DOMAIN-AGNOSTIC DOCUMENT AND QUESTION CLASSIFICATION USING NATURAL LANGUAGE PROCESSING TECHNIQUES

S SUPRAJA

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

A thesis submitted to the Nanyang Technological University

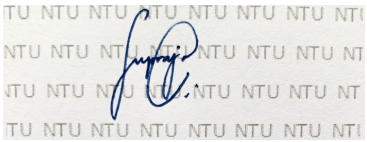
in partial fulfillment of the requirement for the degree of

Doctor of Philosophy

2022

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.



04/05/2022
Date

S Supraja

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

04/05/2022
Date

Andy Khong Wai Hoong

Authorship Attribution Statement

This thesis contains material from 4 papers published/waiting to be published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as the first author.

Chapter 3 is published as S. Supraja, K. Hartman, S. Tatinati, and Andy W. H. Khong, "Toward the automatic labeling of course questions for ensuring their alignment with learning outcomes," in *Proc. 10th Int. Conf. Educational Data Mining (EDM)*, 2017, pp. 56–63.

The contributions of the co-authors are as follows:

- Dr K. Hartman provided the initial project direction.
- I prepared the data for analysis and designed the proposed algorithm.
- Dr K. Hartman and Dr S. Tatinati assisted in the analysis and interpretation of the results.
- I prepared the manuscript drafts. The manuscript was revised by Dr K. Hartman, Dr S. Tatinati, and A/Prof Andy W. H. Khong.

Chapter 4 is published as S. Supraja, S. Tatinati, K. Hartman, and Andy W. H. Khong, "Automatically linking digital signal processing assessment questions to key engineering learning outcomes," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2018, pp. 6996–7000.

The contributions of the co-authors are as follows:

- Dr S. Tatinati provided the initial project direction.

- I designed the experiment and collected the data. A/Prof Andy W. H. Khong assisted in the collection of the data.
- I prepared the data for analysis, developed the proposed algorithm, and analyzed the results.
- I prepared the manuscript drafts. The manuscript was revised by Dr S. Tatinati, Dr K. Hartman, and A/Prof Andy W. H. Khong.

Chapter 5 is published as S. Supraja, Andy W. H. Khong, and S. Tatinati, "Regularized phrase-based topic model for automatic question classification with domain-agnostic class labels," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 29, pp. 3604–3616, 2021.

The contributions of the co-authors are as follows:

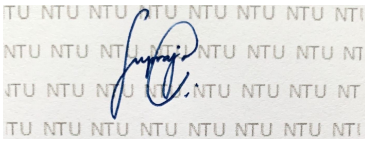
- A/Prof Andy W. H. Khong provided the initial project direction.
- I collected and prepared the data for analysis.
- I formulated the proposed algorithm, conducted experiments, and analyzed the results.
- I prepared the manuscript drafts. The manuscript was revised by A/Prof Andy W. H. Khong and Dr S. Tatinati.

Chapter 6 has been submitted to a journal as S. Supraja and Andy W. H. Khong, "Quad-faceted feature-based graph network for domain-agnostic text classification," *IEEE/ACM Trans. Audio Speech Lang. Proc.*

The contributions of the co-author are as follows:

- A/Prof Andy W. H. Khong provided the initial project direction.
- I collected and prepared the data for analysis.
- I formulated the proposed algorithm, conducted experiments, and analyzed the results.

- I prepared the manuscript drafts. The manuscript was revised by A/Prof Andy W. H. Khong.



04/05/2022	S Supraja
Date	

Acknowledgments

I would like to dedicate this thesis to two special angels: God and my dear parents, without whom I could not have completed this challenging, yet fruitful PhD journey. I have no words to express my gratitude for every single thing my parents have done for my education and for my whole life. I cannot do anything to repay their love, affection, care, and concern for me.

I would like to extend my sincere thanks to my husband, all my family members and friends who have supported me in one way or another.

First and foremost, I would like to sincerely express my gratitude to my advisor A/Prof Andy W. H. Khong for his continuous support and guidance throughout my Ph.D. journey. His constant motivation and encouragement has transformed me tremendously. His advice has always been a great tonic to boost my morale and confidence level.

In addition, I would like to thank my research fellow Dr Sivanagaraja Tatinati and learning science expert Dr Kevin Hartman for their kind mentorship, insights, ideas, and help in my manuscript editing. Special thanks to Dr Sivanagaraja Tatinati for his patience and suggestions to guide my research direction. I would like to thank Dr Jack Sheng Kee from Delta Electronics for his advice.

Last but not least, I would also like to thank my amazing teammates Ng Hongrui Kelvin, Liu Kai, Nguyen Hai Trieu Anh, Divya Venkatraman, Nguyen Quang Hanh, Ho Mun Kit, Cao Zhen, Qiu Wei, Darin Tao Liran, Tan Zhi Wei, and Li Jiawei for their companionship, valuable discussions and feedback throughout the journey.

Supraja

"Not all of us can do great things. But we can do small things with great love."

—Mother Teresa

To my parents and God

Table of Contents

Summary

This thesis addresses the classification of documents and questions to domain-agnostic class labels. Domain refers to the subject matter with which the class labels are associated. Domain-specific document or question classification is commonly applied in articles categorization or in factoid question answering with class labels being defined by subject matter. For instance, considering digital signal processing (DSP) questions, the explicit meaning of the questions will be reflected if the domain-specific class labels consist of *Fourier Transform* or *z-transform*.

In contrast, applications for domain-agnostic document classification include classifying job descriptions into generic skillsets, scientific statements into section types, and sentences into argumentative zone functions. With questions possessing different characteristics, domain-agnostic question classification is applied in information query or dialogue interactions in which the class labels may comprise question types or reasoning capabilities. To enhance the effectiveness of deliberate practice, questions are classified into their respective cognitive complexities for instructors to determine learners' proficiencies. Quite often, in scenarios where the size of the question bank is limited, statistical approaches are adopted for feature extraction. Since domain-agnostic classification takes the implicit substance of a text into account (e.g., learning outcome of the same DSP question irrespective of the content), it relies on a suitable feature extraction process.

This thesis explores the use of topic modeling techniques as feature extractors for questions due to its ability of offering linguistic insights into language patterns by grouping associated words into topics and, thereafter, computing the probabilities of topics occurring in each document. Considering the limitations of employing baseline topic modeling algorithms for automatic question classification (AQC), an algorithm that observes the

effect of pre-processing procedures and word co-occurrence redundancy is proposed. However, the limitation of this method is that it is dataset-specific and requires hand-curated word tagging. To address these shortcomings, a new holistic generalizable regularized phrase-based topic modeling technique is proposed. This technique is driven by the fact that phrases have been shown to be more effective than words to represent questions. Further elements such as nested regular expressions and scaling parameters are being employed to facilitate efficient mapping of questions to class labels.

For documents, the baseline algorithm of graph networks is adopted. This thesis shows that graph networks are suitable since it is important to establish the relationships between documents to better classify them into domain-agnostic categories. In addition, graphs encompass a global perspective compared to conventional deep learning techniques that are both localized and sequential. In the proposed quad-faceted feature-based graph network, this thesis shows that the addition of a new topical layer is vital for observing the impact of topic modeling on generating a meaningful set of features. It also highlights that the use of regular expressions with a domain-agnostic nature is important for co-occurrence statistics while the meaning of a document encapsulated via phrase nodes are crucial for semantic relationships.

List of Figures

List of Tables

List of Abbreviations

Abbreviations	Full name
A-LDA	Asymmetric LDA
ABET	Accreditation Board for Engineering and Technology
ADC	Automatic document classification
AQC	Automatic question classification
BERT	Bidirectional encoder representations from transformers
Bi-LSTM	Bi-directional LSTM
BoW	Bag-of-words
CNN	Convolutional neural network
DSP	Digital signal processing
ELM	Extreme learning machine
GCN	Graph convolutional network
GP	Gaussian process
GRU	Gated recurrent units
KL	Kullback-Leibler
LDA	Latent Dirichlet allocation
LPTM	LDA-based phrase topic model

Abbreviations	**Full name**
LSTM	Long short-term memory network
MDFS	Modified distinguishing feature selector
NER	Named entity recognition
NP	Noun phrase
P-LDA	Phrase-based topic model
PMI	Point-wise mutual information
POS	Parts-of-speech
q-WNTM	Question WNTM
Qu-LDA	Question LDA
RBF	Radial basis function
RNN	Recurrent neural networks
s.TF-IDF	Sorted TF-IDF
SD	Standard deviation
SLFN	Single hidden layer feedforward neural network
SVM	Support vector machine
TF-ICF	Term frequency-inverse class frequency
TF-IDF	Term frequency-inverse document frequency
VP	Verb phrase
W-LDA	Weighted LDA
WNTM	Word network topic model

List of Notations

Symbol	Name
w_i	Word i
p_k	Phrase k
r_k	Regex k
z_j	Topic j
$\mathbf{d}_m$	Concatenated document m
i, j, k, m, n	Generic counters used interchangeably
y_c	Actual class label c
$\widehat{y}$	Predicted class label
N_Q	Total number of questions
N_D	Total number of documents
N_V	Word vocabulary size
N_R	Regex vocabulary size
N_Z	Total number of topics
N_L	Total number of class labels
$\boldsymbol{\Theta}$	Document-topic or question-topic distribution
$\boldsymbol{\Phi}$	Topic-word distribution
$\boldsymbol{\eta}$	Topic-regex distribution
α	Dirichlet prior for document-topic or question-topic distribution
β	Dirichlet prior for topic-word distribution
λ	Dirichlet prior for topic-regex distribution
$\widetilde{q}$	Pre-processed question
$\mathfrak{P}$	Pseudo-question for WNTM and q-WNTM
$\mathbf{q}$	Feature vector from Qu-LDA

Symbol	Name
q	Regex-extracted question
$\mathcal{C}_{p_k}$	C-value for nested phrases
N_{p_k}	Number of times phrase p_k occurs within a corpus
s	Set of N_s phrases that contain p_k as a nested phrase
$N_p^{(s)}$	Number of times each of the phrases in that set s occurs in the corpus
N_{w_i}	Total number of questions in which each word occurs in
$\mathbf{Q}$	Set of question feature vectors
N_w	Number of words per question
Ω_{w_i}	MDFS term weighting per word
$P(\widehat{z}_{j,y_c})$	Regularized topic probability
t_j	Unique value for each topic that is derived from the Newton-Raphson optimization
L_{r_k}	Length of each regex
$\mu_{\mathfrak{N}}$	NP-based regex
$\mu_{\mathcal{V}}$	VP-based regex
$N_{\mu_{\mathfrak{N}}}$	Number of times NP-based regexes occur
$N_{\mu_{\mathcal{V}}}$	Number of times VP-based regexes occur
$\mathcal{C}_{r_k}$	C-value for nested regexes
φ_{r_k}	Scaling parameter for regexes
N_{r_k}	Number of times regex r_k occurs in the entire corpus
S	Set of N_S regexes that contain r_k as a nested regex
$N_r^{(S)}$	Number of times each of the regexes in that set S occurs in the corpus
ρ	Weight ratio of NP- and VP-based regexes
$\mathcal{P}$	Precision
$\mathcal{R}$	Recall
$F1$	F1 score
G	A graph
V	Set of nodes
E	Set of edges

Symbol	Name
$\mathbf{A}$	Adjacency matrix
$\mathbf{D}$	Degree matrix
$\mathbf{W}$	Weight matrix
$\mathbf{H}$	Hidden layer matrix
$\mathbf{F}$	Feature matrix
e	Edge weight
γ	Non-linear activation function
$\mathcal{L}$	Number of layers in GCN
f_l	Number of features for each node in the lth GCN layer
$\mathbf{v}_{\mathbf{p}_i}$	Embedding vector for phrase i
$\mathcal{C}_{p_k}^{(\mathcal{G})}(\mathbf{d}_m)$	Modified C-value for text-phrase relationship
$\mathcal{C}_{r_k}^{(\mathcal{G})}(\mathbf{d}_m)$	Modified C-value for text-regex relationship
ρ_{sem}	Threshold for semantic similarity
ρ_{top}	Threshold for topic convergence consideration
$\mu_{\cos(\theta)}$	Mean of cosine similarity values
$\sigma_{\cos(\theta)}$	Standard deviation among cosine similarity values
$\mu_{KL_{\text{ave}}}$	Mean of average KL-divergence values
σKL_{ave}	Standard deviation among average KL-divergence values
$\mathcal{T}$	Graph tensor
$\mathcal{A}$	Graph adjacency tensor
$\mathcal{H}$	Graph feature tensor
$\Theta_{\text{scl}}(z_j, \mathbf{d}_m)$	Scaled topic probability per text
$\boldsymbol{\gamma}_j$	Weight vector that stores the weights between the hidden and output nodes
b_y	Bias term
L	Number of hidden neurons in ELM
$\mathfrak{p}$	Vector of coefficients in SVM
ϕ	Kernel in SVM
ϵ_j	The parameters that handle the inputs in SVM
σ^2	Kernel bandwidth in GP
κ	Noise covariance in GP

Chapter 1

Introduction

Classification of documents, and in particular questions can be performed in either a domain-specific or domain-agnostic manner. The former requires categories that are in line with the content of the text (explicit relationship) while the latter requires categories that convey the inner meaning of the text (implicit relationship). Classification of documents or questions according to domain-agnostic class labels relies on a suitable feature extraction process. This thesis presents techniques for extracting appropriate features using topic modeling and graph networks for domain-agnostic question and document classification.

1.1 Motivation

Automatic question classification (AQC) has been proposed for several applications and is achieved by defining domain-specific or domain-agnostic class labels. Domain-specific AQC is commonly applied in factoid question answering [1–3] with class labels being defined by subject matter (e.g., science, arts/humanities, business/finance) [4–6]. In contrast, domain-agnostic AQC is applied in information query or dialogue interactions in which the class labels may comprise question types (e.g., true/false, procedural) [7] or reasoning capabilities (e.g., multi-hop, comparison, algebraic) [8, 9]. To enhance the effectiveness of deliberate practice [10], assessment questions are classified into their re-

spective cognitive complexities (e.g., synthesis, evaluation) for instructors to determine learners' proficiencies [11–14]. Quite often, in scenarios where the size of the question bank is limited, statistical approaches are adopted for feature extraction in AQC.

Existing AQC techniques that employ the bag-of-words (BoW) approach represent a question with a vector constructed using syntactic, lexical or semantic features. This feature vector is then classified into its respective class label via a machine learning algorithm [10, 15–18]. Since BoW features are highly sparse and lack diversity [19], topic modeling approaches such as latent Dirichlet allocation (LDA) [20] have been developed. These admixture approaches were originally employed for document classification as they offer linguistic insights into language patterns by grouping associated words into topics and, thereafter, computing the probabilities of topics occurring in each document [21,22]. The ability to capture a document's semantic structure with reduced dimensionality [23–25] has also been exploited for extracting distinct topics to retrieve similar domain-specific questions [26–29].

In terms of sentence classification of which AQC is part of, deep learning has been employed for sentiment analysis (combining topic models and neural networks) [30], question answering [31], and domain-specific AQC [26, 32, 33]. In these contexts, pre-trained sentiment information or datasets with class labels containing details pertaining to the subject matter are used. As shown in this thesis, domain-agnostic AQC requires features not only related to the semantics/content, but also generic markers such as parts-of-speech (POS) tags. One of the key advantages of a topic modeling-based AQC is the consideration of global word co-occurrence patterns across questions that correspond to each class label when the dataset is limited, such as often occurs in practice for questions. The use of topic modeling will therefore allow one to incorporate the distribution of the above occurrences for feature extraction, which is important in providing the degree of association between topics and class labels for accurate AQC.

In domain-agnostic AQC, topics may overlap and comprise all word types; extracting representative features is, therefore, a challenge. Although the use of probabilities corresponding to the question-topic distribution as features may enhance AQC performance, the presence of high-frequency words in questions results in topic homogeneity,

where topics are assigned similar probabilities. While these high-frequency words can be grouped via asymmetric priors [34] and suppressed via term weighting in weighted LDA (W-LDA) [35], directly applying these techniques to AQC results in a similar question-topic distribution being associated with different class labels [36], rendering these features unsuitable for AQC.

Automatic document classification (ADC), on the other hand, is a broad spectrum that consists of a wide range of structures. Despite the fact that any possible technique would (in theory) be applicable for documents, in the context of domain-agnostic class labels, existing deep learning techniques render themselves unsuitable. This is due to the lack of considering the types of terminology being used in the documents that correspond to the generic labels. For instance, mapping job descriptions to skillsets requires the identification of types of phrases and patterns of co-occurrence (regexes) that could be matched against skills such as *Communication* or *Creative Thinking* [37, 38]. Such skills can be applicable to job roles from any industry, hence being domain-agnostic. Recently, graph networks have gained popularity as opposed to the conventional sequence-based or convolutional neural networks due to its ability to capture multi-dimension relational information. However, majority of graph networks developed for text classification only consider word nodes [39, 40] which are insufficient to encapsulate the meaning of a document.

1.2 Main contribution of the thesis

Contributions made by the author are mainly described in Chapters 3, 4, 5, and 6.

In Chapter 3, the main contribution is to enhance the conventional term frequency-inverse document frequency (TF-IDF) by proposing the sorted TF-IDF (s.TF-IDF) to suit questions such that an alternative feature space is used to represent the questions. This work has been published as S. Supraja, K. Hartman, S. Tatinati, and Andy W. H. Khong, "Toward the automatic labeling of course questions for ensuring their alignment with learning outcomes," in *Proc. 10th Int. Conf. Educational Data Mining (EDM)*, 2017, pp. 56–63.

In Chapter 4, the main contributions are to highlight the importance of the choice of stopwords for question classification and enhance the word network topic model (WNTM) by proposing the customized question WNTM (q-WNTM) to suit questions such that word co-occurrence redundancy is being addressed. This work has been published as S. Supraja, S. Tatinati, K. Hartman, and Andy W. H. Khong, "Automatically linking digital signal processing assessment questions to key engineering learning outcomes," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2018, pp. 6996–7000.

In Chapter 5, a new phrase-based topic model that introduces the concept of nested regular expressions for question classification. In this chapter, a new formulation and scaling parameter for determining relevance of regexes, inter- and intra-class-based term-weighting scheme, and a new topic regularization mechanism are described. This work has been published as S. Supraja, Andy W. H. Khong, and S. Tatinati, "Regularized phrase-based topic model for automatic question classification with domain-agnostic class labels," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 29, pp. 3604–3616, 2021.

In Chapter 6, a new quad-faceted feature-based graph network that encompasses four different graphs with different types of nodes and corresponding unique edge weights is presented. This new graph model is evaluated on various document classification datasets that comprise domain-agnostic class labels. This work has been submitted to a journal as S. Supraja and Andy W. H. Khong, "Quad-faceted feature-based graph network for domain-agnostic text classification," *IEEE/ACM Trans. Audio Speech Lang. Proc.*

1.3 Organization of the thesis

This thesis addresses the need to extract suitable features for classification of documents and questions according to domain-agnostic class labels.

Chapter 2 reviews baseline algorithms for both document and question classification, including frequency-based methods, topic modeling, and deep learning approaches. Chapter 3 presents the background of the need to classify questions according to cognitive complexities, details of the proposed s.TF-IDF algorithm, and description of the single course dataset used for subsequent evaluation. Chapter 4 presents the proposed

q-WNTM algorithm and experiment results along with insights generated. Chapter 5 details the generalizability of AQC to domain-agnostic class labels, technicalities of the proposed question LDA (Qu-LDA) phrase-based regularized topic modeling algorithm, the new elements being proposed, followed by various comparison analyses with existing topic modeling variants. Chapter 6 presents a new quad-faceted feature-based graph network (QGN) that encompasses various graphs with different functionalities. Classification results with different datasets are shown. Chapter 7 concludes the thesis and proposes directions of future work.

Chapter 2

Review of Automatic Document and Question Classification Techniques

This chapter reviews existing approaches used for automatic document classification (ADC) and automatic question classification (AQC) as detailed in Figure 2.1. Frequency-based and topic modeling feature extraction techniques are described, followed by various machine learning algorithms that have been employed to process these feature vectors for classification. In addition, the popularly and recently used deep learning techniques have been explored. Notwithstanding that questions are a subset of documents and due to differences in structure and properties of questions as opposed to long texts/documents, topic modeling has been described for AQC and deep learning for ADC.

Conventional methods of AQC according to learning outcomes employ the primitive rule-based approach. Such an approach combines parts-of-speech tagging, identifies verbs associated with Bloom's Taxonomy, and recognizes the presence of particular punctuation marks to create features as inputs to machine learning algorithms. However, for a new or updated set of questions, it is expected that some questions fail to activate any of these rules [41, 42]. Hence, this thesis explores better techniques for feature extraction.

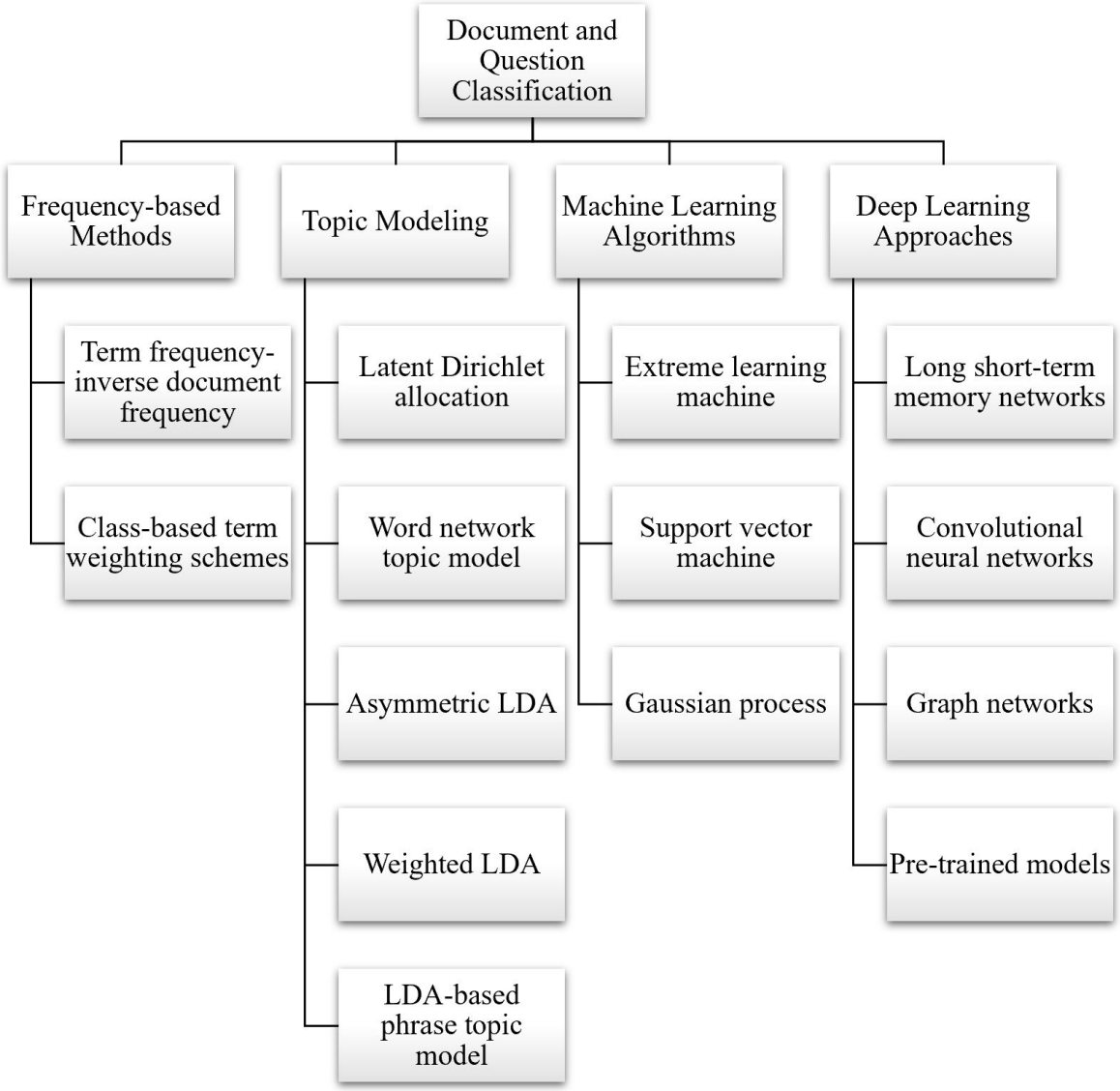


Figure 2.1: Overview of ADC and AQC approaches.

2.1 Frequency-based methods

A bag-of-words (BoW) approach is adopted in which each word in a question is assigned a term weight and a question is represented as a feature vector with dimension corresponding to the total vocabulary size in a corpus.

2.1.1 Term frequency-inverse document frequency

Term frequency-inverse document frequency (TF-IDF) ascertains the nature of a word in terms of uniqueness or rarity of a word in a set of questions. The importance of both the local and global presence of words gives rise to the motivation of the term frequency (TF) and inverse document frequency (IDF) respectively [43]. TF-IDF can be expressed as

$$\Omega_{w_i,q} = \frac{N_{w_i,q}}{N_w} \times log\left(\frac{N_Q}{N_{w_i}+1}\right), \tag{2.1}$$

where $\Omega_{w_i,q}$ denotes the TF-IDF weight for the ith word w_i in a question q, $N_{w_i,q}$ the number of times w_i occurs in q, N_w the number of words per question, N_Q the total number of questions, and N_{w_i} the number of questions in which w_i occurs. The first term in (2.1) therefore models the frequency of a word *within* a question while the second term models the concentration of words *across* all questions. The main objective is to make rare words prominent and ignore common words. The higher the TF-IDF weight of a word, the more it is unique to a question. This makes it easier to distinguish among questions for subsequent classification into the various labels.

The obtained TF-IDF values are subsequently normalized by the Euclidean length of each question [44]. To illustrate the above, the TF-IDF score for each word is divided by the normalization constant $\sqrt{(\Omega^2_{w_1,q} + \Omega^2_{w_2,q} + \ldots + \Omega^2_{w_{L_q},q})}$, where L_q denotes the length of a question. TF-IDF scores are, in general, represented in the traditional BoW manner such that a matrix is created with each row corresponding to a question and each column corresponding to every single unique word in the corpus. Elements in this matrix correspond to the TF-IDF scores for each word in each question. Hence, each question is represented by a vector with dimension belonging to the length of the vocabulary [45].

If a word is not present in a question, the TF-IDF value is zero.

By performing classification using the BoW vector representation, the main disadvantage is that the vector for each question is significantly sparse since the majority are zeros corresponding to the absent words in each question. This implies that the machine learning algorithm identifies the same trend of mostly zeros in all the questions and hence is unable to distinguish among the various classes. From an alternative perspective, the prevalence of zeros causes the lack of diversity among the various questions according to the machine learning algorithm; the important non-zero weights are not sufficiently prominent among the vast space of zeros.

To enhance the TF-IDF performance, the TF-IDF weights could be viewed as interpreting the feature space in terms of the distribution of the nature of words instead of the actual nature of words. The proposed s.TF-IDF feature, which will be described in Chapter 3, sorts the TF-IDF weights and compares the questions in terms of uniqueness and commonality nature of words.

2.1.2 Class-based term weighting schemes

Considering that TF-IDF only takes the corpus-wide frequencies into account, it is insufficient to estimate the significance of a word, therefore, class-based term weighting schemes such as inverse class frequency have been employed for AQC [46, 47]. Other class-based term weighting schemes for general text classification include the inverse gravity moment [48, 49]. Recently, a new term weighting scheme that incorporates inter- and intra-class word distributions has been developed, such that a significant word is determined based on its presence in fewer class labels and largely within a particular class label. The modified distinguishing feature selector (MDFS) term weighting is given by [50]

$$\Omega_{w_i} = \sum_{c=1}^{N_L} \Psi_{w_i,y_c} \Omega_{w_i,y_c}, \tag{2.2}$$

where Ψ_{w_i,y_c} denotes the specific weighting factor imposed on w_i for each class label y_c and Ω_{w_i,y_c} the MDFS weight for that word in every class label such that

$$\Psi_{w_i,y_c} = \log\left(1 + \frac{N_{w_i,y_c}}{\max(1, N_{w_i,\overline{y}_c})} \frac{N_{\overline{w}_i,\overline{y}_c}}{\max(1, N_{\overline{w}_i,y_c})}\right), \tag{2.3}$$

$$\Omega_{w_i,y_c} = \frac{P(y_c|w_i)P(\overline{y}_c|\overline{w}_i)}{P(\overline{w}_i|y_c) + P(w_i|\overline{y}_c) + 1}. \tag{2.4}$$

The variable Ψ_{w_i,y_c} is computed based on the number of documents with w_i as opposed to those without that word $\overline{w}_i$, within a particular class label y_c or in other class labels $\overline{y}_c$. The variable Ω_{w_i,y_c} in (2.4) denotes the conditional probabilities of the presence or absence of a word and each class label that are computed based on (2.3) and a value of 1 is added to avoid division-by-zero error [50]. These formulations reflect the inter- and intra-class attributes, where the former conforms to the criteria of the term weighting being inversely proportional to the spread of frequencies across all class labels, i.e., a word almost equally present in all class labels would receive a lower weight than that concentrated to a particular class label. On the other hand, the intra-class term weighting is proportional to the number of times a word occurs within a class label. These intricacies are not reflected by corpus-wide term weighting which only consider frequencies across all documents [35].

Nevertheless, both TF-IDF and class-based term weighting schemes are represented via a BoW approach that is inefficient due to vector sparsity. To achieve a uniform comparison across the questions by creating clusters of similar words with a common property, topic modeling has been introduced. In topic modeling, the same set of words are grouped together and a common weight is assigned to each group for each question as will be described in the next subsection.

2.2 Topic modeling

Topic modeling aims to uncover hidden patterns of words and connects documents (or questions) with similar patterns. While topic modeling has been proposed for document classification, this technique has also been applied to question classification in this thesis.

Topic modeling observes the correlations among words (occurrences together) and determines their relationship by linking these words across a corpus. However, it is important to note that a topic in document classification implies a group of words which conveys a collective meaning of the content of those words. The labels are not determined by the model; but can be implicitly interpreted by human judgment [51]. Topics are hidden relations that link words within a vocabulary and their occurrence in questions resulting in topics being expressed as a probability distribution over the words. A topic is formed by words which tightly co-occur with each other frequently. Similarly, questions model the probability distribution over the topics based on the words present in each question in comparison with other questions. Clusters formed by latent Dirichlet allocation (LDA) are shared among all the questions, serving as a uniform way of comparing the types of words used in various questions [52]. Each question will be represented by a vector of probabilities assigned to each topic. Observing the combinations of how each topic probability falls within a particular range of values, the machine learning algorithm will be able to clearly differentiate among the three labels. With questions belonging to the category of short texts, topic modeling is more appropriate than sophisticated techniques to obtain accurate features.

2.2.1 Latent Dirichlet allocation

Among various approaches, latent Dirichlet allocation (LDA) is one of the well-known techniques that performs topic modeling primarily for document classification. Questions can be represented via topic modeling by first defining N_Q as the total number of questions, N_Z as the number of topics, and N_V as the size of the word vocabulary. Topics are then sampled from the questions' multinomial distribution $\mathbf{\Theta} \in \mathbb{R}^{N_Q \times N_Z}$ with a Dirichlet prior α and the corpus-wide topic-word multinomial distribution $\mathbf{\Phi} \in \mathbb{R}^{N_Z \times N_V}$ with a Dirichlet prior β [20]. Posterior probabilities of each topic for a question $P(z_j|\widetilde{q})$ and of

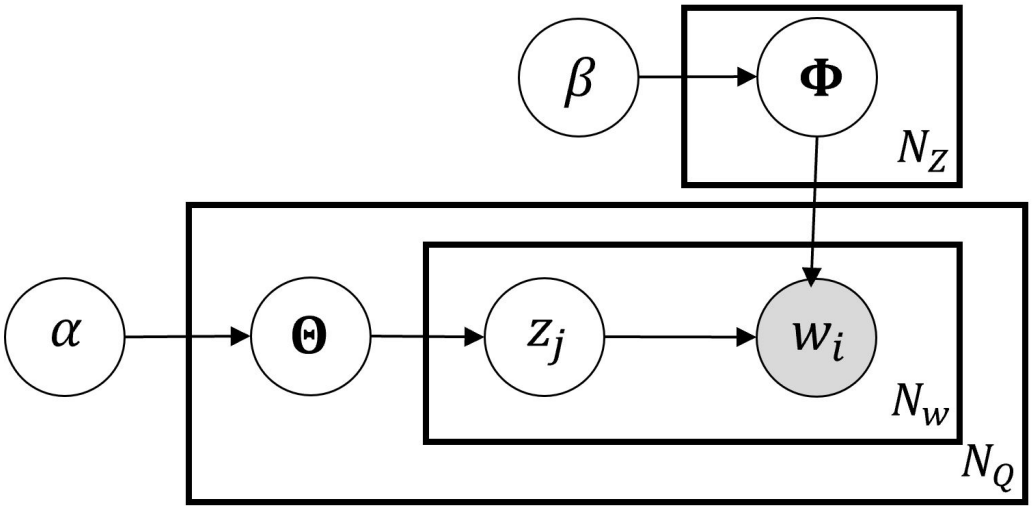


Figure 2.2: Plate diagram of LDA.

each word per topic $P(w_i|z_j)$ are computed as

$$P(z_j|\widetilde{q}) = \frac{\sum_{i=1}^{N_V} N_{w_i,z_j}^{(\widetilde{q})} + \alpha}{\sum_{i=1}^{N_V} N_{w_i}^{(\widetilde{q})} + \alpha N_Z}, \tag{2.5}$$

$$P(w_i|z_j) = \frac{N_{w_i,z_j} + \beta}{\sum_{i=1}^{N_V} N_{w_i,z_j} + \beta N_V}, \tag{2.6}$$

where $\sum_{i=1}^{N_V} N_{w_i,z_j}^{(\widetilde{q})}$ denotes the total number of times each word w_i occurs in $\widetilde{q}$ for a given topic z_j, $\sum_{i=1}^{N_V} N_{w_i}^{(\widetilde{q})}$ the total number of words in that question, N_{w_i,z_j} the number of times each w_i occurs in that topic across all questions, and $\sum_{i=1}^{N_V} N_{w_i,z_j}$ the total number of words in that topic across all questions [20]. Each question consists of N_w words. The plate diagram of LDA is shown in Figure 2.2.

With the objective of grouping words into topics, the first problem involves determining which words belong to which topics and to what extent [53]. The first step is to assign each word in each question with a topic number randomly, based on a pre-defined total number of topics that segregates these words into [54]. This random allocation is performed according to a uniform distribution. Subsequently, two matrices will be formulated: one for counts of words in each topic and another for counts of topics in each question. However, since the initial assignment is performed randomly, it does not pro-

vide an accurate representation of the grouping of words into topics and the probabilities of each topic occurring in a question. LDA employs Gibbs sampling inference techniques to iterate through these topic assignments by re-assigning topics if the product of these local and global analysis is better than the current assignment such that a steady state is achieved eventually. Gibbs sampling is used to obtain a random sample from a posterior distribution, and the eventual final sample serves as a discrete approximation to the posterior distribution after several iterations [55].

In this case, a unique steady state distribution exists, independent of the initial state. In the context of LDA, states refer to topic assignments to words. The requirement in LDA is to design an optimal function that makes a probabilistic choice pertaining to transiting to the next state (topic assignment) according to certain transition probabilities. These transition probabilities ought to be governed by some conditions such that visiting either the current state or transiting to another state will be as desired to obtain the optimal set of topic assignments [56]. Therefore, although any random initialization of topics is performed for the words in all the questions, the final set of topic-word and question-topic proportions will be obtained after several Gibbs sampling iterations.

When scaling down from documents to questions however, there are insufficient number of instances to observe such tight word co-occurrences [57]. In this scenario, there are fewer occurrences of rare words, hence these words are considered to be unimportant and are not taken into consideration. As a result, LDA tends to ignore rare topics that are related to minority of questions in the corpus and identifies the relationships among the remaining words since there are sufficient number of instances in the space of questions to words. Due to this limited word co-occurrence information in questions, this data sparsity results in a question-topic distribution that is not as distinct. Hence, the resultant topics are of lower semantic coherence. If two words are strongly related semantically but rarely co-occur in short texts, LDA is unable to completely capture the semantic relation between these two words. Hence, due to the insufficient number of common words/shared context, this results in the difficulty of determining similarity among questions to perform classification if LDA is applied [58]. The inability of LDA to perform well on questions prompted the development of other techniques to solve the issue of short text topic modeling.

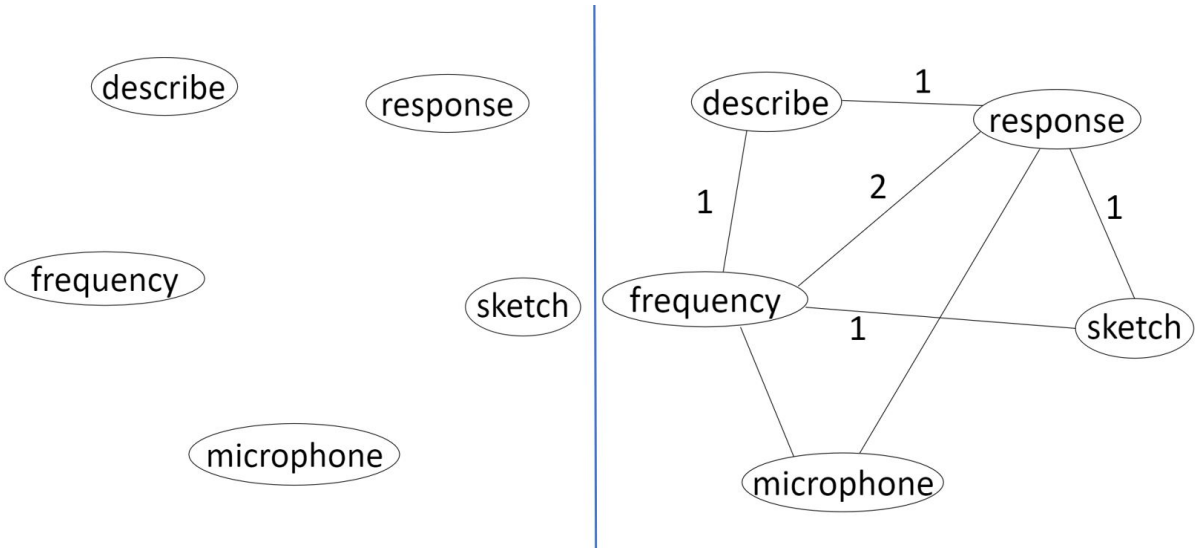
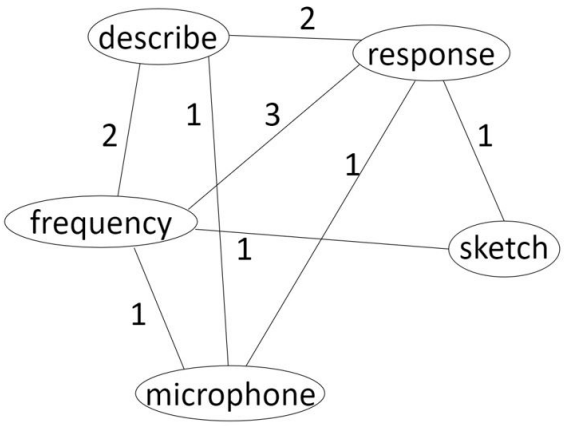


Figure 2.3: Construction of word network diagram in WNTM.

One option to segregate the rare words without introducing new techniques is to increase the number of topics N_Z until the unique rare words belong to particular topics. However, this increases the dimensionality of the vector representation for each question. The grouping of similar words becomes diluted by further segregating the topics, resulting in close to zero values for majority of the columns in the topic probability vector, increasing the sparsity back to the original BoW approach which failed in TF-IDF. Hence, the feature space in which the questions are being analyzed can be changed since there are fewer co-occurrences of words across questions in comparison to long documents. To this end, the feature space among words (reflected by a word-word matrix) is denser compared to the space of questions versus words and hence, rare words can now be made more visible by having relationships with other words. Out of the several methods, word network topic model (WNTM) examines the dense word-word space instead of the question-word space to explore relationships among words [59]. The drawbacks of applying LDA for questions lead to the application of WNTM to generate the feature vector for AQC.



Word	Pseudo-question
describe	frequency response frequency response microphone
response	describe frequency sketch frequency describe frequency microphone
sketch	frequency response
microphone	describe frequency response
frequency	describe response sketch response describe response microphone

Figure 2.4: Word network diagram and pseudo-questions in WNTM.

2.2.2 Word network topic model

Word network topic model (WNTM) models the distribution over topics for each word instead of directly learning the topics for each question. It observes the word co-occurrence information throughout the corpus. The semantic density of data is enriched and the global contextual information is made available through the word-word space. WNTM enables the identification of groups of words that correspond to rare topics as it dwells in the space among words instead of among questions. WNTM operates by firstly creating a word network graph in which the nodes correspond to each unique word present in the corpus. For example, assuming that there are three questions in the corpus as follows:

Question 1: Describe frequency response.

Question 2: Sketch the frequency response.

Question 3: Describe the frequency response of a microphone.

As will be described in Chapter 3, standard text pre-processing is first performed. The unique words are represented as nodes in Figure 2.3 (left). Subsequently, for every co-

occurrence of two words within the same question, the weight between every two nodes of this undirected graph is being incremented. After processing Questions 1 and 2, WNTM determines the partial set of updated weights as illustrated in Figure 2.3 (right). The purpose of having an undirected graph is due to WNTM being a BoW model in which the order of words is not taken into consideration. Instead, the presence of words along with each other in each question is of importance.

The process of updating the edge weights in the word network graph is repeated for every question and Figure 2.4 (left) shows the completed network. After the completed network has been determined, for each word, the neighboring words along with the number of times they have co-occurred are used to generate a set of pseudo-questions (also referred to as the adjacent word list) which constitutes the global set of co-occurrence information for each word as seen in Figure 2.4 (right). Depending upon the edge weight between two nodes, words in the pseudo-question will be repeated accordingly. However, WNTM does not consider the order of words in the pseudo-questions.

After forming the pseudo-questions for every unique word present in the corpus, these pseudo-questions are treated as a new set of questions and the standard LDA Gibbs sampling is applied to iterate through the topic-word allocation counts and pseudo-questions-topic allocation counts. With reference to Section 2.2.1, each topic z_j generated in WNTM is also a multinomial distribution over the vocabulary of words, with a symmetric Dirichlet prior β. However, each new pseudo-question generated by WNTM, denoted by $\mathfrak{P}$, is a multinomial distribution over the topics, with a symmetric Dirichlet prior α. After obtaining the topic probabilities for each pseudo-question corresponding to the global set of co-occurrence relationships for each word, the topic probabilities for each original question based on every individual word $w_i^{(\widetilde{q})}$ are inferred as

$$P(z_j|\widetilde{q}) = \sum_{i=1}^{N_w} P(z_j|w_i^{(\mathfrak{P})}) \times P(w_i^{(\widetilde{q})}|\widetilde{q}), \tag{2.7}$$

where $P(z_j|q)$ refers to the probability of the jth topic z_j in each question $\widetilde{q}$, $P(z_j|w_i^{(\mathfrak{P})})$ refers to the jth topic probability in each pseudo-question $\mathfrak{P}$ belonging to each word $w_i^{(\widetilde{q})}$, and $P(w_i^{(\widetilde{q})}|\widetilde{q})$ refers to the frequency of each unique word in the original question in

Table 2.1: Highlighting rare words through WNTM.

Word	Pseudo-question
A	B C B C D
B	A C A C
C	A B B A D
D	A C

relation to the total number of words present in that question. The sum of this expression is taken for all the words in a question.

The purpose of taking the topic probability represented by each word (first term in (2.7)) is to show how globally that particular word is linked to that topic based on the surrounding words, as well as, to serve as an impact of the repetition of words present in the pseudo-questions. Unlike LDA, WNTM considers each pseudo-question as an extended context that depicts one particular word itself. While constructing the pseudo-questions, repetition of words according to the weights between the edges in the word network graph increases the emphasis for each word in each question, which relates to the higher probability of that particular topic to occur in that question. This concept is similar to TF-IDF with reference to how the more number of times the same word occurs in a question relates to a higher TF value making that word more unique towards that question.

The purpose of taking the frequency of each word in the second term of (2.7) is again similar to the computation of term frequency in TF-IDF in which, within the bound from 0 to 1, a higher value denotes higher emphasis given to the term which occurs more number of times. The intuition behind the creation of pseudo-questions can be interpreted as extending the original question based on the global set of surrounding words according to the word network diagram. It can be interpreted as each word being replaced with the remaining surrounding words to provide the full context of how that word could have occurred in a regular long document. Instead of altering the original question, WNTM considers each word as a separate entity. Based on the surrounding words, it identifies the extent of that word being related to a topic.

In addition, WNTM can identify rare words since implicit connections exist between

these rare words and other words in the word network. Apart from the rare words, implicit connections among all other commonly occurring words are highlighted in the pseudo-questions. To explain how the rare words become prominent in WNTM compared to LDA, an example case can be illustrated with four questions and four unique words A, B, C, and D as follows:

Question 1: A B C

Question 2: A B

Question 3: B C

Question 4: A C D

"D" is a rare word compared to the rest of the words, and hence it becomes ambiguous to which topic "D" should belong to if LDA is being applied to these questions. However, in WNTM, while constructing the pseudo-questions as shown in Table 2.1, "D" can be seen co-occurring not only with "A" and "C", but with "B" as well. This is because the implicit connections among words are emphasized in WNTM due to the pseudo-questions, which represent the global word network relationships, unlike in LDA which considers word co-occurrences only within each question. The final vector of topic probabilities for each question generated through WNTM serves as a better feature vector than LDA. However, directly applying WNTM to the questions may not be appropriate since WNTM generates the word network, which contains the entire set of connections among all the words in the vocabulary. This prompts the need to analyze the types of word connections that are important versus redundant for classifying the questions into the three categories "K," "A," and "T," respectively.

2.2.3 Asymmetric LDA

A hypothetical set of topics with high-probability words and their corresponding topic probabilities for each LDA variant is tabulated in Table 2.2. With symmetric variables α and β, both LDA and WNTM suffer from topic homogeneity described by having all topics comprising high-frequency words (e.g., *the* and *of*) that result in equal topic probabilities $P(z_1) = P(z_j) = P(z_{N_Z}) = 0.02$. This implies that feature vector $\mathbf{q}$ is, to a

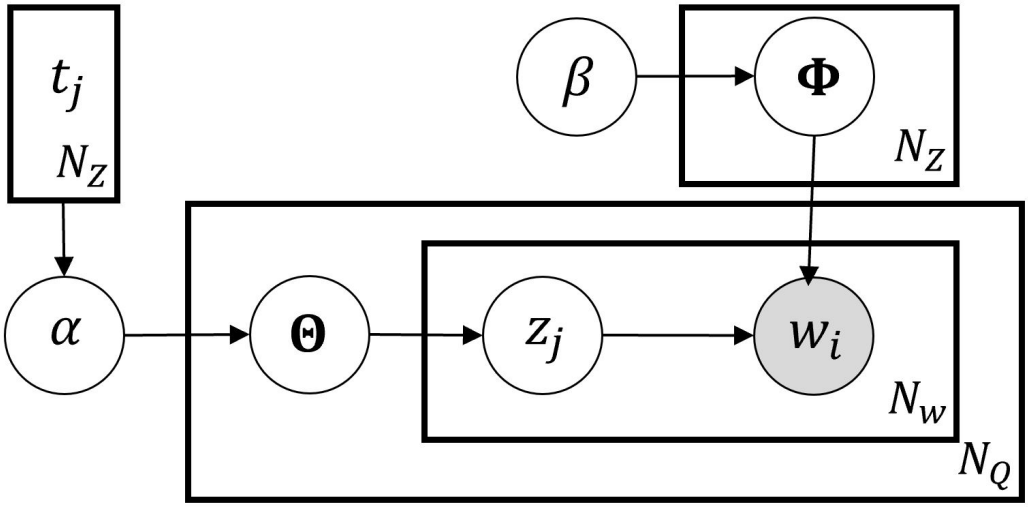


Figure 2.5: Plate diagram of A-LDA.

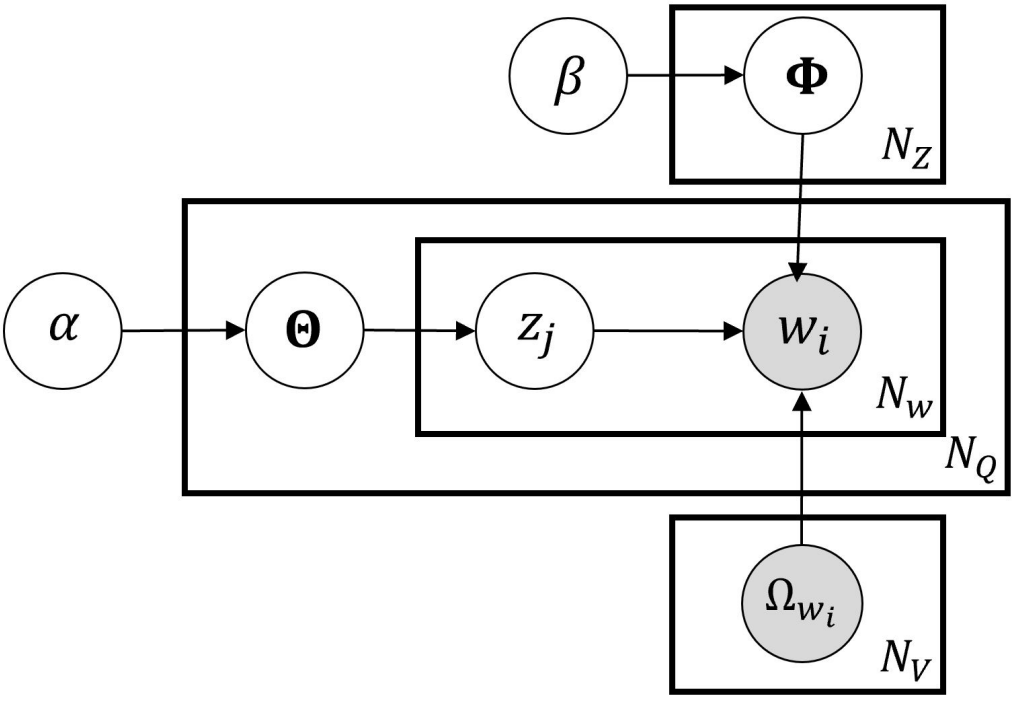


Figure 2.6: Plate diagram of W-LDA.

large extent, agnostic to the class label resulting in poor AQC performance.

To group these high-frequency words into a topic, asymmetric LDA (A-LDA) [34] incorporates asymmetric α priors. With N_Z number of different asymmetric α values being used across all topics, maximizing the likelihood estimate of the Dirichlet distribution to determine an appropriate α value per topic is achieved via the Newton-Raphson method [60–62]. The plate diagram of A-LDA is shown in Figure 2.5.

2.2.4 Weighted LDA

While the use of asymmetric priors in LDA mitigates the homogeneity problem, A-LDA constructs topics based solely on word frequencies. This results in emphasizing the topic associated with high-frequency words as seen in Table 2.2, where $P(z_{N_Z}) = 0.3$, thereby suppressing the prominence of other relevant topics. Weighted LDA (W-LDA) employs term weighting [35] and assigns low probabilities to high-frequency words by replacing N_{w_i} and N_{w_i,z_j} in (2.5) and (2.6) with a pre-computed corpus-wide weight Ω_{w_i} for each w_i and Ω_{w_i,z_j} for each word in each topic, respectively. This results in the topic probability associated with high-frequency words being suppressed as seen via $P(z_{N_Z}) = 0.04$ in Table 2.2. The plate diagram of W-LDA is shown in Figure 2.6.

More importantly, the lack of incorporating class label distribution of each word results in topic probabilities being almost uniform across the remaining topics as seen via $P(z_1) \approx P(z_j)$. This implies that such feature vectors do not discriminate well across the different class labels, rendering poor AQC performance. While phrase-based topic models have recently been proposed to encapsulate the contextualization of words [63–65], application of such models to AQC requires the incorporation of word-based elements (asymmetric α priors and term weighting). Consequently, these phrase-based topic models (denoted by P-LDA) will result in a question-topic distribution similar to W-LDA, with the exception of topics consisting of phrases in lieu of only single words, as shown in Table 2.2.

2.2.5 LDA-based phrase topic model

To generate more meaningful topics, LDA-based phrase topic model (LPTM) first extracts noun phrases (NPs) [66–68] (e.g., *dangerous aspects*—a combination of an adjective and a noun as seen in Table 2.2). It then represents a phrase-extracted question consisting of N_p phrases as $q = \{p_1, p_2, \ldots, p_{N_p}\}$, where $p_k = \{w_{k,1}, \ldots, w_{k,L_{p_k}}\}$ is defined as the kth NP that is made up of L_{p_k} words. The parts-of-speech (POS) tags of all words within the phrase are then grouped together to form a regular expression (regex) $r_k = \{POS(w_{k,1}), \ldots, POS(w_{k,L_{r_k}})\}$ of length L_{r_k}. With reference to the above *dangerous aspects* example, this regex will be denoted by *ADJ_NOUN*. Subsequently, q is now re-defined as a regex-extracted question $q = \{r_1, r_2, \ldots, r_{N_p}\}$.

Such POS-guided phrasal segmentation [69] is employed in LPTM to construct a topic-regex multinomial distribution $\boldsymbol{\eta} \in \mathbb{R}^{N_Z \times N_R}$ with a Dirichlet prior λ [70], where N_R denotes the regex vocabulary size. The posterior probability of each regex per topic is then given by

$$P(r_k|z_j) = \frac{N_{r_k,z_j} + \lambda}{\sum_{k=1}^{N_R} N_{r_k,z_j} + \lambda N_R}, \tag{2.8}$$

where N_{r_k,z_j} is the number of times each regex r_k occurs in z_j and $\sum_{k=1}^{N_R} N_{r_k,z_j}$ the total number of regexes in that topic.

Table 2.2: Hypothetical set of topics and their corresponding topic probabilities using LDA, A-LDA, W-LDA, and P-LDA.

Method	z_1	$P(z_1)$	... ...	z_j	$P(z_j)$	... ...	z_{N_Z}	$P(z_{N_Z})$
LDA	the a	0.02	... ...	the a	0.02	... ...	the a	0.02
A-LDA	what provide	0.0005	... ...	find express	0.007	... ...	the a	0.3
W-LDA	what provide	0.08	... ...	find express	0.1	... ...	the a	0.04
P-LDA	dangerous aspects this problem	0.1	... ...	basic facts one advantage	0.09	... ...	find express	0.04

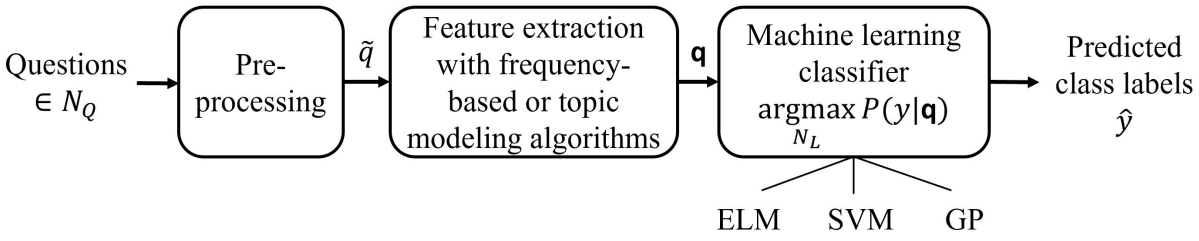


Figure 2.7: AQC framework with frequency-based or topic modeling algorithms for feature extraction and ELM, SVM, or GP machine learning classifiers.

2.3 Machine learning algorithms

The main motivation of Sections 2.1 and 2.2 is to develop feature extraction techniques which are subsequently used by machine learning techniques for AQC. Hence, to provide a comparison analysis, this thesis covers three machine learning techniques – the extreme learning machine (ELM), support vector machine (SVM) and Gaussian process (GP). The AQC framework is shown in Figure 2.7. After generating the feature vector for each question using either frequency-based or topic modeling algorithms, the machine learning techniques utilize the vectors to perform supervised learning and thereafter evaluate the classification performance for the testing dataset [71]. This multi-class classification is performed to compare the performance of the various feature extraction techniques to segregate the class labels.

2.3.1 Extreme learning machine

Extreme learning machine (ELM) is a learning algorithm for a single hidden layer feed-forward neural network (SLFN). The key concept of ELM is that the weights connecting the input and hidden layers are randomly generated while the weights connecting the hidden and output layers are analytically determined using a regularized least squares solution [72]. Given a set of input samples x_k with the respective target/output given by t_k, with an activation function (e.g., sigmoid, sine, radial basis, hard-limit) for the hidden nodes and by specifying the number of hidden neurons L, the weight matrix is computed using the Moore-Penrose generalized inverse [73]. The final predicted class label is based

on the output with the largest value [74].

The SLFN is formulated as

$$\sum_{j=1}^{L} \boldsymbol{\gamma}_j f_j(x_k) = \sum_{j=1}^{L} \boldsymbol{\gamma}_j f(\mathbf{w}_j \cdot x_k + b_j) = o_k, \qquad k = 1, \ldots, L, \tag{2.9}$$

where $\mathbf{w}_j$ denotes the weight vector that stores the weights between the input and hidden nodes, $\boldsymbol{\gamma}_j$ the weight vector that stores the weights between the hidden and output nodes, and b_j the threshold of the jth hidden node. The objective is to achieve $o_k \rightarrow t_k$, i.e., output is close to the target value. ELM learning algorithm differs from the traditional backpropagation in that one only needs to set the number of hidden neurons and the activation function. It avoids learning epochs that exist in the gradient-based backpropagation method. It uses a small number of resources and the parameters are free of tuning [75].

It has been shown that high reliability is achieved for classification tasks using ELM as opposed to SVM when comparing in terms of the standard deviation of training and testing root-mean-square values, time taken, network complexity, as well as performance comparison in actual medical diagnosis applications [72].

2.3.2 Support vector machine

The concept of the support vector machine (SVM) is based on structural risk minimization. SVM maps data in the input space to a feature space using a nonlinear mapping function. Subsequently, a separating hyperplane maximizes the margins of two classes in the new feature space. This process separates the data into groups that have large gaps between them. If the original data is not linearly separable, a kernel function (e.g., radial basis, sigmoid, linear or polynomial) is used to transform these samples into a high-dimensional space [76]. In the case of a multi-class classification, either a one-versus-all (construction of many binary classifiers) or a one-versus-one (classifier for every two classes) approach can be taken [77]. The C-support vector classification type

is used in this thesis. Given a set of inputs and targets, the cost function is given by

$$\min_{\mathfrak{p},K,\epsilon} \frac{1}{2}\mathfrak{p}^T\mathfrak{p} + C\sum_{j=1}^{k}\epsilon_j \tag{2.10}$$

subject to $y_j(\mathfrak{p}^T\phi(v_j) + K) \geq 1 - \epsilon_j,\ \epsilon_j \geq 0, j = 1, \ldots, k$, where $C > 0$ denotes the regularization parameter, K a constant, $\mathfrak{p}$ the vector of coefficients, ϵ_j the parameters that handle the inputs, v the independent variables, ϕ the kernel used that transforms data from the input to the chosen feature space, and y the class labels.

2.3.3 Gaussian process

Since GP is a random process in which any point is assigned a random variable $\mathbf{g}$, the predicted class label is computed via

$$\begin{aligned} P(\mathbf{g}_c|\mathbf{Q}) &= \mathcal{N}(\mathbf{g}_c|\mathbf{0}, \mathbf{K_{QQ}}), c = 1, 2, \ldots, N_L, \\ P(y_c|\mathbf{g_c}) &= \mathcal{N}(y_c|\mathbf{g}_c, \kappa^{-1}\mathbf{I}), c = 1, 2, \ldots, N_L, \end{aligned} \tag{2.11}$$

where $\mathbf{0}$ denotes the mean function, $\mathbf{K_{QQ}} \in \mathbb{R}^{\mathbf{Q}\times\mathbf{Q}}$ the GP covariance matrix for the set of feature vectors in a corpus $\mathbf{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_{N_Q}\}$, and κ the noise covariance. The radial basis function (RBF) given by [78]

$$k(\mathbf{q}_l, \mathbf{q}_{l'}) = \exp\left(-\frac{1}{2\sigma^2}||\mathbf{q}_l - \mathbf{q}_{l'}||_2^2\right) \tag{2.12}$$

is commonly chosen as the kernel function. Here, σ^2 denotes the kernel bandwidth and $||\mathbf{q}_l - \mathbf{q}_{l'}||_2^2$ the squared Euclidean distance between two question feature vectors $\mathbf{q}_l$ and $\mathbf{q}_{l'}$.

For multi-class AQC, the one-versus-all strategy is adopted such that N_L binary classifiers are trained [79]. Each classifier will separate questions of the current class label being considered against the rest and the class membership of an unknown test set question's feature vector $\mathbf{q}_*$ is determined by the class label corresponding to the most confident classifier. The classifier $\mathcal{F}_c$ is then trained on the training set $\mathcal{Q} =$

$\{(\mathbf{q}_1, y_c^1), (\mathbf{q}_2, y_c^2), \ldots, (\mathbf{q}_{N_Q}, y_c^{N_Q})\}$, where y_c^l denotes the corresponding class label y_c for the lth question. Defining $\mathcal{F}(\mathbf{q}_*)$ as the posterior mean according to classifier $\mathcal{F}_c$ that the new input $\mathbf{q}_*$ belongs to that particular class label, the class label of $\mathbf{q}_*$ is predicted via

$$\widehat{y}_* = \underset{c \in \{1,2,\ldots,N_L\}}{\operatorname{argmax}} \mathcal{F}_c(\mathbf{q}_*). \tag{2.13}$$

Therefore, among the probabilities generated for each pair of class labels, the class label that achieves the highest predictive posterior mean is assigned to that unseen (test) question.

2.4 Deep learning approaches

This section reviews the formulations of well-known deep learning architectures such as the long short-term memory network (LSTM), convolutional neural network (CNN), and graph networks that have been widely employed for ADC. Several works that design deep learning based architectures to perform AQC or ADC employ the basic building blocks of CNN or recurrent neural networks (RNN) (LSTM or gated recurrent units (GRU)) in various ways to serve different purposes [33, 80]. Recently, graph networks have been developed for text classification [39, 40].

Techniques that employ deep learning algorithms for documents or AQC utilize word embeddings which convert words into a semantic space such that any unseen word with a similar meaning in the semantic vector space could be used. Words are represented as vectors, i.e., word embeddings, before being used as inputs for the encoder. Each original document is converted to a vector matrix in which the rows represent the vector corresponding to each word. These embeddings are trained instead of obtaining pre-trained word2vec or GloVe vectors as the terminology are significantly different in the collected dataset and might not have appropriate pre-trained vectors.

After applying any of the algorithms that will be presented in the subsequent subsections, the feature vector output is passed to a fully connected softmax layer with dropout

whose output is the probability distribution over the labels given as

$$y = \mathbf{W} \cdot \mathbf{r} + b_y, \tag{2.14}$$

where $\mathbf{W}$ denotes the weight matrix, b_y the bias term and $\mathbf{r} \in \mathbb{R}^{N_L}$ is a masking vector of multinomial random variables (corresponding to the class labels the documents are to be classified into) with total probability of 1.

2.4.1 Recurrent neural networks

One of the most popular sequence encoders in the RNN family that is used for text representation is the LSTM, and particularly the bi-directional LSTM (Bi-LSTM) that can be viewed as a combination of two unidirectional LSTMs, i.e., forward and backward. The forward LSTM computes the current state of the original document's word sequence based on the current embedding and the previous sequence state. The backward LSTM computes the sequence state in the reverse order from the last word to the first. The concatenation of two hidden representations is taken as the representation of the source document [81]. Hence, Bi-LSTM enhances the hidden representation of each word in a document by incorporating contextual information from the surrounding words (right and left). With each word in a document represented by embeddings, the encoder obtains a hidden representation of a document $\mathbf{h}_d = [h_1^d, h_2^d, \ldots, h_{L_d}^d]$, where L_d denotes the length of a document d. With j being an individual time step, the hidden representation for a particular word w_j^d is computed by concatenating the hidden states of both the forward-direction $\overrightarrow{\mathbf{h}}_{w_j^d}$ and backward-direction $\overleftarrow{\mathbf{h}}_{w_j^d}$ LSTMs, given as

$$\mathbf{h}_{w_j^d} = [\overrightarrow{LSTM}(w_j^d, \overrightarrow{\mathbf{h}}_{w_{j-1}^d}); \overleftarrow{LSTM}(w_j^d, \overleftarrow{\mathbf{h}}_{w_{j+1}^d})]. \tag{2.15}$$

Since the attention mechanism is analogous to how humans place emphasis on different segments of a document that exhibit important clues to understand the meaning of that document, an attention layer is applied on top of the encoded vectors to identify key segments of the documents while performing the decoding by assigning different weights to each word in the original document. A non-linear transformation is first applied on

the encoded vectors $\mathbf{h}_{w_j^d} = \tanh(\mathbf{W}\mathbf{h}_{w_j^d} + b)$, where $\mathbf{W}$ and b are the transformation weights and bias respectively. Each encoded vector interacts with a parameterized attention vector $\mathbf{u}_{w_j^d}$, producing an attention coefficient $a_j^d = \mathbf{h}_{w_j^d}^T u_{w_j^d}$ which is subsequently normalized through the softmax operation [82]. The vector representation for a document $\mathbf{d}$ is obtained via a weighted average of the word hidden representations, given by

$$\mathbf{d} = \sum_j \mathbf{h}_{w_j^d} \frac{\exp(a_j^d)}{\sum_j \exp(a_j^d)}. \tag{2.16}$$

The attention distribution is used to compute the context vector which can be considered as a fixed dimensional representation of the document. This representation is then fed through a fully-connected layer.

2.4.2 Convolutional neural networks

Although sequence encoders encompass the longitudinal representation of a document, the latitudinal and dense neighbouring elements are considered by CNNs. Following sequential CNNs, one dimensional convolutions operate the convolutional kernel in sequential order, given as

$$\mathbf{x}_{i,j} = x_i \bigoplus x_{i+1} \bigoplus \ldots \bigoplus x_{i+j}, \tag{2.17}$$

where $\mathbf{x}_i \in \mathbb{R}^e$ represents the e dimensional word representation for the i-th word in the document, and $\bigoplus$ is the concatenation operator. Therefore, $\mathbf{x}_{i,j}$ refers to the concatenated word vector from the ith word to the $(i+j)$-th word in a document.

A convolution operates a filter $w \in \mathbb{R}^{n \times e}$ to a window of n words $x_{i,i+n}$ with bias term b' by $a_i = \sigma(w \cdot x_{i,i+n} + b')$ with non-linear activation function σ to produce a new feature. The filter w is applied to each word in the sentence, generating the feature map $\mathbf{a} = [a_1, a_2, \ldots, a_{L_d}]$ where L_d is the document length. The entire feature map is represented as $\hat{a} = max\{\mathbf{a}\}$ after max-pooling.

To capture different aspects of patterns, CNNs are often initialized randomly via a set of filters with different sizes and values. Each filter will generate a feature as de-

scribed above. To take all the features generated by N different filters into account, $\mathbf{z} = [\widehat{a}_1, \ldots, \widehat{a}_N]$ is used as the final representation. In conventional CNNs, vector $\mathbf{z}$ will be directly fed into the classifiers after the document representation is obtained, e.g., fully-connected neural networks [83, 84]. From an architecture perspective, conventional sequence-based or convolutional neural networks that are often utilized for text classification are limited by their nature to prioritize sequentiality and locality [85, 86]. While these deep learning models capture semantic and syntactic information in the Euclidean space and in local sequences well, they do not account for global word co-occurrences in a corpus that carries non-consecutive and long-distance semantics [39, 87].

2.4.3 Graph networks

This section reviews the graph convolutional network (GCN). In recent years, features for text classification have been generated from non-Euclidean domains and are represented in the form of graphs [87]. These techniques preserve diverse global structural information and capture multi-dimension relational information as meaningful features [88]. In TextGCN, a single graph based on word co-occurrence and document-word relations serves as the input to subsequent convolutional layers for feature extraction and classification [39]. As an extension to sequential-based graphs [89], a TensorGCN triple-graph model has been developed to describe syntactic, semantic, and sequential information among word nodes [40]—in line with SynGCN and SemGCN in terms of the need to incorporate embeddings beyond sequentiality [90]. More recently, the graph fusion network (GFN) addresses the limitation of transductive methods [91] in adapting to new documents by discarding document nodes and constructing four homogeneous graphs [87]. Since the majority of graph models for text classification only consider word nodes [39, 40], direct application of these methods is not suitable for domain-agnostic text classification—they lack representations associated with additional textual features such as phrases, regexes, or topics.

Given the word vocabulary size as N_V and the total number of documents as N_D, $V = \{w_1, \ldots, w_i, \ldots, w_{N_V}, d_1, \ldots, d_m, \ldots, d_{N_D}\}$ is defined as the set of nodes. Here, w_i refers to the ith word node and d_m refers to the mth document node. The set of edges

is given by $E = \{e(w_i, w_j), e(d_m, w_i)\}$, where $e(w_i, w_j)$ denotes the edge between every ith and jth word nodes and $e(d_m, w_i)$ denotes the edge between every mth document and ith word nodes [92]. A graph $G = (V, E)$ consisting of word and document nodes is constructed after performing conventional text pre-processing as will be described for questions in Section 3.3. Defining the feature matrix as $\mathbf{F} \in \mathbb{R}^{(N_V+N_D)\times f}$, where the total number of nodes is $(N_V + N_D)$ and f denotes the feature vector dimension, embedding feature vectors of the nodes are constructed. In addition, the adjacency matrix that encompasses edge weights between all nodes in G is denoted by $\mathbf{A} \in \mathbb{R}^{(N_V+N_D)\times(N_V+N_D)}$.

Hidden layer representations of the node embeddings and edge weights are subsequently obtained by traversing convolutional layers initialized with $\mathbf{F}$. This hidden layer matrix $\mathbf{H}$ then serves as the input to a *softmax* classifier. Defining N_L as the total number of class labels, the classification task is formulated as estimating the mapping between $\mathbf{H}$ and each class label y_c such that the predicted class label is

$$\widehat{y} = \underset{c\in\{1,2,\ldots,N_L\}}{\operatorname{argmax}} \; softmax(y_c|\mathbf{H}). \tag{2.18}$$

Achieving good classification performance, therefore, relies on the types of nodes (to construct $\mathbf{F}$) and the computation of edge weights among the same type of nodes (i.e., word-word) and in relation to document nodes to construct $\mathbf{A}$.

GCN is a multi-layer neural network that operates on a graph with nodes embedded based on properties of their neighborhoods. Figure 2.8 shows the process flow of TextGCN in which all nodes are initialized using a one-hot representation before the joint optimization of embeddings for both words and documents given the class labels to obtain $\mathbf{F}$ [39]. Hidden layer representations are then obtained by encoding the graph structure that comprises the node feature vectors and relationships among nodes (i.e., edge weights) with a layer-wise propagation rule

$$\mathbf{H}^{(l+1)} = \gamma\left(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right), l = 0, 1, \ldots, \mathcal{L}, \tag{2.19}$$

with the non-linear activation function γ defined as a rectified linear unit ($ReLU$) for $0 \leq l < \mathcal{L}$ and *softmax* for the last layer $l = \mathcal{L}$. In (2.19), $\mathbf{H}^{(l)} \in \mathbb{R}^{(N_V+N_D)\times f_l}$ denotes

the feature matrix of the lth layer, f_l denotes the number of features for each node in the lth layer (with $f_{\mathcal{L}} = N_L$), $\mathcal{L}$ denotes the number of layers in GCN, and $\mathbf{W}^{(l)} \in \mathbb{R}^{f_l \times f_{l+1}}$ denotes the layer-specific trainable weight matrix with weight parameters trained via gradient descent. Elements in the degree matrix $\mathbf{D}$ are defined as $D_{ij} = \sum_j A_{ij}$, where A_{ij} denotes each element in $\mathbf{A}$, and the input layer is initialized as $\mathbf{H}^{(0)} = \mathbf{F}$. A fixed-size sliding window is conventionally applied to obtain co-occurrence statistics and PMI (a measure of association between the occurrence of two words) is employed to compute the weights between every pair of word nodes [39]. The document-word edge weight is computed based on the TF-IDF value of each word in a document. The loss function of TextGCN is defined as the cross-entropy error over all labeled documents in the training set to obtain $\widehat{y}$ for an unseen document in the test set.

As an extension to TextGCN, TensorGCN introduces three graphs with different properties for the same set of word nodes comprising edge links computed via dependency parsing, word embedding cosine similarities, and PMI as shown in Figure 2.9. These graphs are synthesized into a graph tensor (i.e., multiple graphs sharing the same nodes) [40] defined as $\mathcal{T} = (G_1, G_2, G_3)$. The corresponding adjacency matrices are similarly synthesized into a graph adjacency tensor $\mathcal{A} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$ with a graph feature tensor $\mathcal{H} = (\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3)$. To derive the representations via the triple graphs in TensorGCN, two types of propagation learning methods have been performed on $\mathcal{T}$. The intra-graph method aggregates information from neighboring nodes within a single graph while the inter-graph method harmonizes heterogeneous information among graphs as illustrated in Figure 2.9. For instance, given the lth layer of GCN, these propagation methods are employed consecutively on the graph feature tensor giving

$$\mathcal{H}^{(l)} \xrightarrow{p_{\text{intra}}} \mathcal{H}^{(l)}_{\text{intra}} \xrightarrow{p_{\text{inter}}} \mathcal{H}^{(l+1)}. \tag{2.20}$$

Here, p_{intra} and p_{inter} denote the application of each learning method and $\mathcal{H}^{(l)}_{\text{intra}}$ denotes the graph feature tensor after performing intra-graph propagation by applying (2.19) on $\mathcal{A}$ and $\mathcal{H}$. Unlike (2.19), the trainable weight matrix $\mathbf{W}^{(l,g)}_{\text{intra}}$ is designed to be graph-specific with g denoting the gth graph. To achieve inter-graph propagation, on the other hand, a series of virtual graphs are constructed by duplicating the same set of nodes and

connecting them across the graphs in the tensor. This results in a new graph adjacency tensor which, along with $\mathcal{H}^{(l)}_{\text{intra}}$ and $\mathbf{W}^{(l,g)}_{\text{inter}}$, constitutes the process p_{inter}. A mean pooling operation is applied over the graphs in the last layer to obtain the final representation of nodes for classification.

In contrast to TensorGCN, GFN [87] consists of four homogeneous word graphs with edge links comprising word embedding cosine similarities and Euclidean distances, PMI, and co-occurrence statistics. To generate document embeddings without document nodes, GFN implements the late fusion paradigm (i.e., the logit-level fusion of word embeddings) [87]. However, removing document nodes is not ideal in this context since the unique relationships between texts and the observable and latent features are vital in determining an accurate graph representation with respect to domain-agnostic class labels. In addition, the use of cosine similarity between embedding vectors for the semantic graph is advantageous when compared to the use of Euclidean distance. This is due to the possibility of two document vectors achieving a smaller cosine angle (i.e., higher similarity) despite being far apart (i.e., lower similarity) by the Euclidean distance that is based on size. Therefore, the proposed QGN that will be presented in Chapter 6 adopts the TensorGCN framework.

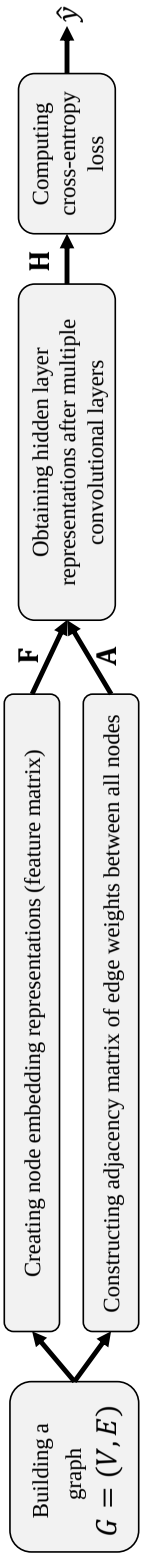


Figure 2.8: Process flow of TextGCN for text classification.

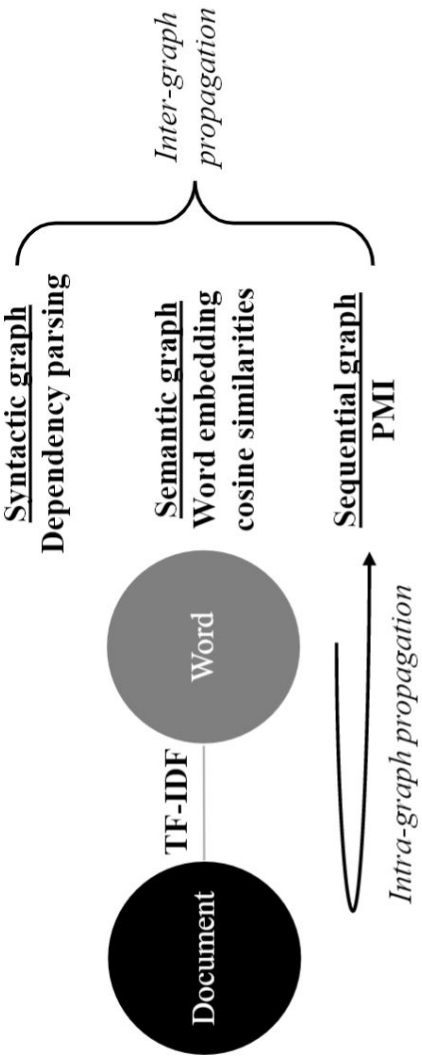


Figure 2.9: Architecture of TensorGCN.

2.4.4 Pre-trained models

Pre-trained language models have been frequently utilized for text classification tasks in recent years. The bidirectional encoder representations from transformers (BERT) model [93] achieves superior performance in various natural language processing applications in comparison to word2vec [94] or GloVE [95] word embeddings as it provides a deeper sense of language context. BERT is a transformer-based architecture—the latter uses a self-attention mechanism suitable for language understanding. The transformer comprises an encoder-decoder architecture with modules that contain feed-forward and attention layers. BERT, on the other hand, is a multi-layered encoder that reads and processes the input text. The decoder separately achieves prediction for a particular task.

BERT achieves bidirectional pre-training mainly via masked language modeling. In this process, the weights are initialized based on the pre-trained English Masked Language Model (MLM), a 12-block decoder-only transformer model trained to predict masked-out words on the Toronto Books Corpus and Wikipedia [96]. In the MLM objective, masking of some words in a sentence is performed, i.e., replacing a word with the token [MASK], randomly swapping/replacing with another word, or leaving it unchanged. To perform masking, words are first being split into subwords, and a certain percentage of words are randomly sampled to be masked [97–101]. By altering the word order of input sentences (noisy input), the shared encoder will be able to learn about the internal structures of a sentence and recover the correct word order (denoising autoencoder).

After the MLM, training is performed simultaneously with the denoising and back-translation objectives. For denoising, a noisy input is produced by randomly masking, dropping, and locally shuffling tokens in the target text, and the model is trained to maximize the probability of the correct word being the output. Similarly, the probability of the output text to be generated is maximized when masking is applied to obtain the predicted text. For back-translation, an original form of a text is generated for the predicted text, and the probability of the original text given the predicted text is maximized. Such language tasks are conventionally evaluated with the bilingual evaluation understudy (BLEU) [102], where BLEU measures the overlap of words between the predicted text and the ground truth (actual text). Hence, the training of BERT is ceased

without supervision when a round-trip BLEU achieves the highest score.

2.5 Chapter summary

In this chapter, various methods have been described to extract features of questions or documents before being utilized by the machine learning algorithms to perform classification. Rule-based methods involve the creation of specific conditions and counting the presence or absence (binary) of those conditions which is not universally applicable to any datasets. TF-IDF or class-based term weighting schemes provide weights to words based on the frequency and concentration. The disadvantage of these schemes is the sparsity of the BoW vector representation. LDA can be described as the clustering of words into topics with a common meaning. However, it does not consider rare word occurrences and fails for short texts (in this context, questions) since questions, in general, do not have sufficient instances of tight word co-occurrences. WNTM, on the other hand, performs topic modeling by considering co-occurrences among words in the corpus. It provides importance to rare words and all word connections but fails to generalize since it contains all connections among words, including possible irrelevant connections. The use of symmetric priors prompted the development of A-LDA and the presence of high-frequency words lead to W-LDA. Nevertheless, word-based approaches are inefficient in encapsulating the meaning of a question unlike a phrase-based approach (in particular, LPTM). However, the under-representation of regexes is a limitation. On the other hand, with the prevalence of deep learning approaches such as LSTM and CNN that have been applied for text classification, the recently emerging graph networks possess the ability of representing a document holistically. However, there is more scope for incorporation of further heterogeneity and diversified graph tensors.

Chapter 3

The Sorted TF-IDF for Enhanced Frequency-based Question Feature Representation

Expertise in a domain of knowledge is characterized by a high fluency for solving problems within that domain and a greater facility for transferring the structure of that knowledge to other domains. Deliberate practice and the feedback that takes place during practice activities serve as gateways for developing domain expertise [103, 104]. However, there is a difficulty in consistently aligning feedback about a learner's practice performance with the intended learning outcomes of those activities —particularly in situations where the person providing feedback is unfamiliar with the intention of those activities [105]. To address this problem, this chapter proposes a sorted TF-IDF (s.TF-IDF) model to automatically label opportunities for practice (assessment questions) according to the learning outcomes intended by the course designers. As a proof of concept, a reduced version of Bloom's Taxonomy has been designed to define the intended learning outcomes. This chapter describes the proposed s.TF-IDF algorithm for feature extraction and the associated single course dataset used for evaluation. The detailed

Part of this chapter has been published as S. Supraja, K. Hartman, S. Tatinati, and Andy W. H. Khong, "Toward the automatic labeling of course questions for ensuring their alignment with learning outcomes," in *Proc. 10th Int. Conf. Educational Data Mining (EDM)*, 2017, pp. 56–63.

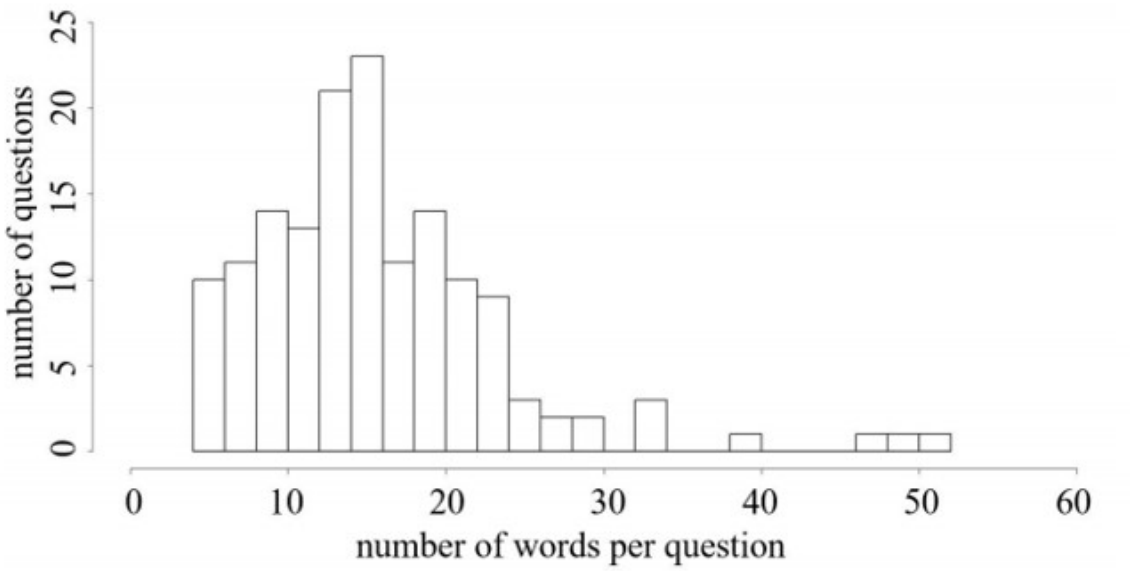


Figure 3.1: Length distribution of questions in the DSP dataset.

analysis of experiment results will be presented in Chapter 4 to facilitate its comparison with the algorithm that will be presented in that chapter.

3.1 Single course dataset for evaluation

The corpus of 150 digital signal processing (DSP) questions underlaying this work aggregates questions published in well-known textbooks [106–108], obtained from online question banks and generated by an instructor of an undergraduate DSP course. The pool of course questions are extracted from a repository of assignment, homework, quiz and exam questions presented to students. All these questions prompt students for a range of answer types, i.e., open-ended, multiple-choice, short-structured and essay. The mean length of the questions was 16.2 words (standard deviation (SD) = 8.01). The frequency distribution of question length in terms of number of words is shown in Figure 3.1.

When looking at every question presented to students over a semester, the subject matter expert identified the number of questions corresponding to *Knowledge*, *Application*, and *Transfer* as shown in Table 3.1. Just by labeling the course questions, the

Table 3.1: Frequency of questions aligned to cognitive complexities

Cognitive complexity	**Frequency (number of questions)**
Knowledge	62
Application	131
Transfer	23

subject matter expert realized how misaligned the course's learning outcomes were with its assessment practices. A significant emphasis on *Application* questions was expected, but the absence of *Transfer* questions was surprising. Of those 23 *Transfer* items, most were presented during the final exam. One of the stated learning outcomes of the course was to prepare students to flexibly transfer course content to novel problems and new situations. However, waiting until the final exam to present students with such opportunities denied them actionable feedback during the semester. In response to the pre-processing labeling efforts, the subject matter expert then added 42 new *Transfer* questions throughout the course for the next semester.

Feature extraction procedures were implemented for all 150 questions. 105 questions (70%) were randomly selected to train the machine learning algorithms while the remaining were used to test the model. Questions used for testing the model is a fixed set that has been held out once due to the small size of the entire dataset. This practice will be followed in the experiments of the remaining Chapters 4, 5, and 6. A subject matter expert manually labeled all of the training questions. To obtain a ground truth when evaluating the classification performance of the testing set, the same instructor manually labeled the test set. Although the questions covered a range of DSP topics such as discrete-time signals and z-transform, the labeling was done solely based on the learning outcome the instructor intended to measure with each question without any analysis of the content.

3.2 Design of customized taxonomy

To comply with the Accreditation Board for Engineering and Technology's (ABET) accreditation criteria and prepare students for the workforce, all engineering programs

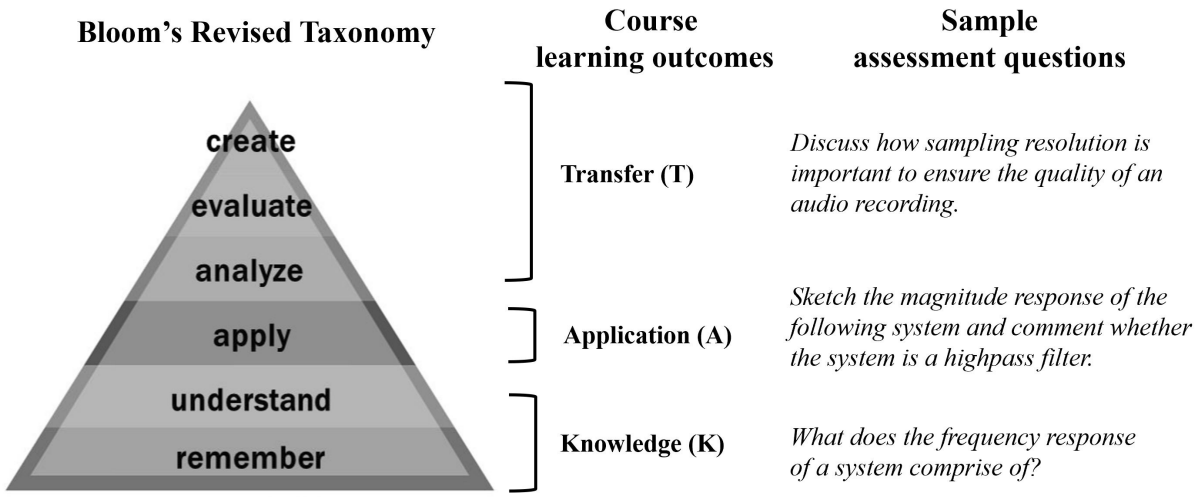


Figure 3.2: Overview of categories for question classification.

must ensure students complete courses that collectively develop eleven categories of learning outcomes [109]. To facilitate students in achieving these outcomes by the end of a program, courses implement learning activities that rely on remembering concepts, applying existing knowledge to tackle problems, and generating viable solutions to real-life scenarios [110]. As a required course in many electrical engineering programs, the design of DSP courses is crucial for achieving compliance with ABET's educational standards. The best practices of outcome-based teaching and learning suggest that course designs should identify the learning outcomes and the assessments that measure those learning outcomes before designing the course's learning activities [105]. For DSP courses, learning activities can include building circuits [111], Matlab programming [112] and laboratory experiments [113] which can be implemented to fulfill a set of learning outcomes. For a host of historical, structural, and policy reasons, the design of DSP courses often deviates from the best practices. Many courses were originally designed decades ago and incrementally updated with new content and assessment items. This slow evolution often translates to the measurement of student outcomes being grafted onto a course that was originally designed for its coverage of content [114].

Integrating Bloom's Taxonomy into ABET's accreditation criteria creates a space that maps the assessment items to the learning outcomes. To this end, this thesis focuses on the space that deals with knowledge facts (K), applying a learned concept (A), and

transferring the learnt concept to another domain (T) as seen in Figure 3.2. Hence, a customized version of Bloom's Revised Taxonomy has been developed in this work to stratify the content related outcomes. The taxonomy starts with the recollection of information at the lowest level, ascends to the application of knowledge, and peaks with creative outcomes [115]. The reduction to Bloom's Revised Taxonomy reflects the philosophy of the DSP course instructor who viewed the different levels of reasoning about course content as pertaining to knowledge, application and transfer. These three categories form the basis of the subsequent analysis and are consistent with ABET's engineering education accreditation criteria. The Biggs' structure [116] maps the Bloom's Taxonomy for formative assessments to the final grades of the summative assessment. Figure 3.2 provides an overview of the customized design of learning outcomes along with sample questions under each category. AQC is performed according to these categories, independent of the actual content or subject matter.

In particular, K-type questions, in general, require students to recall DSP facts, e.g., "How does FFT differ from DFT?" However, A-type questions require students to apply their DSP knowledge to solve a closely related problem, e.g., "Determine the step response of an LTI discrete-time system characterized by the following impulse response." On the other hand, T-type questions require students to transfer their understanding of DSP principles to analyze, evaluate, and generate real-life situations not in the learning materials, e.g., "Why is DSP a natural choice for processing voice information in a digital radio telephony system?" At this juncture, there is a need for an algorithm to automatically label formative assessment questions so that the right set of practice opportunities can be provided throughout the course and prepare them for the final summative assessment. Indeed, there will be a negative impact resulting from the misalignment of formative assessment questions since this can undermine both student motivation and learning [117]. Therefore, there is a need to prevent such misalignment by minimizing misclassifications of formative assessment questions into the various learning outcome categories.

3.3 Question pre-processing

Given a question, conventional pre-processing includes the removal of symbols, diagrams, equations, numbers, and punctuation marks. All characters are set to lower case [118]. While document classification and domain-specific AQC require the removal of stop-words, it has been shown that these words, along with their various forms, possess crucial information pertaining to a sentence's structure [119] that are useful for domain-agnostic AQC [12, 120]. Therefore, conventional stop-word removal, stemming, and lemmatization are not performed. The pre-processed question consisting of N_w words is then defined as

$$\widetilde{q} = \{w_1, w_2, \ldots, w_{N_w}\}, \tag{3.1}$$

where w_i denotes the ith word.

3.4 The proposed s.TF-IDF algorithm

To address the limitation of the large feature space in the conventional TF-IDF BoW method, the proposed sorted TF-IDF (s.TF-IDF) algorithm aims to determine a suitable feature space that is being fed into the machine learning techniques by simulating the grouping of words into clusters. Figure 3.3(a) depicts the conventional TF-IDF weights per question in the order of the vocabulary. For illustrative purpose, only a few values have been plotted instead of all N_V values, where N_V denotes the word vocabulary size. Alternatively, the feature space could be viewed in terms of the distribution of the nature of the words which is reflected by the TF-IDF scores. A high TF-IDF score indicates that the word belongs to one or few questions and occurs many times within those few questions, implying uniqueness. Conversely, a low TF-IDF score indicates that the word is significantly common among the corpus of questions. Hence, there are several alternatives in re-arranging and interpreting the TF-IDF weights in each question.

Since a zero weighted word implies no significance of that word in a question, the importance lies in the non-zero weighted words. By ignoring all the zero weighted words, one option is to consider the remaining words according to the order of how they appear

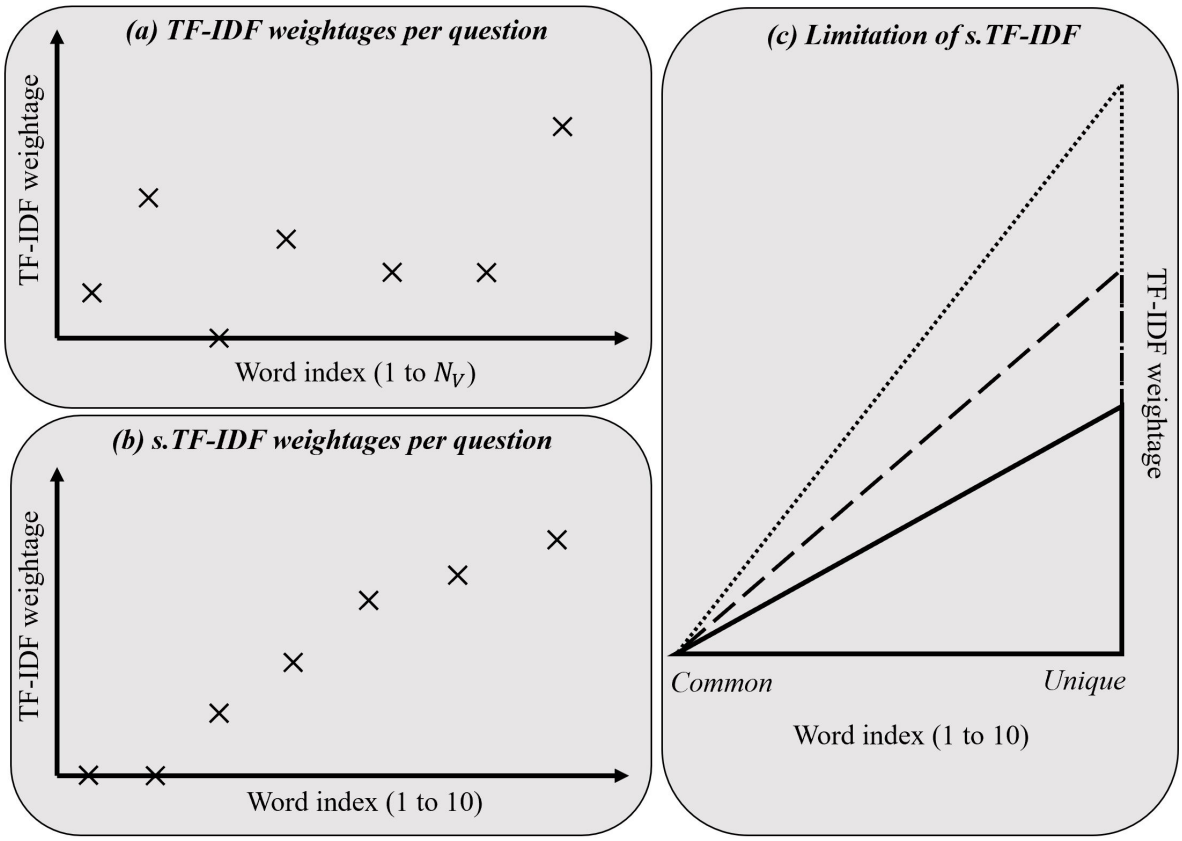


Figure 3.3: Illustration of (a) TF-IDF, (b) s.TF-IDF, and (c) the limitation of s.TF-IDF.

in each question. An alternative option is to consider the remaining words according to the BoW order. The vector length for each question can be based on the question with the largest non-zero vector length by padding zeros if the question length is lesser than the longest question. Alternatively, if there is a need to reduce the dimension of the vector to a fixed smaller size, the concept of max pooling or average pooling can be applied [121]. This can be achieved by gathering the non-zero terms for each question using either the question word order or the BoW order and computing the maximum value or average value for every group of words, which will be considered as a feature. However, both these methods yield a much lower classification performance than the conventional BoW method. In addition, this random order does not indicate a uniform way of comparison across the various questions which have different order of words, as well as, different presence of words according to the vocabulary listing.

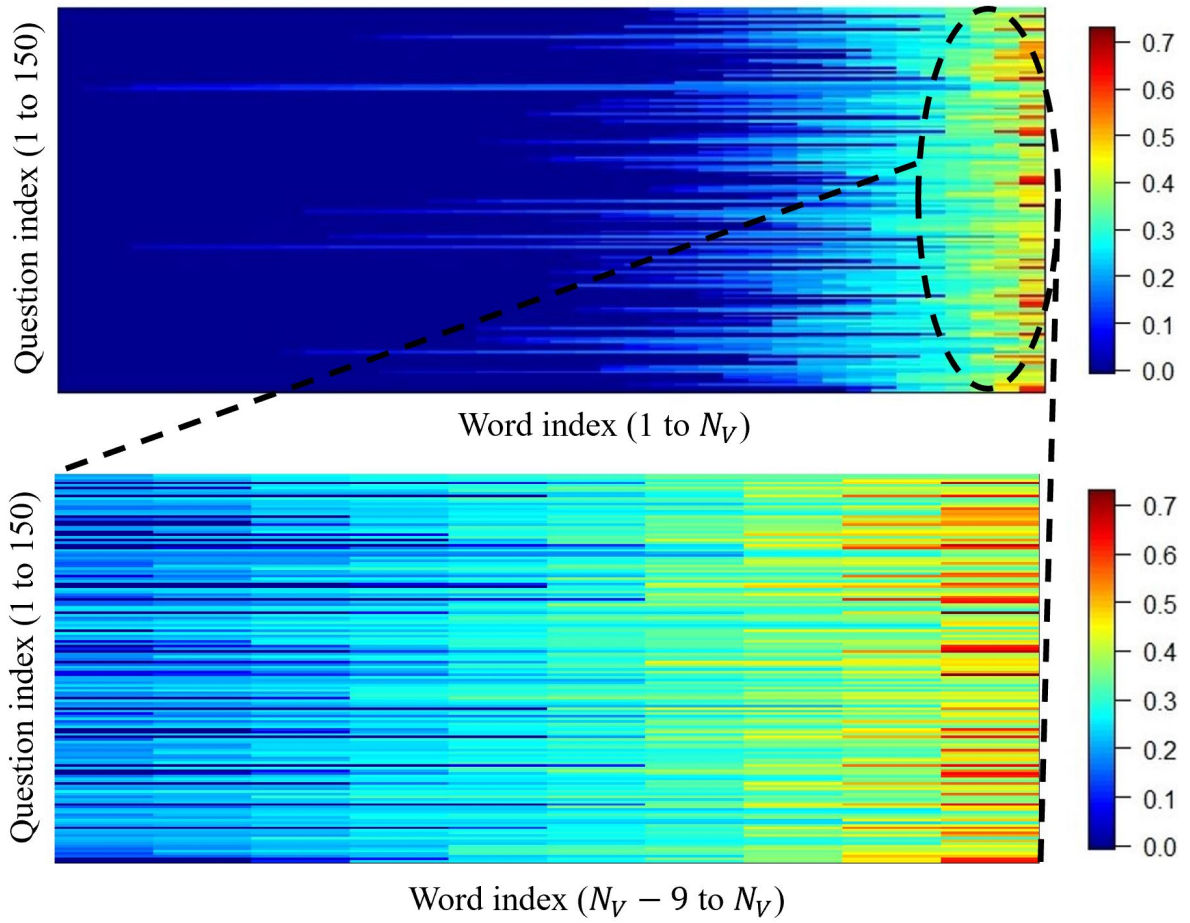


Figure 3.4: Heatmap of sorted TF-IDF weights (above) and Top 10 TF-IDF weights zoomed in (below).

To reflect the changes from highly unique to most common type of words in each question, the proposed s.TF-IDF sorts the non-zero weights of each question in ascending order which indicates the increasing degree of the concentration of word weights. Figure 3.3(b) depicts the sorted TF-IDF weights per question. The data points that have been plotted can be visualized in terms of a triangle with increasing level of uniqueness of words in each question. The top portion of Figure 3.4 illustrates the sorted weights in each question through the heat map with increasing colour gradient. For analysis, an arbitrarily selected top 10 weights have been chosen as the average number of significant non-zero weights in each question. The bottom portion of Figure 3.4 depicts the heat map for the top 10 weights in each question. The column in the extreme right represents

Algorithm 1: Formulation of s.TF-IDF

Input: Matrix $\mathbf{B}$ of TF-IDF values
Output: Sorted TF-IDF matrix $\mathbf{A}$ with top 10 weights
for $i \in N_Q$ **do**
  Sorted $\mathbf{B} \leftarrow a_1^{(i)} < a_2^{(i)} < \ldots < a_n^{(i)}$ using (3.2)
  $\mathbf{a}^{(i)} \leftarrow [a_1^{(i)}, \ldots, a_n^{(i)}]$, $n = 1, \ldots, 10$
end

the highest weight for each question corresponding to the feature of high uniqueness of terms such that with decreasing (sorted) word index, each feature indicates decreasing degree of uniqueness or increasing degree of commonality of terms. Compared to TF-IDF, the feature space has been transformed from the nature of words according to the BoW representation to the distribution of the nature of words.

Algorithm 1 outlines the formulation of s.TF-IDF for AQC. The heatmap in the bottom panel of Figure 3.4 is represented by an $m \times n$ sorted matrix $\mathbf{A}$ where $m = 1, \ldots, 150$ denotes the question index since there are 150 questions while $n = 1, \ldots, 10$ denotes the word index since the top 10 weights are taken compared to the entire set of vocabulary (N_V) as depicted in Figure 3.4. Defining $\mathbf{a}^{(i)} = [a_1^{(i)}, \ldots, a_n^{(i)}]$ as the ith row in $\mathbf{A}$, the sorted values require the condition

$$a_1^{(i)} < a_2^{(i)} < \ldots < a_n^{(i)}. \tag{3.2}$$

On the other hand, if a heatmap is plotted for the conventional TF-IDF, it can be represented by an $m \times r$ matrix $\mathbf{B}$ where $r = 1, \ldots, N_V$ denotes the word index for the whole vocabulary of unique words in the corpus since the BoW representation is being adopted. The DSP dataset consists of a total of 546 unique words, hence $N_V = 546$. In comparison with matrix $\mathbf{A}$ in which each element a could correspond to any particular word, for any row in $\mathbf{B}$, the individual elements b contain values based on a fixed set of column names. For instance, considering the ith row in $\mathbf{B}$ as $\mathbf{b}^{(i)} = [b_1^{(i)}, \ldots, b_r^{(i)}]$ and the jth row in $\mathbf{B}$ as $\mathbf{b}^{(j)} = [b_1^{(j)}, \ldots, b_r^{(j)}]$, both $b_1^{(i)}$ and $b_1^{(j)}$ refer to the TF-IDF weight assigned to the same word in two different questions. Performance of the proposed s.TF-IDF algorithm will be presented in Chapter 4 for comparison with that of topic modeling

Table 3.2: Comparing s.TF-IDF with TF-IDF

Method	K		A		T		Macro-average	
	ELM	SVM	ELM	SVM	ELM	SVM	ELM	SVM
TF-IDF	0.421	0.400	**0.537**	0.364	0.200	**0.250**	0.386	0.338
s.TF-IDF	**0.857**	**0.800**	0.513	**0.596**	**0.333**	0.174	**0.583**	**0.585**

approaches in that chapter.

One of the limitations of this proposed approach is that although the feature space is equivalent in terms of comparing the gradient reduction from unique to common, the actual gradient being reflected across various questions is not equivalent. Figure 3.3(c) shows the various top 10 TF-IDF weights that can be present in three questions, which are represented by the three triangles in dotted, dashed, and solid lines. For instance, the most unique TF-IDF weight in one question can be as high as 0.9 while another question only achieves 0.4. This particular 0.4 weight may represent a common word in the former question. The ambiguity is due to same words having different weights; such discrepancy exists since each question has its own scale of uniqueness or commonality of words that appear in it. Although s.TF-IDF considers the distribution of the nature of words as the feature space, it fails to consider the actual words that belong to each feature.

3.5 Results and discussion

The effectiveness of s.TF-IDF is compared against TF-IDF for the DSP dataset. Table 3.2 shows the F1 scores (individually for each class label, as well as macro-average) for the above methods. It can be seen that the choice of the top set of weights achieves better classification performance due to the filtering of important words.

The desired expectation is that s.TF-IDF should simulate a grouping of words such that each grouping has a weight pertaining to a fixed set of words. However, this is not achieved as the groupings are not uniquely defined across all the questions. As will be seen in Chapter 4, while s.TF-IDF achieves a higher classification performance compared to TF-IDF through the alteration of the feature space and consequently the reduction of vector dimension, the distribution of weights still does not guarantee that a question

uniquely belongs to either class label K, A, or T.

3.6 Chapter summary

The s.TF-IDF algorithm has been proposed to determine a feature space that is more suitable than TF-IDF weighting. This is achieved by sorting the weights in ascending order according to the spread of the nature of words. By selecting the top 10 weights, this in turn reduces the dimension of the vector. While the proposed s.TF-IDF outperforms that of TF-IDF, the concentration of the nature of words is unrelated to the actual words themselves. To achieve a uniform comparison across the questions by creating clusters of words that are similar, topic modeling will be explored in the next chapter.

Chapter 4

The Customized Question WNTM Considering Word Co-occurrence Redundancy in Topic Modeling

To deliver on the potential of outcome-based teaching and learning for engineering education, it is important for engineering courses to provide students with various types of deliberate practice opportunities that are aligned with the program's learning outcomes. Working from these requirements, the design and measurement intentionality of a DSP course has been increased. To align the course's learning outcomes more constructively with its assessment measures, the process of classifying DSP questions has been automated by introducing a model that integrates topic modeling and machine learning. In this chapter, the effect of pre-processing procedures in terms of stop-word selection and word co-occurrence redundancy issue in question classification inferences has been explored. A customized variant of the word network topic model (WNTM), which is able to use its pre-classified DSP questions to reliably classify new questions according to the course's learning outcomes has been proposed. This chapter describes the technicalities of the proposed q-WNTM algorithm that considers word co-occurrence redundancy.

Part of this chapter has been published as S. Supraja, S. Tatinati, K. Hartman, and Andy W. H. Khong, "Automatically linking digital signal processing assessment questions to key engineering learning outcomes," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2018, pp. 6996–7000.

The experiment results of q-WNTM along with s.TF-IDF described in Chapter 3 will be presented. The same DSP dataset has been used in both chapters.

4.1 Problem formulation for AQC with topic modeling

Classification of the pre-processed question $\widetilde{q}$ (defined in (3.1)) via topic modeling can be achieved by first generating a feature vector of topic probabilities $\mathbf{q} = \{P(z_1), P(z_2), \ldots, P(z_{N_Z})\}$, where $P(z_j)$ and N_Z are the probability of the jth topic z_j and the total number of topics, respectively. Defining N_L as the total number of class labels, AQC is formulated as estimating the relationship between $\mathbf{q}$ and each class label y_c such that the predicted class label for that question is given by

$$\widehat{y} = \underset{c \in \{1,2,\ldots,N_L\}}{\operatorname{argmax}} P(y_c|\mathbf{q}). \tag{4.1}$$

The probability $P(y_c|\mathbf{q})$ is estimated based on the ten-fold cross-validation performed by the machine learning algorithm during the training process. The argmax operation is applied to compare among the class labels and select the class label that appears as the prediction for the maximum number of validation runs. Achieving good AQC performance, therefore, relies on the computation of suitable $P(z_j)$ for the construction of $\mathbf{q}$.

4.2 Customized stop-word selection

AQC is a unique domain of interest in comparison to conventional document or short-text classification applications due to the unique structure of questions, thus necessitating a careful selection of words to represent a question. Common practice of pre-processing in text classification requires the sieving of content-based words to convey an explicit meaning (i.e., movie genre, news articles topic, subject or type of questions, e.g., location/numerical), in turn, removing stop-words [32, 122]. However, stop-words could be

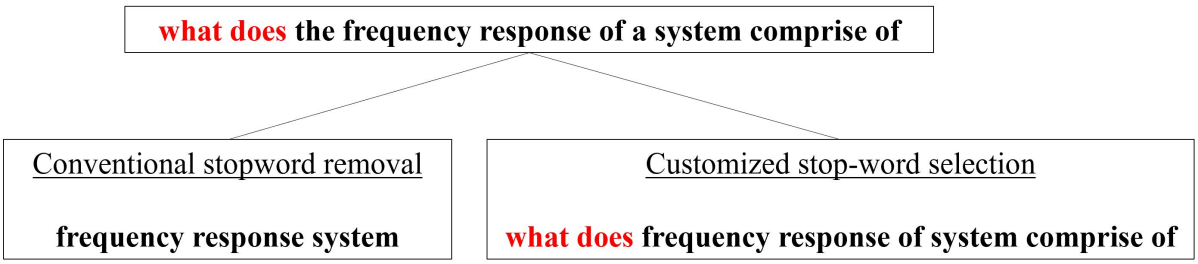


Figure 4.1: Comparison of conventional versus customized stop-word removal.

vital to maintain the intended meaning of a question. From a linguistics perspective of analyzing question structures, the main verbal cues in a question are question words (i.e., *why, what, when, where, which, who, how*) and with verbs (e.g., *explain*, *describe*, *state*) that differentiate among question types such as probing, rhetorical, leading/reflective, true/false or those that express different levels of cognitive complexities [123,124]. Therefore, a question is best represented by various types of word feature set combinations such as an action verb followed by an object, content, or subject and which concludes with a context [125]. Alternatively, the presence of a question word either as the headword (start of a question) or as part of a question that is identified through a parse tree could characterize a typical question [126–128] and detects the type, as well as, focus of a question [129].

Topic models have been used to classify general short texts by sieving out the content words in a document. However, these topic models cannot be directly applied to question classification with respect to stop-word removal. In conventional document classification, stop-words include question words, prepositions, articles, conjunctions and action verbs [130, 131], which are generally defined as high-frequency words that do not contribute to a document's subject matter. However, when performing question classification according to learning outcomes, commonly identified stop-words become key to determining the proper category. There is a need to reduce the list of stop-words for removal to only those words which functionally do not contribute to classifying the questions [132, 133].

The dataset described in Section 3.1 is used for analysis after basic data cleaning

is performed. To preserve the essence of a question, a pre-defined stop-word list was generated. For this particular dataset, only four words *the*, *and*, *a*, and *an* were removed from the corpus. The rationale behind the specific choice of these words is that these words do not affect the ability to decipher the label of the question. To elaborate on the impact of the specific choice of stop-words in an illustrative example of a question "what does the frequency response of a system comprise of", the phrase *what does* serves as a signifier of its class label *Knowledge* (K) as seen in Figure 4.1. The words in red refer to the key terms to define the class label of this question. Assuming that the conventional stop-word removal [130] is being applied, the question will appear as "frequency response system", resulting in the difficulty of identifying its class label. Conversely, the specific choice of stop-word removal transforms the question into "what does frequency response of system comprise of" which does not limit the ability of the algorithm to categorize it as a K-type question.

4.3 The proposed q-WNTM algorithm

There is a need to differentiate between the objective of AQC according to domain-agnostic learning outcome categories versus conventional document classification according to the subject matter. Hence, by applying WNTM, it includes a combination of all word types without considering any possible redundancy of certain word combinations. In this work, inspired by the pseudo-corpus scaling down procedure into topical and general words [58], two different word types have been identified. One type can be defined as content-agnostic words which refer to general words such as verbs and conjunctions (e.g., *what*, *explain*, *how*, and *state*). The second type can be defined as content words. In the context of the DSP dataset in Section 3.1, these words include *filter*, *DTFT*, *Fourier*, and *signal*.

With the above example, it becomes possible to identify the label of a question based on the combinations of content words and content-agnostic words. Using examples of pairs of content-agnostic words, the phrase *explain what* signifies a lower level of thinking compared to the phrase *explain why* which requires a detailed explanation. With respect

Algorithm 2: Formulation of q-WNTM

Input: Set of unique words U in a question
Output: Pseudo-question $\mathfrak{P}$ ignoring content-content word combinations
for $w_i \in U$ **do**
  $U \backslash \{L\}$
  $\mathfrak{P} \backslash \{X\}$ if $w_i \in X$
  $\mathfrak{P}$ if $w_i \in C$
end

to combinations of content-agnostic and content words, the phrase *find DTFT* requires a straightforward computation while *prove DTFT* requires, in general, a more complicated derivation. However, when comparing the phrases *phase response* and *magnitude response* which both consist of content word combinations, there is less significance in observing the co-occurrence of content words when determining the level of thinking expressed by a question. Hence, there is a possible redundancy of content word combinations which could potentially affect the model's classification performance.

Generally, the top set of words in a topic are the words that are informative of that topic and uniquely associated with that topic. The inclusion of content-content word combinations implies that the presence of the groups of content words could constitute a topic without the need for relationships among content-agnostic words. Removing these content word combinations will dampen the dominating nature of the content words under each topic. The proposed q-WNTM model is built upon the framework of WNTM with the difference being the pre-processing procedure of removing word redundancy specifically in the context of question classification during the implementation. To represent the q-WNTM procedure via a set-theory notation, suppose the set C refers to the content-agnostic words and the set X refers to the content words. If a word $w_i \in X$, in the corresponding pseudo-question $\mathfrak{P}$ (refer to Figure 2.4(right) for examples of pseudo-questions), these content words are excluded which can be denoted by $\mathfrak{P} \backslash \{X\}$. No changes are made if $w_i \in C$.

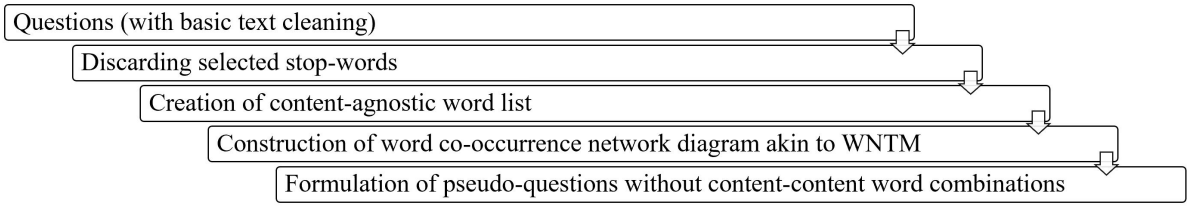


Figure 4.2: Procedure of q-WNTM for question feature extraction.

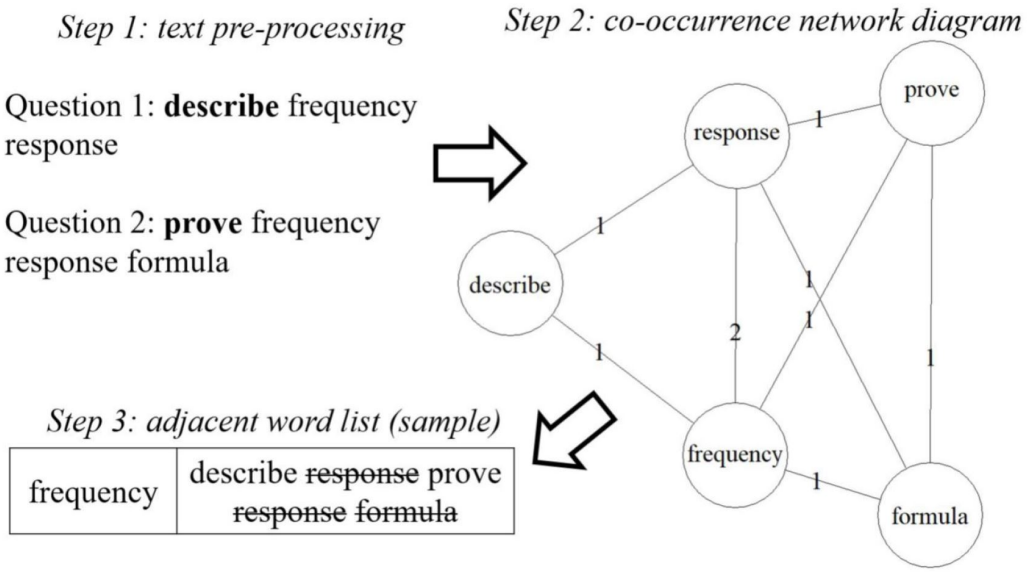


Figure 4.3: Construction of word network diagram in q-WNTM.

4.3.1 Implementation of q-WNTM

Figure 4.2 summarizes the procedure of q-WNTM for question feature extraction. A separate list of content-agnostic words was constructed for the dataset. The corpus's remaining words are considered as content words. With this procedure, there were a total of 109 unique content-agnostic words and 437 unique content words in the dataset. Since q-WNTM complies with the similar procedure of WNTM, the word co-occurrence network diagram as shown in Figure 2.4 (left) is first constructed. After generating the weighted word co-occurrence network diagram, pseudo-questions for every unique word present in the corpus are formulated. Unlike the conventional WNTM algorithm,

for every content word present in the corpus, the other co-occurring content words in the corresponding pseudo-questions are removed. This removal ensures that there will only be combinations of content-agnostic with content-agnostic words, content-agnostic with content words and content with content-agnostic words. This procedure is outlined in Algorithm 2. Subsequently, similar to (2.7), LDA is applied to generate the topic probabilities for each word in the pseudo-question, which are then summed up for the dependent word to then obtain the topic probability vector for the original question.

To highlight this procedure, and with reference to exampler questions in Figure 4.3, words in boldface (i.e., prove and describe) denote the pre-defined set of content-agnostic words while the remaining are content words. With respect to the words in the pseudo-questions corresponding to each content word, it can be observed that the content words have been removed (reflected by the struck out words). By implementing q-WNTM, the topics are now more coherent with respect to the grouping of words. This refers to the clear cut distinction between the grouping of content and content-agnostic words. The top words from each topic now consist of either content-agnostic or content words. On the other hand, LDA and WNTM generated topics which were made up of a mixture of both types of words resulting in poor classification performance as will be shown in Section 4.4.3. As an illustrative example, ten topics were extracted for all the topic models, and the top 10 words were observed to gain insights on the grouping of words under each topic. Table 4.1 highlights the top 10 words in each topic for q-WNTM, with z_j referring to the jth topic. Topics 6, 9 and 10 consist purely of content words while the remaining seven topics consist of content-agnostic words. Subsequent machine learning algorithms would then interpret the range of probabilities assigned to the various topics to perform the question classification according to the three categories. The removal of content-content word combinations in the word network diagram results in mutually exclusive topics. Hence, it facilitates the model in emphasizing Topics 1-5 and 7-8 and allocate the topic probabilities respectively.

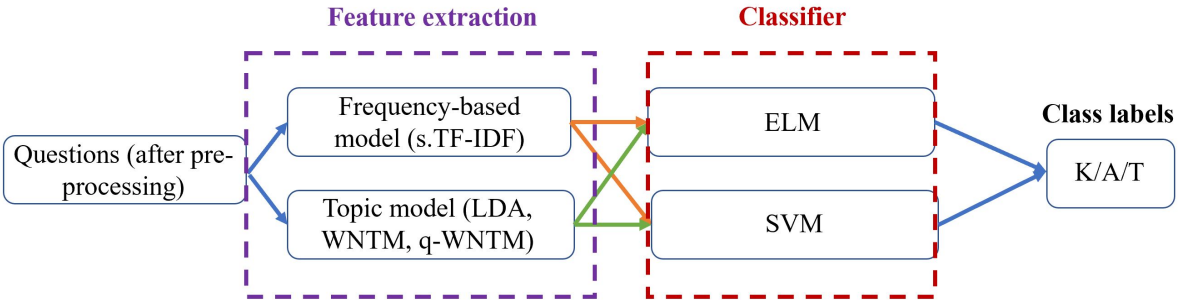


Figure 4.4: Process flow of AQC comparing frequency-based versus topic models for feature extraction passed onto ELM or SVM machine learning classifiers.

4.4 Experiment results and discussion

Figure 4.4 illustrates the process flow of AQC comparing frequency-based versus topic models for feature extraction passed onto ELM or SVM machine learning classifiers.

4.4.1 Performance metric

To evaluate the reliability of classifiers with the subject matter expert's labels, the F1 measure is being used [134, 135]. The F1 measure defined by

$$F1 = \frac{2\mathcal{P}\mathcal{R}}{\mathcal{P} + \mathcal{R}} \tag{4.2}$$

is a harmonic mean of two other metrics: precision and recall. Defining TP, FP, FN as true positive, false positive, and false negative, precision $\mathcal{P}$ refers to the correctness of questions that have been selected as a particular category and can be expressed as

$$\mathcal{P} = \frac{TP}{TP + FP}, \tag{4.3}$$

while recall $\mathcal{R}$ refers to the correctness of selection of the correct category given all the questions that were supposed to be correctly classified and is given as

$$\mathcal{R} = \frac{TP}{TP + FN}. \tag{4.4}$$

Since the minimization of the number of false positives and false negatives was important for accurately assigning questions to the correct labels, the F1 measure is used as the basis for the algorithm comparisons.

4.4.2 Hyperparameter selection

The hyperparameters that require optimal initialization for topic models include the number of topics, prior for questions/pseudo-questions to topic probabilities α, and prior for topic to word probabilities β. An empirically determined $\alpha = 0.1$ and $\beta = 0.01$ were selected as the optimal hyperparameters for this dataset. The number of Gibbs sampling iterations was confined to 2000. The choice of 10 topics is based on an evaluation on the number of topics with a step size of five ranging from 5 to 20. The classification performance was found to be lower with fewer number of topics since most of the words are being clustered together, preventing better segregation. However, having an excessive number of topics results in a sparse matrix as the topics are loosely distributed, resulting in poor classification performance. For this dataset, 10 was chosen as the optimal number of topics for all topic models.

70% of the questions were randomly selected to train ELM. The remaining 30% were used to test the model. A 10-fold cross validation was performed on the training dataset to initialize ELM optimally. A grid search was performed to determine the parameters in ELM and SVM. It was determined that 27 hidden nodes and the sigmoid activation function achieves the best performance for ELM. For SVM, the parameters that achieved the best results corresponded to the sigmoid kernel with a coefficient value of 0.1 and regularization value $C = 1$ according to (2.10).

Table 4.1: Top 10 words for each of the 10 topics in q-WNTM.

z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9	z_{10}
is	of	to	of	of	magnitude	of	in	system	signal
of	in	of	is	in	filter	how	you	output	frequency
what	can	is	by	for	applications	be	with	input	sampling
if	by	would	given	your	response	to	his	response	domain
which	your	what	consider	how	frequencies	describe	of	impulse	transform
following	to	in	find	you	real	used	one	time	fourier
this	out	if	for	may	useful	can	using	signal	analog
find	be	be	where	explain	application	is	your	sequence	discrete
as	will	by	to	is	dft	in	that	ztransform	image
when	using	do	in	one	features	discuss	about	equation	audio

Table 4.2: Comparison of F1 scores for the four methods using both ELM and SVM.

Method	K		A		T		Macro-average		s.d.	
	ELM	SVM	ELM	SVM	ELM	SVM	ELM	SVM	ELM	SVM
s.TF-IDF	0.857	0.800	0.513	0.596	0.333	0.174	0.583	0.585	0.218	0.261
LDA	0.444	0.381	0.941	0.733	0.737	0.718	0.707	0.618	0.204	0.163
WNTM	0.545	0.353	0.800	0.882	0.848	0.821	0.744	0.686	0.133	0.236
q-WNTM	**0.700**	0.609	**0.903**	0.909	**0.923**	0.765	**0.848**	0.775	**0.101**	0.123

Table 4.3: p-values after performing a two-tailed t-test for comparison of topic modeling methods against s.TF-IDF

LDA	WNTM	q-WNTM
0.0277*	0.0277*	0.000426**

* significant at $p < 0.05$ and ** significant at $p < 0.01$

4.4.3 Results and discussions

The F1 scores are compared to evaluate the performance of question classification using various combinations of feature extraction (LDA, s.TF-IDF (described in Chapter 3), WNTM and q-WNTM) and machine learning classifiers (ELM and SVM). Given a model, an individual F1 score for each class label is computed. As described in Chapter 3, these class labels include *Knowledge* K, *Application* A, and *Transfer* T. The macro-average F1 score aggregates the mean of the model's precision and recall values across these class labels and thereafter computes the harmonic mean between them. Table 4.2 shows the F1 measure values (for each individual class label and macro-average) pertaining to the test set for the four combinations with "s.d." denoting the standard deviation. The standard deviation is computed among the three individual F1 scores (pertaining to the three-class labels) for each method using either ELM or SVM.

The aim of computing F1 scores is to differentiate the extent to which each model falsely identifies the true category of a question, thereby hindering the appropriate cognitive level of practice opportunity provided to a student. To provide an example of the impact of misclassification, assuming that an actual A or T-type question is misclassified as K-type, this misestimates a course's prioritization of memorization. The converse implies that students who merely memorized material demonstrated more outcomes than they should have.

Since ELM has shown to outperform SVM for classification in several applications [136], Figure 4.5 illustrates the confusion matrices only for ELM in a grid format with a colour gradient. The actual versus predicted numbers of questions under each category can be seen for the four methods. The TP is reflected via the diagonal elements in each

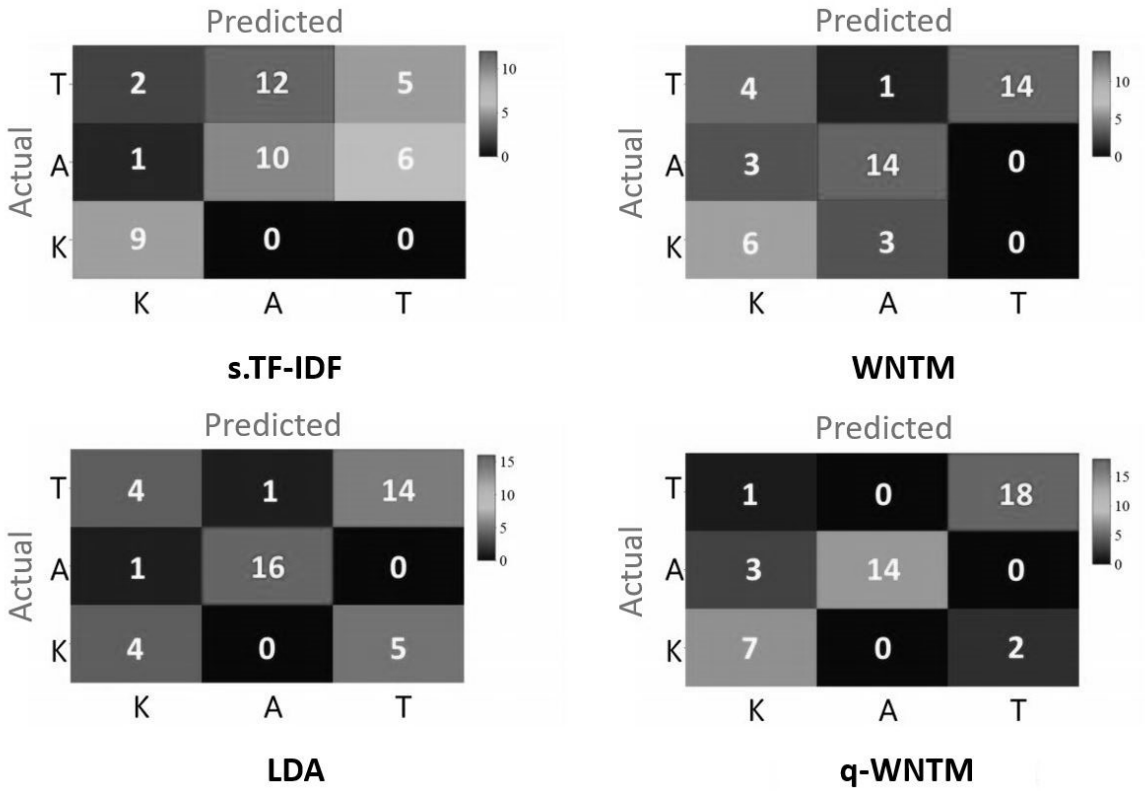


Figure 4.5: Confusion matrices for the four methods using ELM.

confusion matrix. The cells in red indicate good classification if they are present in the diagonal, while the color blue in the diagonal indicates a high rate of misclassification for that class label. It can be seen that q-WNTM, WNTM, LDA, and s.TF-IDF correctly classify 39, 34, 34, and 24 questions out of the total 45 test set questions. In addition, q-WNTM achieves the lowest rate of misclassification as seen via the lowest FP and FN for each class label by summing up the remaining values apart from the diagonal. The cells in red indicate poor classification if they are present as the non-diagonal elements with blue indicating otherwise.

To further evaluate the statistically significant differences, Table 4.3 shows the p-values after performing a two-tailed t-test to compare topic modeling methods against s.TF-IDF. It can be seen that all topic modeling methods achieve statistically significant improvement over s.TF-IDF. In particular, the p-value comparing s.TF-IDF with q-WNTM is significant at $p < 0.01$, indicating the effectiveness of q-WNTM.

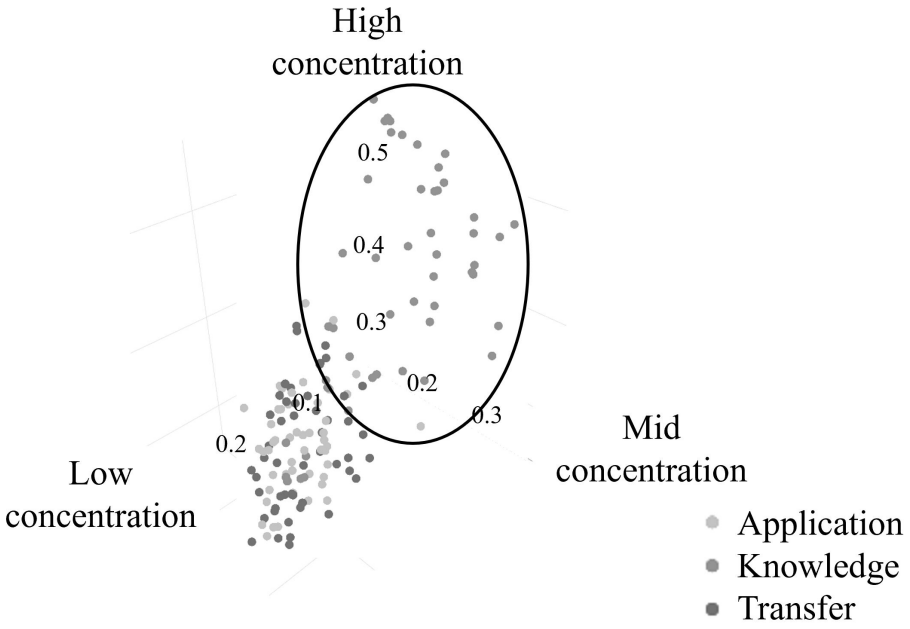


Figure 4.6: Scatter plot for the s.TF-IDF approach.

With respect to the four techniques, results shown in Table 4.2 suggest that the proposed q-WNTM model links assessment questions to learning outcomes more accurately than the alternative models. The proposed s.TF-IDF model achieves the lowest macro-average F1 score of 0.583 using ELM and 0.585 using SVM as shown in Table 4.2. To gain insights into s.TF-IDF, Figure 4.6 depicts the data points corresponding to each question. The red, blue, and green dots refer to A-type, K-type, and T-type questions, respectively. For illustration purposes, the 10-dimensional vector described in Chapter 3 is reduced to a three-dimensional space by taking the average of three or four columns of word weights using the average pooling method. The vector of weights ranges from low (corresponding to common words) to high (corresponding to unique words) for each question. The three axes in Figure 4.6 refer, respectively, to the top few weights in each question (high concentration words), followed by the next few weights (moderate concentration) and the bottom few weights (low concentration). From Figure 4.6, it can be seen that since the groupings of words in the feature space are not defined clearly by the s.TF-IDF approach, the three labels cannot be sufficiently discriminated by ELM or SVM. Although all the questions are represented in an increasing degree of uniqueness,

the features are only in place for the distribution of the nature of words without taking into account that different sets of words could belong to each distribution. This implies that the comparison of word distribution is not performed in the same manner for each question due to the ambiguity of how each question interprets the uniqueness versus commonality of words.

In addition, it can be observed that the K-type questions exhibit better segregation compared to A or T. This correlates with a high individual F1 score for K-type (given by the F1 score of 0.857 using ELM and 0.800 using SVM in Table 4.2) but low F1 values (given by the F1 score of 0.513 and 0.333 using ELM and 0.596 and 0.174 using SVM in Table 4.2) for A and T types respectively which are misinterpreted. One possible reason for the above is that since K-type questions are exceptionally short, this results in several zeros in the vector of the top 10 weights for these questions. Hence, in line with the previous explanation regarding sparsity for the BoW approach, most of the K-type questions consist of the same sparsity trend and thus are easily discriminated against A and T. In addition, the term frequency ratio of each word according to (2.1) in a K-type question is generally higher as the denominator (length of the question) is smaller. Hence, there is a similar pattern of the degree of uniqueness and commonality of words reflected in almost all the K-type questions.

It can also be noted that higher performance in terms of macro-average F1 is achieved by LDA (given the macro-average F1 score of 0.707 using ELM and 0.618 using SVM in Table 4.2) compared to s.TF-IDF (given the macro-average F1 score of 0.583 using ELM and 0.585 using SVM in Table 4.2). However, in terms of class-level individual F1 score, LDA achieves the lowest F1 score of 0.444 using ELM and 0.381 using SVM for K-type questions as seen in Table 4.2. LDA classifies the other two categories of question types more accurately with F1 scores of 0.941 and 0.737 using ELM and 0.733 and 0.718 using SVM as shown in Table 4.2 for A and T types respectively. This modest improvement of LDA over s.TF-IDF underscores the limitations of using LDA for short texts. From the scatter plot in Figure 4.7, it can be seen that A and T types are significantly more segregated while K-type questions are scattered across the feature space. To construct this diagram for illustration purposes, the ten-dimensional topic vectors per question is reduced to a three-dimensional space. Hence, instead of representing each question as a

vector of ten topics, each question is represented as a vector of three topics instead.

The reason for the inability of K-type question being clearly distinguished is due to the short questions lacking question-to-word co-occurrences to a greater extent compared to A and T question types which are longer. Nevertheless, compared to long documents for which LDA was developed, all questions are considered as short text. With reference to the top words under each topic generated by LDA, each topic contains a mixture of both content and content-agnostic words, especially when content words such as *fourier*, *system* and *signal* dominate as the top few words. Due to overlapping topics occurring in LDA, LDA is unable to segregate the words and form clearly defined topics with a common meaning/interpretation reflected by each topic.

The inability of LDA to perform well for questions and the improvement in classification performance by applying WNTM can be seen in Table 4.2. WNTM as a feature extractor achieves a higher macro-average F1 score of 0.744 with ELM and 0.686 with SVM classifier compared to the macro-average F1 score for LDA (0.707 using ELM and 0.618 using SVM), suggesting the importance of using features derived from word level co-occurrences when modeling topics associated with short texts. Figure 4.8 depicts the scatter plot for WNTM with reduced number of topics, in a similar manner to that for LDA in Figure 4.7. Better segregation of data points are exhibited for WNTM compared to LDA. In LDA, since the rare words are being ignored, the words in each topic are the reflection of the commonly occurring words. Many questions have significantly high, i.e., close to 0.95 probability given to one topic and the remaining topics are assigned probabilities that are close to zero, increasing the sparsity within the 10-dimensional vector. This implies that each question is made up of only one or two topics, defeating the purpose of establishing that a question consists of a mixture of topics. LDA is unable to capture this mixture as it does not take the rare words into account.

In comparison, for WNTM, rare words are presented in the pseudo-questions whenever a co-occurring word is seen, implying a greater emphasis on the rare words when applying WNTM. It is not possible to directly co-relate the words present in each topic to the actual questions as WNTM calculates the topic probabilities per question based on the inference from the pseudo-questions. However, it is possible to note that rare words appear much

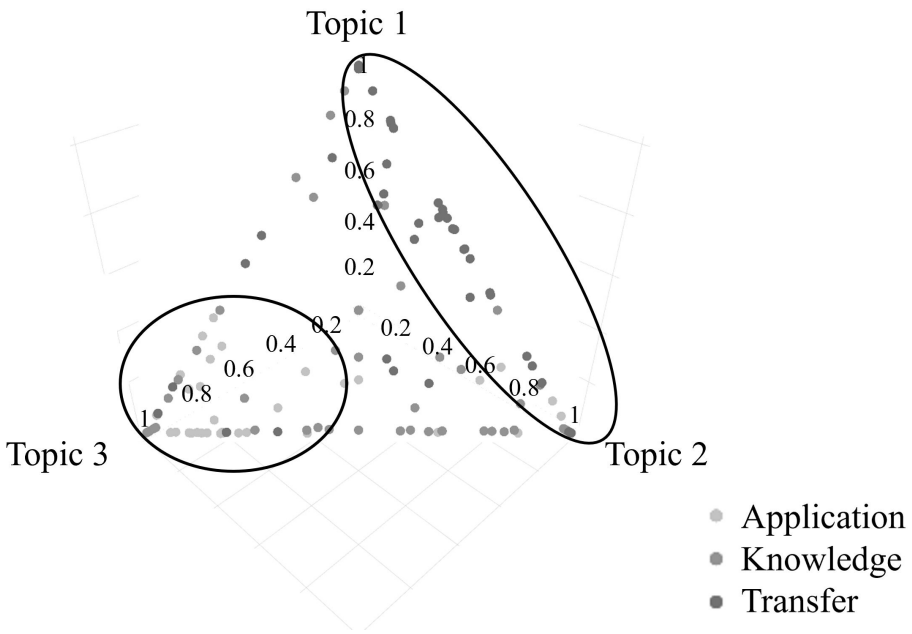


Figure 4.7: Scatter plot for the LDA approach.

more prominently than in LDA. Words such as *advantage* and *useful* have low frequency of occurrence in the corpus of questions but are visible in the top word listing of WNTM unlike in LDA. Hence, by observing the topic probabilities per question, they are more widely distributed as each question now covers more topics due to the coverage of rare words. Nevertheless, to further avoid overlaps in the data points as reflected in Figure 4.8, there is a need to clearly separate the grouping of words.

The solution to this lies in the approach to eliminate redundant word co-occurrences. With reference to results shown for q-WNTM in Table 4.2, the impact of excluding co-occurring content words achieves a high macro-average F1 score of 0.848 and a low standard deviation of 0.101 by using the ELM classifier and 0.775 macro-average F1 score with the low standard deviation of 0.123 by using the SVM classifier. q-WNTM promotes the prevalence of content words to be represented by the surrounding content-agnostic words only and prevents the presence of other content words. With this approach, it is possible to segregate content words into three topics and the remaining content-agnostic words into seven topics as seen in Table 4.1. For instance, to interpret how the top words in each topic could be linked to the original questions, an example of a particular topic

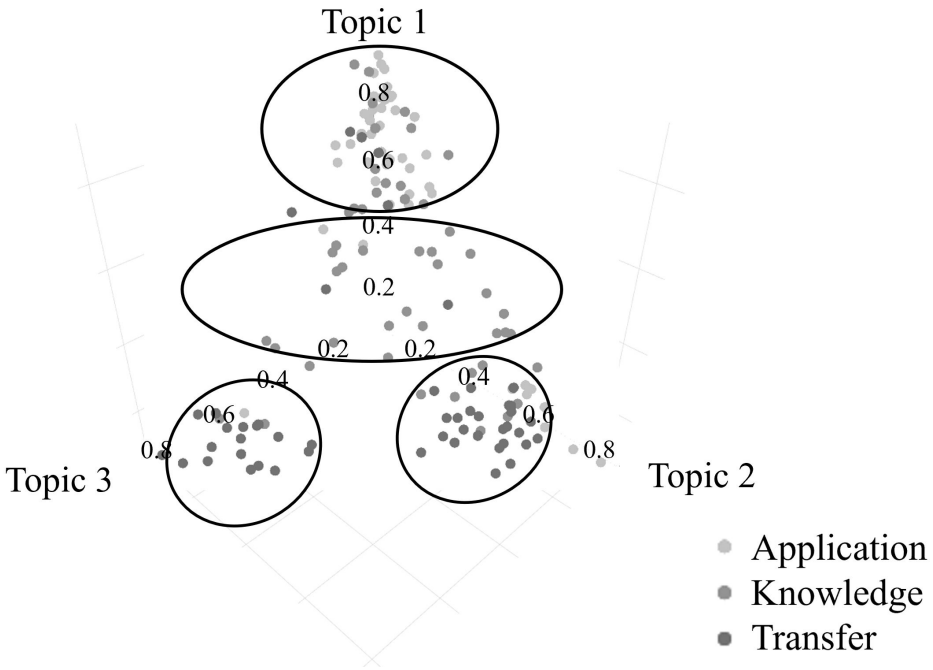


Figure 4.8: Scatter plot for the WNTM approach.

(Topic 7) which consists of top words such as *describe*, *how*, *discuss*, *can* and *used* is taken. This topic can be interpreted as words related to questions that require students to perform inferences. The general trend of questions which gain a high probability for this topic has the structure of "describe/discuss how ___ can be used for ___". This trend is closely related to T-type questions.

The improvement in the results of q-WNTM over WNTM is due to the relative proportion of word occurrence for a question being held as constant since words in the original question are not deleted. However, the major difference is due to the computation of the topic probabilities in the pseudo-question for every content word. Considering that there are 437 content words compared to 109 content-agnostic words, if the co-occurring content words were not excluded, topic probabilities in the pseudo-question for each of the 437 content words would be skewed towards the mixture of majority of content words, resulting in a content-based representation instead of representing each question by the words related to the learning outcome it exhibits. However, in q-WNTM, topics which contain only the content words are assigned low probabilities since for a content word, higher probability is allocated towards the surrounding content-agnostic words. Hence,

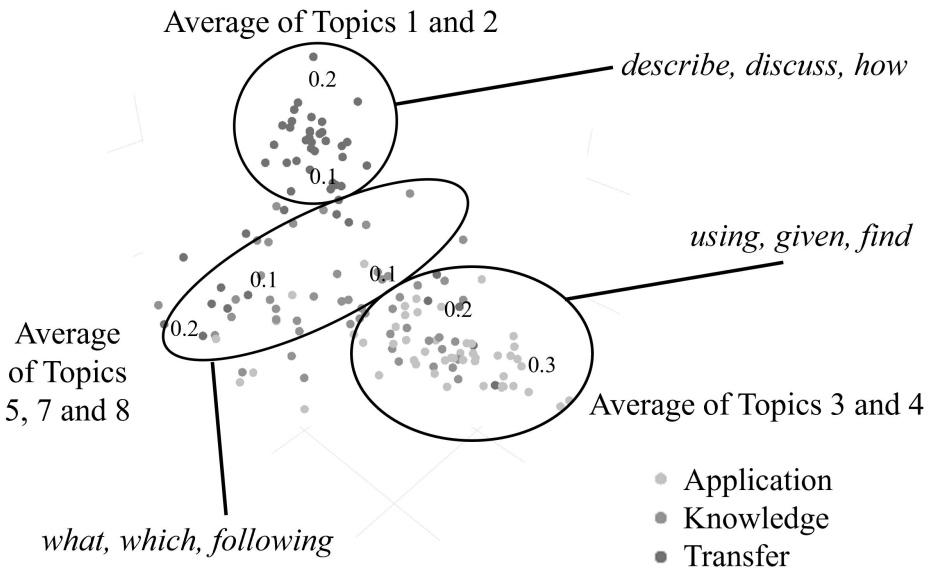


Figure 4.9: Scatter plot for the q-WNTM approach.

the importance for combinations related to solely content words are diminished. As seen in the confusion matrix of q-WNTM in Figure 4.5, almost all questions in the test dataset are correctly classified (diagonal values). Results obtained with q-WNTM highlight that for this dataset, the proposed stop-words list and the consideration of word combination redundancy is necessary for enhancing the question classification performance. Figure 4.9 illustrates the scatter plot for q-WNTM. Since there is an enhanced segregation of words, significant words that identify the categories are mentioned beside each label. However, unlike how LDA and WNTM were plotted based on three topics and since q-WNTM is able to filter out the content words, the three topics containing only content words were not taken into consideration for the plotting. The remaining seven topics were taken and average values for every two or three topics were taken for plotting purpose; only the columns with content-agnostic words contain significant probabilities.

4.5 Chapter summary

A different set of stop-word selection for removal has been proposed for question classification. The proposed q-WNTM algorithm incorporates the impact of irrelevant word

combinations. Although content-agnostic words contribute largely to a question's class label in the context of domain-agnostic AQC, the content words cannot be completely removed as they establish the relationships between content and content-agnostic words. However, since question classification is not based on the subject matter (i.e., pertaining to domain-specific class labels), content-content word combinations have been removed to reduce ambiguity of topics.

The F1 scores have been compared for all four methods of feature extraction along with ELM and SVM classifiers. As one method improves over the other to generate meaningful features to represent the questions, the corresponding improvement of classification results can be observed from s.TF-IDF to LDA to WNTM and finally to q-WNTM. The purpose of plotting scatter plots is to provide a form of visualization of how each question can be represented by a data point in the respective feature spaces for each method. The clustering of data points using the same colour denotes the segregation of the various classes performed through the feature extraction procedures. Subsequently, when ELM or SVM classifies the questions into the various class labels, the features extracted by q-WNTM achieve the best performance.

The main limitation of q-WNTM is that it is dataset-specific such that the list of content-agnostic words requires manual effort of hand-curation based on observation of each question. Chapter 5 introduces a generalized topic model that is not only applicable for a single dataset (i.e., DSP dataset) or a single type of class labels (i.e., cognitive complexities), but instead performs well for several datasets with different types of domain-agnostic class labels. While the algorithm in Chapter 5 has been verified for K, A, T class labels, formulation of the algorithm has not been confined to within such class labels.

Chapter 5

Regularized Phrase-based Topic Model for Domain-Agnostic Question Classification

This chapter describes the use of phrases that is more effective than using words to represent questions. The proposed phrase-based topic modeling technique employs asymmetric priors that are scaled with a new C-value for nested regular expressions. The original C-value for nested phrases identifies the relevancy of phrases by computing a value for each phrase according to whether it is a nested phrase (i.e., sub-phrase within a longer phrase). In addition, to suppress high-frequency words in phrases, term weights computed using the modified distinguishing feature selector are deployed. The proposed approach also incorporates a new topic regularization mechanism to facilitate efficient mapping of questions to class labels. Performance of the above approach is validated via four datasets across different domain-agnostic class labels comprising question types, reasoning capabilities, and cognitive complexities. Results obtained highlight that the proposed technique outperforms existing methods in terms of macro-average F1 score.

Part of this chapter has been published as S. Supraja, Andy W. H. Khong, and S. Tatinati, "Regularized phrase-based topic model for automatic question classification with domain-agnostic class labels," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 29, pp. 3604–3616, 2021.

5.1 Importance of phrases for domain-agnostic AQC

Notwithstanding that techniques such as WNTM, A-LDA or W-LDA rely on word-based representations, phrases are, in general, multi-word terms with contextual information that can achieve meaningful and coherent text representation via a constituency structure [137–139]. In this regard, the embedding of nouns and verbs have shown to yield good representations of grammatical phrases for question retrieval [140]. In recent years, topical phrase mining has been proposed for document summarization and recommendation systems. Such models extract high-quality phrases from documents and dynamically assign a topic to each phrasal word [141]. The LDA-based phrase topic model (LPTM) [70] extends topical phrase mining by constructing a topic-regular expression (regex) distribution akin to LDA's topic-word distribution. Each phrase-generalizable regex is formed by concatenating the parts-of-speech (POS) tag corresponding to each phrasal word.

In the context of this thesis, and exploiting the domain-agnostic nature of POS tags, LPTM may be applied for AQC to identify groups of regexes corresponding to a class label. Utilizing regexes is vital for effective question representation as they contribute to a question's syntax, capture long range dependencies between function words, and are associated with the labeling taxonomy [142–145]. It is useful to note that, for the clustering of keywords extracted from documents, LPTM utilizes regexes formed from only noun phrases (NPs) since these NPs carry significant information about a document [146]. In addition, placing equal emphasis on all regexes in the computation of LPTM results in bias toward high-frequency regexes, leading to improper topic assignment.

This thesis presents the newly proposed phrase-based question-LDA (Qu-LDA) algorithm that pre-extracts both NPs and verb phrases (VPs) for AQC, where the latter has been shown to convey the intention of a question [147]. Inspired by the ability of nested phrases to differentiate among phrase decomposition [148], the concept of nested regex for AQC which considers subsets of regexes based on POS tag combinations has been introduced. Besides identifying such sub-regex structure, a degree of relevance to each regex that, in turn, defines the class label, has been assigned. This assignment is based on corpus statistics and linguistic heuristics [68,149] and is achieved by formulating a new

C-value that was originally developed for nested phrases, i.e., sub-phrases that appear within other longer phrases [148]. The proposed C-value is then incorporated into LPTM by replacing the prior for the topic-regex distribution with asymmetric values, such that unique nested regexes appear with higher probabilities in the derived topics. Since NPs and VPs contribute differently to the semantics of a question, a scaling parameter that suppresses NP-based regexes taking the relative importance between NP- and VP-based regexes into account is introduced. This parameter is designed to vary inversely with the frequency of occurrence of each regex, thereby suppressing any abnormalities caused by frequently-occurring short regexes.

In spite of the above relevance-based topic assignment to the regexes, determining the importance of words to form suitable phrases (and, in turn, regexes) is not a trivial task. Due to the presence of high-frequency words (e.g., articles such as *the*) in phrases and to overcome W-LDA's inability to encapsulate the importance of a word in relation to the class labels, the modified distinguishing feature selector (MDFS) [50] technique has been employed. MDFS encompasses both inter- and intra-class word distributions to compute term weights that constitute the phrases before generalizing the NPs and VPs to regexes. Using these term-weighted regexes, a term-weighted topic-regex distribution influenced by the topic-word distribution (that includes label-relevant words) is constructed.

Finally, the question-topic distribution is computed from the above proposed term-weighted topic-regex distribution scaled by the new C-value. The dependency between topics and class labels is exploited by introducing a topic regularization mechanism. This mechanism takes the label distribution of each word within a topic into account when determining that topic's label proportion. This allows the question-topic distribution to consider the impact of class labels on words, in turn, affecting the POS tags that constitute the regexes. The resultant regularized vector of topic probabilities per question obtained with Qu-LDA are then used as features for the Gaussian process classifier (GPC) [78] to achieve AQC. Performance of AQC algorithms is evaluated via four datasets comprising questions categorized according to various domain-agnostic class labels. Results obtained highlighted that Qu-LDA achieves significant performance improvement compared to existing techniques for the above datasets.

5.2 Nested phrase mining

Nested phrases have been used for the retrieval [141, 150] and removal of irrelevant phrases in collocation or contextual word extraction tasks [63]. Nested phrases are defined as sub-phrases [151] that appear within other longer phrases. The C-value is defined as

$$\mathcal{C}_{p_k} = \begin{cases} N_{p_k} \log_2 L_{p_k}, & \text{if } p_k \text{ is not a nested phrase;} \\ \left(N_{p_k} - \frac{\sum_{p_k(s)} N_p^{(s)}}{N_s} \right) \log_2 L_{p_k}, & \\ & \text{if } p_k \text{ is a nested phrase.} \end{cases} \tag{5.1}$$

The variable N_{p_k} denotes the number of times phrase p_k occurs within a corpus, s the set of N_s phrases that contain p_k as a nested phrase, and $N_p^{(s)}$ the number of times each of the phrases in that set s occurs in the corpus. Defining $\sum_{p_k(s)}$ as the summation across all longer phrases in the set s in which p_k occurs, (5.1) implies that high-frequency nested phrases are assigned a higher C-value. This is viable given that the larger number of longer phrases that a phrase appears as nested in, the higher is the certainty about that phrase being a strong building block upon which other phrases depend on, thus exhibiting independence [148].

5.3 The proposed phrase-based question-LDA (Qu-LDA)

The proposed AQC framework with Qu-LDA is shown in Figure 5.1. Unlike designing a customized set of stop-words as described in Chapter 4, no stop-words are being removed to avoid ambiguity. After pre-processing a question to obtain $\widetilde{q}$, regexes of both NPs and VPs are extracted to achieve q before being used for the computation of the regularized phrase-based topic probabilities. The obtained feature vector $\mathbf{q}$ is then fed to the Gaussian process classifier (GPC) [78] for predicting a class label $\widehat{y}$. Computation of features with Qu-LDA is illustrated via the plate diagram of Qu-LDA in Figure 5.2 with the shaded boxes and dotted arrows denoting the newly introduced elements and links,

respectively.

To determine the phrase-based topic probabilities, Qu-LDA incorporates asymmetric λ priors computed with the proposed C-value $\mathcal{C}_{r_k}$ for each regex. Computation of this C-value is achieved via a scaling parameter φ_{r_k} that takes both the relative importance of regexes associated with NPs to VPs and the frequency of occurrence defined with respect to the length of each regex L_{r_k} into account. Since the weighting of the words that constitute the phrases influence the importance of phrases (and, in turn, regexes), Qu-LDA also incorporates asymmetric α priors and the MDFS term weighting per word Ω_{w_i}. As seen in Figure 5.2, distributions $\boldsymbol{\Theta}$, $\boldsymbol{\eta}$, and $\boldsymbol{\Phi}$ are associated with the term-weighted topic, term-weighted regex, and term-weighted words, respectively. Qu-LDA finally employs a topic regularization mechanism that relies on the word-label association. This mechanism regularizes the probability computed for each topic z_j and representing it as $P(\widehat{z}_{j,y_c})$ for each class label y_c.

5.3.1 Extraction of NP- and VP-based regexes

Both NPs and VPs are extracted based on the POS tag syntactic structure [94]. A predominant predicate that is selected heuristically in questions is verbs, especially if a particular verb attains the highest level of embedding in a constructed dependency tree [152–155]. This is due to the ability of verbs paraphrasing a question accurately as opposed to nouns. Therefore, although the baseline identifies key phrases corresponding to both NPs and VPs for questions, more emphasis is placed on VPs [147].

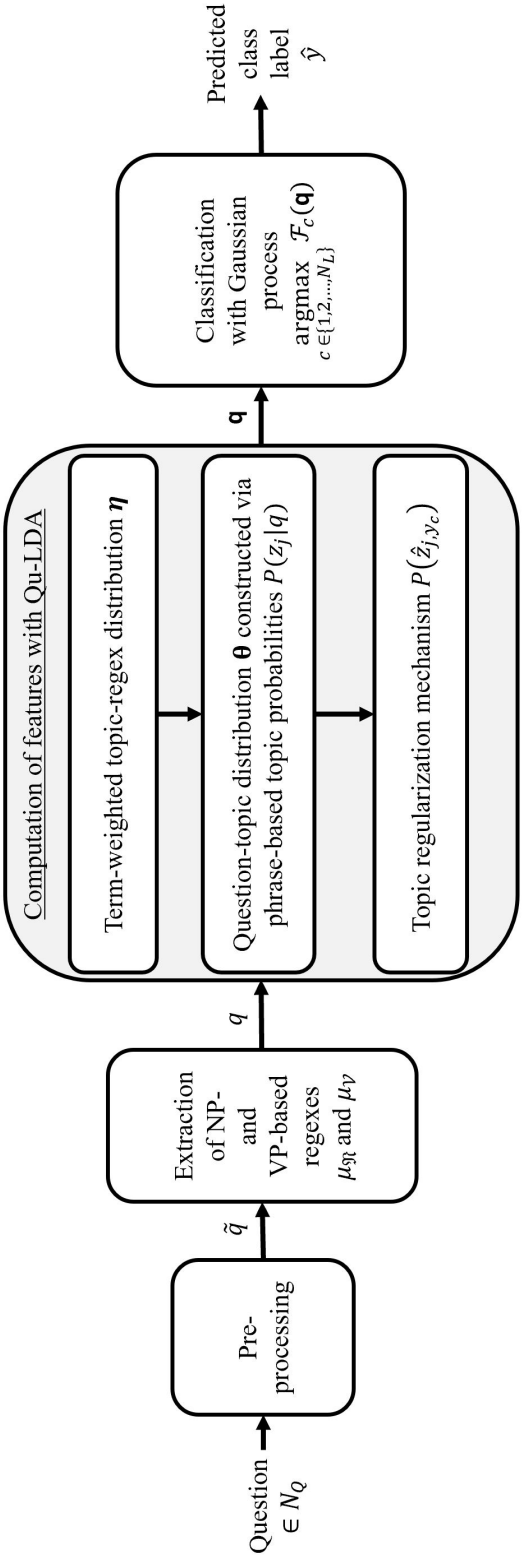


Figure 5.1: Process flow of the proposed AQC framework with Qu-LDA.

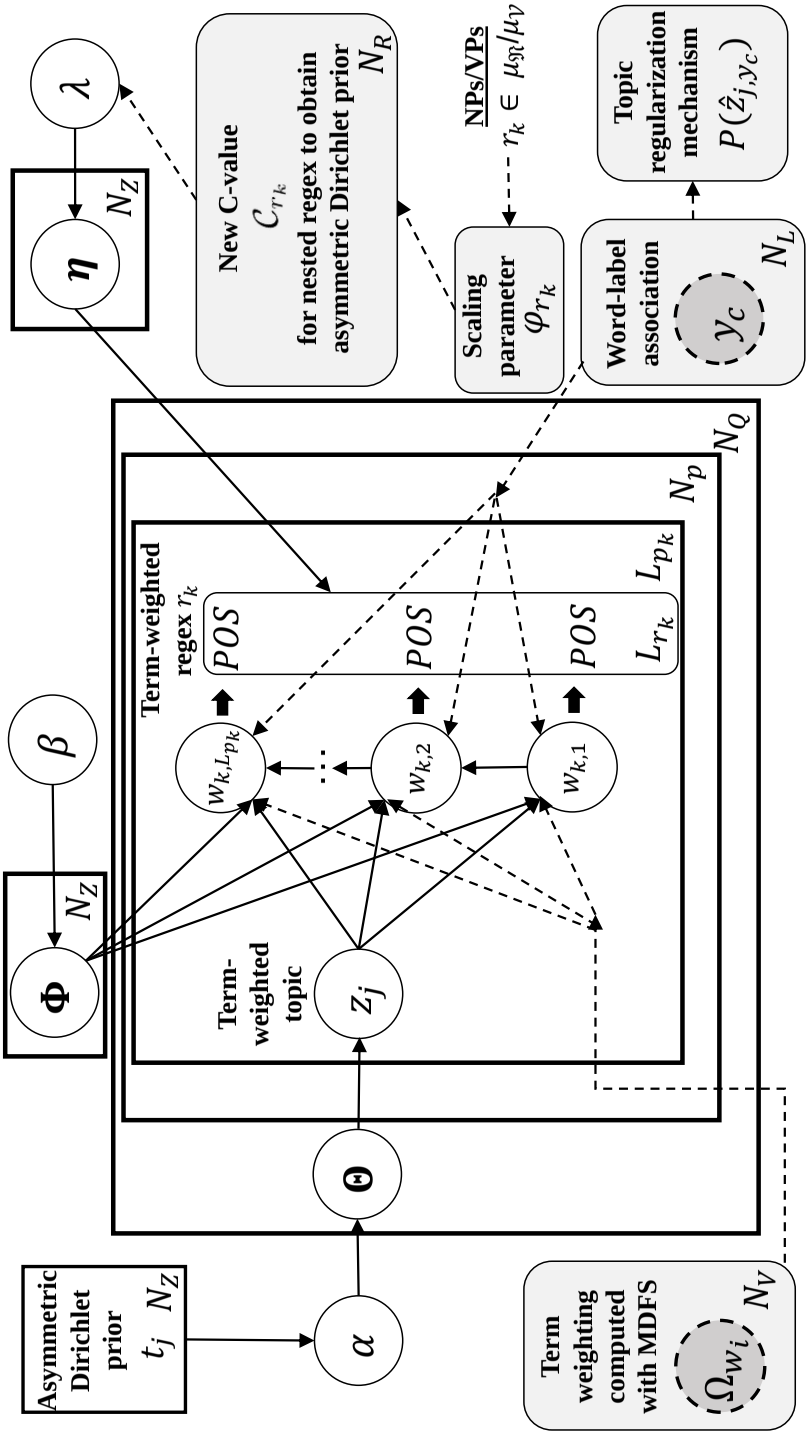


Figure 5.2: Plate diagram of the proposed Qu-LDA with the shaded boxes and dotted arrows denoting the newly introduced elements and links, respectively. Asymmetric λ priors are computed with the new C-value C_{r_k} that incorporates a scaling parameter φ_{r_k}. To address the high frequencies of words that constitute the phrases, asymmetric α priors are used and the term weight Ω_{w_i} for each word is computed with MDFS. The topic regularization mechanism is based on the word-label association.

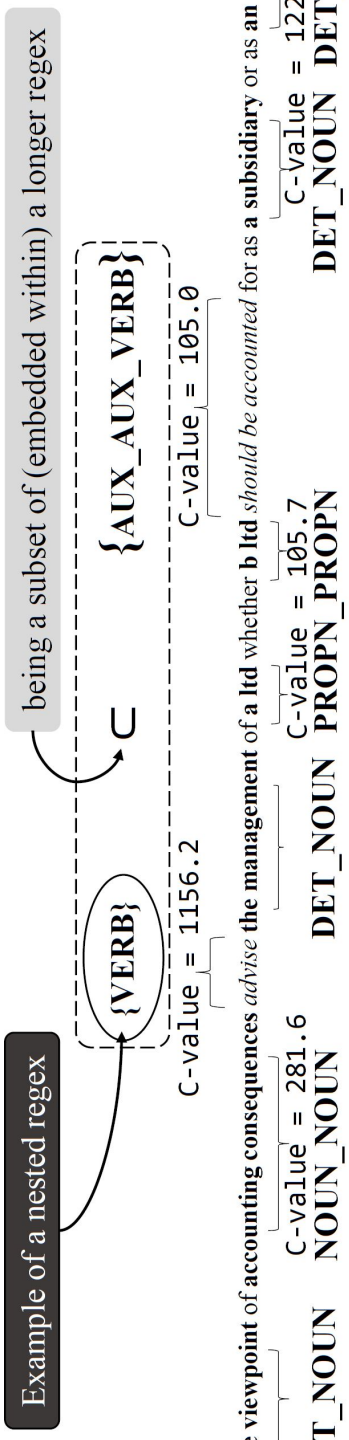


Figure 5.3: Illustration of nested regex (e.g., *VERB* within *AUX_AUX_VERB*) with new C-values. Phrases in bold refer to NPs while those in italics refer to VPs.

The combinations and order of words belonging to different POS tags are used to represent NPs and VPs. The unique amalgamation of a word (e.g., *the*) belonging to a particular POS tag (e.g., *DET*, i.e., determiner) with one or more word(s) belonging to associated POS tag(s) constitutes a phrase p_k being either an NP or a VP. The POS tags corresponding to the NPs and VPs are then clustered into generalizable regexes. The above can be implemented by the *textacy* software package, where a model was pre-trained on a randomized and stratified subset of approximately 375,000 texts extracted from various sources that include Wikipedia, crowd-sourced sentences, and journal articles. The character n-grams that have been extracted from the lower-cased text are embedded into a vector of a hundred dimensions. The resulting feature vectors are then concatenated into a single embedding layer, before being fed into a dense layer that uses the *ReLu* activation and a *softmax* output layer. Such a trained model has shown to achieve an average F1 score of 0.97 over several languages [156].

The combination of POS tags, i.e., the patterns follow

$$\begin{aligned}\mu_{\mathfrak{N}} = &<\text{DET}>?<\text{NUM}>* \\ &(<\text{ADJ}><\text{PUNCT}>?<\text{CONJ}>?)* \\ &(<\text{NOUN}>|<\text{PROPN}><\text{PART}>?)+, \qquad (5.2)\\ \mu_{\mathcal{V}} = &<\text{AUX}>*<\text{ADV}>*<\text{VERB}>, \qquad (5.3)\end{aligned}$$

for NP- and VP-based regexes, respectively. The punctuation marks ?, *, and + denote zero or one, zero or more, and one or more occurrences of the preceding POS tag, respectively. A corresponding regex r_k for an NP (denoted by $\mu_{\mathfrak{N}}$) and for a VP (denoted by $\mu_{\mathcal{V}}$) is then defined as any suitable combination of POS tags according to (5.2) or (5.3).

5.3.2 Computation of term-weighted topic-regex and question-topic distributions

Algorithm 3 provides a formal description of Qu-LDA. Due to the skewed nature of high-frequency regexes brought about by the general structure of questions (i.e., presence of several NPs and standalone verbs), different regexes are emphasized through the

Algorithm 3: Proposed Qu-LDA algorithm.

Input: Pre-processed and regex-extracted question q
Output: Feature vector with regularized topic probabilities $\mathbf{q}$ for each question
for *label* $y_c \in N_L$ **do**
  $P(y_c|w_i) \leftarrow N_{w_i,y_c}, N_{w_i}$ using (5.13)
  $P(w_i|z_j) \leftarrow P(w_i|r_k, z_j)$
  $P(y_c|z_j) \leftarrow P(y_c|w_i), P(w_i|z_j)$ using (5.12)
  $P(\widehat{z}_{j,y_c}) \leftarrow P(y_c|z_j)P(z_j|q)$ using (5.11)
  $\mathbf{q} \leftarrow P(\widehat{z}_{j,y_c})$ using (5.10)
  for *topic* $z_j \in N_Z$ **do**
    $P(z_j|q) \leftarrow \boldsymbol{\Theta}(\Omega_{w_i}, \alpha, t_j)$ using (5.6)
    for *regex* $r_k \in N_R$ **do**
      $\varphi_{r_k} \leftarrow N_{\mu\mathfrak{N}}, N_{\mu\mathcal{V}}, L_{r_k}$ using (5.5)
      $\mathcal{C}_{r_k} \leftarrow \varphi_{r_k}, L_{r_k}, N_{r_k}$ using (5.4)
      $P(r_k|z_j) \leftarrow \boldsymbol{\eta}(\Omega_{w_i}, \lambda, \mathcal{C}_{r_k})$ using (5.7)
      for *word* $w_i \in N_V$ **do**
        $\Omega_{w_i} \leftarrow \text{MDFS}(w_i)$ using (2.2)
        $P(w_i|r_k, z_j) \leftarrow \boldsymbol{\Phi}(\Omega_{w_i}, \beta)$ using (5.8)
      end
    end
  end
  $P(z_j|w_i, r_k, z_{-j}) \leftarrow P(w_i|r_k, z_j)P(r_k|z_j)P(z_j|q)$
  Assign z_j to each r_k and w_i based on (5.9)
end

nested regex concept, where regexes are divided into their sub-structure. This allows the algorithm to discern among regex decomposition, which, as a consequence, better defines the distribution of regexes within each topic. This, in turn, facilitates the representation of topic distribution for a question to achieve better association of that question with its corresponding class label.

The anatomy of nested regex in Qu-LDA is highlighted in Figure 5.3 using an exemplar question belonging to one of the datasets to be described in the section on domain-agnostic question datasets. Phrases in bold refer to NPs while phrases in italics refer to VPs. An instance of a nested regex is shown by $VERB \subset AUX\_AUX\_VERB$, where $\subset$ indicates that the former is a subset of (embedded within) the latter. To compute the relevance of regex decomposition, a new C-value has been proposed, defined for each r_k,

as

$$\mathcal{C}_{r_k} = \begin{cases} \varphi_{r_k} L_{r_k} N_{r_k}, & \text{if } r_k \text{ is not a nested regex;} \\ \varphi_{r_k} L_{r_k} \left(N_{r_k} - \frac{\sum_{r_k{}^{(S)}} N_r^{(S)}}{N_S} \right), & \\ & \text{if } r_k \text{ is a nested regex,} \end{cases} \tag{5.4}$$

where the scaling parameter is defined by

$$\varphi_{r_k} = \begin{cases} \left(\frac{1}{10 \times \frac{N_{\mu_{\mathfrak{N}}}}{N_{\mu_{\mathcal{V}}}}} \right) \left(1 - \exp(-10^{(L_{r_k}-3)}) \right), & \text{if } r_k \in \mu_{\mathfrak{N}}; \\ 1 - \exp(-10^{(L_{r_k}-3)}), & \text{if } r_k \in \mu_{\mathcal{V}}. \end{cases} \tag{5.5}$$

In (5.4), N_{r_k} denotes the number of times regex r_k occurs in the entire corpus, S the set of N_S regexes that contain r_k as a nested regex, and $N_r^{(S)}$ the number of times each of the regexes in that set S occurs in the corpus. The proposed $\mathcal{C}_{r_k}$ values are, therefore, based on the occurrences of all phrases that belong to each regex. Defining $\sum_{r_k{}^{(S)}}$ as the summation over all longer regexes in the set S in which r_k occurs, the above formulation implies that high-frequency nested regexes are assigned a higher $\mathcal{C}_{r_k}$ similar to nested phrases. As opposed to (5.1), where $\mathcal{C}_{p_k}$ has been defined for phrases that only considers bi-grams and beyond [148], the logarithmic function has been removed in (5.4) since regex may also be derived from a single-word phrase if it is a constituent in a sentence's syntax [63,69]. Removing this logarithmic function, therefore, prevents these single-word regexes (e.g., *NOUN* or *VERB*) that are vital for the construction of NPs and VPs from receiving a null value.

To appreciate the effect of φ_{r_k} in (5.4), it can be noted from Figure 5.3 that, without φ_{r_k}, a high C-value is exhibited for NP-based regexes (e.g., *DET_NOUN*) and regexes with $L_{r_k} = 1$ (e.g., *VERB*), both which have the highest frequencies. Defining $N_{\mu_{\mathfrak{N}}}$ and $N_{\mu_{\mathcal{V}}}$ as the number of times NP- and VP-based regexes occur, respectively, the proposed formulation in (5.5) results in higher emphasis of a regex if $r_k \in \mu_{\mathcal{V}}$. Figure 5.4 illustrates the variation of φ_{r_k} with regex length L_{r_k} for an illustrative case of $N_{\mu_{\mathfrak{N}}}/N_{\mu_{\mathcal{V}}} = 2.05$. It can be seen that regexes corresponding to NPs are significantly suppressed compared to VPs. In addition, φ_{r_k} increases with L_{r_k} such that shorter regexes are de-emphasized. For $L_{r_k} \geq 4$, no de-emphasis is required since long regexes are generally absent from a corpus

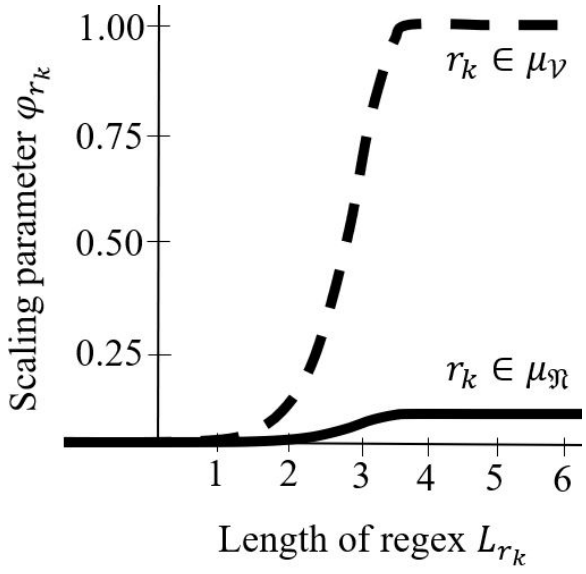


Figure 5.4: Variation of scaling parameter φ_{r_k} with L_{r_k} for the suppression of NP-based and shorter regexes. The dotted curve refers to a VP-based regex while the solid curve refers to an NP-based regex.

of questions; the number of words within a phrase (phrase length) generally follows a long-tailed distribution, indicating that most phrases have relatively short lengths [141]. Further experiments show that the ratio $N_{\mu_{\mathfrak{N}}}/N_{\mu_{\mathcal{V}}}$ obtained for the datasets described in the subsequent section ranges from 1.26 to 2.05 with a modest difference of 0.04 $< \triangle\varphi_{r_k} <$ 0.08 across the datasets. Hence, the value of φ_{r_k} does not vary significantly across datasets.

Besides scaling the importance of regexes appropriately, term weights and asymmetric α priors pertaining to words that constitute the phrases are incorporated. This allows the algorithm to take the prevalence of words across and within class labels into account. With reference to Figure 5.2, Qu-LDA incorporates inter- and intra-class word distributions by employing the MDFS term weighting per word Ω_{w_i} which, in turn, influence the word probabilities.

With words weighted by Ω_{w_i} and relevance of the regexes defined by $\mathcal{C}_{r_k}$, the proposed Qu-LDA employs a new term-weighted topic-regex distribution that considers all the phrase weights constituting each regex. These phrase weights are, in turn, the sum of the MDFS term weighting of the words that each phrase consists of. Accordingly, the

three posterior probability distributions of Qu-LDA are computed, respectively, as

$$\begin{aligned} P(z_j|q) &= \boldsymbol{\Theta}(\Omega_{w_i}, \alpha, t_j) \\ &= \frac{\sum_{k=1}^{N_p} \sum_{i=1}^{L_{r_k}} \Omega^{(q)}_{w_i,r_k,z_j} + t_j\alpha}{\sum_{k=1}^{N_p} \sum_{i=1}^{L_{r_k}} \Omega^{(q)}_{w_i,r_k} + \sum_{j=1}^{N_Z} t_j\alpha}, \end{aligned} \tag{5.6}$$

$$\begin{aligned} P(r_k|z_j) &= \boldsymbol{\eta}(\Omega_{w_i}, \lambda, \mathcal{C}_{r_k}) \\ &= \frac{\sum_{r_k} \sum_{i=1}^{L_{r_k}} \Omega_{w_i,z_j,r_k} + \mathcal{C}_{r_k}\lambda}{\sum_{k=1}^{N_R} \sum_{i=1}^{L_{r_k}} \Omega_{w_i,z_j,r_k} + \sum_{k=1}^{N_R} \mathcal{C}_{r_k}\lambda}, \end{aligned} \tag{5.7}$$

$$\begin{aligned} P(w_i|r_k, z_j) &= \boldsymbol{\Phi}(\Omega_{w_i}, \beta) \\ &= \frac{\Omega_{w_i,z_j} + \beta}{\sum_{i=1}^{N_V} \Omega_{w_i,z_j} + \beta N_V}, \end{aligned} \tag{5.8}$$

while the probability for the optimal topic allocation to each phrase after removing the particular phrase of interest in each Gibbs sampling iteration conforms to the relationship

$$P(z_j|w_i, r_k, z_{-j}) \propto P(w_i|r_k, z_j)P(r_k|z_j)P(z_j|q). \tag{5.9}$$

The variable t_j in (5.6) denotes the unique value for each topic that is derived from the Newton-Raphson optimization. The term $\sum_{k=1}^{N_p} \sum_{i=1}^{L_{r_k}} \Omega^{(q)}_{w_i,r_k,z_j}$ denotes the sum of term weights for all regexes in a question for a topic z_j, $\sum_{k=1}^{N_p} \sum_{i=1}^{L_{r_k}} \Omega^{(q)}_{w_i,r_k}$ the sum of term weights for all regexes in a question across all topics, and $\sum_{j=1}^{N_Z} t_j\alpha$ the sum of asymmetric prior values for the question-topic distribution. In (5.7), the term $\sum_{r_k} \sum_{i=1}^{L_{r_k}} \Omega_{w_i,z_j,r_k}$ denotes the sum of term weights for all phrases under the same regex for that topic, $\sum_{k=1}^{N_R} \sum_{i=1}^{L_{r_k}} \Omega_{w_i,z_j,r_k}$ the sum of all term weights for all regexes for that topic, and $\sum_{k=1}^{N_R} \mathcal{C}_{r_k}\lambda$ the sum of asymmetric prior values for the topic-regex distribution. The term $\sum_{i=1}^{N_V} \Omega_{w_i,z_j}$ in (5.8) is defined as the sum of term weights for all words belonging to all regexes in that topic.

With reference to notations used in LPTM [70], z_{-j} in (5.9) denotes the exclusion of the corresponding existing topic assignment of that phrase, implying the negation of the words belonging to it and the regex associated with it. The topic z_j is, therefore, assigned to each r_k and w_i based on (5.9) in each iteration. In addition, since $P(w_i, r_k, z_j|q) = P(w_i|r_k, z_j)P(r_k|z_j)P(z_j|q)$ and that $P(z_j|w_i, r_k, z_{-j}) \propto P(w_i, r_k, z_j|q)$,

the topic assignment is dependent on the term weights and the scaling of regexes. This, in turn, affects the question-topic distribution that the classification depends upon. The vector of topic probabilities corresponding to the question-topic distribution $\mathbf{\Theta}$ can, therefore, be considered as the set of intermediate features that represent each question as shown in Figure 5.1.

5.3.3 Topic regularization mechanism

Despite establishing the mapping between the question-topic distribution and class labels via a rigid topic-label dependency based on available prior information, such an approach is suitable for domain-specific class labels directly associated with their topical information. For domain-agnostic class labels, a topic regularization mechanism that incorporates the impact of each class label on the topic probabilities has been proposed. This dependency is modeled via the word-label association since a word may occur in different class labels with different proportions. This is akin to the construction of domain-independent lexicons via statistical co-occurrence information between candidates and sentiment labels [157]. The proposed mechanism regularizes the question-topic distribution based on the relationship between words that each topic consists of and the respective class labels. As seen in Figure 5.2, this is achieved via the class label information $y_c \in N_L$ in relation to each word that regularizes each topic's probability into $P(\widehat{z}_{j,y_c})$.

The regularized topic probabilities are computed for each class label and concatenated into

$$\begin{aligned}\mathbf{q} = \{&P(\widehat{z}_{1,y_1}), \ldots, P(\widehat{z}_{N_Z,y_1}),\\ &P(\widehat{z}_{1,y_2}), \ldots, P(\widehat{z}_{N_Z,y_2}), \ldots\\ &P(\widehat{z}_{1,y_{N_L}}), \ldots, P(\widehat{z}_{N_Z,y_{N_L}})\},\end{aligned} \tag{5.10}$$

which denotes a regularized vector of topic probabilities with length $N_L N_Z$ based on each class label. The regularized probability of the jth topic belonging to each class label y_c

is defined as

$$P(\widehat{z}_{j,y_c}) = P(y_c|z_j)P(z_j|q), \tag{5.11}$$

where $P(z_j|q)$ is computed using (5.6) and

$$P(y_c|z_j) = \sum_{i=1}^{N_V} P(y_c|w_i)P(w_i|z_j) \tag{5.12}$$

denotes the probability of each class label for that topic. It is useful to note that $P(w_i|z_j) = P(w_i|r_k, z_j)$ since word probabilities in a topic can be considered without that of regexes based on which each word belongs to; the presence of each word and each regex are conditionally independent given that the topic is assigned to each phrase (in turn, the corresponding regex and each word within the phrase). With N_{w_i} denoting the total number of questions in which each word occurs in and N_{w_i,y_c} the number of times word w_i occurs in a class label y_c for a given training dataset, the word-label association which denotes the probability of each class label y_c for that word is given by

$$P(y_c|w_i) = \frac{N_{w_i,y_c}}{N_{w_i}}. \tag{5.13}$$

The formulation in (5.10) implies that the class label that obtains the maximum number of high regularized topic probabilities based on the corresponding word-label proportions would most likely be assigned to that question.

In conventional binary classification, the set of Qu-LDA feature vectors $\mathbf{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_{N_Q}\}$ will be fed into a Gaussian process (GP) regressor as the input [78]. Here, $\mathbf{q}_l$ (derived from (5.10)) denotes the lth question's feature vector in the training corpus.

5.4 Domain-agnostic question datasets and labeling taxonomies

Datasets corresponding to various domain-agnostic class labels have been used to evaluate the performance of the proposed Qu-LDA model against baseline AQC feature extraction techniques. Details of these datasets, along with examples of NPs and VPs

for each domain-agnostic category are tabulated in Table 5.1.

The gradual evolvement of cognitive complexity required for solving questions involves recalling previously taught materials (*Knowledge* (K)), examining new materials (*Application* (A)), or drawing connections between old and new learning materials by means of mental processes (*Transfer* (T)) [120, 158, 159]. Two datasets of questions across various disciplines from six Nanyang Technological University (NTU) courses and Najran University (NU) have been labeled by course instructors according to this taxonomy [44].

Questions have also been labeled according to reasoning abilities. The ARC dataset of the AI2 Reasoning Challenge has been annotated by subject-matter experts according to several knowledge and reasoning types [8, 9]. Due to overlapping categories, questions belonging to three mutually exclusive class labels (*Basic facts*, *Linguistic matching*, *Hypothetical*) have been selected.

An alternative approach to labeling questions is via question types. The middle school science classroom educational questions dataset published in LREC [7] has a total of sixteeen categories. Since multi-label classification is outside the scope of this thesis, questions with single labels (*Very short answer*, *Context sensitive*, *Answers will vary*) have been extracted.

Table 5.1: Details on the various datasets used for AQC performance evaluation

	NTU dataset	**NU dataset [44]**
Source	Nanyang Technological University (NTU)	Najran University (NU)
Number of questions	1023	596
Domain-agnostic category	Cognitive complexities	
Class labels	Knowledge (K) (46.9%) and (33.5%) Application (A) (37.7%) and (16.8%) Transfer (T) (15.4%) and (49.7%)	
Examples of NPs	the main purpose plausible explanations a true representation	
Examples of VPs	briefly describe can be characterized would be required	
	ARC dataset [8, 9]	**LREC dataset [7]**
Source	AI2 Reasoning Challenge	Middle schools
Number of questions	279	345
Domain-agnostic category	Reasoning capabilities	Question types
Class labels	Basic facts (28.0%) Linguistic matching (47.0%) Hypothetical (25.0%)	Very short answer (41.6%) Context sensitive (36.0%) Answers will vary (22.4%)
Examples of NPs	the same amount a cold air mass a plastic bottle	an acceptable answer experimental group the two data tables
Examples of VPs	carefully measure has been most affected most likely caused	being transferred might be directly related give examples

Table 5.2: Macro-average F1 scores for each dataset. LDA+ denotes appropriate combinations of existing LDA variants

Method type	**Feature extraction technique**	**NTU**	**NU**	**ARC**	**LREC**
BoW	TF-IDF [44]	0.509	0.585	0.554	0.342
	TF-ICF [47]	0.462	0.409	0.674	0.344
LDA variants	LDA [20]	0.487	0.336	0.499	0.386
	A-LDA [34] and W-LDA [35]	0.329	0.213	0.491	0.165
	A-LDA and W-LDA (with MDFS [50])	0.567	0.410	0.542	0.478
	Modified LPTM [70] (with word-based elements)	0.597	0.655	0.719	0.575
Proposed	**Qu-LDA (phrase-based)**	**0.628**	**0.759**	**0.803**	**0.710**

Table 5.3: Impact of different distributions on degree of word probabilities for exemplar question shown in Figure 5.3

Word	**K**	**A**	**T**	**Corpus-wide**	**Inter-class**	**Intra-class**
advise	0	1	2	High	Low	Low
associate	0	3	0	High	High	Low
ltd	1	20	3	Low	High	High

Table 5.4: Comparing position of regexes in topics with symmetric λ priors as opposed to asymmetric λ priors

Method	**Many topics**	**Many topics**
Symmetric λ	NOUN DET_NOUN	VERB ADV_VERB
Method	**A topic**	**Another topic**
Asymmetric λ	DET_NOUN DET_ADJ_NOUN	VERB AUX_AUX_VERB

5.5 Hyperparameter selection

Each dataset was divided into 70/30 training/testing split. The number of topics was evaluated from 5 to 60 in intervals of 5 with the optimal number being one that achieves the highest macro-average F1 score. The optimal number of topics for the NTU dataset was found to be 50 while it was 20 for the remaining datasets; the higher number of topics found in the NTU dataset was due to the large number of questions. In this work, symmetric $\alpha = 0.1$ and $\beta = 0.01$ values have been employed for LDA while a symmetric value $\lambda = 0.1$ is used in LPTM. 1000 Gibbs sampling iterations have been used for both training and testing.

For the Gaussian process classifier (GPC), the RBF kernel was used and bandwidth/ smoothness values were determined by a grid search from 0.25 to 1.0 with an interval of 0.25. The identified optimal value was 1.0.

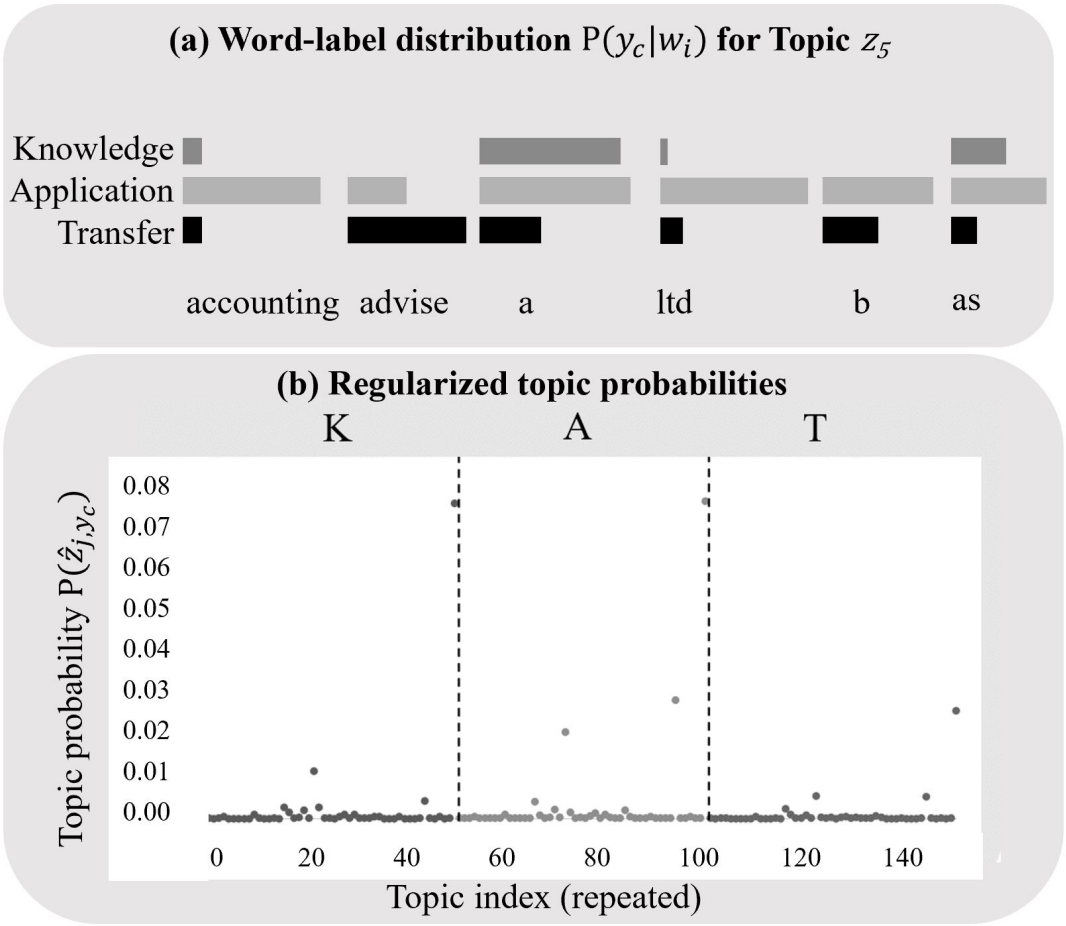


Figure 5.5: Impact of the topic regularization mechanism as reflected in (b) which is based on the word-label association illustrated in (a).

5.6 Comparison analysis

Similar to Chapter 4, given a model, an individual F1 score for each label and the macro-average F1 score are computed to evaluate its efficacy for AQC. A high macro-average F1 score indicates a large number of questions that are predicted to belong to the correct label (true positives) and a large number of questions that are predicted not to belong to the incorrect label (true negatives). The macro-average F1 is preferred over micro-average F1 for AQC to avoid bias against the class label that consists of the largest number of questions.

Table 5.2 illustrates performance achieved by BoW methods with feature vectors

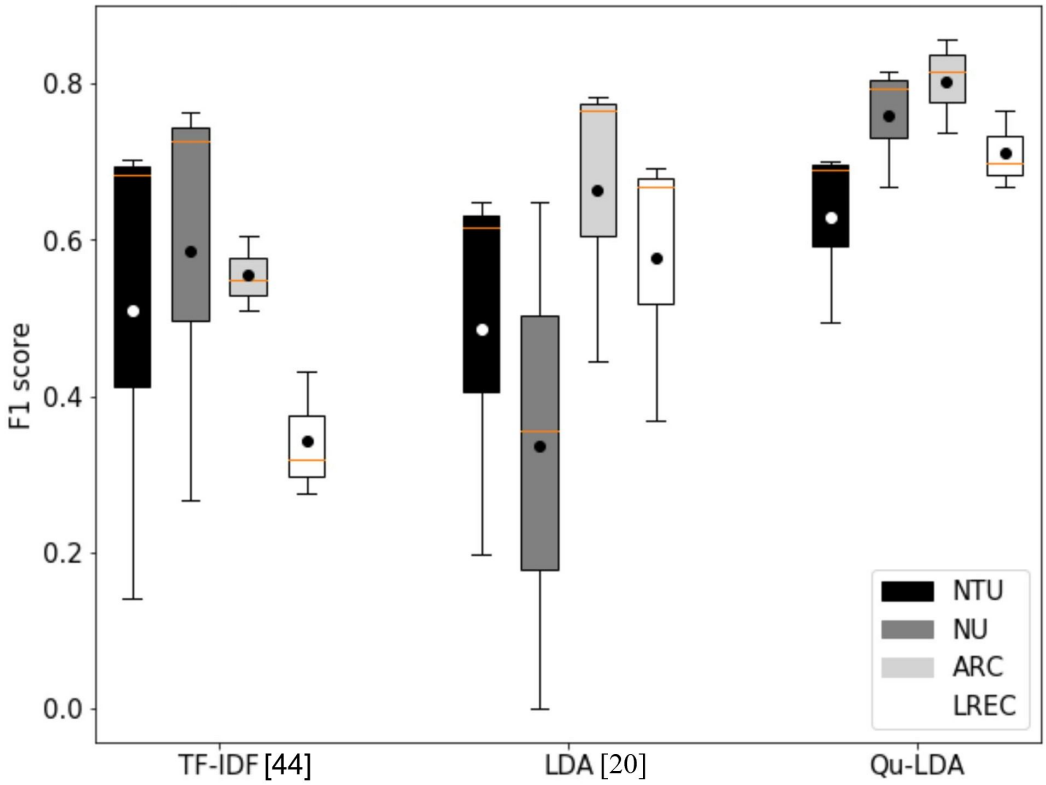
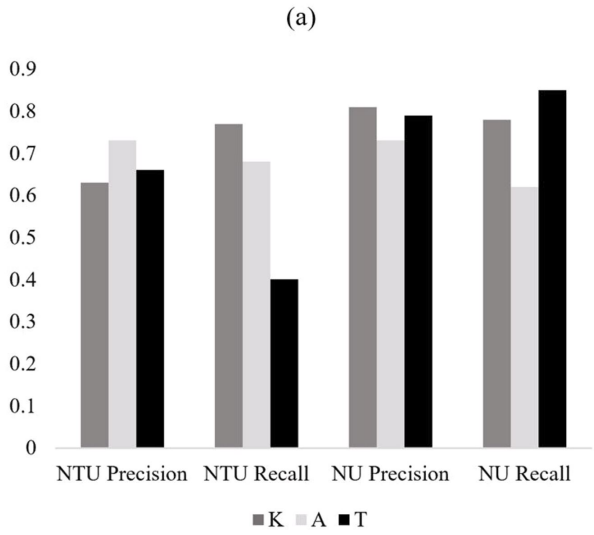


Figure 5.6: Box-plots to incorporate standard deviation information for individual F1 scores pertaining to each class label for all datasets. The mean values among the class labels are denoted by the dots in each box-plot.

extracted using term frequency-inverse document frequency (TF-IDF) [44] and term frequency-inverse class frequency (TF-ICF) [47]. As can be seen, these techniques achieved a low macro-average F1 score for each dataset ranging from 0.342 to 0.585 for TF-IDF and a modest increase for TF-ICF. These results underpin that prediction capabilities of these methods are limited by the use of word frequencies. While LDA addresses the sparsity limitation of the above approaches, it suffers from poor AQC performance due to topic homogeneity. Although the use of asymmetric α priors and term weighting in A-LDA and W-LDA address the homogeneity problem of LDA, the macro-average F1 scores of between 0.165 and 0.491 are lower than that of LDA due to the use of corpus-wide term weighting.

It is useful to note that employing MDFS term weighting in W-LDA increases the



(b)

NTU dataset

Actual \ Predicted	K	A	T
K	101	23	8
A	36	82	3
T	24	8	21

NU dataset

Actual \ Predicted	K	A	T
K	54	3	12
A	3	16	7
T	10	3	71

Figure 5.7: Performance of Qu-LDA for the NTU and NU datasets (a) via precision and recall scores and (b) via the confusion matrices.

macro-average F1 score significantly since MDFS encapsulates both inter- and intra-class word distributions that is suitable for AQC. To highlight the importance of considering inter- and intra-class word distributions in the computation of word probabilities, words corresponding to the pre-processed exemplar question highlighted in Figure 5.3 are tabulated in Table 5.3. The number of times these words appear in each class label (K, A, or T) along with their respective degree of weights (*High* or *Low*) for corpus-wide, inter-class, and intra-class distributions has also been shown. A *High* corpus-wide term weight implies that the word occurs rarely in the entire corpus. For instance, for the word *ltd* which occurs mainly in class label A (Application-type question), a *High* inter-class weight is assigned, implying concentration toward a particular class label. The same word occurs twenty times in class label A, indicating a *High* intra-class weight for high frequency in that class label. However, *ltd* exhibits a *Low* corpus-wide weight. This shows that relying on word frequencies alone is not sufficient to determine the importance of a word given the class label information; the corpus-wide method is, therefore, incapable of providing accurate term weights for AQC. Incorporating MDFS term weighting according to (2.2) addresses this bias in W-LDA and is numerically verified by an increase in macro-average F1 scores to 0.567, 0.410, 0.542, and 0.478 for the respective datasets

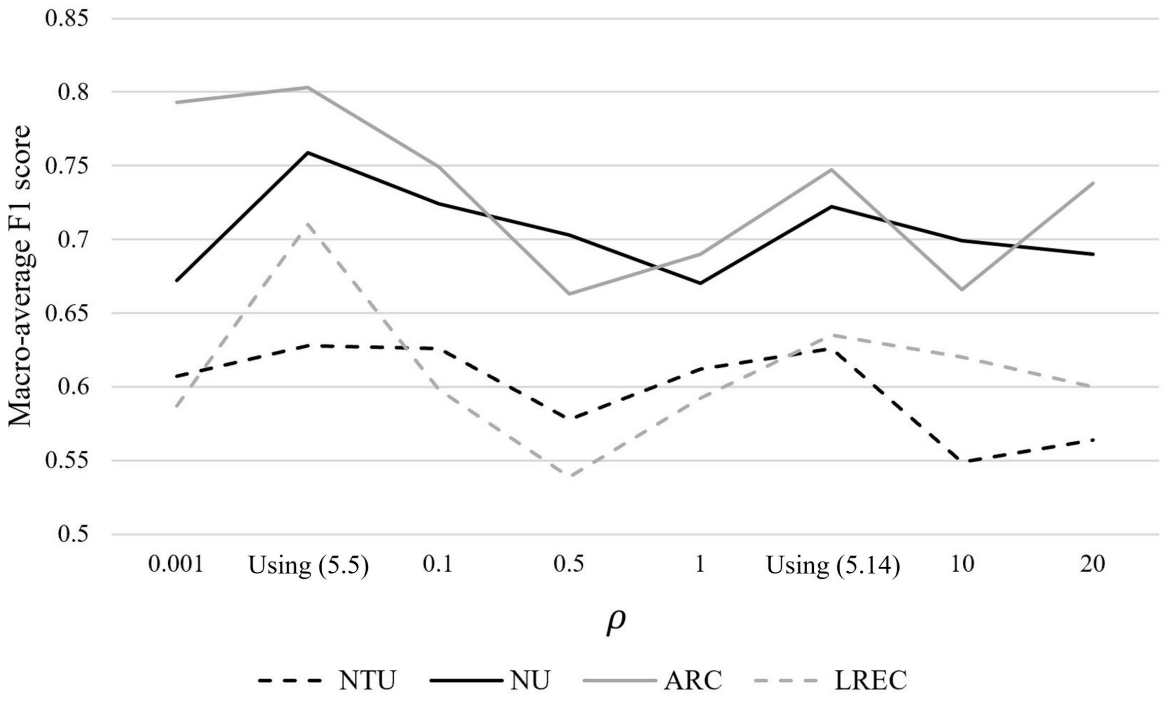


Figure 5.8: Sensitivity of the macro-average F1 score with respect to the weight ratio ρ of NP- and VP-based regexes.

as seen in Table 5.2.

Although the modified LPTM (that extracts both NP- and VP-based regexes and incorporates word-based elements) adopts a phrase-based approach, the classification performance is significantly lower than Qu-LDA due to the lack of any emphasis given to different types of regexes. This can be seen from the macro-average F1 scores of 0.597, 0.655, 0.719, and 0.575 achieved by the modified LPTM for the respective datasets. The proposed Qu-LDA algorithm achieves improved performance by incorporating the nested regex concept with the new scaled C-value. In addition, with asymmetric λ priors, the topic-regex distribution results in an almost unique set of regexes per topic which, in turn, provides higher association to each class label. This effect can be seen in Table 5.4, where many topics consist of high-frequency regexes for the case of symmetric λ priors.

The impact of the topic regularization mechanism in Qu-LDA described by (5.11) is illustrated in Figure 5.5 for the same question listed in Figure 5.3.The number of topics was determined via the hyperparameter selection process as described in Section 5.5.

Once the topics were generated, the same order of topics was maintained while applying the topic regularization mechanism. Taking the word-label distribution within a topic into account is important for AQC to determine the likelihood of that topic belonging a particular class label. For this exemplar question, the high-probability words for the most representative topic z_5 is depicted in Figure 5.5(a). With reference to (5.13), the bar-plots highlight the probability of each word belonging to each class label $P(y_c|w_i)$, with the length of each bar representing the degree of belongingness. It can be inferred that almost all words (except *advise*) belong to the class label A. This implies that z_5 is associated toward class label A, described via (5.12). Since z_5 achieves the highest probability for this question, the topic regularization mechanism suggests that the class label is likely to be A for this question. This is illustrated by the scatter plot in Figure 5.5(b), where the concatenated vector of regularized topic probabilities is plotted for this question according to (5.10). The dotted lines indicate the separation of the fifty regularized topic probabilities corresponding to each class label. It can be seen that the number of significant points are three in K, four in A, and three in T, resulting in the question being classified as class label A. With the above characteristics, Qu-LDA achieves the highest macro-average F1 scores of 0.628, 0.759, 0.803, and 0.710 for the respective datasets as seen in Table 5.2.

The F1 scores across three algorithms implemented specifically for AQC have also been compared as seen in Figure 5.6. The mean values among the class labels (macro-average F1 scores) are denoted by the dots in each box-plot. It can be noted that Qu-LDA achieves an average of approximately 62% improvement in macro-average F1 scores over TF-IDF [44] and LDA [20] across all datasets. It can be inferred from the length of the box-plots that Qu-LDA achieves a low standard deviation of individual F1 scores pertaining to each class label. The short box-plots of Qu-LDA highlight an approximately 51% average reduction in standard deviation over the longer box-plots of TF-IDF and LDA across all datasets. This implies almost equal F1 scores for each class label in comparison to skewed classifications for TF-IDF and LDA. In addition, it can be inferred that TF-IDF and LDA are sensitive to the datasets while Qu-LDA is less sensitive. This is verified through the approximately 17% average reduction in standard deviation among the macro-average F1 scores for the four datasets for Qu-LDA

in comparison to TF-IDF and LDA.

It can be seen from Figure 5.6 that the proposed Qu-LDA algorithm achieves a lower performance for the NTU dataset compared to the other three datasets. This is due to the structure of questions in the NTU dataset that is different from the others. Figure 5.7 shows the precision and recall scores achieved by Qu-LDA along with the confusion matrices for the NTU and NU datasets. The other two datasets are not considered in this discussion since the class labels are different—the context of the questions differs from the cognitive complexities class labels of both NTU and NU datasets. It can be noted from Figure 5.7(a) that while the precision scores are consistent across the class labels, the NTU dataset suffers from a low recall score of 0.4 for the T class label. On the other hand, the NU dataset achieves consistently high precision and recall scores across all class labels, resulting in a higher macro-average F1 score than the NTU dataset. Upon further investigation via the confusion matrices shown in Figure 5.7(b), it can be seen that 24 questions belonging to the T class label were incorrectly classified as K for the NTU dataset. Upon further examining questions within the NU dataset, it was noted that they comprise a distinguishable question structure for each class label that is represented by the choice of words, i.e., class-specific Bloom's Taxonomy verbs. On the contrary, for the NTU dataset, similar question structure results in the model facing difficulty in discriminating between appropriate class-specific co-occurrence patterns. This results in a lower macro-average F1 score for the NTU dataset.

To examine the importance of different weight ratios (applied to NP- and VP-based regexes) on the macro-average F1 score, the ratio $\rho = (\varphi_{r_k} \text{ if } r_k \in \mu_{\mathfrak{N}})/(\varphi_{r_k} \text{ if } r_k \in \mu_{\mathcal{V}})$ is first defined such that a high value of ρ implies more emphasis given to NP-based regexes. Figure 5.8 shows the variation of the macro-average F1 score with ρ for the four datasets. It can be seen that Qu-LDA suffers from poor performance if NP-based regexes are not sufficiently emphasized (as seen when $\rho = 0.001$) as opposed to the formulation of φ_{r_k} in (5.5). In addition, if the emphasis is instead placed on NP-based regexes given by

$$\varphi_{r_k} = \begin{cases} 1 - \exp(-10^{(L_{r_k}-3)}), & \text{if } r_k \in \mu_{\mathfrak{N}}; \\ \left(\frac{1}{10 \times \frac{N_{\mu_{\mathfrak{N}}}}{N_{\mu_{\mathcal{V}}}}}\right)\left(1 - \exp(-10^{(L_{r_k}-3)})\right), & \text{if } r_k \in \mu_{\mathcal{V}}, \end{cases} \tag{5.14}$$

Table 5.5: Comparison with deep learning methods

Method	NTU	NU	ARC	LREC
LSTM	**0.680**	0.650	0.740	0.560
CNN	**0.680**	0.700	0.730	0.620
BERT	0.304	0.337	0.303	0.194
Qu-LDA	0.628	**0.759**	**0.803**	**0.710**

the macro-average F1 score reduces in particular when there is an over-emphasis on NP-based regexes with larger ρ values beyond the formulation in (5.14). This experiment, therefore, validates the computation of φ_{r_k} in (5.5) and the importance of VP-based regexes for achieving high AQC performance with Qu-LDA.

To compare the effectiveness of Qu-LDA over deep learning techniques, experiments with a long short-term memory network (LSTM) and convolutional neural network (CNN) that employ embeddings trained on each dataset, as well as the bidirectional encoder representations from transformers (BERT) model were performed. Table 5.5 shows results for the above methods with the AQC performance verified via the same four datasets. It can be seen that LSTM and CNN achieve higher performance than BERT. This is due to the embeddings being trained on the words in each dataset—rare technical words may not be found in the conventional dictionary of pre-trained models such as BERT. In addition, it can also be noted that the proposed Qu-LDA achieves the highest performance for the smaller NU, ARC, and LREC datasets, while Qu-LDA suffers modest performance degradation for the larger NTU dataset. This is because, with a larger dataset, the deep learning algorithms are able to extract semantic information by prioritizing locality and sequentiality [160] that topic modeling algorithms fail to do.

5.7 Chapter summary

In this chapter, the proposed phrase-based topic modeling approach to represent a question for AQC is being described. In addition to pre-extracting both NPs and VPs, the concept of nested regex is introduced and a new C-value is proposed to assess the relevance of each regex. This C-value scales the regexes according to their type (via

the suppression of NP-based regexes) and frequency of occurrence in relation to regex length (via the suppression of short regexes). Term weights that incorporate both inter- and intra-class distributions are then employed to suppress the weights of high-frequency words. The resultant term-weighted topic-regex distribution, therefore, offers a set of feature vectors that represent a question. The dependency between topics and class labels is then taken into account by incorporating a topic regularization mechanism based on the word-label association for the words under each topic. Experiment results show that the proposed approach outperforms existing AQC techniques across four datasets.

Chapter 6

Quad-faceted Feature-based Graph Network for Domain-Agnostic Document Classification

This chapter presents a new quad-faceted feature-based graph network that incorporates four types of graphs that operate on different set of nodes with unique computations of edge weights. The benefits of term weighting from Chapter 3, topic modeling from Chapter 4, and phrases, as well as, regexes, from Chapter 5 particularly for domain-agnostic classification will be used as the foundation for the development of a diverse heterogeneous graph for document classification. Various types of documents including scientific statements, journal articles, and job descriptions that correspond to the domain-agnostic class labels of nature of statements, argumentative zones, and general skillsets, respectively are used for performance evaluation of the proposed model. Experiment results show that when compared with other conventional deep learning techniques such as LSTM and CNN or with graph models such as TextGCN [39] and TensorGCN [40], the proposed model outperforms due to its ability of encompassing different node types with appropriate edge weights.

Part of this chapter has been submitted to a journal as S. Supraja and Andy W. H. Khong, "Quad-faceted feature-based graph network for domain-agnostic text classification," *IEEE/ACM Trans. Audio Speech Lang. Proc.*

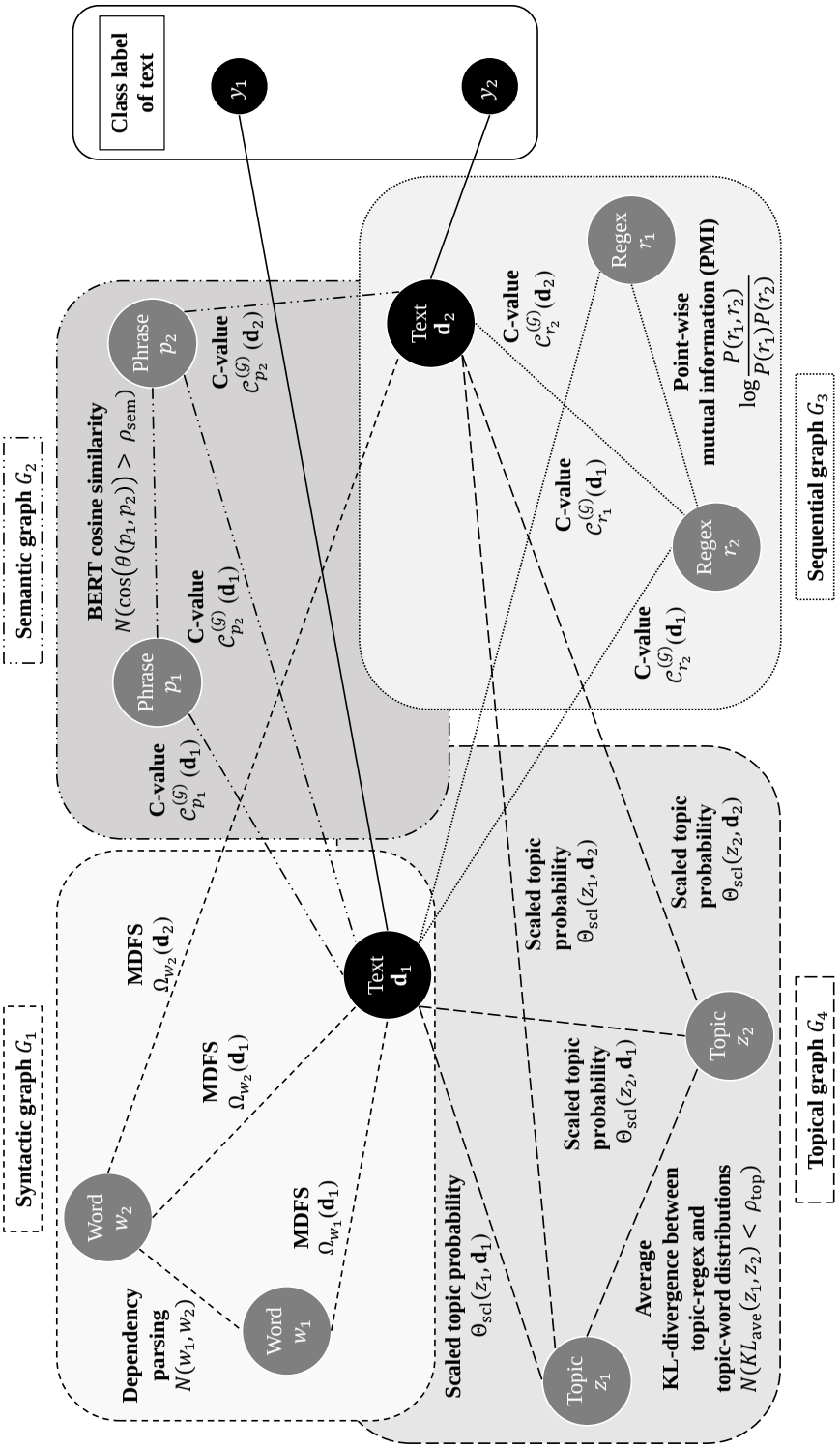


Figure 6.1: Architecture of the proposed quad-faceted feature-based graph network.

6.1 Diversity in heterogeneity

Direct application of existing machine learning techniques may not be suitable for domain-agnostic classification since these methods do not consider the impact of observable and latent features (beyond words) on deriving appropriate representations for accurate classification. Observable features include phrases [161] and their associated regular expressions (regexes) with the latter formed by the concatenation of parts-of-speech (POS) tags [162]. The use of phrases (as elaborated in Chapter 5) is beneficial since these multi-word terms contain contextual information that can achieve meaningful and coherent text representations via a constituency structure [137–139]. Utilizing symbolic rules such as regexes, on the other hand, is important for effective text representation as they are domain-agnostic (as highlighted in Chapter 5), interpretable, contribute to a text's syntax, and provide pattern matching capability [163–165]. As opposed to observable features, latent features can be derived via topic modeling techniques that have shown to provide a global perspective by assigning probability values to word groupings (topics). Such distribution of word co-occurrences across texts offers linguistic insights into language patterns and is important in providing the degree of association between topics and class labels [22]. From an architecture perspective, conventional sequence-based or convolutional neural networks that are often utilized for text classification are limited by their nature to prioritize sequentiality and locality [85]. While these deep learning models capture semantic and syntactic information in the Euclidean space and in local sequences well, they do not account for global word co-occurrences in a corpus that carries non-consecutive and long-distance semantics [39, 87].

Inspired by the use of multiple aspects such as topic, sentence, mention, and entity for relation extraction [166] and information retrieval [167], a quad-faceted feature-based graph network (QGN) is proposed. Similar to the heterogeneous graph convolutional network (GCN) in HeteGCN [168] or SHINE [162], QGN incorporates word, phrase, regex, and topic nodes for domain-agnostic text classification, with each type of node being activated to achieve a different representation. As shown in Figure 6.1, the proposed QGN first comprises a syntactic graph that considers the dependency parsing between word nodes similar to TensorGCN [40]. Unlike the semantic graph applied to word nodes

in TensorGCN and GFN, motivated by the ability of phrases to encapsulate the semantics of a document [141, 169] for information retrieval [69], phrase nodes are employed. The second graph (i.e., semantic graph), therefore, employs cosine similarities between vector representations [170] of phrases derived from bidirectional encoder representations from transformers (BERT) [93] for the edge weights. To further account for regex co-occurrences that contribute to a domain-agnostic class label and similar to the sequential graph employed by TensorGCN and GFN, QGN incorporates the point-wise mutual information (PMI) between all regex nodes in the third graph. The fourth graph (i.e., topical graph) considers the average Kullback–Leibler (KL)-divergence between the word and regex probability vectors across topics for the edge weights between topic nodes. These vectors are derived from the respective co-occurrence frequencies in texts that are used to construct the topic-word and topic-regex distributions. The KL-divergence measure identifies topics that are convergent to each other based on their distributions [171] (e.g., two topics containing noun phrase (NP)-based regexes). This work shows that beyond the previous three graphs based on observable features, this topical graph is also important to generate a meaningful set of latent features for classification via the modeling of topics.

Since observable (words, phrases, and regexes) and latent (topics) features from a text (i.e., document or question) are concatenated, a document node (defined in existing works) is defined as a text node in QGN. With the above quad-faceted graphs, the relationships between the text nodes and the word, phrase, regex, or topic nodes are established. As opposed to the term frequency-inverse document frequency (TF-IDF) value according to (2.1) for the document-word edge weights in TensorGCN, the text-word edge weights in QGN are computed via the modified distinguishing feature selector (MDFS). This allows the model to incorporate both the inter- and intra-class term weights of each word [50] such that a significant word is determined based on its presence in fewer class labels and largely within a particular class label. The text-phrase and text-regex edge weights are computed from the C-value of nested phrases [148] and nested regexes [172], respectively, with the addition of parameters that depend on the inverse frequency of phrase or regex usage across texts to consider the corpus-wide significance. QGN uses the text-topic distribution that signifies the probability of each topic occurring

in a text to compute the text-topic edge weights in QGN. A scaling parameter is then formulated to regularize the range of these text-topic edge weights in line with the other edge weights that are related to the text nodes. The selection of threshold values to determine whether two phrases are similar or a pair of topics are convergent has also been formulated based on the distribution of cosine similarity or KL-divergence values in a given dataset.

6.2 Formulation of the quad-faceted feature-based graph network

QGN, as shown in Figure 6.1, comprises five types of nodes such that

$$\begin{aligned} V = \{&w_1, \ldots, w_i, \ldots, w_{N_V},\ p_1, \ldots, p_k, \ldots, p_{N_P}, \\ &r_1, \ldots, r_k, \ldots, r_{N_R},\ z_1, \ldots, z_j, \ldots, z_{N_Z}, \\ &\mathbf{d}_1, \ldots, \mathbf{d}_m, \ldots, \mathbf{d}_{N_D}\}, \end{aligned} \tag{6.1}$$

where N_P and N_R denote the phrase vocabulary and regex vocabulary sizes, respectively. The nodes w_i, p_k, r_k, z_j, and $\mathbf{d}_m$ are defined as the ith word, kth phrase, kth regex, jth topic, and mth concatenated text, respectively. A phrase $p_k = \{w_{k,1}, \ldots, w_{k,L_{p_k}}\}$ is defined as the kth noun or verb phrase that is made up of L_{p_k} words. The POS tags of all words within the phrase are then grouped together to form a regex $r_k = \{POS(w_{k,1}), \ldots, POS(w_{k,L_{r_k}})\}$ of length $L_{p_k} = L_{r_k}$ [156] and the same index k is used for each phrase and regex. The index j is based on a suitable (pre-defined) number of topics N_Z per dataset. With reference to (6.1),

$$\begin{aligned} \mathbf{d}_m = \{&w_{m,1}, \ldots, w_{m,i}, \ldots, w_{m,L_d},\ p_{m,1}, \ldots, \\ &p_{m,k}, \ldots, p_{m,N_\mathcal{P}},\ r_{m,1}, \ldots, r_{m,k}, \ldots, \\ &r_{m,N_r},\ z_{m,1}, \ldots, z_{m,j}, \ldots, z_{m,N_Z}\} \end{aligned} \tag{6.2}$$

is defined as the mth text with length $L_{\mathbf{d}_m} = L_d + (2 \times N_{\mathcal{P}}) + N_Z$, where L_d is defined as the length of the original text containing only words and $N_{\mathcal{P}}$ denotes the number of extracted phrases. Here, since the number of regexes in a text $N_r = N_{\mathcal{P}}$ (due to the POS-guided phrasal segmentation [69],) the number of phrases and regexes are considered as $2 \times N_{\mathcal{P}}$.

To determine $\mathbf{A}$, QGN encompasses

$$E = \{e_{\text{syn}}(w_i, w_j),\ e_{\text{sem}}(p_i, p_j),\ e_{\text{seq}}(r_i, r_j), e_{\text{top}}(z_i, z_j),\\ e(\mathbf{d}_m, w_i),\ e(\mathbf{d}_m, p_k),\ e(\mathbf{d}_m, r_k),\ e(\mathbf{d}_m, z_j)\}, \tag{6.3}$$

where the subscripts "syn," "sem," "seq," and "top" denote for the syntactic, semantic, sequential, and topical graphs, respectively. The corresponding edge weights between similar nodes are for the ith and jth word nodes, phrase nodes, regex nodes, and topic nodes. On the other hand, the corresponding edge weights with reference to text nodes are between every mth text node and ith word, phrase, regex, or topic node.

As opposed to TensorGCN that considers bi-grams of words for computing word-word edge weights, QGN employs bi-terms that are irrespective of adjacency or order of appearance [173] that allow for cross-referencing against different terms across a text. For instance, a term could refer to a phrase (e.g., *quantitatively_oriented*) or a regex (e.g., *ADV_VERB* that is the concatenation of an adverb and a verb) conjoined by underscore symbol(s). Adopting bi-terms is beneficial for the computation of KL-divergence per topic pair [171], similarities among various phrases in a text, and capturing domain-agnostic relationships between every two regexes. Algorithm 4 provides a formal description of edge weight computations between similar nodes and in relation to text nodes in QGN.

Algorithm 4: Formulation of edge weights in the proposed quad-faceted feature-based graph network.

Input: Concatenated text $\mathbf{d}_m$ using (6.2)
Output: $\mathcal{A}$
for $w_i \in L_d$ **do**
  $e_{\text{syn}}(w_i, w_j) \leftarrow N(w_i, w_j)$ using (6.4)
  $e(\mathbf{d}_m, w_i) \leftarrow \Omega_{w_i}(\mathbf{d}_m)$ using (6.14)
end
for $p_k \in N_{\mathcal{P}}$ **do**
  $e_{\text{sem}}(p_i, p_j) \leftarrow N\Big(\cos\big(\theta(p_i, p_j)\big) > \rho_{\text{sem}}\Big)$ using (6.6)
  $\rho_{\text{sem}} \leftarrow \big(\mu_{\cos(\theta)} - \sigma_{\cos(\theta)}\big)$ using (6.7)
  $e(\mathbf{d}_m, p_k) \leftarrow \mathcal{C}^{(\mathcal{G})}_{p_k}(\mathbf{d}_m)$ using (6.16)
end
for $r_k \in N_r$ **do**
  $e_{\text{seq}}(r_i, r_j) \leftarrow \log\frac{P(r_i, r_j)}{P(r_i)P(r_j)}$ using (6.8)
  $e(\mathbf{d}_m, r_k) \leftarrow \mathcal{C}^{(\mathcal{G})}_{r_k}(\mathbf{d}_m)$ using (6.18)
end
for $z_j \in N_Z$ **do**
  $e_{\text{top}}(z_i, z_j) \leftarrow N\big(KL_{\text{ave}}(z_i, z_j) < \rho_{\text{top}}\big)$ using (6.12)
  $\rho_{\text{top}} \leftarrow \big(\mu_{KL_{\text{ave}}} + \sigma_{KL_{\text{ave}}}\big)$ using (6.13)
  $e(\mathbf{d}_m, z_j) \leftarrow \Theta_{\text{scl}}(z_j, \mathbf{d}_m)$ using (6.19)
end

6.2.1 Edge weight computations between similar nodes for syntactic, semantic, sequential, and topical graphs

The syntactic graph G_1 that activates only the word nodes and the corresponding edges employs dependency parsing to compute the word-word edge weight

$$e_{\text{syn}}(w_i, w_j) = N(w_i, w_j), \tag{6.4}$$

where $i \neq j$ and $N(w_i, w_j)$ denotes the number of times dependent word pairs occur in the corpus. Following the assumption made in TensorGCN, although the extracted dependency is directed, it can be treated as an undirected relationship. Each pair of words is considered to be dependent on each other unless they have the same parse (including *root*) tags [40]. Syntactic structures such as dependency trees have been

shown to accurately encode the correlation between words, in particular with graph neural networks [174].

The semantic graph G_2 activates only the phrase nodes and the corresponding edges. Graph G_2 employs BERT to generate an embedding vector for each phrase node [175]. In comparison to word2vec [94] or GloVE [95] used in GFN, BERT possesses the ability to better depict the contextual information (i.e., semantics) of a text. Discovering relationships through the use of phrases is more effective than words since phrases convey a more holistic meaning [175, 176]. With the above phrase nodes, the cosine similarity value between vector representations of each pair of phrases p_i and p_j is then computed via [161]

$$\cos\big(\theta(p_i, p_j)\big) = \frac{\mathbf{v}_{p_i} \cdot \mathbf{v}_{p_j}}{||\mathbf{v}_{p_i}||||\mathbf{v}_{p_j}||}, \tag{6.5}$$

where $\mathbf{v}_{p_i}$ denotes the BERT embedding vector corresponding to p_i. With $0 \leq \cos\big(\theta(p_i, p_j)\big) \leq 1$, a value of 1 implies an almost semantically similar pair of phrases. The edge weight of each pair of phrase nodes in G_2 is then computed as

$$e_{\text{sem}}(p_i, p_j) = N\Big(\cos\big(\theta(p_i, p_j)\big) > \rho_{\text{sem}}\Big), \tag{6.6}$$

where $i \neq j$ and $N\Big(\cos\big(\theta(p_i, p_j)\big) > \rho_{\text{sem}}\Big)$ denotes the number of times $\cos\big(\theta(p_i, p_j)\big)$ exceeds a threshold ρ_{sem}, implying the co-occurrence frequency of phrases deemed similar.

The threshold for semantic similarity ρ_{sem} is estimated based on the distribution of cosine similarity values among all phrase pairs $\cos(\theta) = \{\cos\big(\theta(p_1, p_2)\big), \ldots, \cos\big(\theta(p_{N_{P-1}}, p_{N_P})\big)\}$ in a given corpus. Figure 6.2 shows, for an illustrative corpus extracted from the Argumentative (Arg.) Zones dataset described in Section 6.3, the probability density function of $\cos\big(\theta(p_i, p_j)\big)$ estimated using parametric density function [177]. In addition, the skewness and kurtosis of the distribution have been measured. The skewness is a measure of symmetry, i.e., a distribution is symmetric if it is the same to the right and left of the center point and has a skewness value close to zero. The skewness of the distribution in Figure 6.2 is 0.06, implying that the distribution is almost symmetric. The kurtosis, which measures whether the distribution is heavy-tailed (positive value) or light-tailed (negative value) relative to a normal distribution, has a value of -0.15 for this distribu-

tion. This value implies that the distribution is light-tailed and moderately close to a normal distribution.This corpus consists of approximately 1000 documents with mean value of $\mu_{\cos(\theta)} = 0.61$ and standard deviation of $\sigma_{\cos(\theta)} = 0.13$. A threshold value

$$\rho_{\text{sem}} = \left(\mu_{\cos(\theta)} - \sigma_{\cos(\theta)}\right) \tag{6.7}$$

is proposed implying that phrase pairs with a cosine similarity value higher than one standard deviation away from the mean are considered as being similar and will be used to compute the phrase-phrase edge weights according to (6.6). This computation is in line with the effectiveness of semantically similar phrases contributing to a higher text classification performance [178].

The sequential graph G_3 computes the co-occurrence between various regex nodes in each text $\mathbf{d}_m$ via the PMI [85]. The PMI of a pair of of regex nodes quantifies the discrepancy between the probability of their coincidence given their joint distribution and their individual distributions, assuming independence. The objective of G_3 is for the subsequent classifier to determine whether resemblance among patterns of regex co-occurrences constitute a class label. The edge weight of each pair of regex nodes r_i and r_j in G_3 is computed as

$$e_{\text{seq}}(r_i, r_j) = \log \frac{P(r_i, r_j)}{P(r_i)P(r_j)}, \tag{6.8}$$

where $i \neq j$. The variable $P(r_i)$ denotes the marginal distribution of occurrence for r_i and

$$P(r_i, r_j) = \frac{N(r_i, r_j)}{L_{\mathbf{d}_m}}. \tag{6.9}$$

denotes the joint probability of the regex node pair co-occurring in the same text. Here, $N(r_i, r_j)$ denotes the number of co-occurrences between every pair of regexes within the length of the mth concatenated text $L_{\mathbf{d}_m}$. The proposed QGN model does not take a sliding window length into account since the objective is to consider all possible combinations of regex occurrences within the entire text.

The topical graph G_4 activates only the topic nodes and the corresponding edges to compute the topic-topic edge weights. These weights are computed from the KL-divergence between the vectors of regex and word probabilities across topics derived

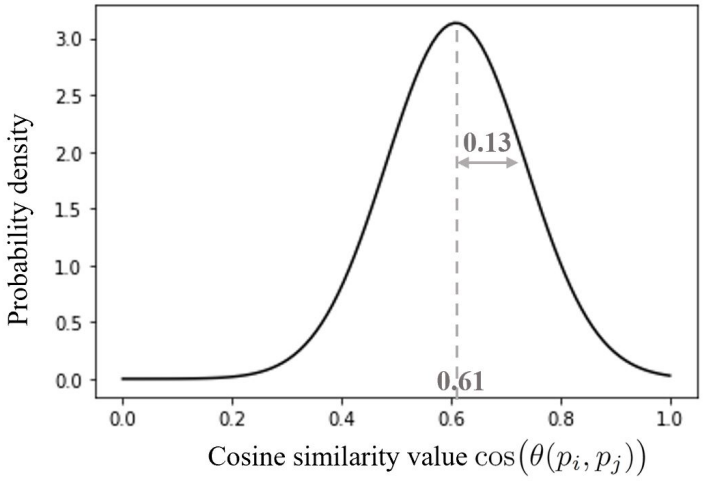


Figure 6.2: Estimated probability density function of pairwise cosine similarity values (phrase pairs) across a corpus. The distribution fit values are 0.61 for the mean and 0.13 for the standard deviation.

from the topic-regex distribution $\boldsymbol{\eta} \in \mathbb{R}^{N_Z \times N_R}$ and the topic-word distribution $\boldsymbol{\Phi} \in \mathbb{R}^{N_Z \times N_V}$, respectively. The framework of Qu-LDA [172] is adopted in this work such that $\boldsymbol{\eta}$ considers the weights of all phrases constituting each regex. These phrase weights are, in turn, the sum of the MDFS term weighting of the words that each phrase consists of. Similarly, $\boldsymbol{\Phi}$ takes the importance of each word based on class label information into account. Given the efficiency of these distributions for domain-agnostic question classification, they are employed in G_4 for comparing between topics.

The KL-divergence, which quantifies how one probability distribution differs from another, is applied for $\boldsymbol{\eta}$ and $\boldsymbol{\Phi}$, and is computed between topics z_i and z_j as

$$KL(\boldsymbol{\eta}_{z_i} || \boldsymbol{\eta}_{z_j}) = \int_{-\infty}^{\infty} \boldsymbol{\eta}_{z_i} \log\left(\frac{\boldsymbol{\eta}_{z_i}}{\boldsymbol{\eta}_{z_j}}\right) d\boldsymbol{\eta}_{z_i} \tag{6.10}$$

for the topic-regex distribution and similarly for the topic-word distribution by replacing instances of $\boldsymbol{\eta}$ with $\boldsymbol{\Phi}$. In (6.10), $\boldsymbol{\eta}_{z_i}$ corresponds to the vector of all regex probabilities for topic z_i. Hence, (6.10) is integrating over the distribution of regex occurrences in each topic from (5.7) and similarly, the distribution of word occurrences in each regex present in each topic from (5.8) to obtain $KL(\boldsymbol{\Phi}_{z_i} || \boldsymbol{\Phi}_{z_j})$. It can be noted from the above that

the KL-divergence is computed in both directions since the relationships between topics is considered irrespective of the order of appearance. Subsequently, the average of both KL-divergence values

$$KL_{\text{ave}}(z_i, z_j) = \frac{KL(\boldsymbol{\eta}_{z_i} || \boldsymbol{\eta}_{z_j}) + KL(\boldsymbol{\Phi}_{z_i} || \boldsymbol{\Phi}_{z_j})}{2} \tag{6.11}$$

is computed to consider the impact of both word and regex occurrences in a topic. The edge weight of each pair of topic nodes in G_4 is determined via

$$e_{\text{top}}(z_i, z_j) = N\big(KL_{\text{ave}}(z_i, z_j) < \rho_{\text{top}}\big), \tag{6.12}$$

where $i \neq j$ and $N\big(KL_{\text{ave}}(z_i, z_j) < \rho_{\text{top}}\big)$ denotes the number of times each topic pair has converging (i.e., low) average KL-divergence values below the threshold ρ_{top} in a corpus.

Similar to G_2, the threshold for convergence consideration ρ_{top} is estimated based on the distribution of average KL-divergence values among all topic pairs $(KL_{\text{ave}}) = \{KL_{\text{ave}}(z_1, z_2), \ldots, KL_{\text{ave}}(z_{N_{Z-1}}, z_{N_Z})\}$ in a given corpus. In contrast to cosine similarities that favor larger values, KL-divergence favors smaller values that correspond to converging/similar distributions. Hence, a threshold value

$$\rho_{\text{top}} = \big(\mu_{KL_{\text{ave}}} + \sigma_{KL_{\text{ave}}}\big) \tag{6.13}$$

is proposed, where $\mu_{KL_{\text{ave}}}$ denotes the mean and $\sigma_{KL_{\text{ave}}}$ denotes the standard deviation of the average KL-divergence values. Therefore, (6.13) implies that topic pairs that are assigned an average KL-divergence value less than one standard deviation away from the mean are considered as being similar and will be used to compute the topic-topic edge weights according to (6.12). This low or convergent KL-divergence value highlights that the similar distribution of either words or regexes, or both words and regexes between two texts is analogous to these two texts likely being classified into the same class label.

6.2.2 Edge weight computation with reference to text nodes

With reference to Figure 6.1, the text-word edge weight between the mth text node and ith word node in G_1 is given by the class-based term weighting scheme MDFS [50] (as discussed in Chapter 2)

$$e(\mathbf{d}_m, w_i) = \Omega_{w_i}(\mathbf{d}_m). \tag{6.14}$$

The intricacies of inter- and intra-class attributes are not reflected by the use of TF-IDF (in TextGCN or TensorGCN) which only considers corpus-wide frequencies. Hence, MDFS is suitable to compute $e(\mathbf{d}_m, w_i)$ in QGN for representing the relation of words to class labels.

The text-phrase edge weight between the mth text node and kth phrase node in G_2 is given by

$$e(\mathbf{d}_m, p_k) = \mathcal{C}_{p_k}^{(\mathcal{G})}(\mathbf{d}_m). \tag{6.15}$$

The modified C-value for nested phrases is formulated as

$$\mathcal{C}_{p_k}^{(\mathcal{G})}(\mathbf{d}_m) = \begin{cases} \left(\frac{N_D}{N_{p_k}}\right)\log_2 L_{p_k}, & \text{if } p_k \text{ is not a nested phrase;} \\ \left(\left(\frac{N_D}{N_{p_k}}\right) - \frac{\sum_{p_k(s)} N_p^{(s)}}{N_s}\right)\log_2 L_{p_k}, & \\ & \text{if } p_k \text{ is a nested phrase.} \end{cases} \tag{6.16}$$

This C-value is modified from the original ($\mathcal{C}_{p_k}$ computed using (5.1)) by replacing the frequency of occurrence N_{p_k} with the inverse frequency ($1/N_{p_k}$) extracted partially from the TF-IDF computation in (2.1). This replacement is necessary to take the importance of each phrase across the corpus into account such that a rare phrase obtains higher significance. The above formulation also considers whether each phrase is a nested phrase before assigning a high C-value to a phrase.

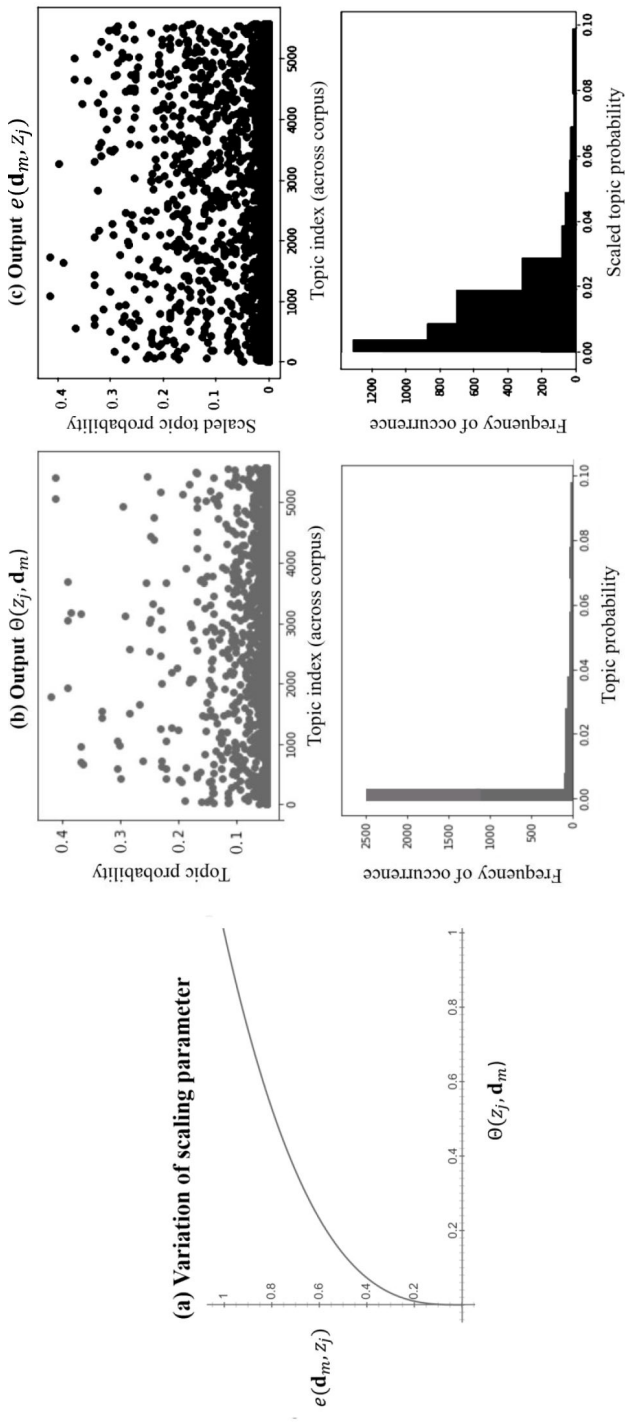


Figure 6.3: (a) Variation of scaling parameter and impact of scaling parameter on topic probabilities comparing the cases (b) for output $\Theta(z_j, \mathbf{d}_m)$ (without scaling) versus (c) for output $e(\mathbf{d}_m, z_j)$ (with scaling).

The C-value for nested regexes [172] evaluates the relevance of regexes. The text-regex edge weight between the mth text node and kth regex node in G_3 given by

$$e(\mathbf{d}_m, r_k) = \mathcal{C}_{r_k}^{(\mathcal{G})}(\mathbf{d}_m) \tag{6.17}$$

is computed via the modified C-value for nested regexes

$$\mathcal{C}_{r_k}^{(\mathcal{G})}(\mathbf{d}_m) = \begin{cases} (0.01)\varphi_{r_k} L_{r_k}\left(\frac{N_D}{N_{r_k}}\right), \\ \qquad\qquad \text{if } r_k \text{ is not a nested regex;} \\ (0.01)\varphi_{r_k} L_{r_k}\left(\left(\frac{N_D}{N_{r_k}}\right) - \frac{\sum_{r_k(S)} N_r^{(S)}}{N_S}\right), \\ \qquad\qquad \text{if } r_k \text{ is a nested regex.} \end{cases} \tag{6.18}$$

As opposed to Qu-LDA which employs a different φ_{r_k} for regexes associated with verb phrases [172], the formulation of (6.18) computes the scaling parameter $\varphi_{r_k} = 1 - \exp(-10^{(L_{r_k}-3)})$ by only considering the regex length into account. It is worth noting that the variable φ_{r_k} is not applied in favor of verb phrase-based regexes since texts do not require a specific conditioning toward verb phrase-based regexes. This allows equal importance to be given to both noun and verb phrase-based regexes for determining combinations of regexes that represent a text. In addition, instead of frequency N_{r_k}, $1/N_{r_k}$ similar to (6.16) has been employed, implying the significance of a rare regex toward identifying a class label. The scaling constant 0.01 that serves as a substitute for the logarithmic function in (6.16) is in place for $e(\mathbf{d}_m, r_k)$ to conform to the range of other edge weights in relation to text nodes for compatibility reasons. Following the formulation of Qu-LDA [172], (6.18) does not contain the logarithmic function to consider regexes derived from single-word phrases that aid more accurate classification.

For topical graph G_4, the text-topic edge weight between the mth text node and jth topic node is computed via a scaled topic probability $\Theta_{\text{scl}}(z_j, \mathbf{d}_m)$ that has a logarithmic relationship with its original topic probability and can be modeled as

$$e(\mathbf{d}_m, z_j) = \big(\Theta(z_j, \mathbf{d}_m)\big)(N_Z)^{(-0.5)\log\left(\Theta(z_j, \mathbf{d}_m)\right)}, \tag{6.19}$$

where $\Theta(z_j, \mathbf{d}_m)$ denotes the probability of topic z_j for the mth text. Each topic probability forms the vector of topic probabilities per text (conventionally used as features for classification) that is obtained from the text-topic distribution $\mathbf{\Theta} \in \mathbb{R}^{N_D \times N_Z}$ constructed based on asymmetric priors [34] due to stop-words not being removed. However, with direct application of these topic probability values, the range of $\Theta(z_j, \mathbf{d}_m)$ is incompatible when compared to $e(\mathbf{d}_m, w_i)$, $e(\mathbf{d}_m, p_k)$, and $e(\mathbf{d}_m, r_k)$. This is because, topics that are insignificant in representing a text obtain low probabilities close to zero; these values are being ignored albeit having an impact (to a smaller extent.) Hence, a new scaling formulation has been employed such that the insignificant values are better represented while the higher probabilities that correspond to the most relevant topics are being less emphasized. Despite scaling the values for better compatibility and comparison, the desired differences between the high and low topic probabilities are still maintained to discriminate among the class labels.

The relationship between $\Theta_{\text{scl}}(z_j, \mathbf{d}_m)$ and $\Theta(z_j, \mathbf{d}_m)$ is illustrated in Figure 6.3(a). Figure 6.3(b) illustrates the range of probabilities across topics (with indices of length $N_D \times N_Z$) before performing the proposed scaling operation while Figure 6.3(c) highlights the probabilities after applying (6.19) for a given corpus. Once the topics were generated, the same order of topics was maintained while plotting these figures. From the scatter plot in Figure 6.3(b), it can be noted that majority of $\Theta(z_j, \mathbf{d}_m)$ are low; only topics with prominent values are seen as anomaly points. This is highlighted via the histogram (in the bottom panel) that exhibits a heavy-tailed distribution. The scatter plot in Figure 6.3(c), on the other hand, exhibits a less skewed set of points based on the scaling in (6.19) and is reflected via the histogram with a lighter-tailed distribution. The above implies that moderate emphasis is given to less significant topic probabilities which is subsequently shown to result in better feature representation for classification.

With all quad-faceted graphs, the graph tensor in QGN that consists of text, word, phrase, regex, and topic nodes is then defined as $\mathcal{T} = (G_1, G_2, G_3, G_4)$. Similar to the TensorGCN architecture as described in Section 2.4.3, the edge weights computed from Sections 6.2.1 and 6.2.2 serve as inputs to the graph adjacency tensor $\mathcal{A} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4)$. The corresponding graph feature tensor is then defined as $\mathcal{H} = (\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \mathbf{H}_4)$.

6.3 Domain-agnostic document datasets and labeling taxonomies

Two different document datasets with class labels that do not depend on the domain have been used for evaluation. Details of the datasets, along with examples of statements/extracts corresponding to each domain-agnostic category are tabulated in Table 6.1. Although these datasets have not been widely used in document classification tasks, this thesis adopts the domain-agnostic properties of the class labels belonging to the respective documents for analysis. In addition, three of the question datasets (NU, ARC, and LREC) as described in Chapter 5 are used for performance evaluation.

In the Argumentative (Arg.) Zones dataset, documents have been labeled into argumentative zones by extracting sentences from the abstract and introduction of more than 3000 articles from biology, machine learning, and psychology journals [179–181]. This thesis selects 1257 documents with class labels *Aim, Own, Contrast*, and *Miscellaneous*. These labels correspond, respectively, to a specific research goal of the paper, description of own work presented in the paper, statements of comparison with other works along with the limitations of existing works, and scientific background along with descriptions of other researchers' works.

The SkillsFuture Singapore (SSG) dataset involves the mapping of 610 job descriptions across various industries to skills. These skills are general competencies expected of employees across several job scopes [37]. Thirty-four sectors with several sub-roles that have been mapped to critical core skills (*Thinking critically, Interacting with others*, and *Staying relevant*) [38] have been used in this thesis.

6.4 Hyperparameter selection

The same intra- and inter-propagation methods established in TensorGCN according to (2.20) have been employed in QGN. Hyperparameters for the embeddings and convolutional networks are, therefore, adopted from the TensorGCN implementation. Two layers of TensorGCN are used with the first layer dimension of node embedding being

defined as 200 and the second equal to N_L. During training, the dropout rate is set as 0.8 and the L_2 loss weight is set as 1e-4. Seventy percent of each dataset is used for training with the remaining thirty percent for testing. Ten percent of the training set is randomly selected as the validation set. The Adam optimizer with a learning rate of 0.002 is used along with a maximum of 1000 training epochs. The window size in QGN is equivalent to the length of each concatenated text $L_{\mathbf{d}_m}$.

Phrase embeddings are trained using BERT with a vector dimension of 768. The remaining topic modeling-related hyperparameters are obtained from Qu-LDA [172]. The number of topics was evaluated from 5 to 60 in intervals of 5 with the optimal number being one that achieves the highest macro-average F1 score. The optimal number of topics for the Arg. Zones and SSG datasets was found to be 50 while it was 20 for the remaining two datasets; the higher number of topics found in the first two datasets was due to the relatively longer documents. Since questions in the NU, ARC, and LREC datasets are shorter, these datasets require a lower number of topics to avoid over-fitting. Symmetric $\beta = 0.01$ values are employed while the asymmetric α values were derived via Newton-Raphson optimization [60]. The asymmetric priors λ for $\boldsymbol{\eta}$ were computed via the original C-value for nested regexes in Qu-LDA [172]. 1000 Gibbs sampling iterations for both training and testing have been used.

Table 6.1: Details of datasets used for ADC performance evaluation (the abbreviation "Arg." refers to Argumentative)

	Arg. Zones dataset [179–181]	**SSG dataset [37, 38]**
Source	University of California	SkillsFuture Singapore
Number of documents	1257	610
Type of documents	Journal articles	Job descriptions
Domain-agnostic category	Argumentative zones	Generic skillsets
Class labels	Aim (15.4%) Own (34.8%) Contrast (13.5%) Miscellaneous (36.3%)	Thinking critically (37.2%) Interacting with others (53.4%) Staying relevant (9.4%)
Examples of NPs	our contribution traditional approaches the mechanism	good time management the initiative technical feasibility
Examples of VPs	are estimated experimentally show identically distributed	also perform practice change is required

Table 6.2: Macro-average F1 scores for each dataset

Deep learning technique	Arg. Zones	SSG	NU	ARC	LREC
Bi-LSTM	0.481	0.387	0.680	0.687	0.689
CNN	0.498	0.384	0.700	0.663	0.651
TextGCN [39]	0.496	**0.395**	0.693	0.703	0.651
TensorGCN [40]	0.533	0.373	0.748	0.706	0.701
QGN	**0.586**	0.373	**0.758**	**0.742**	**0.756**

6.5 Quantitative analysis

Performance of the proposed QGN is compared against baseline deep learning methods for text classification such as bidirectional long short-term memory network (Bi-LSTM), convolutional neural network (CNN), TextGCN [39], and TensorGCN [40]. GFN is not used for comparison since it does not incorporate text nodes that are required for domain-agnostic classification. To evaluate the classification reliability of the above methods with the actual class labels, the F1 measure is used to observe the extent of how each technique minimizes false positives and false negatives. The macro-average F1 scores for the five datasets are shown in Table 6.2.

The Arg. Zones dataset highlights the efficacy of employing graph networks for domain-agnostic document classification. The overall classification performance is comparable for Bi-LSTM, CNN, and TextGCN due to the lack of incorporating unique features for domain-agnostic classification. This is reflected by the macro-average F1 scores of 0.481, 0.498, and 0.496, respectively in Table 6.2. The increase in performance of TensorGCN (with a macro-average score of 0.533) is due to the consideration of syntactic, semantic, and sequential computation methods of word-word edge weights. QGN augments TensorGCN by extending word nodes to other observable and latent nodes such as phrases, regexes, and topics that holistically represent a text. The use of specific computations between similar node types based on unique properties (e.g., KL-divergence between topics since they are represented as probability distributions) and for the relationships with text nodes beyond TF-IDF (e.g., C-value for nested regexes) allows better discrimination of class labels since it accounts for distinguishable features being extracted. This is reflected by an approximate 10% increase in the macro-average F1

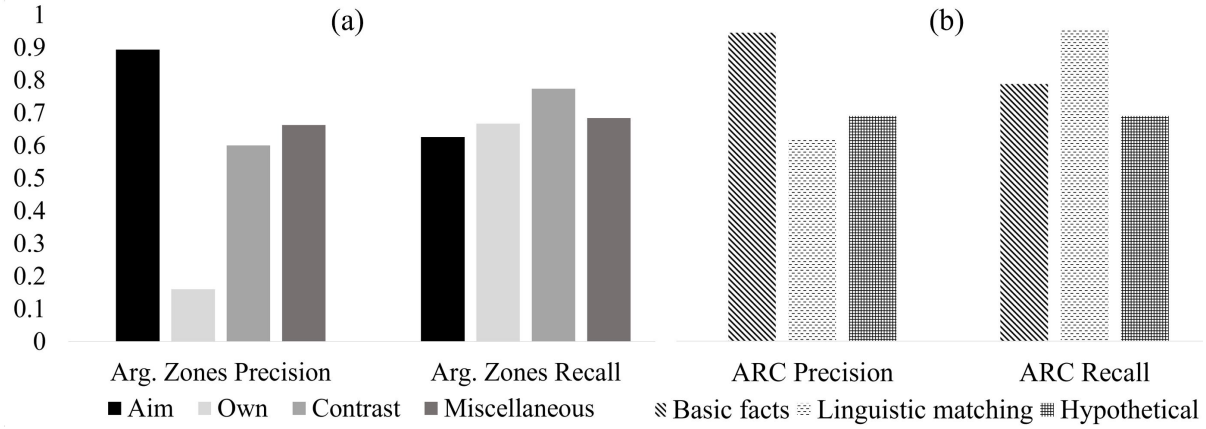


Figure 6.4: Performance of QGN for the Arg. Zones and ARC datasets via precision and recall scores.

score from 0.533 for TensorGCN to 0.586 for QGN as seen in Table 6.2.

Similar performance improvement is observed for the other three question datasets. For the ARC and LREC datasets, the proposed QGN model achieves approximately 5-8% increase (from 0.706 to 0.742 and from 0.701 to 0.756, respectively) in the macro-average F1 scores in comparison to TensorGCN. It is also useful to note that the models exhibit higher performance for these datasets compared to the Arg. Zones dataset. This is due to the existence of overlapping content between the description of an author's own work across other class labels resulting in mis-classification for the *Own* class label. This effect can be observed in Figure 6.4(a), where the Arg. Zones dataset suffers from a low precision score of 0.16 for the *Own* class label when processed via QGN. As opposed to the Arg. Zones dataset, the ARC dataset achieves consistently high precision and recall scores across all three class labels due to distinguishable class labels as seen in Figure 6.4(b), resulting in a higher macro-average F1 score than the Arg. Zones dataset.

It is useful to note that poor classification performance is exhibited across all models for the SSG dataset. On further investigation, it was determined that this poor performance is attributed to the nature of the job descriptions in relation to the domain-agnostic class labels. Since there exist overlaps with multiple skills being involved in each job role, confining a job description to one skill achieves low macro-average F1 scores.

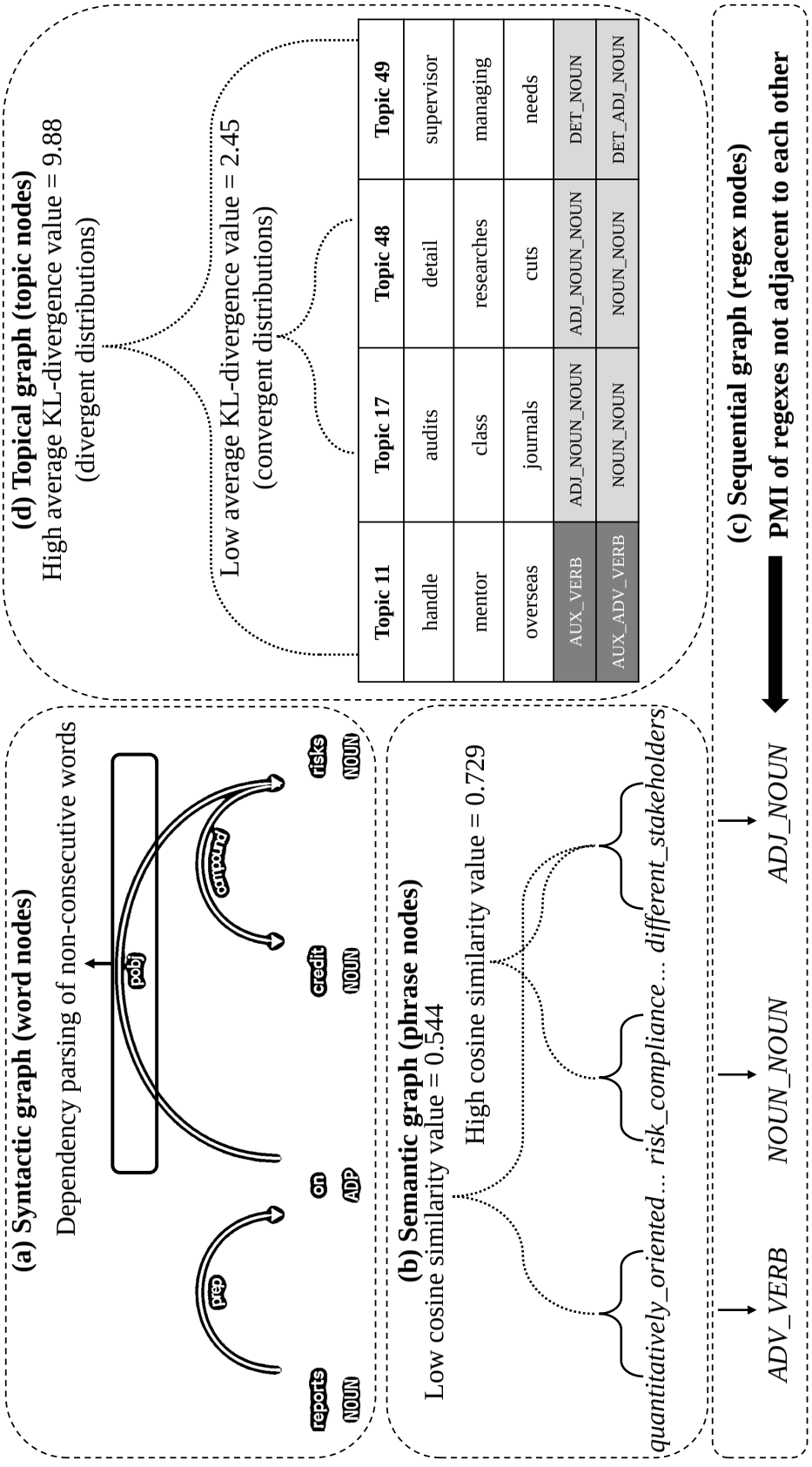


Figure 6.5: Illustrative examples of the four types of graphs in the proposed quad-faceted feature-based graph network for the SSG dataset. The syntactic graph comprising word nodes is shown in (a), the semantic graph made up of phrase nodes is depicted in (b), the sequential graph consisting of regex nodes is shown in (c), while the topical graph that constitutes topic nodes can be seen in (d).

Figure 6.6: Illustrative examples of the four types of graphs in the proposed quad-faceted feature-based graph network for the NU dataset. The syntactic graph comprising word nodes is shown in (a), the semantic graph made up of phrase nodes is depicted in (b), the sequential graph consisting of regex nodes is shown in (c), while the topical graph that constitutes topic nodes can be seen in (d).

Table 6.3: Examples of phrases and cosine similarity values within and across class labels in the ARC dataset

Phrase p_i	**Class label**	**Phrase** p_j	**Class label**	$\cos\big(\theta(p_i, p_j)\big)$
the_least_likely_way	Basic facts	which_layer	Basic facts	0.705
her_thinking	Hypothetical	an_investigation	Hypothetical	0.749
the_following	Linguistic matching	the_best_prediction	Hypothetical	0.577
what_part	Basic facts	the_best_explanation	Linguistic matching	0.568

Table 6.4: Examples of edge weights in relation to text nodes using TF-IDF versus QGN computations for the Arg. Zones dataset

Node type	**Node name**	**TF-IDF value**	**Computation in QGN**	**New value**
Regex	$NOUN\_NOUN$	0.466	$e(\mathbf{d}_m, r_k) = \mathcal{C}_{r_k}^{(\mathcal{G})}(\mathbf{d}_m)$ using (6.18)	2.216
Topic	Topic 36	10.122	$e(\mathbf{d}_m, z_j) = \Theta_{\text{scl}}(z_j)(\mathbf{d}_m)$ using (6.19)	2.262

6.6 Qualitative analysis

To gain further insights on the reasons for QGN achieving the highest classification performance, the impact of unique edge weights between similar types of nodes, as well as, the difference in edge weights with reference to text nodes if TF-IDF was instead being employed have been examined. Such node interaction patterns have shown to provide informative components associated with the class labels [182].

With reference to the computations as described in Section 6.2.1, illustrative examples of the quad-faceted graphs are shown in Figure 6.5 and Figure 6.6 using an exemplar document that describes the job description of a risk compliance and legal credit risk officer/manager from the SSG dataset and an exemplar pre-processed question "*create a storyboard for a sequel to your book use the same characters*" from the NU dataset, respectively. The syntactic graph comprising word nodes, the semantic graph made up of phrase nodes, the sequential graph consisting of regex nodes, and the topical graph that constitutes topic nodes are depicted in Figure 6.5(a)-(d), respectively. The impact of each graph highlights the need to consider the unique properties of each node type and the corresponding relationships, particularly in a bi-term context. From Figure 6.5(a) and Figure 6.6(a), it can be seen that by considering bi-terms, the dependency parsing of non-consecutive words is extracted (highlighted via the rectangle). If bi-grams were instead extracted, only adjacent relationships will be identified. However, the use of bi-grams has been shown to only be suitable for long texts but ineffective for short texts such as questions since the frequency of bi-grams is low; this leads to inefficient modeling of word co-occurrence and dependency for subsequent classification [183].

In Figure 6.5(b), three different phrases (two noun phrases and one verb phrase) from the document are shown. In Figure 6.6(b), four different phrases (three noun phrases and one verb phrase) from the question are shown. In Figure 6.5(b), the BERT cosine similarity has a relatively high value of 0.729 between the phrases *risk_compliance* and *different_stakeholders* as both are noun phrases referring to a common context on performing a real-life application that deals with people involved in it. On the other hand, the BERT cosine similarity has a lower value of 0.544 between the phrases *quantitatively_oriented* and *different_stakeholders* since the former is a verb phrase that requires

an ability while the latter is a noun phrase stating an object. Similarly, in Figure 6.6(b), the BERT cosine similarity has a relatively high value between the phrases *a_storyboard* and *a_sequel* as both are noun phrases referring to a common context. On the other hand, the BERT cosine similarity attains a lower value between the phrases *create* and *the_same_characters* since the former is a verb phrase that requires an ability while the latter is a noun phrase with reference to an object. Hence, relationships that are strongly built among such similar pairs of phrases are important in determining the features of a document that contribute toward the classification into respective class labels.

To further substantiate Figure 6.5(b), the effect of bi-terms in providing more contextual information has been investigated. Applying cosine similarities between BERT vectors of phrases is shown to be effective in differentiating among class labels. Examples of phrases and cosine similarity values within and across class labels in the ARC dataset are shown in Table 6.3. It can be seen that phrases that are strongly indicative of a class label (e.g., *the_least_likely_way* and *which_layer* belonging to *Basic facts*) achieve high $\cos\big(\theta(p_i, p_j)\big)$ values according to (6.5), whereas phrases in different class labels (e.g., *the_following* that is strongly toward *Linguistic matching* and *the_best_prediction* that is highly associated with *Hypothetical*) achieve low $\cos\big(\theta(p_i, p_j)\big)$ values. As opposed to the use of bi-grams and word nodes in the semantic graph of TensorGCN, considering bi-terms and phrase nodes in QGN enables a wider range of co-occurrences and more meaningful relationships that result in good classification performance.

Using the same set of phrases in Figure 6.5(b) and Figure 6.6(b), Figure 6.5(c) and Figure 6.6(c), respectively highlight that the PMI is computed for regexes not adjacent to each other. For PMIs computed between every regex pair derived from bi-terms, meaningful collocation pairs that are discriminative features (e.g., *NOUN_NOUN* and *ADJ_NOUN*) achieve high PMI since the probability of co-occurrence is only modestly lower than the marginal probabilities of occurrence of each regex within the text. Conversely, a pair of regexes whose marginal probabilities of occurrence are considerably higher than their probability of co-occurrence achieves a low PMI. For instance, verb phrase-based regexes and noun phrase-based regexes do not necessarily occur together. Similar to the identification of dependent pairs of words, the use of bi-terms allows for detection of specific regex pairs across a text that achieve high PMI to distinguish such

features apart from other class labels.

Figure 6.5(d) and Figure 6.6(d) show tables of four sample topics from the respective datasets that combine the outputs from the topic-word (unshaded) and topic-regex (shaded) distributions. It is worth noting that the words and regexes within each topic might not have a direct relationship with each other; the algorithm groups these items into topics based on frequencies of co-occurrence determined by the two distributions. From Figure 6.5(d), it can be noted that Topic 11 and Topic 49 achieve a high average KL-divergence value of 9.88 (divergent distributions). This is mainly due to verb phrase-based regexes largely belonging to the former topic while the latter topic consists of noun phrase-based regexes although the presence of the words in both topics (e.g., "mentor" and "supervisor") suggests similar contextual information. In contrast, Topic 17 and Topic 48 achieve a low average KL-divergence value of 2.45 (convergent distributions) since both topics comprise noun phrase-based regexes and words with similar contextual information (e.g., "audits" and "researches".) From Figure 6.6(d), it can be noted that Topic A and Topic D achieve a high average KL-divergence value (divergent distributions). This is due to verb phrase-based regexes largely belonging to the former topic while the latter topic consists of noun phrase-based regexes. In addition, the presence of the words in both topics suggests dissimilar contextual information. For instance, words such as *inspect* and *defend* are of higher cognitive complexities in comparison to *state* or *illustrate* in the NU dataset. In contrast, Topic B and Topic C achieve a low average KL-divergence value (convergent distributions) since both topics comprise noun phrase-based regexes and words with similar contextual information such as *create* and *construct*. Given the combinations of highly associated topics in a text reflected via low KL-divergence values, QGN can differentiate among the topical patterns corresponding to each class label.

In terms of the edge weights computed with respect to text nodes described in Section 6.2.2, Table 6.4 shows examples of edge weights using TF-IDF compared to unique computations for the Arg. Zones dataset. For instance, a particular regex $NOUN\_NOUN$ achieves a text-regex TF-IDF edge weight of 0.466, indicating a commonly used regex that receives a low weight. In contrast, the modified C-value $\mathcal{C}_{r_k}^{(\mathcal{G})}(\mathbf{d}_m) = 2.216$ computed using (6.18) suggests a higher importance given to nested regexes. This higher C-value in

Table 6.5: Ablation test results (macro-average F1 scores)

Removal of graph	ARC	LREC
Syntactic (G_1)	0.702	0.714
Semantic (G_2)	0.705	0.690
Sequential (G_3)	0.668	0.699
Topical (G_4)	0.710	0.717
None (proposed model)	**0.742**	**0.756**

a text indicates better discriminability against other texts with respect to G_3. Similarly, Topic 36 achieves an exceptionally high edge weight of 10.122 via TF-IDF while a relatively similar value of $\Theta_{\text{scl}}(z_j, \mathbf{d}_m) = 2.262$ is obtained for QGN. Therefore, a significantly overemphasized topic probability via TF-IDF leads to an incorrect representation of G_4, in turn, resulting in inaccurate classification. The above illustrates that the relevance of edge weights with respect to text nodes according to each node type is important in determining meaningful feature representations of a text in QGN compared to TensorGCN or TextGCN that utilize an undesirable one-size-fits-all approach.

6.7 Ablation test

An ablation study was performed to analyze the impact of each type of graph when it was removed from the QGN architecture. The macro-average F1 scores for each scenario of QGN without each graph type for two of the datasets (ARC and LREC) are tabulated in Table 6.5. It can be seen that the sensitivity of each graph varies for different datasets. For instance, the removal of G_3 results in an approximate 10% drop in classification performance compared to the proposed model for the ARC dataset. It is worth noting that G_3 seems to be the most important graph for the ARC dataset since the structure of questions (i.e., regexes) are different across the reasoning capability class labels. For example, the *Basic facts* class label tends to have more noun phrase-based regexes than the other two class labels. The sequential presence of regexes, therefore, has the most significant contribution toward the classification performance.

However, removing G_2 achieves the lowest performance for the LREC dataset. This

could be due to the semantic information that differentiates among the question types. For instance, phrases such as *many_situations*, *two_examples*, and *many_reasons* contain a common meaning related to the *Answers will vary* class label. On the other hand, phrases such as *the_right*, *the_room*, *this_picture*, and *this_mini_lab*, which refer to a particular situation, belong to the *Context sensitive* class label. This observation is in line with the importance of semantic similarity in knowledge-driven graphs for text classification applications beyond domain-agnostic class labels such as sentiment analysis that is based on sentence polarity [184]. Nevertheless, it is worth noting that all four graphs are required for good classification performance due to the advantages of each dimension.

6.8 Chapter summary

This chapter introduces four graphs with different features to accurately represent a text for classification according to domain-agnostic class labels. The proposed QGN augments TensorGCN by incorporating term weighting, nested phrases and regexes, and topic modeling for domain-agnostic text classification. With observable and latent node types along with the corresponding edge weights between the same type of nodes and in relation to text nodes, the proposed model outperforms state-of-the-art graph-based methods for text classification.

Chapter 7

Conclusions and Recommendations

7.1 Conclusions

The developed question classification models in this thesis are able to effectively map assessment questions to domain-agnostic class labels such as learning outcomes, question types, or reasoning capabilities as long as the course designers or subject matter experts provide enough examples that are explicitly aligned to the intended learning outcomes when training the model.

To assist instructors match their assessment questions to learning outcomes, the proposed q-WNTM model in Chapter 4 is compared with previously implemented methods and the proposed s.TF-IDF model presented in Chapter 3. The q-WNTM algorithm augments the performance of question classification beyond the previously described work by addressing issues concerning the selection of stopwords and considering redundant edges in the network of word co-occurrences. Beyond a single dataset and a single taxonomy, the proposed Qu-LDA algorithm in Chapter 5 surpasses such restrictions and generalizes to other dataset and class labels being considered in this thesis.

Hence, with the reliability of the proposed techniques presented in this thesis, the process for calibrating the algorithm could be used in both academic or industrial settings to provide the right set of formative assessment opportunities to students (for enhancing subject knowledge) or employees (for professional development). Once the learning

outcomes of assessment questions are labeled reliably, it is easier to engage learners in deliberate practice to reach those learning outcomes and develop their expertise. Once opportunities for deliberate practice that align to the course learning outcomes are implemented into a course, it facilitates the provision of appropriate feedback based on the performance of students in the various categories of questions.

On the other hand, the developed quad-faceted feature-based graph network in Chapter 6 for document classification poses benefits in areas such as recommendation of suitable job roles based on skillsets acquired by employees, opportunities for skills upgrade when transferrable skills are required, or analyzing different types of scientific articles based on their structures.

7.2 Recommendations for future research

The following are some suggestions for future research:

- Dependencies among topics is a potential area that could provide more in-depth knowledge on why certain words or regexes are grouped together and how the presence of one topic affects another.
- Exploring the suitability of other datasets with class labels of a domain-agnostic nature would be useful for extensive comparison analysis in Chapter 6.
- Having different weights for each graph in QGN can be considered instead of having equal importance for each of the four graphs. This could be naturally determined by a learnable parameter via a weighted fusion layer. In addition, heterogeneity within each graph in QGN could be further explored beyond the use of GCN.
- The work on question classification is part of the research ecosystem for skill sets identification. Beyond question classification, an additional step of evaluation of a learner's knowledge in the summative assessment is important to determine the level of proficiency attained. Based on the existing mastery level of the learners, appropriate measures can be established to guide learners in their journey towards

gaining new relevant skills sets for their education or to climb up the career ladder. Hence, after identifying the learning outcome label of questions, knowledge tracing can be applied to track students' answering skills in the different types of questions using their scores or results. It would subsequently be less challenging to pinpoint where a learner lies in the various levels of cognitive development.

Author's Publications

The author's publications are summarized as follows.

Journal Articles

- **S. Supraja**, Andy W. H. Khong, and S. Tatinati, "Regularized phrase-based topic model for automatic question classification with domain-agnostic class labels," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 29, pp. 3604–3616, 2021.
- **S. Supraja** and Andy W. H. Khong, "Quad-faceted feature-based graph network for domain-agnostic text classification," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, Submitted.

Conference Proceedings

- **S. Supraja**, K. Hartman, S. Tatinati, and Andy W. H. Khong, "Toward the automatic labeling of course questions for ensuring their alignment with learning outcomes," in *Proc. 10th Int. Conf. Educational Data Mining (EDM)*, 2017, pp. 56–63.
- **S. Supraja**, S. Tatinati, K. Hartman, and Andy W. H. Khong, "Automatically linking digital signal processing assessment questions to key engineering learning outcomes," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2018, pp. 6996–7000.

Bibliography

[1] A. Chernov, V. Petukhova, and D. Klakow, "Linguistically motivated question classification," in *Proc. Nordic Conf. Comp. Ling.*, 2015, pp. 51–59.

[2] D. Roth, C. Cumby, X. Li, P. Morie, R. Nagarajan, N. Rizzolo, K. Small, and W. t. Yih, "Question-answering via enhanced understanding of questions," in *Proc. TREC*, 2002, pp. 1–10.

[3] X. Li and D. Roth, "Learning question classifiers," in *Proc. Intl. Conf. Comp. Ling.*, 2002, pp. 1–7.

[4] D. Xu, P. Jansen, J. Martin, Z. Xie, V. Yadav, H. T. Madabushi, O. Tafjord, and P. Clark, "Multi-class hierarchical question classification for multiple choice science exams," in *Proc. Int. Conf. Lang. Resources Eval.*, 2020, pp. 5370–5382.

[5] M. Wasim, M. N. Asim, M. U. G. Khana, and W. Mahmoodb, "Multi-label biomedical question classification for lexical answer type prediction," *J. Biomedical Info.*, vol. 93, p. 103143, 2019.

[6] M. J. Blooma, D. H.-L. Goh, A. Y. K. Chua, and Z. Ling, "Applying question classification to Yahoo! Answers," in *Proc. Int. Conf. Appl. Digital Info. Web Tech. (ICADIWT)*, 2008, pp. 229–234.

[7] A. Godea and R. Nielsen, "Annotating educational questions for student response analysis," in *Proc. Int. Conf. Lang. Resources Eval.*, 2018, pp. 3557–3561.

[8] M. Boratko, H. Padigela, D. Mikkilineni, P. Yuvraj, R. Das, A. McCallum, M. Chang, A. F. Nkoutche, P. Kapanipathi, N. Mattei, R. Musa, K. Talamadupula, and M. Witbrock, "A systematic classification of knowledge, reasoning, and context within the ARC dataset," in *Proc. Mach. Reading Question Answering (MRQA) Workshop ACL 2018*, 2018, pp. 60–70.

[9] M. Boratko, H. Padigela, D. Mikkilineni, P. Yuvraj, R. Das, A. McCallum, M. Chang, A. F. Nkoutche, P. Kapanipathi, N. Mattei, R. Musa, K. Talamadupula, and M. Witbrock, "An interface for annotating science questions," in *Proc. Empirical Methods Nat. Lang. Proc. (EMNLP) 2018 Syst. Demo. Prog.*, 2018, pp. 102–107.

[10] S. Supraja, K. Hartman, S. Tatinati, and A. W. H. Khong, "Toward the automatic labeling of course questions for ensuring their alignment with learning outcomes," in *Proc. 10th Int. Conf. Educational Data Mining (EDM)*, 2017, pp. 56–63.

[11] S. Das, S. K. D. Mandal, and A. Basu, "Identification of cognitive learning complexity of assessment questions using multi-class text classification," *Contemporary Edu. Tech.*, vol. 12, no. 2, pp. 1–14, 2020.

[12] R. Meissner, D. Jenatschke, and A. Thor, "Evaluation of approaches for automatic e-assessment item annotation with levels of Bloom's Taxonomy," in *Proc. Int. Conf. Advances Web-Based Learn. (ICWL)*, 2020, pp. 1–12.

[13] M. Liu, V. Rus, and L. Liu, "Automatic chinese multiple choice question generation using mixed similarity strategy," *IEEE Trans. Learn. Technol.*, vol. 11, no. 2, pp. 193–202, 2018.

[14] M. Liu, Y. Li, W. Xu, and L. Liu, "Automated essay feedback generation and its impact on revision," *IEEE Trans. Learn. Technol.*, vol. 10, no. 4, pp. 502–513, 2017.

[15] T. Parshakova, F. Rameau, A. Serdega, I. S. Kweon, and D.-S. Kim, "Latent question interpretation through variational adaptation," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 27, no. 11, pp. 1713–1724, 2019.

[16] V. A. Silva, I. I. Bittencourt, and J. C. Maldonado, "Automatic question classifiers: A systematic review," *IEEE Trans. Learn. Technol.*, vol. 12, no. 4, pp. 485–502, 2019.

[17] Hardy and Y.-N. Cheah, "Question classification using extreme learning machine on semantic features," *J. ICT Res. Appl.*, vol. 7, pp. 36–58, 2013.

[18] A. Sahu and P. K. Bhowmick, "Feature engineering and ensemble-based approach for improving automatic short-answer grading performance," *IEEE Trans. Learn. Technol.*, vol. 13, no. 1, pp. 77–90, 2020.

[19] H. Chen, L. Xie, C.-C. Leung, X. Lu, B. Ma, and H. Li, "Modeling latent topics and temporal distance for story segmentation of broadcast news," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 25, no. 1, pp. 112–123, 2017.

[20] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learning Res.*, vol. 3, pp. 993–1022, 2002.

[21] C.-H. Lee and J.-T. Chien, "Deep unfolding inference for supervised topic model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2016, pp. 2279–2283.

[22] K. S. Prabhudesai, B. O. Mainsah, L. M. Collins, and C. S. Throckmorton, "Augmented latent Dirichlet allocation (LDA) topic model with Gaussian mixture topics," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2018, pp. 2451–2455.

[23] J.-T. Chien, "Bayesian nonparametric learning for hierarchical and sparse topics," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 26, no. 2, pp. 422–435, 2018.

[24] R. Zhao and K. Mao, "Topic-aware deep compositional models for sentence classification," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 25, no. 2, pp. 248–260, 2017.

[25] K. Chen, R. Wang, M. Utiyama, E. Sumita, and T. Zhao, "Neural machine translation with sentence-level topic context," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 27, no. 12, pp. 1970–1984, 2019.

[26] D. Li, J. Zhang, and P. Li, "Representation learning for question classification via topic sparse autoencoder and entity embedding," in *Proc. Int. Conf. Big Data*, 2018, pp. 126–133.

[27] S. K. Maity, A. Kharb, and A. Mukherjee, "Language use matters: Analysis of the linguistic structure of question texts can characterize answerability in Quora," in *Proc. Int. Conf. Web Social Media (ICWSM, AAAI)*, 2017, pp. 612–615.

[28] P. Chahuara, T. Lampert, and P. Gancarski, "Retrieving and ranking similar questions from question-answer archives using topic modelling and topic distribution regression," in *Proc. Research Adv. Tech. Digital Libraries*, 2016, pp. 41–53.

[29] C. M. Intisar, Y. Watanobe, M. Poudel, and S. Bhalla, "Classification of programming problems based on topic modeling," in *Proc. Int. Conf. Info. Edu. Tech.*, 2019, pp. 275–283.

[30] Y. Zhou, L. Liao, Y. Gao, R. Wang, and H. Huang, "TopicBERT: A topic-enhanced neural language model fine-tuned for sentiment classification," *IEEE Trans. Neural Networks Learn. Syst.*, vol. Early Access, pp. 1–14, 2021.

[31] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning–based text classification: A comprehensive review," *ACM Comp. Surveys*, vol. 54, no. 3, pp. 62:1–62:40, 2021.

[32] B. Sun, Y. Zhu, Y. Xiao, R. Xiao, and Y. Wei, "Automatic question tagging with deep neural networks," *IEEE Trans. Learn. Technol.*, vol. 12, no. 1, pp. 29–43, 2018.

[33] W. Xia, W. Zhu, B. Liao, M. Chen, L. Ca, and L. Huang, "Novel architecture for long short-term memory used in question classification," *Neurocomputing*, vol. 299, pp. 20–31, 2018.

[34] H. M. Wallach, D. Mimno, and A. McCallum, "Rethinking LDA: Why priors matter," in *Proc. Neural Info. Proc. Syst. (NIPS)*, 2009, pp. 1973–1981.

[35] A. T. Wilson and P. A. Chew, "Term weighting schemes for latent Dirichlet allocation," in *Proc. Human Lang. Tech.: Annual Conf. North American Chap. ACL (NAACL)*, 2010, pp. 465–473.

[36] S. Momtazi and I. Gurevych, "Unsupervised latent Dirichlet allocation for supervised question classification," *Info. Proc. Mgmt.*, vol. 54, no. 3, pp. 380–393, 2018.

[37] SSG | Critical Core Skills. Available at https://www.skillsfuture.gov.sg/skills-framework/criticalcoreskills, Jul. 3, 2021.

[38] SSG | Skills Framework. Available at https://www.skillsfuture.gov.sg/skills-framework#whicharethesectors, Jul. 3, 2021.

[39] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. AAAI Conf. Artificial Intell.*, 2019, pp. 7370–7377.

[40] X. Liu, X. You, X. Zhang, J. Wu, and P. Lv, "Tensor graph convolutional networks for text classification," in *Proc. AAAI Conf. Artificial Intell.*, 2020, pp. 8409–8416.

[41] S. S. Haris and N. Omar, "Bloom's Taxonomy question categorization using rules and n-gram approach," *J. Theoretical Appl. Inform. Technol.*, vol. 76, pp. 401–407, 2015.

[42] K. Jayakodi, M. Bandara, and I. Perera, "An automatic classifier for exam questions in engineering: A process for Bloom's Taxonomy," in *Proc. IEEE Int. Conf. Teaching, Assessment, Learn. Eng. (TALE)*, 2015, pp. 12–17.

[43] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, pp. 513–523, 1988.

[44] A. A. Yahya, A. Osman, A. Taleb, and A. A. Alattab, "Analyzing the cognitive level of classroom questions using machine learning techniques," in *Proc. 9th Int. Conf. Cognitive Sci.*, 2013, pp. 587–595.

[45] M. Lan, C.-L. Tan, H.-B. Low, and S.-Y. Sung, "A comprehensive comparative study on term weighting schemes for text categorization with support vector machines," in *Proc. Intl. World Wide Web Conf.*, 2005, pp. 1032–1033.

[46] D. Wang and H. Zhang, "Inverse-category-frequency based supervised term weighting schemes for text categorization," *J. Info. Sci. Engg.*, vol. 29, no. 2, pp. 209–225, 2013.

[47] X. Quan, W. Liu, and B. Qiu, "Term weighting schemes for question categorization," *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 33, no. 5, pp. 1009–1021, 2011.

[48] K. Chen, Z. Zhang, J. Long, and H. Zhang, "Turning from TF-IDF to TF-IGM for term weighting in text classification," *Expert Systems Appl.*, vol. 66, pp. 245–260, 2016.

[49] T. Dogan and A. K. Uysal, "Improved inverse gravity moment term weighting for text classification," *Expert Systems Appl.*, vol. 130, pp. 45–59, 2019.

[50] L. Chen, L. Jiang, and C. Li, "Modified DFS-based term weighting scheme for text classification," *Expert Systems Appl.*, vol. 168, p. 114438, 2021.

[51] J. Chang, J. Graber, S. Gerrish, C. Wang, and D. Blei, "Reading tea leaves: how humans interpret topic models," in *Proc. Intl. Conf. Neural. Info. Proc. Sys.*, 2009, pp. 288–296.

[52] D. Blei and J. Lafferty, "Correlated topic models," in *Proc. Intl. Conf. Neural. Info. Proc. Syst.*, 2005, pp. 147–154.

[53] Y. Zhuang, H. Gao, F. Wu, S. Tang, Y. Zhang, and Z. Zhang, "Probabilistic word selection via topic modeling," *IEEE Trans. Knowledge Data Engg.*, vol. 27, no. 6, pp. 1643–1655, 2015.

[54] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proc. Nat. Acad. Sci.*, vol. 101, pp. 5228–5235, 2004.

[55] G. Casella and E. George, "Explaining the Gibbs sampler," *The American Stat.*, vol. 46, pp. 167–174, 1992.

[56] O. Papaspiliopoulos and G. Roberts, "Stability of the gibbs sampler for bayesian hierarchical models," *The Annals Stat.*, vol. 36, pp. 95–117, 2008.

[57] M. Xu, Y. Cai, H. Wu, C. Wang, and N. Li, "Intensity of relationship between words: using word triangles in topic discovery for short texts," in *Proc. Web Big Data*, 2017, pp. 642–649.

[58] L. Jiang, M. X. H. Lu, and C. Wang, "Biterm pseudo document topic model for short text," in *Proc. Intl. Conf. Tools. Artificial. Intell.*, 2016, pp. 865–872.

[59] Y. Zuo, J. Zhao, and K. Xu, "Word network topic model: A simple but general solution for short and imbalanced texts," *Knowledge, Inform. Syst.*, vol. 48, pp. 379–398, 2016.

[60] S. Syed and M. Spruit, "Selecting priors for latent Dirichlet allocation," in *Proc. IEEE Int. Conf. Semantic. Comp.*, 2018, pp. 194–202.

[61] N. Wicker, J. Muller, R. Kalathur, and O. Poch, "A maximum likelihood approximation method for Dirichlet's parameter estimation," *Comp. Stat. Data Analysis*, vol. 52, pp. 1315–1322, 2008.

[62] M. Giordan and R. Wehrens, "A comparison of computational approaches for maximum likelihood estimation of the Dirichlet parameters on high-dimensional data," *SORT*, vol. 39, no. 1, pp. 109–126, 2015.

[63] Z. Yu, T. R. Johnson, and R. Kavuluru, "Phrase based topic modeling for semantic information processing in biomedicine," in *Proc. Int. Conf. Mach. Learn. Appl.*, 2013, pp. 440–445.

[64] A. ElKishky, Y. Song, C. Wang, C. R. Voss, and J. Han, "Scalable topical phrase mining from text corpora," *VLDB Endowment (in Proc. Int. Conf. Very Large Databases)*, vol. 8, no. 3, pp. 305–316, 2014.

[65] G. M. d. B. Wenniger and K. Sima'an, "Labeling hierarchical phrase-based models without linguistic resources," *Mach. Translat*, vol. 29, pp. 225–265, 2015.

[66] O. Vechtomova, "Noun phrases in interactive query expansion and document ranking," *Info. Retrieval*, vol. 9, no. 4, pp. 399–420, 2006.

[67] W. Zhang, S. Liu, C. Yu, C. Sun, F. Liu, and W. Meng, "Recognition and classification of noun phrases in queries for effective retrieval," in *Proc. Conf. Info. Knowledge Mgmt.*, 2007, pp. 711–720.

[68] D. A. Evans and C. Zhai, "Noun-phrase analysis in unrestricted text for information retrieval," in *Proc. Annual Meeting Assoc. Comp. Ling.*, 1996, pp. 17–24.

[69] J. Shang, J. Liu, M. Jiang, X. Ren, C. R. Voss, and J. Han, "Automated phrase mining from massive text corpora," *IEEE Trans. Knowledge Data Engg.*, vol. 30, no. 10, pp. 1825–1837, 2018.

[70] B. Li, W. Xu, Y. Tian, and J. Chen, "A phrase topic model for large-scale corpus," in *Proc. Int. Conf. Cloud Comp. Big Data Analytics*, 2019, pp. 634–639.

[71] F. Rodrigues, M. Lourenco, B. Ribeiro, and F. C. Pereira, "Learning supervised topic models for classification and regression from crowds," *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 39, no. 12, pp. 2409–2422, 2017.

[72] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, pp. 489–501, 2006.

[73] M. A. A. Albadra and S. Tiuna, "Extreme learning machine: a review," *Intl. J. App. Eng. Res.*, vol. 12, pp. 4610–4623, 2017.

[74] A. Akusok, K.-M. Bjork, Y. Miche, and A. Lendasse, "High performance extreme learning machines: a complete toolbox for big data applications," *IEEE Access*, vol. 3, pp. 1011–1025, 2015.

[75] G.-B. Huang, "What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle," *Cogn. Comput.*, vol. 7, pp. 263–278, 2015.

[76] J. Choworski, J. Wang, and J. Zurada, "Review and performance comparison of SVM- and ELM-based classifiers," *Neurocomputing*, vol. 128, pp. 507–516, 2014.

[77] Y. Liu and Y. Zheng, "One-against-all multi-class SVM classification using reliability measures," in *Proc. Intl. Conf. Neural. Net.*, 2005, pp. 849–854.

[78] M. Kandemir, T. Kekec, and R. Yeniterzi, "Supervising topic models with Gaussian processes," *Pattern Recognition*, vol. 77, pp. 226–236, 2018.

[79] B. Fröhlich, E. Rodner, M. Kemmler, and J. Denzler, "Large-scale Gaussian process multi-class classification for semantic segmentation and facade recognition," *Mach. Vision Appl.*, vol. 24, no. 5, pp. 1043—1053, 2013.

[80] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, 2019.

[81] Y. Zhang and W. Xiao, "Keyphrase generation based on deep seq2seq model," *IEEE Access: Special Section Big Data Discovery*, vol. 6, pp. 46 047–46 057, 2018.

[82] Z. Liang, J. Du, and C. Li, "Abstractive social media text summarization using selective reinforced seq2seq attention model," *Neurocomputing*, vol. 410, pp. 432–440, 2020.

[83] M. Ma, L. Huang, B. Xiang, and B. Zhou, "Group sparse CNNs for question classification with answer sets," in *Proc. Assoc. Comp. Ling. (ACL)*, 2017, pp. 335–340.

[84] T. Lei, Z. Shi, D. Liu, L. Yang, and F. Zhu, "A novel CNN-based method for question classification in intelligent question answering," in *Proc. Intl. Conf. Algorithms, Comp. Artificial Intelligence (ACAI)*, 2018, pp. 1–6.

[85] H. Chen, Q. Ma, L. Yu, Z. Lin, and J. Yan, "Corpus-aware graph aggregation network for sequence labeling," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 29, pp. 2048–2057, 2021.

[86] K. Zhao, L. Huang, R. Song, Q. Shen, and H. Xu, "A sequential graph neural network for short text classification," *Algorithms*, vol. 14, no. 352, pp. 1–14, 2021.

[87] Y. Dai, L. Shou, M. Gong, X. Xia, Z. Kang, Z. Xu, and D. Jiang, "Graph fusion network for text classification," *Knowledge-Based Syst.*, vol. 236, p. 107659, 2022.

[88] H. Peng, J. Li, S. Wang, L. Wang, Q. Gong, R. Yang, B. Li, P. S. Yu, and L. He, "Hierarchical taxonomy-aware and attentional graph capsule RCNNs for large-scale multi-label text classification," *IEEE Trans. Knowledge Data Engg.*, vol. 33, no. 6, pp. 2505–2519, 2021.

[89] L. Huang, D. Ma, S. Li, X. Zhang, and H. Wang, "Text level graph neural network for text classification," in *Proc. Conf. Empirical Methods Nat. Lang. Proc.*, 2019, pp. 3444–3450.

[90] S. Vashishth, M. Bhandari, P. Yadav, P. Rai, C. Bhattacharyya, and P. Talukdar, "Incorporating syntactic and semantic information in word embeddings using graph convolutional networks," in *Proc. Assoc. Comp. Ling. (ACL)*, 2019, pp. 3308–3318.

[91] Y. Lin, Y. Meng, X. Sun, Q. Han, K. Kuang, J. Li, and F. Wu, "BertGCN: Transductive text classification by combining GCN and BERT," in *Proc. Assoc. Comp. Ling. (ACL-IJCNLP 2021)*, 2021, pp. 1456–1462.

[92] F. Feng, X. He, H. Zhang, and T.-S. Chua, "Cross-GCN: Enhancing graph convolutional network with k-order feature interactions," *IEEE Trans. Knowledge Data Engg.*, vol. Early Access, pp. 1–11, 2021.

[93] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proc. Human Lang. Tech.: Annual Conf. North American Chap.(NAACL-HLT)*, 2019, pp. 4171–4186.

[94] M. Mohd, R. Jan, and M. Shah, "Text document summarization using word embedding," *Expert Systems Appl.*, vol. 143, p. 112958, 2020.

[95] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Empirical Methods Nat. Lang. Proc. (EMNLP)*, 2014, pp. 1532–1543.

[96] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, "Unsupervised neural machine translation," in *Proc. Intl. Conf. Learn. Rep. (ICLR)*, 2018, pp. 1–12.

[97] G. Lample, A. Conneau, L. Denoyer, and M. A. Ranzato, "Unsupervised machine translation using monolingual corpora only," in *Proc. Intl. Conf. Learn. Rep. (ICLR)*, 2018, pp. 1–14.

[98] G. Lample, M. Ott, A. Conneau, L. Denoyer, and M. A. Ranzato, "Phrase-based and neural unsupervised machine translation," in *Proc. Conf. Empirical Methods Nat. Lang. Proc. (EMNLP)*, 2018, pp. 5093–5049.

[99] P. Ramachandran, P. J. Liu, and Q. V. Le, "Unsupervised pretraining for sequence to sequence learning," in *Proc. Conf. Empirical Methods Nat. Lang. Proc. (EMNLP)*, 2017, pp. 383–391.

[100] A. Conneauand and G. Lample, "Cross-lingual language model pre-training," in *Proc. Neural Info. Proc. Syst. (NIPS)*, 2019, pp. 7059–7069.

[101] K. Song, X. Tan, T. Qin, J. Lu, and T. Y. Liu, "Mass: masked sequence to sequence pre-training for language generation," in *Proc. Intl. Conf. Mach. Learn. (ICML)*, 2019, pp. 1–11.

[102] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proc. Assoc. Comp. Ling.*, 2002, pp. 311–318.

[103] P. Black and D. William, "Assessment and classroom learning," *Assessment Edu. Principles, Policy, Prac.*, vol. 5, pp. 7–74, 1998.

[104] K. A. Ericsson, R. T. Krampe, and C. T.-Romer, "The role of deliberate practice in the acquisition of expert performance," *Psych. Review*, vol. 100, pp. 363–406, 1993.

[105] D. Boud and N. Falchikov, "Aligning assessment with long-term learning," *Assessment, Evaluation Higher Edu.*, vol. 31, pp. 399–413, 2006.

[106] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach - 3rd Edition.* McGraw-Hill Companies, 2005.

[107] E. C. Ifeachor and B. W. Jervis, *Digital Signal Processing: A Practical Approach - 2nd Edition.* Prentice Hall, 2001.

[108] L. Chaparro, *Signals and Systems using MATLAB - 2nd Edition.* Academic Press, 2004.

[109] A. Kenimer and ABET, "Proposed revised to the criteria for accrediting engineering programs general criteria introduction, criterion 3, student outcomes," 2017.

[110] R. M. Felder and R. Brent, "Designing and teaching courses to satisfy the ABET engineering criteria," *J. Eng. Edu.*, vol. 92, pp. 7–25, 2003.

[111] R. S.-Nom, "Interactive teaching and assessment using recycled SP concepts," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2013, pp. 4354–4358.

[112] C. H. G. Wright, T. B. Welch, and M. G. Morrow, "Signal processing concepts help teach optical engineering," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2016, pp. 6275–6279.

[113] M. F. Bugallo and A. M. Kelly, "An outreach after-school program to introduce high-school students to electrical engineering," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2015, pp. 5540–5544.

[114] M. B-Sacre, M. F. Cox, M. Borrego, K. Beddoes, and J. Zhud, "Changing engineering education: Views of U.S. faculty, chairs, and deans," *J. Eng. Edu.*, vol. 103, pp. 193–213, 2014.

[115] D. Krathwohl, "A revision of Bloom's Taxonomy: An overview," *Theory into Practice*, vol. 41, pp. 212–218, 2002.

[116] J. Biggs, "Enhancing teaching through constructive alignment," *Higher Edu.*, vol. 32, pp. 347–364, 1996.

[117] R. Siddiqi, C. J. Harrison, and R. Siddiqi, "Improving teaching and learning through automated short-answer marking," *IEEE Trans. Learn. Technol.*, vol. 3, no. 3, pp. 237–249, 2010.

[118] M. M. Brut, F. Sedes, and S. D. Dumitrescu, "A semantic-oriented approach for organizing and developing annotation for e-learning," *IEEE Trans. Learn. Technol.*, vol. 4, no. 3, pp. 239–248, 2011.

[119] A. Khoo, Y. Marom, and D. Albrecht, "Experiments with sentence classification," in *Proc. Australasian Lang. Tech. Workshop (ALTW)*, 2006, pp. 18–25.

[120] S. Supraja, S. Tatinati, K. Hartman, and A. W. H. Khong, "Automatically linking digital signal processing assessment questions to key engineering learning outcomes," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2018, pp. 6996–7000.

[121] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Nat. Lang. Proc.*, 2014, pp. 1746–1751.

[122] A. Sangodiah, M. Muniandy, and L. E. Heng, "Question classification using statistical approach: A complete review," *J. Theoretical Appl. Info. Tech.*, vol. 71, no. 3, pp. 386–395, 2015.

[123] J. Pomerantz, "A linguistic analysis of question taxonomies," *J. American Soc. Info. Sci. Tech.*, vol. 56, pp. 715–728, 2005.

[124] A. Mohasseb, M. Bader-El-Den, and M. Cocea, "Classification of factoid questions intent using grammatical features," *ICT Express*, vol. 4, pp. 239–242, 2018.

[125] A. Kapelner, J. Soterwood, S. Nessaaiver, and S. Adlof, "Predicting contextual informativeness for vocabulary learning," *IEEE Trans. Learn. Technol.*, vol. 11, no. 1, pp. 13–26, 2018.

[126] M. Pota, A. Fuggi, M. Esposito, and G. D. Pietro, "Extracting compact sets of features for question classification in cognitive systems: A comparative study," in *Proc. Int. Conf. P2P Parallel Grid Cloud Internet Comp.*, 2015, pp. 551–556.

[127] C. B. Jacinoa and S. DeDeoa, "Opacity, obscurity, and the geometry of question-asking," *Cognition*, vol. 196, pp. 1–13, 2020.

[128] L. Karttunen, "Syntax and semantics of questions," *Ling. Philosophy*, vol. 1, pp. 3–44, 1977.

[129] H. Duan, Y. Cao, C. Y. Lin, and Y. Yu, "Searching questions by identifying question topic and question focus," in *Proc. Conf. Human Lang. Tech. (Assoc. Comp. Ling.)*, 2008, pp. 156–164.

[130] C. Fox, "A stop list for general text," *ACM-SIGIR Forum*, vol. 24, pp. 19–35, 1989.

[131] M. Gerlach, H. Shi, and L. A. N. Amaral, "A universal information theoretic approach to the identification of stopwords," *Nature Mach. Intell.*, vol. 1, pp. 606–612, 2019.

[132] D. Khashabi, T. Khot, A. Sabharwal, and D. Roth, "Learning what is essential in questions," in *Proc. Conf. Comp. Nat. Lang. Learning (CoNLL)*, 2017, pp. 80–89.

[133] J. Ni, C. Zhu, W. Chen, and J. McAuley, "Learning to attend on essential terms: An enhanced retriever-reader model for open-domain question answering," in *Proc. Int. Conf. North American Chap. Assoc. Comp. Ling.: Human Lang. Tech. (NAACL-HLT)*, 2019, pp. 335–344.

[134] A. K. Santra and C. J. Christy, "Genetic algorithm and confusion matrix for document clustering," *IJCSI Int. Journ. Comp. Sci. Iss.*, vol. 9, pp. 322–328, 2012.

[135] Q. Liu, S. Zhang, Q. Wang, and W. Chen, "Mining online discussion data for understanding teachers' reflective thinking," *IEEE Trans. Learn. Technol.*, vol. 11, no. 2, pp. 243–254, 2018.

[136] L. Zhang and D. Zhang, "SVM and ELM: who wins? Object recognition with deep convolutional features from ImageNet," pp. 1–7, 2015.

[137] G. Zhou, Z. Xie, T. He, J. Zhao, and X. T. Hu, "Learning the multilingual translation representations for question retrieval in community question answering via non-negative matrix factorization," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 24, no. 7, pp. 1305–1314, 2016.

[138] P. Le-Hong, X.-H. Phan, and T.-D. Nguyen, "Using dependency analysis to improve question classification," *Knowledge Syst. Engg. Adv. Intell. Syst. Comp. (Springer)*, vol. 326, pp. 653–665, 2015.

[139] D. Xiong, M. Zhang, and X. Wang, "Topic-based coherence modeling for statistical machine translation," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 23, no. 3, pp. 483–493, 2015.

[140] G. Zhou, L. Cai, J. Zhao, and K. Liu, "Phrase-based translation model for question retrieval in community question answer archives," in *Proc. Annual Meeting Assoc. Comp. Ling.*, 2011, pp. 653–662.

[141] B. Li, X. Yang, R. Zhou, B. Wang, C. Liu, and Y. Zhang, "An efficient method for high quality and cohesive topical phrase mining," *IEEE Trans. Knowledge Data Engg.*, vol. 31, no. 1, pp. 120–137, 2019.

[142] H. T. Madabushi and M. Lee, "High accuracy rule-based question classification using question syntax and semantics," in *Proc. Int. Conf. Comp. Ling. (COLING)*, 2016, pp. 1220–1230.

[143] F. Bu, X. Zhu, Y. Hao, and X. Zhu, "Function-based question classification for general QA," in *Proc. Conf. Empirical Methods Nat. Lang. Proc.*, 2010, pp. 1119–1128.

[144] Z. Huang, M. Thint, and Z. Qin, "Question classification using head words and their hypernyms," in *Proc. Conf. Empirical Methods Nat. Lang. Proc.*, 2008, pp. 927–936.

[145] S. Beyer, C. Macho1, M. D. Penta, and M. Pinzger, "What kind of questions do developers ask on Stack Overflow? A comparison of automated approaches to classify posts into question categories," *Empirical Software Engg.*, vol. 25, pp. 2258–2301, 2020.

[146] A. Handler, M. J. Denny, H. Wallach, and B. O. Connor, "Bag of what? Simple noun phrase extraction for text analysis," in *Proc. Workshop Nat. Lang. Proc. Comp. Soc. Sci. (Conf. Empirical Methods Nat. Lang. Proc.)*, 2016, pp. 114–124.

[147] W.-N. Zhang, Z.-Y. Ming, Y. Zhang, T. Liu, and T.-S. Chua, "Capturing the semantics of key phrases using multiple languages for question retrieval," *IEEE Trans. Knowledge Data Engg.*, vol. 28, no. 4, pp. 888–900, 2016.

[148] K. Frantzi, S. Ananiadou, and H. Mima, "Automatic recognition of multi-word terms: The C-value/NC-value method," *Int. J. Digital Libraries*, vol. 3, no. 2, pp. 115–130, 2000.

[149] A. Mohasseb, M. B.-E-Den, and M. Cocea, "Question categorization and classification using grammar based approach," *Info. Proc. Mgmt.*, vol. 54, pp. 1228–1243, 2018.

[150] C. Wang, M. Danilevsky, N. Desai, Y. Zhang, P. Nguyen, T. Taula, and J. Han, "Phrase mining framework for recursive construction of a topical hierarchy," in *Proc. Conf. Knowledge Discovery Data Mining (KDD)*, 2013, pp. 437–445.

[151] S. Gerdjikov and K. U. Schulz, "Corpus analysis without prior linguistic knowledge – unsupervised mining of phrases and subphrase structure," pp. 1–56, 2016.

[152] D. Shen and M. Lapata, "Using semantic roles to improve question answering," in *Proc. Joint Conf. Empirical Methods Nat. Lang. Proc. Comp. Nat. Lang. Learn.*, 2007, pp. 12–21.

[153] J. E. Boland, M. K. Tanenhaus, S. M. Garnsey, and G. N. Carlson, "Verb argument structure in parsing and interpretation: Evidence from wh-questions," *J. Memory Lang.*, vol. 34, pp. 774–806, 1995.

[154] J. Ginzburg, "Questions: Logic and interactions," *Handbook Logic Lang.*, pp. 1–15, 2010.

[155] P. Ozdzynski and D. Zakrzewska, "A search of significant phrases for building topic models in text documents," *Info. Syst. Mgmt.*, vol. 5, no. 2, pp. 205–214, 2016.

[156] textacy · pypi. Available at https://pypi.org/project/textacy/, Aug. 30, 2020.

[157] D. Deng, L. Jing, J. Yu, S. Sun, and M. K. Ng, "Sentiment lexicon construction with hierarchical supervision topic model," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 27, no. 4, pp. 704–718, 2019.

[158] T. Andre, "Does answering higher-level questions while reading facilitate productive learning?" *Review Edu. Research*, vol. 49, pp. 280–318, 1979.

[159] S. G. Bull, "The role of questions in maintaining attention to textual material," *Review Edu. Research*, vol. 43, pp. 83–88, 1973.

[160] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, and R. Faulkner, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, pp. 1–14, 2018.

[161] K. M. Hammouda and M. S. Kamel, "Efficient phrase-based document indexing for web document clustering," *IEEE Trans. Knowledge Data Engg.*, vol. 16, no. 10, pp. 1279–1296, 2004.

[162] Y. Wang, S. Wang, Q. Yao, and D. Dou, "Hierarchical heterogeneous graph representation learning for short text classification," in *Proc. Empirical Methods Nat. Lang. Proc. (EMNLP)*, 2021, pp. 3091–3101.

[163] C. Jiang, Y. Zhao, S. Chu, L. Shen, and K. Tu, "Cold-start and interpretability: Turning regular expressions into trainable recurrent neural networks," in *Proc. Conf. Empirical Methods Nat. Lang. Proc.*, 2020, pp. 3193–3207.

[164] J. Liu, R. Bai, Z. Lu, P. Ge, U. Aickelin, and D. Liu, "Data-driven regular expressions evolution for medical text classification using genetic programming," in *Proc. IEEE Congress Evolutionary Comp. (CEC)*, 2020, pp. 1–8.

[165] M. Cui, R. Bai, Z. Lu, X. Li, U. Aickelin, and P. Ge, "Regular expression based medical text classification using constructive heuristic approach," *IEEE Access*, vol. 7, pp. 147 892–147 904, 2019.

[166] Y. Shi, Y. Xiao, P. Quan, M. Lei, and L. Niu, "Document-level relation extraction via graph transformer networks and temporal convolutional networks," *Pattern Recognition Letters*, vol. 149, pp. 150–156, 2021.

[167] T. Yang, L. Hu, C. Shi, H. Ji, X. Li, and L. Nie, "HGAT: Heterogeneous graph attention networks for semi-supervised short text classification," *ACM Trans. Info. Syst.*, vol. 39, no. 3, pp. 32:1–32:29, 2021.

[168] R. Ragesh, S. Sellamanickam, A. Iyer, R. Bairi, and V. Lingam, "HeteGCN: Heterogeneous graph convolutional networks for text classification," *arXiv preprint arXiv:2008.12842*, pp. 1–10, 2020.

[169] H. Chim and X. Deng, "Efficient phrase-based document similarity for clustering," *IEEE Trans. Knowledge Data Engg.*, vol. 20, no. 9, pp. 1217–1229, 2008.

[170] P. Wei, J. Zhao, and W. Mao, "A graph-to-sequence learning framework for summarizing opinionated texts," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 29, pp. 1650–1660, 2021.

[171] S. Koltcov, V. Ignatenko, and O. Koltsova, "Estimating topic modeling performance with Sharma–Mittal entropy," *Entropy*, vol. 21, no. 660, pp. 1–29, 2019.

[172] S. Supraja, A. W. H. Khong, and S. Tatinati, "Regularized phrase-based topic model for automatic question classification with domain-agnostic class labels," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 29, pp. 3604–3616, 2021.

[173] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proc. Intl. World Wide Web Conf.*, 2013, pp. 1445–1455.

[174] X. Bai, P. Liu, and Y. Zhang, "Investigating typed syntactic dependencies for targeted sentiment classification using graph attention neural network," *IEEE/ACM Trans. Audio Speech Lang. Proc.*, vol. 29, pp. 503–514, 2021.

[175] Y. Arase and J. Tsujii, "Transfer fine-tuning: A BERT case study," in *Proc. Empirical Methods Nat. Lang. Proc. (EMNLP)*, 2019, pp. 5393–5404.

[176] Y. Wang, J. Liu, Y. Huang, and X. Feng, "Using hashtag graph-based topic model to connect semantically-related words without co-occurrence in microblogs," *IEEE Trans. Knowledge Data Engg.*, vol. 28, no. 7, pp. 1919–1933, 2016.

[177] P. Kontkanen and P. Myllymäki, "MDL histogram density estimation," in *Proc. Int. Conf. Artificial Intell. Statistics (PMLR)*, 2007, pp. 2:219–226.

[178] Y. Wu, S. Zhao, and W. Li, "Phrase2Vec: Phrase embedding based on parsing," *Info. Sci.*, vol. 517, pp. 100–127, 2020.

[179] S. Teufel, "Argumentative zoning: information extraction from scientific text," *PhD thesis, Sch. Informatics, Univ. Edinburgh*, 1999.

[180] S. Teufel and M. Moens, "Summarizing scientific articles: experiments with relevance and rhetorical status," *Comp. Ling.*, vol. 28, no. 4, pp. 409–445, 2002.

[181] S. Teufel, "Statistical models for text classification: Applications and analysis," *PhD thesis, Univ. California, Irving*, 2013.

[182] X. Li, J. Saude, P. Reddy, and M. Veloso, "Classifying and understanding financial data using graph neural network," in *Proc. AAAI Conf. Artificial Intell. Workshop Knowledge Discovery Unstructured Data Fin. Services*, 2020, pp. 1–8.

[183] A. P. Tuan, B. Tran, T. H. Nguyen, L. N. Van, and K. Than, "Bag of biterms modeling for short texts," *Knowledge Info. Syst.*, vol. 62, pp. 4055—-4090, 2020.

[184] N. Shanavas, H. Wang, Z. Lin, and G. Hawe, "Knowledge-driven graph similarity for text classification," *Int. J. Machine Learn. Cybernetics*, vol. 12, pp. 1067—1081, 2021.