L. E. Barnes, Hdltex: Hierarchical deep learning for text classification, in: Proceedings of the IEEE International Conference on Machine Learning and Applications, IEEE, 2017, pp. 364–371. doi:10.1109/ICMLA.2017.0-134.

[2] A. R. Perez, L. M. Martinez, J. M. Delfino, Physicochemical stability and rheologic properties of a natural hydrating and exfoliating formulation beneficial for the treatment of skin xeroses, Latin American Journal of Pharmacy 36 (2017) 157–164.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 6000—-6010. doi:10.5555/3295222.3295349.

[4] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450 (2016).

[5] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: Proceedings of the International Conference on Learning Representations, 2015, pp. 1–15. doi:10.48550/arXiv.1409.0473.

[6] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, SemEval-2015 task 12: Aspect based sentiment analysis, in: Proceedings of the International Workshop on Semantic Evaluation, 2015, pp. 486–495. doi:10.18653/v1/S15-2082.
URL https://aclanthology.org/S15-2082

[7] D. W. Otter, J. R. Medina, J. K. Kalita, A survey of the usages of deep learning for natural language processing, IEEE Transactions on Neural Networks and Learning Systems 32 (2) (2021) 604–624. doi:10.1109/TNNLS.2020.2979670.

[8] J. Hirschberg, C. D. Manning, Advances in natural language processing, Science 349 (6245) (2015) 261–266. doi:10.1126/science.aaa8685.

[9] M. Sharp, R. Ak, T. Hedberg, A survey of the advancing use and development of machine learning in smart manufacturing, Journal of Manufacturing Systems 48 (2018) 170–179. doi:10.1016/j.jmsy.2018.02.004.

[10] I. Mantegh, N. S. Darbandi, Knowledge-based task planning using natural language processing for robotic manufacturing, in: Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 3, 2010, pp. 1309–1316. doi:10.1115/DETC2010-29123.

[11] I. E. Fisher, M. R. Garnsey, M. E. Hughes, Natural language processing in accounting, auditing and finance: A synthesis of the literature with a roadmap for future research, Intelligent Systems in Accounting, Finance and Management 23 (3) (2016) 157–214. doi:10.1002/isaf.1386.

[12] J. M. Ho, A. Shahid, Financial Data Analytics: Theory and Application, Springer, 2022, Ch. Natural Language Processing for Exploring Culture in Finance: Theory and Applications, pp. 269–291. doi:10.1007/978-3-030-83799-0_9.

[13] F. Bozyiğit, D. Kılınç, The Impact of Artificial Intelligence on Governance, Economics and Finance, Volume 2, Springer, 2022, Ch. Practices of Natural Language Processing in the Finance Sector, pp. 157–170. doi:10.1007/978-981-16-8997-0_9.

[14] F. Popowich, Using text mining and natural language processing for health care claims processing, SIGKDD Exploration Newsletter 7 (1) (2005) 59–66. doi:10.1145/1089815.1089824.

[15] C. Friedman, N. Elhadad, Biomedical Informatics: Computer Applications in Health Care and Biomedicine, Springer, 2014, Ch. Natural Language Processing in Health Care and Biomedicine, pp. 255–284. doi:10.1007/978-1-4471-4474-8_8.

[16] A. Sleimi, N. Sannier, M. Sabetzadeh, L. Briand, J. Dann, Automated extraction of semantic legal metadata using natural language processing, in: IEEE International Requirements Engineering Conference, 2018, pp. 124–135. doi:10.1109/RE.2018.00022.

[17] M. Dragoni, S. Villata, W. Rizzi, G. Governatori, Combining natural language processing approaches for rule extraction from legal documents, in: U. Pagallo, M. Palmirani, P. Casanovas, G. Sartor, S. Villata (Eds.), AI Approaches to the Complexity of Legal Systems, 2018, pp. 287–300.

[18] H. Jiang, P. He, W. Chen, X. Liu, J. Gao, T. Zhao, SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2020, pp. 2177–2190. doi:10.18653/v1/2020.acl-main.197.

[19] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, in: Advances in Neural Information Processing Systems, Vol. 32, 2019, pp. 1–11.

[20] Z. Sun, C. Fan, Q. Han, X. Sun, Y. Meng, F. Wu, J. Li, Self-explaining structures improve nlp models, Computing Research Repository (2020).

[21] S. Takase, S. Kiyono, Lessons on parameter sharing across layers in transformers, Computing Research Repository (2021).

[22] X. Liu, K. Duh, L. Liu, J. Gao, Very deep transformers for neural machine translation, Computing Research Repository (2020).

[23] A. Aghajanyan, A. Gupta, A. Shrivastava, X. Chen, L. Zettlemoyer, S. Gupta, Muppet: Massive multi-task representations with pre-finetuning, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2021, pp. 5799–5811. doi:10.18653/v1/2021.emnlp-main.468.

[24] A. Aghajanyan, A. Shrivastava, A. Gupta, N. Goyal, L. Zettlemoyer, S. Gupta, Better fine-tuning by reducing representational collapse, in: Proceedings of the International Conference on Learning Representations, 2021, pp. 1–12.

[25] D. S. Sachan, M. Zaheer, R. Salakhutdinov, Revisiting lstm networks for semi-supervised text classification via mixed objective function, in: Proceedings of AAAI Conference on Artificial Intelligence, 2019, pp. 6940–6948.

[26] X. Liu, J. Gao, X. He, L. Deng, K. Duh, Y. Y. Wang, Representation learning using multi-task deep neural networks for semantic classification and information retrieval, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, ACL, 2015, pp. 912–921. doi:10.3115/v1/n15-1092.

[27] S. P. Panda, J. P. Mohanty, A domain classification-based information retrieval system, in: Proceedings of the IEEE International Women in Engineering Conference on Electrical and Computer Engineering, IEEE, 2020, pp. 122–125. doi:10.1109/WIECON-ECE52138.2020.9398018.

[28] S. Park, J. Cho, K. Park, H. Shin, Customer sentiment analysis with more sensibility, Engineering Applications of Artificial Intelligence 104 (2021) 104356. doi:10.1016/j.engappai.2021.104356.

[29] J. Yang, X. Zou, W. Zhang, H. Han, Microblog entiment analysis via embedding social contexts into an attentive lstm, Engineering Applications of Artificial Intelligence 97 (2021) 104048. doi:10.1016/j.engappai.2020.104048.

[30] B. Qu, G. Cong, C. Li, A. Sun, H. Chen, An evaluation of classification models for question topic categorization, Journal of the American Society of Information Science and Technology 63 (5) (2012) 889–903. doi:10.1002/asi.22611.

[31] W. Yu, Z. Sun, H. Liu, Z. Li, Z. Zheng, Multi-level deep learning based e-commerce product categorization, in: Proceedings of the International ACM SIGIR Conference on Research and Development of Information Retrieval: Workshop on E-Commerce, ACM, 2018, pp. 1–6.

[32] D. D. Lewis, Y. Yang, T. Russell-Rose, F. Li, Rcv1: A new benchmark collection for text categorization research, Journal of Machine Learning Research 5 (Apr) (2004) 361–397. doi:10.5555/1005332.1005345.

[33] E. Sandhaus, The new york times annotated corpus (Oct 2008).
URL https://catalog.ldc.upenn.edu/LDC2008T19

[34] J. C. Gomez, M. F. Moens, A survey of automated hierarchical classification of patents, in: Professional search in the modern world, Springer, 2014, pp. 215–249. doi:10.1007/978-3-319-12511-4_11.

[35] L. Abdelgawad, P. Kluegl, E. Genc, S. Falkner, F. Hutter, Optimizing neural networks for patent classification, in: Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2019, pp. 688–703. doi:10.1007/978-3-030-46133-1_41.

[36] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, K. Brinker, Multilabel classification via calibrated label ranking, Machine Learning 73 (2) (2008) 133–153. doi:10.1007/s10994-008-5064-8.

[37] T. Joachims, Text categorization with support vector machines: Learning with many relevant features, in: C. Nédellec, C. Rouveirol (Eds.), Proceedings of European Conference on Machine Learning, 1998, pp. 137–142.

[38] R. Johnson, T. Zhang, Effective use of word order for text categorization with convolutional neural networks, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, ACL, 2015, pp. 103–112. doi:10.3115/v1/N15-1011.

[39] H. Chen, M. Sun, C. Tu, Y. Lin, Z. Liu, Neural sentiment classification with user and product attention, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, ACL, 2016, pp. 1650–1659. doi:10.18653/v1/D16-1171.

[40] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, ACL, 2016, pp. 1480–1489. doi:10.18653/v1/N16-1174.

[41] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, B. Xu, Attention-based bidirectional long short-term memory networks for relation classification, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL, 2016, pp. 207–212. doi:10.18653/v1/P16-2034.

[42] S. Dumais, H. Chen, Hierarchical classification of web content, in: Proceedings of the International ACM SIGIR Conference on Research and Development of Information Retrieval, ACM, 2000, pp. 256–263. doi:10.1145/345508.345593.

[43] J. M. Moyano, E. L. Gibaja, K. J. Cios, S. Ventura, Review of ensembles of multi-label classifiers: Models, experimental study and prospects, Information Fusion 44 (2018) 33–45. doi:10.1016/j.inffus.2017.12.001.

[44] J. Lu, L. Du, M. Liu, J. Dipnall, Multi-label few/zero-shot learning with knowledge aggregated from multiple label graphs, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, ACL, 2020, pp. 2935–2943. doi:10.18653/v1/2020.emnlp-main.235.

[45] J. Zhou, C. Ma, D. Long, G. Xu, N. Ding, H. Zhang, P. Xie, G. Liu, Hierarchy-aware global model for hierarchical text classification, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL, 2020, pp. 1106–1117. doi:10.18653/v1/2020.acl-main.104.

[46] H. Chen, Q. Ma, Z. Lin, J. Yan, Hierarchy-aware label semantics matching network for hierarchical text classification, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL, 2021, pp. 4370–4379. doi:10.18653/v1/2021.acl-long.337.

[47] Z. Deng, H. Peng, D. He, J. Li, S. Y. Philip, Htcinfomax: A global model for hierarchical text classification via information maximization, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, ACL, 2021, pp. 3259–3265. doi:10.18653/v1/2021.naacl-main.260.

[48] J. Lanchantin, A. Sekhon, Y. Qi, Neural message passing for multi-label classification, in: Proceedings of Joint European Conference on Machine Learn-

ing and Knowledge Discovery in Databases, Springer, 2019, pp. 138–163. doi:10.1007/978-3-030-46147-8_9.

[49] Z. Wang, P. Wang, L. Huang, X. Sun, H. Wang, Incorporating hierarchy into text encoder: A contrastive learning approach for hierarchical text classification, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL, 2022, pp. 7109–7119. doi:10.48550/arXiv.2203.03825.

[50] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Transactions on Neural Networks 20 (1) (2008) 61–80. doi:10.1109/TNN.2008.2005605.

[51] S. Gopal, Y. Yang, Recursive regularization for large-scale classification with hierarchical and graphical dependencies, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2013, pp. 257–265. doi:10.1145/2487575.2487644.

[52] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, Q. Yang, Large-scale hierarchical text classification with recursively regularized deep graph-cnn, in: Proceedings of the World Wide Web Conference, ACM, 2018, pp. 1063–1072. doi:10.1145/3178876.3186005.

[53] Y. Yu, Z. Sun, C. Sun, W. Liu, Hierarchical multilabel text classification via multitask learning, in: Proceedings of the IEEE International Conference on Tools for Artificial Intelligence, IEEE, 2021, pp. 1138–1143. doi:10.1109/ICTAI52525.2021.00180.

[54] R. Wang, S. Long, X. Dai, S. Huang, J. Chen, et al., Meta-lmtc: Meta-learning for large-scale multi-label text classification, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, ACL, 2021, pp. 8633–8646. doi:10.18653/v1/2021.emnlp-main.679.

[55] Y. Mao, J. Tian, J. Han, X. Ren, Hierarchical text classification with reinforced label assignment, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, ACL, 2019, pp. 445–455. doi:10.18653/v1/D19-1042.

[56] S. Chatterjee, A. Maheshwari, G. Ramakrishnan, S. N. Jagaralpudi, Joint learning of hyperbolic label embeddings for hierarchical multi-label classification, in: Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, ACL, 2021, pp. 2829–2841. doi:10.48550/arXiv.2101.04997.

[57] B. Chen, X. Huang, L. Xiao, Z. Cai, L. Jing, Hyperbolic interaction model for hierarchical multi-label classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 7496–7503.

[58] O. Ganea, G. Bécigneul, T. Hofmann, Hyperbolic neural networks, Advances in neural information processing systems 31 (2018).

[59] M. Nickel, D. Kiela, Poincaré embeddings for learning hierarchical representations, Advances in neural information processing systems 30 (2017).

[60] D. Chai, W. Wu, Q. Han, F. Wu, J. Li, Description based text classification with reinforcement learning, in: Proceedings of the International Conference on Machine Learning, PMLR, 2020, pp. 1371–1382.

[61] J. Y. Hang, M. L. Zhang, Collaborative learning of label semantics and deep label-specific features for multi-label classification, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021). doi:10.1109/TPAMI.2021.3136592.

[62] G. E. Hinton, A. Krizhevsky, S. D. Wang, Transforming auto-encoders, in: Proceedings of the International Conference on Artificial Neural Networks, Springer, 2011, pp. 44–51. doi:10.1007/978-3-642-21735-7_6.

[63] S. Sabour, N. Frosst, G. E. Hinton, Dynamic routing between capsules, in: Advances in Neural Information Processing Systems, 2017, pp. 3859—-3869. doi:10.5555/3294996.3295142.

[64] W. Zhao, H. Peng, S. Eger, E. Cambria, M. Yang, Towards scalable and reliable capsule networks for challenging nlp applications, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL, 2019, pp. 1549–1559. doi:10.18653/v1/P19-1150.

[65] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, G. Eryiğit, SemEval-2016 task 5: Aspect based sentiment analysis, in: Proceedings of the International Workshop on Semantic Evaluation,

2016, pp. 19–30. doi:10.18653/v1/S16-1002.
URL https://aclanthology.org/S16-1002

[66] C. N. Silla, A. A. Freitas, A survey of hierarchical classification across different application domains, Data Mining and Knowledge Discovery 22 (1) (2011) 31–72. doi:10.1007/s10618-010-0175-9.

[67] P. Saigal, V. Khanna, Multi-category news classification using support vector machine based classifiers, SN Applied Sciences 2 (3) (2020) 1–12.

[68] Y. He, J. Li, Y. Song, M. He, H. Peng, Time-evolving text classification with deep neural networks, in: Proceedings of the International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 2241–2247. doi:10.24963/ijcai.2018/310.

[69] J. Liu, W. C. Chang, Y. Wu, Y. Yang, Deep learning for extreme multi-label text classification, in: Proceedings of the International ACM SIGIR Conference on Research and Development of Information Retrieval, ACM, 2017, pp. 115–124. doi:10.1145/3077136.3080834.

[70] T. Fagni, F. Sebastiani, Selecting negative examples for hierarchical text classification: An experimental comparison, Journal of the American Society of Information Science and Technology 61 (11) (2010) 2256–2265. doi:10.5555/1869064.1869084.

[71] S. Banerjee, C. Akkaya, F. Perez-Sorrosal, K. Tsioutsiouliklis, Hierarchical transfer learning for multi-label text classification, in: Proceedings of the An-

nual Meeting of the Association for Computational Linguistics, ACL, 2019, pp. 6295–6300. doi:10.18653/v1/P19-1633.

[72] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (3) (1995) 273–297.

[73] M. Krendzelak, F. Jakab, Hierarchical text classification using cnns with local classification per parent node approach, in: Proceedings of the International Conference on Emerging eLearning Technologies and Applications, IEEE, 2019, pp. 460–464. doi:10.1109/ICETA48886.2019.9040022.

[74] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Computation 1 (4) (1989) 541–551. doi:10.1162/neco.1989.1.4.541.

[75] K. Shimura, J. Li, F. Fukumoto, Hft-cnn: Learning hierarchical category structure for multi-label short text categorization, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, ACL, 2018, pp. 811–816. doi:10.18653/v1/D18-1093.

[76] J. Wehrmann, R. Cerri, R. Barros, Hierarchical multi-label classification networks, in: Proceedings of the International Conference on Machine Learning, PMLR, 2018, pp. 5075–5084.

[77] W. Huang, E. Chen, Q. Liu, Y. Chen, Z. Huang, Y. Liu, Z. Zhao, D. Zhang, S. Wang, Hierarchical multi-label text classification: An attention-based recurrent network approach, in: Proceedings of the ACM International Confer-

ence on Information and Knowledge Management, ACM, 2019, pp. 1051–1060. doi:10.1145/3357384.3357885.

[78] X. Zhang, J. Xu, C. Soh, L. Chen, La-hcn: Label-based attention for hierarchical multi-label text classification neural network, Expert Systems with Applications 187 (2022) 115922. doi:10.1016/j.eswa.2021.115922.

[79] L. Xu, S. Teng, R. Zhao, J. Guo, C. Xiao, D. Jiang, B. Ren, Hierarchical multi-label text classification with horizontal and vertical category correlations, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, ACL, 2021, pp. 2459–2468. doi:10.18653/v1/2021.emnlp-main.190.

[80] J. Wu, W. Xiong, W. Y. Wang, Learning to learn and predict: A meta-learning approach for multi-label classification, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, ACL, 2019, pp. 4354–4364. doi:10.18653/v1/D19-1444.

[81] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of the International Conference on Learning Representations, 2017, pp. 1–14. doi:10.48550/arXiv.1609.02907.
URL https://openreview.net/forum?id=SJU4ayYgl

[82] J. Fan, Y. Yu, Z. Wang, Partial label learning with competitive learning graph neural network, Engineering Applications of Artificial Intelligence 111 (2022) 104779. doi:10.1016/j.engappai.2022.104779.

[83] J. Jo, J. Baek, S. Lee, D. Kim, M. Kang, S. J. Hwang, Edge representation learning with hypergraphs, in: Advances in Neural Information Processing Systems, 2021, pp. 1–13. doi:10.48550/arXiv.2106.15845.
URL https://openreview.net/forum?id=vwgsqRorzz

[84] J. Luo, C. Li, Q. Fan, Y. Liu, A graph convolutional encoder and multi-head attention decoder network for tsp via reinforcement learning, Engineering Applications of Artificial Intelligence 112 (2022) 104848. doi:10.1016/j.engappai.2022.104848.

[85] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, J. Leskovec, Hierarchical graph representation learning with differentiable pooling, in: Advances in Neural Information Processing Systems, 2018, pp. 4805—-4815. doi:10.48550/arXiv.1806.08804.

[86] Y. Ma, S. Wang, C. C. Aggarwal, J. Tang, Graph convolutional networks with eigenpooling, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2019, pp. 723–731. doi:10.1145/3292500.3330982.

[87] C. Gallicchio, A. Micheli, Graph echo state networks, in: International Joint Conference on Neural Networks, IEEE, 2010, pp. 1–8. doi:10.1109/IJCNN.2010.5596796.

[88] Y. Li, R. Zemel, M. Brockschmidt, D. Tarlow, Gated graph sequence neural networks, in: Proceedings of the International Conference on Learning Representations, 2016, pp. 1–20. doi:10.48550/arXiv.1511.05493.

[89] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S. Y. Philip, A comprehensive survey on graph neural networks, IEEE transactions on neural networks and learning systems 32 (1) (2020) 4–24.

[90] C. Gallicchio, A. Micheli, Tree echo state networks, Neurocomputing 101 (2013) 319–337. doi:10.1016/j.neucom.2012.08.017.

[91] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and deep locally connected networks on graphs, in: Proceedings of the International Conference on Learning Representations, 2014, pp. 1–14. doi:10.48550/arXiv.1312.6203.
URL https://openreview.net/forum?id=DQNsQf-UsoDBa

[92] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: Proceedings of the International Conference on Learning Representations, 2018, pp. 1–12. doi:10.48550/arXiv.1710.10903.
URL https://openreview.net/forum?id=rJXMpikCZ

[93] C. Xiang, L. Zhang, Y. Tang, W. Zou, C. Xu, Ms-capsnet: A novel multi-scale capsule network, IEEE Signal Processing Letter 25 (12) (2018) 1850–1854. doi:10.1109/LSP.2018.2873892.

[94] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research 15 (56) (2014) 1929–1958. doi:10.5555/2627435.2670313.

[95] J. Gu, V. Tresp, Improving the robustness of capsule networks to image affine transformations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2020, pp. 7285–7293. doi:10.1109/CVPR42600.2020.00731.

[96] T. Jeong, Y. Lee, H. Kim, Ladder capsule network, in: Proceedings of the International Conference on Machine Learning, PMLR, 2019, pp. 3071–3079.

[97] W. Huang, F. Zhou, Da-capsnet: Dual attention mechanism capsule network, Scientific Reports 10 (1) (2020) 1–13. doi:10.1038/s41598-020-68453-w.

[98] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, Z. Zhao, Investigating capsule networks with dynamic routing for text classification, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, ACL, 2018, pp. 3110–3119. doi:10.18653/v1/D18-1350.

[99] R. Aly, S. Remus, C. Biemann, Hierarchical multi-label classification of text with capsule networks, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics: Stud. Res. Workshop, ACL, 2019, pp. 323–330. doi:10.18653/v1/P19-2045.

[100] H. Peng, J. Li, S. Wang, L. Wang, Q. Gong, R. Yang, B. Li, S. Y. Philip, L. He, Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification, IEEE Transactions on Knowledge and Data Engineering 33 (6) (2019) 2505–2519. doi:10.1109/TKDE.2019.2959991.

[101] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2015, pp. 2267–2273. doi:10.5555/2886521.2886636.

[102] B. Wang, X. Hu, P. Li, S. Y. Philip, Cognitive structure learning model for hierarchical multi-label text classification, Knowledge-Based Systems 218 (2021) 106876. doi:10.1016/j.knosys.2021.106876.

[103] Y.-N. Chen, D. Hakkani-Tür, X. He, Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2016, pp. 6045–6049. doi:10.1109/ICASSP.2016.7472838.

[104] Y. Shen, X. He, J. Gao, L. Deng, G. Mesnil, A latent semantic model with convolutional-pooling structure for information retrieval, in: Proceedings of the ACM International Conference on Information and Knowledge Management, 2014, pp. 101–110.

[105] R. Puri, B. Catanzaro, Zero-shot text classification with generative language models, arXiv preprint arXiv:1912.10165 (2019).

[106] Y. Hou, W. Che, Y. Lai, Z. Zhou, Y. Liu, H. Liu, T. Liu, Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2020, pp. 1381–1393. doi:10.18653/v1/2020.acl-main.128.

[107] J. D. Lafferty, A. McCallum, F. C. N. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the International Conference on Machine Learning, 2001, p. 282–289.

[108] K. Halder, A. Akbik, J. Krapac, R. Vollgraf, Task-aware representation of sentences for generic text classification, in: Proceedings of the International Conference on Computational Linguistics, 2020, pp. 3202–3213. doi:10.18653/v1/2020.coling-main.285.

[109] Q. Luo, L. Liu, Y. Lin, W. Zhang, Don't miss the labels: Label-semantic augmented meta-learner for few-shot text classification, in: Findings of the Association for Computational Linguistics, 2021, pp. 2773–2782. doi:10.18653/v1/2021.findings-acl.245.

[110] Q.-W. Zhang, X. Zhang, Z. Yan, R. Liu, Y. Cao, M.-L. Zhang, Correlation-guided representation for multi-label text classification, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2021, pp. 3363–3369. doi:10.24963/ijcai.2021/463.

[111] N. Pappas, J. Henderson, GILE: A generalized input-label embedding for text classification, Transactions of the Association for Computational Linguistics (2019) 139–155doi:10.1162/tacl_a_00259.

[112] L. Xiao, X. Huang, B. Chen, L. Jing, Label-specific document representation for multi-label text classification, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2019, pp. 466–475. doi:10.18653/v1/D19-1044.

[113] L. Cai, Y. Song, T. Liu, K. Zhang, A hybrid bert model that incorporates label semantics via adjustive attention for multi-label text classification, IEEE Access 8 (2020) 152183–152192. doi:10.1109/ACCESS.2020.3017382.

[114] X. Zhang, Q. W. Zhang, Z. Yan, R. Liu, Y. Cao, Enhancing label correlation feedback in multi-label text classification via multi-task learning, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL, 2021, pp. 1190–1200. doi:10.18653/v1/2021.findings-acl.101.

[115] C. Yu, Y. Shen, Y. Mao, L. Cai, Constrained sequence-to-tree generation for hierarchical text classification, in: Proceedings of the International ACM SIGIR Conference on Research and Development of Information Retrieval, ACM, 2022, pp. 1865–1869. doi:10.48550/arXiv.2204.00811.

[116] W. Zhang, X. Li, Y. Deng, L. Bing, W. Lam, A survey on aspect-based sentiment analysis: Tasks, methods, and challenges (2022). doi:10.48550/ARXIV.2203.01054.
URL https://arxiv.org/abs/2203.01054

[117] J. Bu, L. Ren, S. Zheng, Y. Yang, J. Wang, F. Zhang, W. Wu, ASAP: A Chinese review dataset towards aspect category sentiment analysis and rating prediction, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 2069–2079. doi:10.18653/v1/2021.naacl-main.167.
URL https://aclanthology.org/2021.naacl-main.167

[118] M. Hu, S. Zhao, L. Zhang, K. Cai, Z. Su, R. Cheng, X. Shen, CAN: Constrained attention networks for multi-aspect sentiment analysis, in: Proceedings of the

Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019, pp. 4601–4610. doi:10.18653/v1/D19-1467.
URL https://aclanthology.org/D19-1467

[119] H. Wan, Y. Yang, J. Du, Y. Liu, K. Qi, J. Z. Pan, Target-aspect-sentiment joint detection for aspect-based sentiment analysis, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 9122–9129. doi:10.1609/aaai.v34i05.6447.
URL https://ojs.aaai.org/index.php/AAAI/article/view/6447

[120] H. Cai, Y. Tu, X. Zhou, J. Yu, R. Xia, Aspect-category based sentiment analysis with hierarchical graph convolutional network, in: Proceedings of the International Conference on Computational Linguistics, 2020, pp. 833–843. doi:10.18653/v1/2020.coling-main.72.
URL https://aclanthology.org/2020.coling-main.72

[121] M. Schmitt, S. Steinheber, K. Schreiber, B. Roth, Joint aspect and polarity classification for aspect-based sentiment analysis with end-to-end neural networks, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2018, pp. 1109–1114. doi:10.18653/v1/D18-1139.
URL https://aclanthology.org/D18-1139

[122] J. Liu, Z. Teng, L. Cui, H. Liu, Y. Zhang, Solving aspect category sentiment analysis as a text generation task, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2021, pp. 4406–4416.

doi:10.18653/v1/2021.emnlp-main.361.
URL https://aclanthology.org/2021.emnlp-main.361

[123] K. S. Kalyan, A. Rajasekharan, S. Sangeetha, Ammus : A survey of transformer-based pretrained models in natural language processing (2021). doi:10.48550/ARXIV.2108.05542.
URL https://arxiv.org/abs/2108.05542

[124] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the International Conference on Learning Representations, 2010, pp. 1–8. doi:10.5555/3104322.3104425.
URL https://openreview.net/forum?id=rkb15iZdZB

[125] M. L. Zhang, L. Wu, Lift: Multi-label learning with label-specific features, IEEE transactions on pattern analysis and machine intelligence 37 (1) (2014) 107–120.

[126] J. Huang, G. Li, Q. Huang, X. Wu, Learning label specific features for multi-label classification, in: Proceedings of the IEEE International Conference on Data Mining, 2015, pp. 181–190. doi:10.1109/ICDM.2015.67.

[127] X. Y. Jia, S. S. Zhu, W. W. Li, Joint label-specific features and correlation information for multi-label learning, Journal of Computer Science and Technology 35 (2) (2020) 247–258.

[128] A. L. Maas, A. Y. Hannun, A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proceedings of the International Conference on

Machine Learning Workshop Deep Learn. Audio Speech Lang. Process., 2013, pp. 1–6. doi:10.1.1.693.1422.

[129] R. M. Pereira, Y. M. Costa, C. N. Silla, Handling imbalance in hierarchical classification problems using local classifiers approaches, Data Mining and Knowledge Discovery 35 (4) (2021) 1564–1621. doi:10.1007/s10618-021-00762-8.

[130] T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, IEEE, 2017, pp. 2980–2988. doi:10.1109/ICCV.2017.324.

[131] Y. Yang, An evaluation of statistical approaches to text categorization, Information Retrieval 1 (1) (1999) 69–90. doi:10.1023/A:1009982220290.

[132] J. L. Elman, Finding structure in time, Cognitive Science 14 (2) (1990) 179–211. doi:10.1207/s15516709cog1402_1.

[133] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation 9 (8) (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[134] L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, Z. Jin, How transferable are neural networks in NLP applications?, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2016, pp. 479–489. doi:10.18653/v1/D16-1046.

[135] K. Cho, B. van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder–decoder approaches, in: Proceedings of

the Workshop on Syntax, Semantics and Structure in Statistical Translation, 2014, pp. 103–111. doi:10.3115/v1/W14-4012.

[136] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, ACL, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[137] M. T. Nguyen, D. T. Le, L. Le, Transformers-based information extraction with limited data for domain-specific business documents, Engineering Applications of Artificial Intelligence 97 (2021) 104100. doi:10.1016/j.engappai.2020.104100.

[138] H. A. Uymaz, S. K. Metin, Vector based sentiment and emotion analysis from text: A survey, Engineering Applications of Artificial Intelligence 113 (2022) 104922. doi:10.1016/j.engappai.2022.104922.

[139] I. Beltagy, K. Lo, A. Cohan, Scibert: A pretrained language model for scientific text, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, ACL, 2019, pp. 3615–3620. doi:10.18653/v1/D19-1371.

[140] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations, 2015, pp. 1–15. doi:10.48550/arXiv.1412.6980.
URL https://openreview.net/forum?id=8gmWwjFyLj

[141] G. Abuselidze, Modern challenges of monetary policy strategies: inflation and devaluation influence on economic development of the country, Academy of Strategic Management Journal 18 (4) (2019) 1–10.

[142] J. Wanner, L. V. Herm, K. Heinrich, C. Janiesch, P. Zschech, White, grey, black: Effects of xai augmentation on the confidence in ai-based decision support systems, in: Proceedings of the International Conference on Information Systems, 2020, pp. 2089–2097.

[143] A. Makkuva, S. Oh, S. Kannan, P. Viswanath, Learning in gated neural networks, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2020, pp. 3338–3348.

[144] R. Dey, F. M. Salem, Gate-variants of gated recurrent unit (gru) neural networks, in: Proceedings of the IEEE International Midwest Symposium on Circuits and Systems, 2017, pp. 1597–1600. doi:10.1109/MWSCAS.2017.8053243.

[145] X. Li, Z. Li, H. Xie, Q. Li, Merging statistical feature via adaptive gate for improved text classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 13288–13296.

[146] S. Ramasinghe, C. Athuraliya, S. H. Khan, A context-aware capsule network for multi-label classification, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018, pp. 1–9.

[147] M. K. Patrick, A. F. Adekoya, A. A. Mighty, B. Y. Edward, Capsule networks – a survey, Journal of King Saud University - Computer and Information Sciences 34 (1) (2022) 1295–1310. doi:10.1016/j.jksuci.2019.09.014.

[148] Y. LeCun, L. Bottou, G. B. Orr, K.-R. Müller, Efficient backprop, in: Neural Networks: Tricks of the Trade, Springer, 1998, pp. 9–50.

[149] S. Wang, L. Thompson, M. Iyyer, Phrase-bert: Improved phrase embeddings from bert with an application to corpus exploration, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2021, pp. 10837–10851.

[150] A. Cohan, S. Feldman, I. Beltagy, D. Downey, D. Weld, SPECTER: Document-level representation learning using citation-informed transformers, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2020, pp. 2270–2282. doi:10.18653/v1/2020.acl-main.207.

[151] S. Mysore, A. Cohan, T. Hope, Multi-vector models with textual guidance for fine-grained scientific document similarity, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2022, pp. 4453–4470.
URL https://aclanthology.org/2022.naacl-main.331

[152] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, et al., Exploring the limits of transfer learning with a unified text-to-text transformer., Journal of Machine Learning Research 21 (140) (2020) 1–67.

[153] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mpnet: Masked and permuted pre-training for language understanding, Advances in Neural Information Processing Systems 33 (2020) 16857–16867.

[154] S. Brody, U. Alon, E. Yahav, How attentive are graph attention networks?, in: Proceedings of the International Conference on Learning Representations, 2021, pp. 1–26.

[155] I. Androutsopoulos, D. Galanis, S. Manandhar, H. Papageorgiou, J. Pavlopoulos, M. Pontiki, Semeval-2015 task 12 (May 2014).
URL https://alt.qcri.org/semeval2015/task12/

[156] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: Proceedings of the International Conference on Learning Representations, 2019, pp. 1–18.
URL https://openreview.net/forum?id=Bkg6RiCqY7

[157] R. Shimizu, Y. Mukuta, T. Harada, Hyperbolic neural networks++, in: Proceedings of the International Conference on Learning Representations, 2021, pp. 1–25.
URL https://openreview.net/forum?id=Ec85b0tUwbA

[158] S. Dai, Z. Gan, Y. Cheng, C. Tao, L. Carin, J. Liu, APo-VAE: Text generation in hyperbolic space, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 416–431. doi:10.18653/v1/2021.naacl-main.36.

[159] Q. Liu, M. Nickel, D. Kiela, Hyperbolic graph neural networks, in: Proceedings of Advances in Neural Information Processing Systems, Vol. 32, 2019, pp. 1–12.

[160] I. Chami, Z. Ying, C. Ré, J. Leskovec, Hyperbolic graph convolutional neural networks, in: Proceedings of Advances in Neural Information Processing Systems, Vol. 32, 2019, pp. 1–12.

[161] B. Chen, X. Huang, L. Xiao, L. Jing, Hyperbolic capsule networks for multi-label classification, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3115–3124. doi:10.18653/v1/2020.acl-main.283.

[162] T. Singh, M. Kumari, Role of text pre-processing in twitter sentiment analysis, Procedia Computer Science 89 (2016) 549–554.

[163] S. Zhuang, G. Zuccon, Dealing with typos for BERT-based passage retrieval and ranking, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2021, pp. 2836–2842. doi:10.18653/v1/2021.emnlp-main.225.

[164] S. Kobayashi, Contextual augmentation: Data augmentation by words with paradigmatic relations, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 452–457. doi:10.18653/v1/N18-2072.

[165] Z. Xie, S. Wang, J. Li, D. Levy, A. Nie, D. Jurafsky, A. Ng, Data noising as smoothing in neural network language models, in: Proceedings of the International Conference on Learning Representations, 2017, pp. 1–12.

[166] J. Wei, K. Zou, EDA: Easy data augmentation techniques for boosting performance on text classification tasks, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2019, pp. 6382–6388. doi:10.18653/v1/D19-1670.

[167] S. Ren, J. Zhang, L. Li, X. Sun, J. Zhou, Text AutoAugment: Learning compositional augmentation policy for text classification, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2021, pp. 9029–9043. doi:10.18653/v1/2021.emnlp-main.711.

국문초록

계층적 문서 분류는 다양한 분야의 실제 산업의 자연어 처리 관련 과업에 적용될 수 있어 큰 관심을 받고 있는 과업이다. 딥러닝의 발전과 함께 자연어 처리 분야에서도 딥러닝 기반 기법들이 좋은 성능을 기록하고 있으며, 계층적 문서 분류 역시 딥러닝 기반 기법이 기존 기법 대비 최고 성능을 거두고 있다. 그러나 기존 연구는 라벨의 계층 구조에 대한 분석을 효과적으로 수행하거나 계층 구조로는 표현되지 않는 라벨 간 관계의 도출에 집중할 뿐 이 두 접근을 결합한 연구는 많지 않다. 이에 본 논문은 계층 구조를 효과적으로 표현할 뿐만 아니라 내재된 라벨 간 관계 역시 학습하여 분류에 반영하는 딥러닝 기반 기법인 그래프 어텐션 캡슐망(graph attention capsule network for hierarchical text classification, GACaps-HTC)를 제안한다. 그래프 신경망의 일종인 그래프 어텐션 신경망은 문서에서 추출한 표현에 라벨의 계층 구조에 대한 정보를 주입하기 위해 사용되며, 캡슐망은 임의의 두 라벨 사이의 관계를 학습함과 동시에 각 라벨에 대한 분류 확률을 추론하기 위해 활용된다. 본 논문이 제안하는 기법은 계층적 문서 분류 과업이 가지는 라벨 불균형 문제를 해소하기 위한 손실함수로 학습되며 과업에 특화된 다양한 후처리 방법을 도입한다. 계층적 문서 분류 성능을 평가하기 위해 많이 사용되는 두 개의 데이터 셋으로 수행한 실험의 결과, 제안 기법은 기존 기법 대비 성능을 향상시킴을 확인할 수 있었으며, 제안 기법의 각 요소는 해당 성능 향상에 기여함을 확인하였다.

또한, 본 논문은 라벨의 명칭 혹은 문서형 설명을 통해 캡슐망의 연결 계수 초기화 및 갱신을 수행하는 의미 기반 동적 라우팅 알고리즘(semantic-aware dynamic routing algorithm)을 제안한다. 캡슐망 내 두 캡슐 사이의 연결 계수는 두 캡슐 내의 정보 간 유사성을 표현하기에 라벨의 명칭 혹은 문서형 설명에서 추출한 표현 사이의 유사성으

로 이를 초기화한다. 실험 결과 해당 알고리즘으로 라벨 정보를 주입하는 방법은 다른 방식으로 GACaps-HTC에 라벨 정보를 주입하는 방법에 비해 좋은 성능을 거두었으며, 제안 알고리즘을 활용할 경우 기존 동적 라우팅 알고리즘에 비해 빠르게 학습이 수렴함을 확인하였다.

마지막으로 본 논문은 제안 기법의 확장 가능성을 평가하고자 속성 카테고리 감성 분석을 계층적 문서 분류로 치환하여 제안 기법을 적용한다. 감성 분석 데이터 셋으로 수행한 실험의 결과, 문서 내 의미적 정보에 대한 분석은 물론 감성적 정보에 대한 분석을 위해 제안 기법을 적용할 수 있음을 확인하였다.

주요어: 계층적 문서 분류, 그래프 신경망, 캡슐망, 딥러닝, 자연어 처리

학번: 2017-28575

감사의 글

대학원 과정을 마무리하며 이렇게 저의 연구를 정리한 학위논문을 작성하고 제출할 생각을 하니 감회가 새롭습니다. 이런 결과가 있기까지 많은 분들의 조언과 가르침이 있었기에, 그리고 많은 격려와 응원이 있었기에 그 과정이 즐거웠고 알찼습니다. 저에게 도움을 주신 많은 분들께 정말로 감사했음을 이번 기회에 전달드립니다.

우선 학부 때부터 대학원 과정까지 저를 지도해주시고 더 나은 연구자로서 나아갈 수 있게 해주신 박종헌 교수님께 진심으로 감사드립니다. 교수님의 가르침 덕분에 넓은 산업공학이라는 학문에서 제가 어떤 것을 공부하고 연구하고 싶은지 깨달을 수 있었으며, 어떻게 하면 더 나은 질문을 던지고 더 좋은 답변을 할 수 있는지 배웠습니다. 저를 성장시켜주었던 여러 연구 분야의 많은 경험들을 선물해주셨으며, 연구를 다듬어가며 스스로를 발전시킬 수 있는 태도를 가르쳐주셨습니다. 저의 졸업이 배움의 끝이라는 생각을 하지 않고, 교수님께 배운대로 계속 질문하며 배우고 성장할 수 있도록 하겠습니다. 앞으로 건강하시길, 그리고 행복한 일들만 가득하시길 바라겠습니다.

그리고 저희 기수에 특별한 애정을 가져주시고, 저와 많은 대화를 나눠주시며 저의 진로에 대해서도 조언해주시던 이경식 교수님, 훌륭한 학생의 마음가짐에 대한 가르침을 주신 이재욱 교수님, 저의 연구가 어떻게 가치를 가질 수 있을지 고민해보라는 조언을 해주신 조성준 교수님, 바쁘신 와중에 지도위원을 맡아주셔서 감사드립니다. 교수님들 덕분에 스스로 저의 연구에 대해 더 많이 고민할 수 있었고, 더 나은 결과를 낼 수 있었습니다. 현재는 은퇴하셨지만 저에게 산업공학의 본질에 대해 가르침을 주신 박진우 교수님, 오형식 교수님, 그리고 박용태 교수님께도 이 기회에 감사의 마음 전달드립니다. 학부 과정 중 면담을 통해 저의 고민을 들어주시던 장우진 교수님, 저에게는 생소하던 주제에 대해서 가르침을 주시고 식견을 넓혀주신 문일경 교수님, 이덕주 교수님, 그리고 윤명환 교수님, 언제나 열정이 가득한 강의를 해주시던 홍성필 교수님

감사드립니다. 아쉽게 강의를 통해 가르침을 받을 기회가 없었던 홍유석 교수님, 박우진 교수님, 박건수 교수님, 그리고 이성주 교수님, 언젠가 교수님들과 말씀나누며 성장할 수 있는 기회가 있길 바라겠습니다.

연구실에서 함께 생활하며 고민을 들어줄 귀가 되어주기도, 스승이 되어주기도 했던 많은 선배님들과 동기들, 후배들에게도 감사함을 전합니다. 처음 뵀을 때부터 환하게 웃으며 환영해주신 용석이형과 재홍이형, 언제나 멋진 모습으로 연구실을 환하게 비춰주시던 수빈이형과 중균이형, 그리고 저와 함께 입학해 동고동락했던 상우형 모두 감사합니다. 저에게는 스승같았던 재석이형, 인범이형, 그리고 승욱이형 덕분에 많은 것을 배울 수 있었고 연구자로서 성장할 수 있었습니다. 시시콜콜한 궁금증까지 친절하게 답변해주셔서 감사드립니다. 이제 스스로의 여정을 시작하신 종권이형과 교윤이형, 학부 때부터 저에게 좋은 선배였던 두분에게 감사를 전하며, 앞으로 형들이 걷는 길이 눈부시길 응원하고 있겠습니다. 그리고 철이 없던 모습도 이해해주시며 선배임에도 친구처럼 저와 지내주신 종혁이형과 석현이형, 형들 덕분에 대학원 생활이 너무나도 즐거웠습니다. 현재 박사 과정의 마무리를 위해 노력 중이신 모두들, 석현이형과 저의 첫 프로젝트 팀장이셨던 문정이형, 함께 너무나 생소했던 주제에 대해 연구했었던 도균이형, 그리고 후배임에도 너무나 배울 모습이 많았던 완이 모두 좋은 결과가 있을 것이라고 믿고 응원합니다. 저의 마지막 프로젝트에서 미숙했던 저를 따라 다양한 일들을 하느라 고생했던 유민이, 석기, 그리고 유리님 덕분에 프로젝트 잘 마무리할 수 있었고, 즐겁게 진행할 수 있었으며, 그 결과로 학위 논문까지 작성할 수 있었습니다. 그리고 마지막 프로젝트에서, 그리고 그 외에도 저와 친하다는 이유만으로 제가 누군가에게 부탁하기 어려운 부탁들을 모두 들어줘야했던 동현이형에게도 감사의 마음을 전합니다. 저보다 먼저 졸업하셔서 각자의 분야에서 멋진 모습을 보여주고 계신 희웅이형, 보형이형, 성은, 설아, 재희, 뒷자리였던 해성이와 옆자리였던 창진이까지, 훌륭했던 여러분과 함께 얘기를 나누며 교류할 수 있던 시간은 저에게는 소중한 경험이었습니다. 마지막으로 연구실에서 지금 이 시간에도 노력 중일 민재, 지문님, 은채, 재룡이,

희웅이와 이경이에게는 함께 보낸 시간이 저에게는 즐거웠음을 꼭 말씀드리고 싶었고, 저와 보낸 시간이 여러분께도 그렇게 기억되었으면 좋겠습니다. 학부부터 대학원 과정을 하는 기간동안 힘이 되어준 나의 친구들, 민재, 세호형, 세영이, 범호형, 범국이형, 상엽이형에게도 감사의 마음을 전합니다

사랑하는 저의 가족들이 계속 지지해주고 응원해준 덕분에 이렇게 잘 마무리할 수 있었습니다. 선배 연구자로서 길잡이가 되어주신 저의 어머니, 어머니의 말씀은 언제나 큰 가르침이 되었습니다. 언제나 저에게 웃는 모습을 보여주시던 저의 아버지, 아버지의 말씀에 있던 따스함 덕분에 저도 함께 웃을 수 있었습니다. 그리고 저를 자랑스러워하던 저의 형, 형은 언제나 나에게 닮고 싶은 사람이었음을 이 기회에 전합니다. 학사모를 쓴 모습을 보며 자랑스러워하실 저의 할머니들, 그리고 하늘에서 언제나 저를 응원하고 계실 저의 할아버지들에게도 너무나도 감사드립니다. 모두가 주신 사랑 덕분에 지금의 제가 있을 수 있었고, 모두의 응원 덕분에 이렇게 결실을 맺을 수 있었습니다. 제가 받은 사랑 다시 가족 모두에게 드리며 앞으로 더욱 멋진 아들, 동생, 그리고 손주가 되겠습니다. 마지막으로 함께 미래를 그려나가기로 약속한 지현이와 따스하게 가족으로 맞아주시고 저를 응원해주셨던 어머님, 아버님께도 감사드립니다. 스스로를 의심하며 불안해하던 저에게 확신을 주며 응원해주는 지현이가 있었기에 계속 나아갈 수 있었고, 이토록 빛나는 사람에게 어울리는 사람이 되고자 하는 마음 덕분에 더 나은 사람이 될 수 있었습니다. 앞으로 어떤 일이 있어도 함께 행복할 수 있게 노력하고, 함께 있는 매 순간을 소중히 여기겠습니다. 저의 가족 모두에게 진심으로 감사하고, 사랑합니다.

감사할 분들이 너무나 많기에 지면으로는 다 표현하지 못했지만 저에게 힘이 되어 주었던 모든 분들에게 감사의 마음을 전합니다.