Enhancing the Reasoning Capabilities of Natural Language Inference Models with Attention Mechanisms and External Knowledge

GAJBHIYE, AMIT

Enhancing the Reasoning Capabilities of Natural Language Inference Models with Attention Mechanisms and External Knowledge

Amit Gajbhiye

A Thesis presented for the degree of

Doctor of Philosophy

Department of Computer Science
Durham University
United Kingdom
August 2020

Abstract

Natural Language Inference (NLI) is fundamental to natural language understanding. The task summarises the natural language understanding capabilities within a simple formulation of determining whether a natural language hypothesis can be inferred from a given natural language premise. NLI requires an inference system to address the full complexity of linguistic as well as real-world commonsense knowledge and, hence, the inferencing and reasoning capabilities of an NLI system are utilised in other complex language applications such as summarisation and machine comprehension. Consequently, NLI has received significant recent attention from both academia and industry. Despite extensive research, contemporary neural NLI models face challenges arising from the sole reliance on training data to comprehend all the linguistic and real-world commonsense knowledge. Further, different attention mechanisms, crucial to the success of neural NLI models, present the prospects of better utilisation when employed in combination. In addition, the NLI research field lacks a coherent set of guidelines for the application of one of the most crucial regularisation hyper-parameters in the RNN-based NLI models – dropout.

In this thesis, we present neural models capable of leveraging the attention mechanisms and the models that utilise external knowledge to reason about inference. First, a combined attention model to leverage different attention mechanisms is proposed. Experimentation demonstrates that the proposed model is capable of better modelling the semantics of long and complex sentences. Second, to address the limitation of the sole reliance on the training data, two novel neural frameworks utilising real-world commonsense and domain-specific external knowledge are introduced. Employing the rule-based external knowledge retrieval from the knowledge graphs, the first model takes advantage of the convolutional encoders and factorised bilinear pooling to augment the reasoning capabilities of the state-of-the-art NLI models. Utilising the significant advances in the research of contextual word representations, the second model, addresses the existing crucial challenges of external knowledge retrieval, learning the encoding of the retrieved knowledge and the fusion of the learned encodings to the NLI representations, in unique ways. Experimentation demonstrates the efficacy and superiority of the proposed models over previous state-of-the-art approaches. Third, for the limitation on dropout investigations, formulated on exhaustive evaluation, analysis and validation on the proposed RNN-based NLI models, a coherent set of guidelines is introduced.

Declaration

The work in this thesis is based on research carried out at the Department of Computer Science, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

I wish to express my deepest gratitude to my supervisors, Dr Steven Bradley and Dr Noura Al Moubayed, for the opportunity to pursue my PhD at Durham University. Steven, I am deeply indebted for your sensible advice and selfless patience towards my growth as a researcher. Noura, thank you for your kind support, technical insights and for fostering a productive research environment in the research group.

I would like to express my sincere gratitude to the National Overseas Scholarships, Ministry of Social Justice and Empowerment, Government of India, for supporting my PhD studies.

The research in this thesis has been possible because of my incredible fellow labmates. I am very grateful to them for the technical discussions which helped me build on my research ideas.

Finally, I owe my deepest gratitude to my beloved parents, Shrimati Sushila and late Shri Sukhdev Gajbhiye, for their boundless love, blessings and countless sacrifices. I wouldn't be where I am today without their blessings. Their dedication and love have always been selfless, and I do not think I can ever repay for their sacrifices.

Contents

List of Figures

List of Tables

Dedication

To my lovely parents

CHAPTER 1

Introduction

"... inferential ability is not only central manifestation of semantic competence but it is in fact centrally constitutive of it, it shouldn't be a surprise that we regard inferencing tasks as the best way of testing an NLP system's semantic capacity."

— Cooper et al., The FRaCaS Consortium, Ch. 3, 1996

Language is fundamental to communication. Since the advent of modern computers, one of the crucial research questions has been - Can computers learn, understand, and produce human language? However, with some initial success of computer programs that can understand simple natural language instructions to solve algebra word problems [6] or undertake superficial dialogues [7], it turned out that making computers understand human language is a difficult task because of the inherent ambiguities in natural language, the variability of semantic expressions and the context-dependent interpretations of natural languages [8]. The research and development of theory-motivated automatic computational techniques that explores how computers can be made to learn, understand, and produce human languages is called Natural Language Processing (NLP) [9]. NLP is an umbrella term encompassing various theories and technologies for analysing natural language in written

or spoken form. The focus of this research work is in the written form of natural language i.e. text sequences, to reason and understand the inference between the input sequences.

In text-based NLP, the text can be analysed at different levels to understand and extract the meaning [8]. At the lexical level, the NLP system interprets the meaning of individual words. At the syntactic level, the goal is to identify the structure of the input sentences and ascertain the validity according to the grammatical rules of the language. The semantic level is concerned with interpreting or understanding the (literal) meanings of the sentence by focussing on the interactions among the word-level meanings in the sentence. The discourse level works at the inter-sentence level and concentrates on the properties of the text as a whole that convey meaning by making connections between the interrelated sentences. Finally, at the highest level, the text can be analysed at the pragmatic level where the goal is to explain the text over and above the contents of the text. The analysis of the text at this level requires significant world knowledge, including the understanding of intentions, plans and goals [8]. NLP systems involved in the pragmatic analysis of text may utilise real-world knowledge sources such as Knowledge Graphs (KGs) to enrich the context of the analysed text.

The goal of the NLP is to achieve true natural language understanding by means of analysing the linguistic data at the aforementioned levels [8]. Over the last decade, the availability of a large amount of linguistic data, development of sophisticated machine learning methods, a vast increase in computing power, and a richer understanding of the structure of human language has contributed exceedingly to the advancement of the NLP field [10], however natural language understanding still remains the goal of NLP.

This thesis contributes to the field of natural language understanding on the task of Natural Language Inference − also called Recognising Textual Entailment, through the development of deep-learning–based sequence models.

Natural Language Inference (NLI) task encapsulates natural language understanding capabilities within very simple formulation − determining whether a natural language hypothesis can be inferred from a given premise [11]. The example

in Table 1.1, presents a premise and hypothesis from the first recognising textual entailment challenge [2], where the hypothesis is regarded to be entailed from the premise.

Premise: Norway's most famous painting, "The Scream" by Edvard Munch, was recovered Saturday, almost three months after it was stolen from an Oslo museum.
Hypothesis: Edvard Munch painted "The Scream".

Table 1.1: Inference: Premise and Hypothesis sentences from the first recognising textual entailment challenge [2].

1.1 NLI Definitions

The inference is generally defined as deriving a conclusion on the basis of evidence and reasoning. For example, the Oxford[1] dictionary defines the first sense of "infer" as, "to reach an opinion or decide that something is true on the basis of information that is available". More formally, the inference can be defined as *"the act of passing from one proposition, statement, or judgement considered as true to another whose truth is believed to follow from that of the former"* [11].

In NLP, the inference can be defined as the process of concluding the truth value of a *textual* statement based on (the truth of) another given piece of text [11]. NLI captures this language-oriented view of the inference. Dagan et al. [12] refer to NLI as Recognizing Textual Entailment (RTE) formulates the task as follows:

Definition 1.1: *Textual entailment is defined as a directional relationship between pairs of text expressions, denoted by T (the entailing "Text") and H (the entailed "Hypothesis"). We say that T entails H if humans reading T would typically infer that H is most likely true.*

The NLI task definition relies on common human understanding of language and the real-world commonsense knowledge on which the (human) entailment judgement

[1] `https://www.oxfordlearnersdictionaries.com/` – As on August 9, 2020

relies [11]. Thus, for an NLI system to succeed on this task, it must address the full complexity of compositional semantics at all levels of language analysis (lexical, syntactic, semantic, discourse, and pragmatic) as well as learn, remember and apply the real-world commonsense knowledge from the training data.

For example, consider the simplistic premise-hypothesis pair in Table 1.2 from the popular NLI dataset, Stanford Natural Language Inference (SNLI) [3].

Premise: A man inspects the uniform of a figure in some East Asian country.
Hypothesis: The man is sleeping.

Table 1.2: Inference: Premise and Hypothesis sentences from the SNLI dataset [3].

For an inference system to successfully classify that the hypothesis is not entailed from the premise, ideally the system must learn the commonsense fact that when humans inspect something their eyes are open and they can not be asleep (and humans can not do conscious activities like inspection while asleep). However, this commonsense knowledge may or may not be available to the inference system.

Machine learning has been a dominant approach to solving NLI [13]. However, the machine learning research for NLI is severely limited in performance by the lack of gold-standard premise-hypothesis pairs [3, 13]. In the year 2015, the field has renewed prosperity by the introduction of a large human-annotated corpus — SNLI [3]. The public availability of this big dataset has allowed the application of a class of machine learning algorithm called - deep learning.

Deep learning method learns multiple levels of representations from the raw input data and composes the representations learnt at the lower levels into a representation that is more abstract than the lower-level representations [14]. To learn these representations, deep learning employs neural networks with multiple layers and requires a large amount of raw input data [15]. The public availability of the large SNLI dataset has considerably advanced the case of deep learning for NLI. In fact, in the NLI literature, deep-learning-based models are the dominant approach to NLI and claim the state-of-the-art results [16–18]. Our research in the thesis focuses on the development and evaluation of deep-learning-based NLI architectures that are

robust, generalisable and are grounded in real-world knowledge.

In the recent deep-learning-based literature, NLI is predominantly [19–21] defined as

DEFININTION 1.2: *Given a premise and hypothesis sentence, NLI aims to determine whether the logical relationship between premise and hypothesis sentences is among entailment (if the premise is true, then the hypothesis must be true), contradiction (if the premise is true, then the hypothesis must be false) and neutral (neither entailment nor contradiction).*

The formulation of NLI as a simple decision problem over sentence pairs conveniently place the NLI into the standard classification task in the machine learning field [22] while capturing the essence of the definition by Dagan et al. [11].

1.2 Motivation

NLI is crucial to natural language understanding. The task attests the natural language understanding capabilities of a system. As pointed out by Cooper et al. [23] that inferential ability of a system is the best way to evaluate its language understanding competency. The argument is further emphasised by MacCartney and Manning [24], and they state — *"Any system which can reliably identify implications of natural language sentences must have a good understanding of how language works: it must be able to deal with all manner of linguistic phenomena and broad variability of semantic expression."*.

As discussed in the context of NLI Definition 1.1, that for an NLI system to succeed, it must address the full complexity of lexical and compositional semantics at all levels of language analysis (lexical, syntactic, semantic, discourse, and pragmatic), developing such systems considerably advances the developments towards true natural language understanding in NLP.

In addition to advancing the research area of natural language understanding, the inferencing and reasoning abilities of NLI systems are also employed in other complex neural natural language understanding tasks. For example, NLI is used to generate

short and accurate abstractive summaries in the summarisation task [25, 26], the machine comprehension [27] task employs NLI to rank the candidate answers, in neural machine translation, NLI is utilised to investigate the quality of sentence representations encoded by the translation models [28].

Another important area where NLI has been particularly effective is in learning the supervised universal sentence representations [29–31]. Universal sentence representations are trained on a particular task and are subsequently used in other NLP tasks. For example, InferSent [29], the model trained on the SNLI dataset in a supervised manner have outperformed the established unsupervised method, SkipThought [32], on a wide range of transfer tasks such as sentiment classification, paraphrase detection and caption-image retrieval.

As a consequence of its significance to natural language understanding, a broad range of applications and the availability of large corpora amenable to train deep neural networks, NLI has received considerable attention from both academia and industry. A substantial amount of neural NLI literature (Chapter 2) is accumulated in a considerable short time span of half a decade (since the public availability of the SNLI dataset). The attention mechanism [33], which allows a neural model to identify and selectively focus on the important parts of the input, has been a crucial component of this vast NLI literature. Consequently, the research literature can be categorised into two broad categories of - sentence encoding-based models and joint sentence encoding-based models, depending on whether the models employ, intra-attention (when the attention mechanism is applied to the individual premise or hypothesis sentence) [34, 35] or inter-attention mechanisms (when the attention mechanism is applied across the premise and hypothesis sentence) [36, 37] respectively.

Although the models in both the categories utilise the attention mechanism individually, the combination of intra-attention and inter-attention mechanisms is understudied. There are two main challenges. First, the identification of the attention mechanisms that works in synergy and achieve higher performance than the individual attention mechanisms. Second, the investigation of an effective deep neural architecture that can accommodate the combination of the identified attention

mechanisms.

As noted in the NLI task Definition 1.1, learning linguistic as well as real-world commonsense knowledge is central to the success of an NLI system. However, a vast majority of neural NLI models discussed in Chapter 2, bank solely on the training data to learn the full ranges of the linguistic as well as real-world commonsense knowledge required for reasoning inference between the premise and hypothesis. Consequently, the models fail to generalise [38–40]. Particularly, the shortcoming intensifies on the domain-specific datasets, such as SciTail [5], on which the performance of the state-of-the-art models, for example, ESIM [41] and Decomposable Attention Model [42] declines by 18.0% and 14.5% respectively. The state-of-the-art models lack robustness across NLI datasets.

The shortcoming is attributed to the assumption that all the linguistic and commonsense knowledge required for inference is learnable from the provided training data. The assumptions may not be valid, especially for smaller and domain-specific NLI datasets such as SciTail [5] and MedNLI [43]. Particularly, the assumption is not valid, first, because humans (and the NLI datasets annotators) do not express the implicit [44] linguistic and commonsense knowledge, however, they use it all the time. For example, when humans judge the relationship of the premise-hypothesis illustrated in Table 1.2, they intuitively utilise the full range of linguistic and real-world commonsense knowledge and do not just rely on the premise and hypothesis context [45]. Second, Grasser [46] estimates the ratio of explicit:implicit information is up to 1 : 8.22, which implies that a vast majority of information is not mentioned in texts [47]; and hence automated dataset generation techniques utilising Web texts also do not have the implicit linguistic and commonsense knowledge required for reasoning and inference.

As the significant amount of linguistic and commonsense knowledge required for reasoning and inference is not explicitly expressed in the datasets from which the deep neural networks learn, it is unreasonable to expect the neural networks to perform well on the commonsense reasoning tasks such as NLI. The solution is to supply the neural networks with the external linguistic and real-word commonsense knowledge required for reasoning and inference. However, incorporating external knowl-

edge into the neural NLI models is challenging. The main challenges are, first, the **Structured Knowledge Retrieval:** Given a premise-hypothesis pair how to effectively retrieve specific and relevant external knowledge from the massive amounts of data in external knowledge sources such as KGs. Existing models [5, 39–41, 48, 49], use heuristics and word surface forms of the premises and hypothesis, which may be biased and may not be contextually relevant for reasoning over premise and hypothesis. Second, **Encoding Retrieved Knowledge:** Learning the representations of the retrieved external knowledge amenable to be fused with the learned representations of the premise and hypothesis. Various KG embedding techniques [50] are employed to learn these representations. However, while learning the KG embeddings, the embeddings are required to be valid within the individual KG fact and hence might not be predictive enough for the downstream tasks [50]. Third, **Feature Fusion:** How to fuse the learned external knowledge representations with the premise-hypothesis representations. This feature fusion requires substantial NLI model adaptations with marginal performance gains [41].

Hyper-parameter optimisation is highly significant to the performance of Recurrent Neural Network (RNN) based NLI models [51, 52], however, it is surprisingly overlooked in the NLI literature. RNNs (Long Short-Term Memory (LSTM) [53] and Bidirectional Long Short-Term Memory (BiLSTM) [54] networks) owing to their large number of parameters are susceptible to overfitting [55] — the case when the neural network learns the exact patterns present in the training data but fails to generalise to unseen data [56]. One of the crucial hyper-parameters utilised to prevent overfitting in RNNs [55] is dropout [56]. However, the location of dropout applications in the RNN-based NLI models varies considerably and is based on the trial-and-error experiments. The dropout rates are also crucial to the use of dropout regularisation [57]. However, the RNN-based NLI models [58, 59] that apply the dropout at the same layers, for example at the embedding and to the final classifier (refer Section 2.2), also differs in the dropout rates. There is a considerable ambiguity with regards to the application of the crucial regularisation parameter dropout in RNN-based NLI models and the research field lacks a set of coherent guidelines for the dropout application.

The work presented in this thesis addresses the above-mentioned challenges and limitations representing advances in deep neural models for NLI.

1.3 Thesis Contributions

The main contributions of the thesis are as follows:

- We introduce a generic neural architecture that encompasses the contemporary layered neural NLI architectures and present a comprehensive review of the existing literature in the field of deep learning for NLI. (Chapter 2)

- To leverage the benefits of intra-attention and inter-attention mechanisms, we propose a new combined attention model which employs the intra-attention in conjunction with the inter-attention mechanism. Exhaustive evaluation on the SNLI and SciTail datasets show that intra-attention and inter-attention mechanisms work in synergy and achieve higher accuracy when they are combined together in the same model than using them independently. Further, the accuracy analysis for the varying premise-hypothesis lengths shows the model's effectiveness on longer inputs. (Chapter 3)

- To address the limitation of learning the required linguistic commonsense knowledge solely from the training data, we investigate a bilinear feature fusion-based neural NLI model which incorporates real-world commonsense knowledge in the NLI models. Combined with convolutional feature detectors and bilinear feature fusion, the proposed model provides a conceptually simple mechanism that generalizes well and achieves significant performance gains compared to contemporary external knowledge-based models. (Chapter 4)

- We investigate the RNN-based NLI models proposed in Chapter 3 and Chapter 4 for the application of the dropout at different locations in the models with varying dropout rates. As a result, we develop an empirically guided and validated set of guidelines for the application of the dropout regularisation to the deep neural RNN-based NLI models. (Chapter 5)

- A unique method to enrich the contextual representations of the recently proposed pre-training based BERT [16] model with the external knowledge to improve BERT's grounding in the real-world knowledge and reinforce its reasoning and inference capabilities for NLI. Based on the state-of-the-art developments in the field of contextual word representations [16,60,61], the proposed model overcomes the three main challenges of the external knowledge incorporation discussed in Section 1.2, in unique ways and addresses the limitations of the models proposed in Chapter 4. Extensive experiments on the challenging SciTail and SNLI datasets demonstrate the effectiveness of the proposed model: in comparison to the previous state-of-the-art, our model obtain an accuracy of 95.9% on the SciTail dataset and 91.5% on the SNLI dataset. (Chapter 6)

1.4 Publications

The work contained within this thesis has been previously published or is in communication in the following peer-review publications, and is used in the chapters as indicated below:

- **CAM: A Combined Attention Model for Natural Language Inference**, Amit Gajbhiye, Sardar Jaf, Noura Al Moubayed, Steven Bradley, and A. Stephen McGough, In IEEE International Conference on Big Data (Big Data). IEEE, 2018. (Published, Contributing to Chapter 3) [62]

- **Bilinear Fusion of Commonsense Knowledge with Attention-Based NLI Models**, Amit Gajbhiye, Thomas Winterbottom, Noura Al Moubayed, and Steven Bradley, In 29$^{\text{th}}$ International Conference on Artificial Neural Networks, Springer, Cham, 2020. (Accepted for publication, Contributing to Chapter 4)

- **An Exploration of Dropout with RNNs for Natural Language Inference**, Amit Gajbhiye, Sardar Jaf, Noura Al Moubayed, Steven Bradley,

and A. Stephen McGough, In International Conference on Artificial Neural Networks. Springer, Cham, 2018. (Published, Contributing to Chapter 5) [63]

- **ExBERT: An External Knowledge Enhanced BERT for Natural Language Inference**, Amit Gajbhiye, Noura Al Moubayed, and Steven Bradley, In Proceedings of COLING 2020, the 28th International Conference on Computational Linguistics: Technical Papers, 2020 (In communication, Contributing to Chapter 6)

1.5 Thesis Scope and Structure

Development of new neural architectures and their empirical evaluation on the NLI data constitute the primary mode of research of this thesis. The goal is to propose and implement robust, generalisable and knowledge-grounded neural architectures for NLI via the incorporation of external knowledge and attention mechanisms. Towards this goal, the thesis presents the research work structured into the following chapters.

Chapter 2 sets the background knowledge required for the thesis by introducing a generic architecture for the existing deep neural NLI models in the literature. The chapter tabulates the available NLI and external knowledge sources and elaborates on the SNLI and SciTail NLI datasets and the external knowledge sources the ConceptNet [64] and Aristo Tuple [65] KGs.

The chapter, then thoroughly reviews the deep neural NLI literature by categorising the field into sentence encoding-based models and joint sentence encoding-based models. The deep learning models are further divided and discussed according to the structure of the encoder the models employ.

Chapter 3 explores the use of combined attention mechanisms in a neural NLI model. Exploiting the benefits of the intra-attention and inter-attention mechanisms, the experimental result on the SNLI and SciTail datasets demonstrate that the two attention mechanisms work in synergy. The chapter further investigates ablation, premise-hypothesis lengths and the qualitative analysis showing that our model effectively learns to reason between the premise and hypothesis and does not

depend on the word overlap between them. The research presented in this chapter is published in a peer-reviewed conference [62].

Chapter 4 addresses the limitation of inadequate learning of the linguistic and commonsense knowledge from the training dataset by incorporating linguistic as well as real-world commonsense knowledge into the NLI models. The chapter further presents the quantitative and qualitative results using the SNLI and SciTail datasets in combination with a general real-world commonsense knowledge KG ConceptNet and (science) domain-specific KG Aristo Tuple. The research presented in this chapter is accepted for publication in a peer-reviewed conference[2].

Chapter 5 presents the empirical evaluations, analysis and discussions on the RNN-based NLI models presented in Chapter 3 and Chapter 4. The chapter first formulates the different locations of dropout application for the model proposed in Chapter 3 and then based on the formulations evaluates the model with varying dropout rates at each location. The evaluation results are analysed to observe distinct patterns. The observations are then validated on the RNN-based NLI model proposed in Chapter 4. Finally, the validated observations are established as the guidelines for the application of the dropout in RNN-based NLI models. The research presented in this chapter is the part of the publication [63].

Chapter 6 investigates enriching the contextual representations of the pre-trained BERT model with the real-world commonsense knowledge from the external knowledge sources to enhance its grounding in real-world knowledge and augment the reasoning capabilities for NLI.

The chapter proposes a novel external knowledge retrieval mechanism utilising contextual representations to retrieve relevant external knowledge. The retrieved external knowledge is incorporated in the BERT model via a unique approach. The experimental results on the SNLI and SciTail datasets in conjunction with the ConceptNet and Aristo Tuple KGs show that the proposed approach achieves significant performance improvements over the previous state-of-the-art methods, including those which are enhanced by the $\text{BERT}_{\text{LARGE}}$ model. The proposed model

[2] `https://e-nns.org/icann2020/` – As on August 9, 2020

also outperforms all our previous approaches to NLI. The research work presented in this chapter is in communication with a peer-reviewed conference[3].

Finally, Chapter 7 concludes and lays out promising directions for future work.

[3] `https://coling2020.org/` – As on August 9, 2020

CHAPTER 2

Deep Learning for NLI - Generic Model & Literature Review

2.1 Introduction

In this chapter, we introduce a generic neural NLI model from which most of the existing neural NLI models can be derived (Section 2.2). This model sets the stage for the discussions about the NLI models discussed subsequently. We tabulate the available NLI datasets for the model evaluation. We elaborate on the NLI datasets and the external knowledge sources that are utilised in the NLI model evaluations presented in this thesis (Section 2.3). Further, we discuss the performance metric used to evaluate deep neural NLI models (Section 2.4).

We present a taxonomy and review the current neural NLI literature by categorising the field into sentence encoding- and joint sentence encoding-based models (Section 2.5). We further classify the literature based on the structure of the encoders the NLI model employs to encode the context of the premise and hypothesis. Finally, we present the conclusions of the chapter (Section 2.6).

2.2 Neural NLI Models: A Generic Architecture

In this section, we present a generic architecture for neural NLI models. The architecture is depicted in Fig.2.1. The layered architecture consists of the embedding, encoding, intra-attention, inter-attention, enhancement, composition, pooling, matching and output layers and an external knowledge source component. We elaborate on each of the layers in the following sections.

2.2.1 Embedding Layer

The embedding layer, also known as word representation layer, maps each word/token in the premise and hypothesis to a d-dimensional vector representation. In the model, the embedding layer can represent words as vectors using pre-trained word embeddings/representations such as Word2Vec [66], GloVe [67], FastText [68], ConceptNet Numberbatch [64] or contextual word embeddings [69] such as CoVE [70], ELMo [60], or the BERT [16] embeddings.

The representation capabilities of the word embeddings can be augmented with the character embeddings of words (word's embedding learned from its individual characters), part-of-speech, semantic role, named-entity recognition tag embeddings, and with the parsing information of the individual words to incorporate more lexical, syntactical and semantic information [52, 71, 72]. Further, word embeddings are also fine-tuned dynamically with external knowledge such as Wikipedia[1], to incorporate commonsense and background knowledge into the NLI models [73]. The pre-trained word embeddings may or may not be fine-tuned jointly with the other parameters of the model while training.

Let us represent the NLI task as a triple $(\mathrm{P}, \mathrm{H}, y)$, where $\mathrm{P} = \{p_1, p_2, \ldots, p_n\}$ is the premise sentence with length n, $\mathrm{H} = \{h_1, h_2, \ldots, h_m\}$ is the hypothesis sentence with length m, and $y \in \mathcal{Y}$ is the label, for example, in $(entailment, contradiction, neutral)$, representing the relationship between the premise and hypothesis. The

[1] `http://wiki.dbpedia.org/downloads-2016-10` – As on August 9, 2020

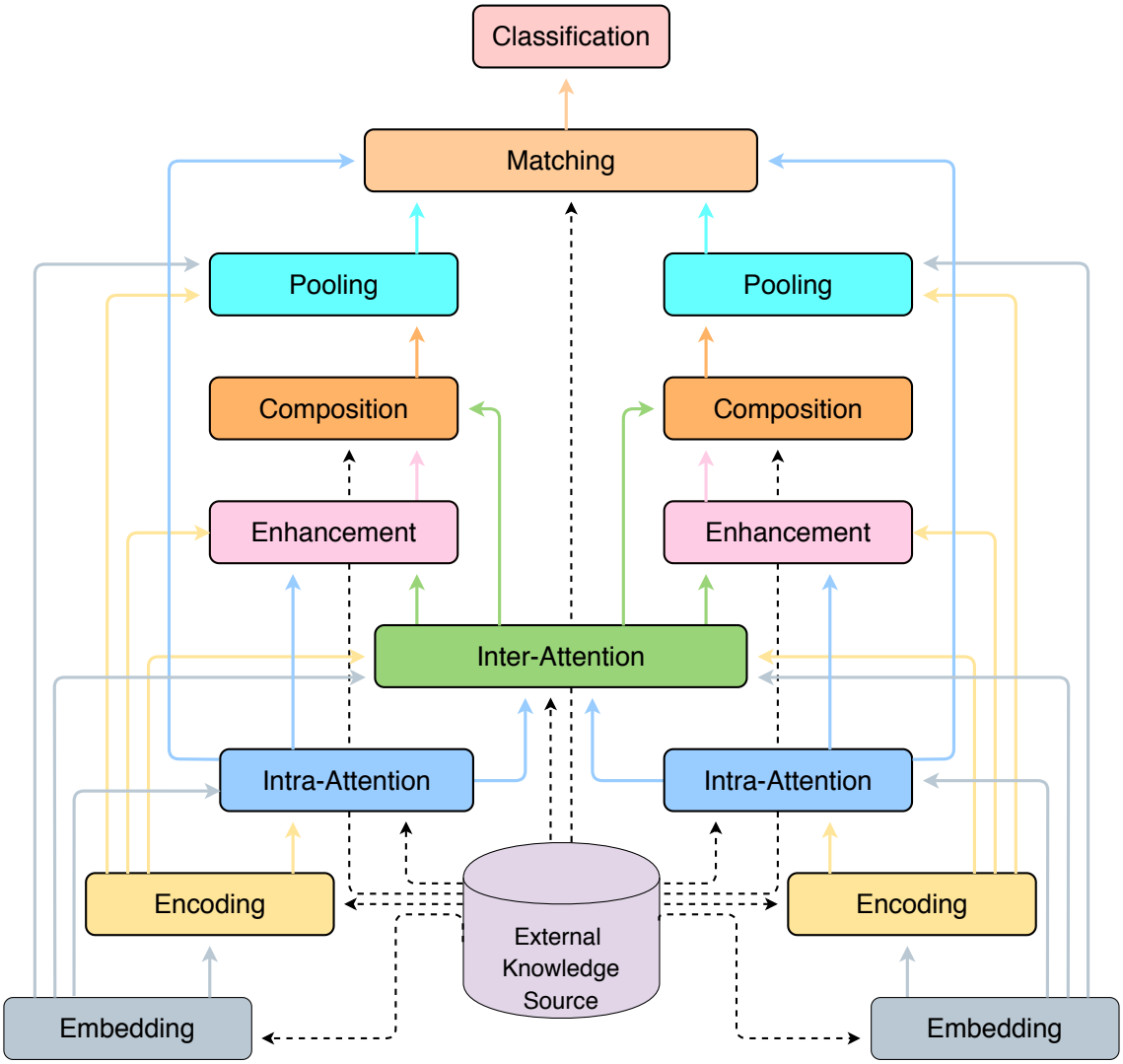


Figure 2.1: The generic architecture of neural NLI models. The connecting coloured arrows join a layer to the other possible layers to which the arrow emanating layer can be connected to. The layers and the emanating arrows are coded in the same colour. The dotted lines from the external knowledge source show the layers at which the external knowledge is incorporated in the literature.

embedding layer can be defined as

$$P^{emb} = (\mathbf{p}_1^{emb}, \mathbf{p}_2^{emb}, \ldots, \mathbf{p}_n^{emb}) = \text{Embedding}(p_1, p_2, \ldots, p_n) \tag{2.1}$$

$$H^{emb} = (\mathbf{h}_1^{emb}, \mathbf{h}_2^{emb}, \ldots, \mathbf{h}_m^{emb}) = \text{Embedding}(h_1, h_2, \ldots, h_m) \tag{2.2}$$

where $P^{emb} \in \mathbb{R}^{n \times d^{emb}}$ and $H^{emb} \in \mathbb{R}^{m \times d^{emb}}$ are the matrices representing each

of the premise word embeddings $(\mathbf{p}_1^{emb}, \mathbf{p}_2^{emb}, \ldots, \mathbf{p}_n^{emb})$ and the hypothesis word embeddings $(\mathbf{h}_1^{emb}, \mathbf{h}_2^{emb}, \ldots, \mathbf{h}_m^{emb})$ respectively. d^{emb} is the dimension of the pre-trained embedding.

2.2.2 Encoding Layer

The encoding layer accepts the sequence of word embeddings as input and encodes them by incorporating the context information from the word embeddings in the surrounding context. Different encoders such as chain- and tree-structured RNNs [74] (LSTMs [53], BiLSTMs [54] and GRUs [75]), Convolutional Neural Networks (CNNs) [76], highway networks [77], encoders without RNN and CNN [16, 78] have been employed in various models proposed in the NLI literature. Different techniques such as stacking of encoding layers [78, 79], short-cut connections from the preceding layers [80], external memory augmentation [81] and infusing external knowledge [82] are applied to enhance the representation capabilities of the sentence encodings.

Recently, Pre-Trained Language Model (PTLM) based encoders such as ULMFiT [83], OpenAI GPTs [61, 84, 85], and the BERT [16] model have become popular encoders for natural language understanding tasks including the NLI. Formally, the encoding process can be defined as

$$P^{enc} = (\mathbf{p}_1^{enc}, \mathbf{p}_2^{enc}, \ldots, \mathbf{p}_n^{enc}) = \text{Encoding}(\mathbf{p}_1^{emb}, \mathbf{p}_2^{emb}, \ldots, \mathbf{p}_n^{emb}) \tag{2.3}$$

$$H^{enc} = (\mathbf{h}_1^{enc}, \mathbf{h}_2^{enc}, \ldots, \mathbf{h}_m^{enc}) = \text{Encoding}(\mathbf{h}_1^{emb}, \mathbf{h}_2^{emb}, \ldots, \mathbf{h}_m^{emb}) \tag{2.4}$$

where $P^{enc} \in \mathbb{R}^{n \times d^{enc}}$ and $H^{enc} \in \mathbb{R}^{m \times d^{enc}}$ are the matrices representing context-aware representation of each of the tokens in the premise and hypothesis respectively. d^{enc} is the dimension of the hidden states of the encoding layer.

2.2.3 Interaction Layers

NLI models utilise the interactions between the words of the individual premise and hypothesis or across the two sentences to learn the dependencies between words and to link and fuse the local features. Intra-attention (self-attention) [34] and inter-attention (cross-sentence) [36] mechanisms facilitate these interactions. In the

following sections, we elaborate on the intra-attention and inter-attention layers.

Intra-Attention Layer

The intra-attention (self-attention) layer models the dependencies between the words from the same sequence. The layer learns the relevance and similarity of each word with respect to the entire sequence, capturing long-distance dependencies and the global context of the entire sequence. The relevance and similarity of the words are measured by an attention score between the words at different positions in the sequence by computing an attention function (also called compatibility function or similarity function) between each pair of words. Different attention functions such as additive [34], gated [86] and dot-product [20] have been utilised in intra-attention layer.

The intra-attention layer can be applied to the output of the encoding layer [20, 34, 86] or to the output of the embedding layer bypassing the encoding layers [78, 87, 88].

For the sake of brevity, let $\acute{P}$ and $\acute{H}$ represent either of the embedded premise (P^{emb}) and embedded hypothesis (H^{emb}) or the encoded premise (P^{enc}) and encoded hypothesis (H^{enc}) respectively.

$$C^{p\_intra\_atten},\ A^{p\_intra\_atten} = \text{IntraAttention}(\acute{P}) \tag{2.5}$$

$$C^{h\_intra\_atten},\ A^{h\_intra\_atten} = \text{IntraAttention}(\acute{H}) \tag{2.6}$$

where $C^{p\_intra\_atten} \in \mathbb{R}^{n \times \acute{d}}$ is the context-aware encoding obtained by linear combination (weighted sum) of each representation in $\acute{P}$ and attention probabilities $A^{p\_intra\_attn} \in \mathbb{R}^{n \times n}$. Similarly, the context-aware encodings ($C^{h\_intra\_atten}$) and attention probabilities ($A^{h\_intra\_atten}$) are obtained for the hypothesis sentence. $\acute{d}$ is the hidden state dimension depending on whether the embedding or encoding representations are used in intra-attention layer.

Although the RNNs dominate the class of encoders in NLI literature, RNNs process the input sequence sequentially, word-by-word and hence preclude parallelisation at training time. Recently, a class of RNN-/CNN-free models such as

Transformers [78], DiSAN [87], ReSAN [88] and distance-based intra-attention networks [89] have been proposed that utilise solely the intra-attention mechanism to encode the input sequence. These models benefit from the parallelisable computation, reduced training time, and flexibility in modelling long-distance dependencies in the input sequence [90].

Inter-Attention Layer

The inter-attention layer is set-up on the intra-attention layer or the encoding layer and it learns the alignment between the relevant words of the premise and hypothesis. The alignment provides the local relevance and dependencies between the words of the premise and the hypothesis. Different attention functions such as additive [34,36], dot-product [21,91], scaled dot-product [78] and bilinear [92] attentions are employed at inter-attention layer. The dot product and recently, the scaled dot-product attention is dominantly used in NLI models due to its fast computation speed and better results. Let us consider when the inter-attention layer is applied after the encoding layer as

$$C^{p\_inter\_atten}, C^{h\_inter\_atten}, A^{p\_inter\_atten}, A^{h\_inter\_atten} = \text{InterAttention}(P^{enc}, H^{enc}) \tag{2.7}$$

where $C^{p\_inter\_atten} \in \mathbb{R}^{n \times d^{enc}}$ is the attention probability ($A^{h\_inter\_atten} \in \mathbb{R}^{n \times m}$) weighted summation of the encoded hypothesis representations (H^{enc}). Intuitively, $C^{p\_inter\_atten}$ represents the contents in H^{enc} which are relevant to premise. Similarly, $C^{h\_inter\_atten} \in \mathbb{R}^{m \times d^{enc}}$ is the attention probability weighted ($A^{p\_inter\_atten} \in \mathbb{R}^{m \times n}$) summation of P^{enc}, highlights the representations relevant to hypothesis.

2.2.4 Enhancement Layer

The enhancement layer further captures a variety of similarities between the learned attentional information from the inter-attention layer and the encoded representations from the encoding layer or the intra-attention layer with the intention of better identifying the entailments and contradictions [21, 91, 93]. For example, the difference and element-wise product of the attentional information with the corresponding

encoded representations are used to calculate the similarity and closeness between the two representations [21]. The layer generally generates a feature vector by the concatenation of the outputs from the different similarity measures employed, the corresponding encoded representations and the representations from the attention layers (intra-attention layer or inter-attention layer). Let us consider the case when the encoded representations are enhanced with the attentional information from the inter-attention layer. The layer, in this case, can be represented as

$$P^{enhance} = \text{Enhancement}(P^{enc}, C^{p\_inter\_atten}) \tag{2.8}$$

$$H^{enhance} = \text{Enhancement}(H^{enc}, C^{h\_inter\_atten}) \tag{2.9}$$

where $P^{enhance} \in \mathbb{R}^{n \times num\_sim * d^{enc}}$ and $H^{enhance} \in \mathbb{R}^{m \times num\_sim * d^{enc}}$ and num_sim is the number of similarity measures. A feed-forward neural network may be employed in this layer to lower the dimensionality of the output feature vectors in order to reduce model complexity and prevent overfitting [21, 91] (refer Chapter 5). The output of the layer is fed to the composition layer as discussed below.

2.2.5 Composition Layer

The composition layer learns and aggregates the local alignment information learned through the inter-attention layer. The layer essentially determines the overall inference information between the premise and the hypothesis. The majority of NLI models [20, 21, 91] uses RNNs (BiLSTMs and LSTMs) for aggregation to avoid losing any information that might rely on the sequence of local inference vectors. The models such as DIIN [19] employ a CNN architecture to aggregate the local inference information. The composition layer is also known as aggregation layer in the NLI literature.

$$P^{comp} = \text{Composition}(P^{enhance}) \tag{2.10}$$

$$H^{comp} = \text{Composition}(H^{enhance}) \tag{2.11}$$

where $P^{comp} \in \mathbb{R}^{n \times d^{comp}}$ and $H^{comp} \in \mathbb{R}^{m \times d^{comp}}$ are the matrices representing the

overall inference information in the premise and hypothesis respectively and d^{comp} is the dimensionality of the composition layer hidden states.

2.2.6 Pooling Layer

The pooling layer computes a fixed-length sentence representation (sentence embedding) from the output of the composition layer. Standard maximum, minimum, and mean pooling operations are used to generate the fixed-length sentence embedding [21, 29, 70, 94]. Special pooling strategies such as attention pooling [86, 92], weighted pooling [41] and generalised pooling [72] are also proposed in the literature.

$$\mathbf{p}^{pool} = \text{Pooling}(P^{comp}) \tag{2.12}$$

$$\mathbf{h}^{pool} = \text{Pooling}(H^{comp}) \tag{2.13}$$

where $\mathbf{p}^{pool} \in \mathbb{R}^{d^{pool}}$ and $\mathbf{h}^{pool} \in \mathbb{R}^{d^{pool}}$ are the fixed-length sentence embeddings of premise and hypothesis.

2.2.7 Matching Layer

The matching layer defines how the individual sentence embeddings created in the pooling layer are combined to capture the relationship between the premise and hypothesis. Most of the NLI models concatenate the maximum- and mean-pooled outputs of the premise and hypothesis representation created in the pooling layer [20, 21, 70, 91]. However, different matching heuristics such as the concatenation of premise sentence embedding, hypothesis sentence embedding, their element-wise subtraction and their element-wise multiplication are also evaluated in the literature [95].

$$\mathbf{f}^{final} = \text{Matching}(\mathbf{p}^{pool}, \mathbf{h}^{pool}) \tag{2.14}$$

where $\mathbf{f}^{final} \in \mathbf{R}^{d^{final}}$ is the final joint sentence representation of the premise and hypothesis that is input to the classification layer for NLI class prediction.

2.2.8 Classification Layer

The classification layer employs a multilayer perceptron (MLP) classifier composed of multiple hidden layers, usually with tanh activation and a final softmax layer outputting the probabilities (y^{prob}) of each NLI class as

$$y^{prob} = \text{softmax}(\text{MLP}(\mathbf{f}^{final})) \tag{2.15}$$

The NLI models are trained using the following standard cross-entropy objective.

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log y_i^{prob} \tag{2.16}$$

where y_i denotes the true class label and C is the number of classes in NLI. NLI can be a binary or multi-class (three) classification depending on the available classes in the dataset.

2.3 Datasets

This section discusses NLI datasets and KGs. We first tabulate the different NLI datasets and elaborate on two well-established – SNLI [3] and SciTail [5] NLI datasets. We use these datasets to evaluate the NLI models that are presented in the thesis. Similarly, we tabulate various KGs and elaborate on the two KGs – ConceptNet [64] and Aristo Tuple [65] KGs employed in the presented models to retrieved external knowledge.

2.3.1 NLI Datasets

Recently, a large number of specialised datasets focussing on evaluating the particular abilities, such as generalisation capability [38], cross-lingual language understanding [96] and the quantitative reasoning abilities [97] of the NLI models have been proposed. Table 2.1 presents the available NLI datasets.

NLI Dataset	Description	Size	NLI Classes
FraCaS [23]	Dataset targeted for logical entailment.	346	3 (YES, NO, UNK)
RTE-n [2]	Different RTE challenge[2] datasets.	≈ 1000	2 (E, NE), 3 (E, C, UNK)
SICK [98]	Dataset aimed at evaluating the lexical, syntactic and semantic phenomena that a neural model should handle for modelling natural language sentence.	9,800	3 (E, C, N)
SNLI [3]	The most evaluated dataset, publicly released to advance the case of neural models research in NLI. Created from the image description of Flickr30K [99] image captions. The reasoning in SNLI is tied to specific situations of image caption descriptions.	570,000	3 (E, C, N)
MPE [100]	NLI task dataset that requires inference over multiple premise sentences.	10,000	3 (E, C, N)
SPR, FN+, DPR [101]	Leverage existing Semantic Proto-Roles (SPR) [102], FrameNet Plus (FN+) [103] and Definite Pronoun Resolution (DPR) [104] datasets as a source of inference examples.	312,873	2 (E, NE)

[2] `https://aclweb.org/aclwiki/Textual_Entailment_Resource_Pool` – As on August 9, 2020.

MultiNLI [105]	Dataset designed to increase the coverage and difficulty of NLI datasets. The premise and hypothesis are derived from ten different genres of written and spoken English to capture complex phenomena such as belief and temporal reasoning.	433,000	3 (E, N, C)
SciTail [5]	Science domain targeted dataset. The premises and hypotheses are derived from the end task of science question-answering.	27,000	2 (N, E)
Adversarial NLI [38]	Adversarial SNLI test set aimed at evaluating the generalisation capability of NLI models. The test set contains sentences that differ by at most one word from the sentences in the training set.	8,193	3 (E, C, N)
DNC [106]	DNC recasts 13 existing datasets from 7 NLP tasks into NLI examples.	570,459	2 (N, NE)
GLUE Diagnostic Dataset [107]	Test set to analyse the types of knowledge (such as lexical semantics, logic, commonsense, etc.) learned by the models evaluated on GLUE benchmark.	1100	3 (E, C N)
XNLI [96]	XNLI extends the development and test sets of MultiNLI to 15 languages to evaluate the cross-lingual language understanding of the NLI models.	112,500	3 (E, C, N)
e-SNLI [108]	e-SNLI annotates the SNLI dataset with human explanations.	570,000	3 (E, C, N)
MedNLI [43]	NLI dataset in clinical domain annotated by doctors.	14, 049	3 (E, C, N)

HANS [109]	Test set concentrating on evaluating the learned syntactic heuristics in NLI models.	4827	2 (E, NE)
SherLic [110]	Test set focussed on testing the lexical inference in context in NLI models.	3985	2 (E, NE)
EQUATE [97]	Test set to evaluate quantitative reasoning of NLI models.	9606	2 (E, NE), 3 (E, C, N)
IMPPRES [111]	Dataset to investigate two pragmatic inference types - scalar implicatures and presuppositions.	32,000	3 (E, C, N)

Table 2.1: NLI Datasets. For the NLI classes, the label E stands for the Entailment, C for Contradiction, N for Neutral, UNK for Unknown, NE for Not Entailed.

In the following sections, we elaborate on the — SNLI and SciTail datasets, that are employed to evaluate the models presented in this thseis.

Stanford Natural Language Inference Dataset (SNLI)

The Stanford Natural Language Inference (SNLI) dataset [3], publicly released in the year 2015, is the most popular dataset that is utilised to train and evaluate deep neural NLI models. Almost all the state-of-the-art NLI models are trained on the SNLI dataset and the performance of a model on this dataset is a yardstick[3] for the effectiveness of the model. Before SNLI dataset, the PASCAL Recognising Textual Entailment (RTE) challenges [2] was the primary source of annotated NLI data[4]. Although these hand-labelled datasets were of high-quality, the small dataset size limits their utility for learning the semantic representations via deep neural models. Inspired by these limitations, Bowman et al. [3] created the SNLI dataset to facilitate the learning and evaluation of deep learning approaches to NLI.

[3]SNLI dataset Leaderboard: `https://nlp.stanford.edu/projects/snli/` – As on July 15, 2020

[4]`https://aclweb.org/aclwiki/Textual_Entailment_Resource_Pool` – As on July 15, 2020

Data Creation Rules and Collection The authors collected 570,152 premise-hypothesis pairs through Amazon Mechanical Turk[5], a widely used crowdsourcing marketplace to outsource processes and jobs to a distributed workforce. The workers were presented with premise sentences and were asked to supply hypotheses for each of the NLI classes. For the premise sentence, the captions from the Flicker30K [99] corpus, a collection of approximately 160,000 captions, were used. To provide the hypotheses for each of the entailment, neutral, and contradiction class respectively, the workers were instructed as follows.

- Write one alternate caption that is definitely a true description of the photo.
- Write one alternate caption that might be a true description of the photo.
- Write one alternate caption that is definitely a false description of the photo.

Table 2.2 illustrates the hypotheses corresponding to a premise written by the workers according to each instruction enlisted above.

Premise	
A senior is waiting at the window of a restaurant that serves sandwiches.	
Hypotheses Written by Workers for Each Instruction	
Entailment	A person waits to be served his food.
Neutral	A man is looking to order a grilled cheese sandwich.
Contradiction	A man is waiting in line for the bus.

Table 2.2: SNLI dev set premise with the hypotheses written by Amazon Mechanical Turk workers for each inference class according to the instructions provided.

The SNLI dataset is balanced among the three NLI classes and is available with a prespecified train/development/test splits of $549,367/9,842/9,824$ examples respectively. The dataset is publicly available[6] under Creative Commons Attribution-ShareAlike license[7].

[5] `https://www.mturk.com/` – As on August 9, 2020.
[6] `https://nlp.stanford.edu/projects/snli/` – As on August 9, 2020.
[7] `https://creativecommons.org/licenses/by-sa/4.0/` – As on August 9, 2020.

Textual Entailment Dataset from Science Question Answering (SciTail)

The SciTail [5] dataset consists of science domain-specific premise-hypothesis pairs. Khot et al. [5] argue that NLI datasets should be created in conjunction with an end task to capture the kind of entailment queries that naturally arises in the task. The authors created the SciTail dataset, derived from the task of multiple-choice science question answering and that consists of naturally occurring premises and hypotheses which are not authored specifically for the NLI task.

Data Creation Rules and Collection Given a multiple-choice question and the correct answer, the hypothesis in the SciTail dataset was created by converting the question and the answer into an assertive statements. The premise sentence was retrieved from a large text corpus using the words from the question and the answer. Consider the following multiple-choice question with correct answer (C) from the 4^{th} grade science test [5].

Which of the following best explains how stems transport water to other parts of the plant?

(A) *through a chemical called chlorophyll.*
(B) *by using photosynthesis.*
(C) *through a system of tubes.*
(D) *by converting water to food.*

The assertive sentence for the hypothesis is then manually created as - "*Stems transport water to other parts of the plant through a system of tubes*". The premise retrieved from the text corpus, for example can be - "*Water and other materials necessary for biological activity in trees are transported throughout the stem and branches in thin, hollow tubes in the xylem, or wood tissue.*".

The authors used the aforementioned annotation scheme to create the SciTail premise-hypothesis pairs. For the multiple-choice questions the authors used the publicly released 4^{th} grade and 8^{th} grade exams[8] and the crowd-sourced questions

[8]Using AI2 Science Questions v1 from `http://allenai.org/data/science-exam-questions.html` – As on August 9, 2020

from SciQ dataset [112]. For premise sentence, a set of K probable sentences were retrieved from the Web Corpus [4], containing 280 GB of plain text, using the question and answer choice as query [113].

To collect the NLI class labels for the premise-hypothesis pair the authors utilised Amazon Mechanical Turk. For each question and correct answer a batch of 10 retrieved premises were shown to the workers who were instructed as follows to classify each premise into one of the three categories.

- *Complete Support*, if the premise fully supports the answer choice.
- *Unrelated*, if the premise is unrelated to the question; or
- *Partial support*, if the premise is related to the question but only provides partial support for the answer.

Table 2.3 shows a hypothesis and the retrieved premises which are labelled according to the instructions enlisted above.

Hypotheis	
One way to change water from a liquid to a solid is to decrease the temperature.	
Retrieved Premises with Annotator Label	
Complete Support	A liquid becomes a solid if its temperature decreases.
Unrelated	At one temperature and pressure, called the Triple Point , all three phases of water (liquid), water vapor (gas), and ice (solid) coexist at equilibrium.
Partial support	In particular, if the temperature of a sample of materials is changed, the material may change from one state to another (liquid to solid, liquid to gas, and so on).

Table 2.3: SciTail development set hypothesis with the retrieved premises from the Web Corpus [4] and the corresponding Amazon Mechanical Turk annotator labels according to the provided instructions.

The examples with *Complete Support* label are used to create *entailment* class and *Unrelated* label creates *neutral* class examples in the dataset. The *Partial Support* examples were ignored.

The SciTail dataset contains a total of $27,026$ premise-hypothesis pairs with

10,101 examples of entailment class and 16,925 of neutral class. The train/development/test splits is prespecified and contains 23,596/1,304/2,126 examples respectively. The dataset is publicly available[9].

On the Use of SNLI and SciTail Datasets

We utilise the SNLI and SciTail datasets to train and evaluate the NLI models presented in this thesis. The following are the major considerations for these datasets:

- **Established Benchmarks** The SNLI and SciTail datasets are well-established and are widely employed in the neural NLI literature for training and evaluation. This facilitates model performance comparisons and helps in putting the model performance into perspective.
- **Dataset Heterogeneity** The SNLI and SciTail datasets differs significantly in sizes (SNLI is approximately twice the order of magnitude than SciTail), genres (SNLI contains general descriptive premise-hypothesis pairs whereas SciTail has science domain specialised premise-hypothesis pairs), sentence lengths and complexity (SNLI premise-hypothesis pairs are semantically simpler and are shorter in token length than the long and the semantically complex premise-hypothesis pairs of SciTail [5]). Training and evaluation on such diverse datasets provide an excellent indication of the model generalisability and learning capabilities.

Enhancing the language understanding and reasoning capabilities of NLI models trained on the aforementioned datasets via incorporating external lexical, commonsense and domain-specific knowledge is a significant theme of the presented thesis. Thus, in addition to NLI datasets, external knowledge sources are crucial to the models developed as a part of the thesis. The following section discusses the external knowledge sources utilised.

[9] `https://allenai.org/data/scitail` – As on August 9, 2020.

2.3.2 External Knowledge Sources

The external knowledge sources utilised in NLI literature are presented in Table 2.4. Most openly available external knowledge sources are organised in the form of KGs [114].

Knowledge Source	**Description**
WordNet [115]	A is a lexical database, where English words are grouped into different sets (synsets), each expressing a distinct concept. The 117,000 WordNet synsets are linked to other synsets by means of a small number of lexical relations such as hypernyms, hyponyms and meronym.
DbPedia [116]	DBPedia extracts knowledge from Wikipedia, providing a large number of facts ($\approx$ 1.95M), mainly focussed on named entities (for example, persons, places, music albums and films) that have Wikipedia articles.
FreeBase [117]	Freebase is contains general human knowledge mainly focussed on named entities (people, palces and things). Freebase contains $\approx$ 4.9M facts [82].
UMLS [118]	The largest publicly available database focussed on biomedical domain containing $\approx$ 12M biomedical concepts.
ConceptNet [64]	ConceptNet is a multilingual KG that relates words and phrases through lexical as well as commonsense knowledge used in real-world. The KG consists of $\approx$ 21M lexical and commonsense facts about real-world. (refer Section 2.3.2)
Aristo Tuple [65]	Aristo Tuple is domain-targeted KG containing facts relevant to elementary science. The KG contains 283K science facts. (refer Section 2.3.2)

Table 2.4: External knowledge sources utillised in neural NLI models.

A KG models the real-world to represent factual knowledge that connects real-world objects, events and abstract concepts. It is organised as a multi-relational graph that is composed of entities (nodes) and relations (different types of edges) representing the knowledge as *(head relation tail)* triples [119]. The relation specifies that the head and tail entities are associated by the relation, for example, the triple from ConceptNet [64] KG - *employee partof company*, represents the fact that an employee is a part of a company.

KGs are used widely in modern deep-learning approach to create knowledge-grounded natural language understanding applications such as conversational agents [120,121], reading comprehension systems [49], question answering systems [122] and inference systems [123].

KGs are the source of external linguistic, commonsense and domain-specific knowledge for the models presented in the thesis. In the following sections, we discuss two widely used KGs - ConceptNet 5.5 [64] and Aristo Tuple v4 [122]. We will utilise these KGs to incorporate linguistic, general commonsense knowledge and science domain-specific knowledge into the deep neural NLI models that are proposed in this thesis.

ConceptNet 5.5

ConceptNet[10] [64] is a multilingual KG that includes linguistic and real-world commonsense knowledge. Figure 2.2 depicts an extract from the ConceptNet KG. It connects words and phrases of natural language with the lexical and commonsense relationship between them. For example, *performing synonym acting, liquid antonym solid, shirt UsedFor wearing, car IsA vehicle or umbrella AtLocation closet.* Natural language applications such as reading comprehension [48], conversational systems [124] and commonsense question answering [125], leverage ConceptNet to incorporate rich real-world knowledge into the deep neural models.

Data Collection ConceptNet 5.5 acquires knowledge from the following sources:

- Open Mind Common Sense (OMCS) [126] and sister projects in other languages [127].
- Parsed information from Wiktionary[11].
- "Games with a purpose", designed to collect common knowledge [128] [129].
- Linked-data representation of WordNet [115] - the Open Multilingual WordNet [130].
- A Japanese-multilingual dictionary - JMDict [131].

[10] `http://conceptnet.io/` – As on August 9, 2020.
[11] `https://en.wiktionary.org/wiki/Wiktionary:Main_Page` – As on August 9, 2020.

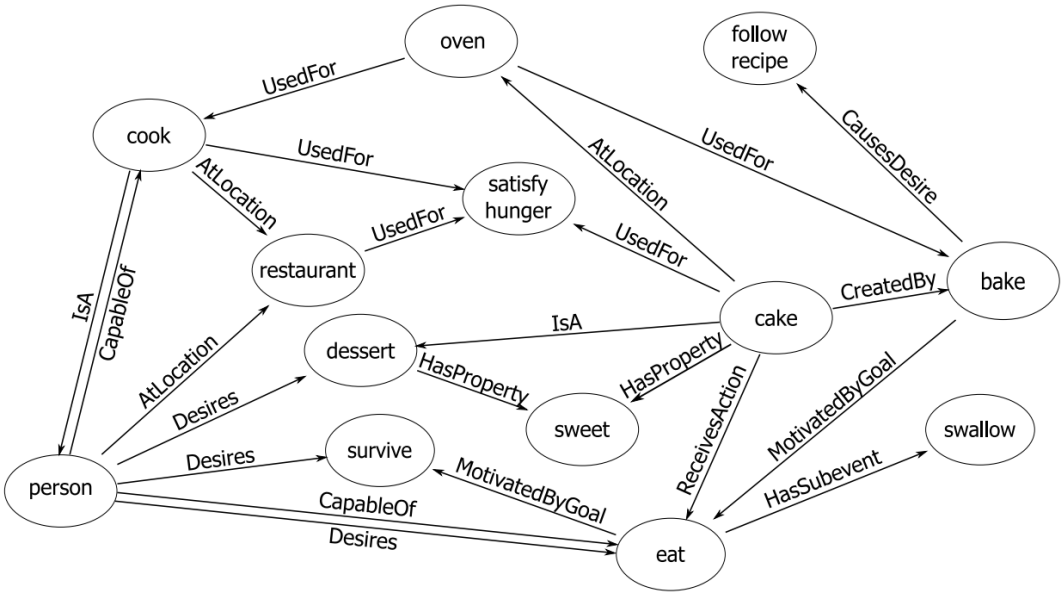


Figure 2.2: An extract of the commonsense knowledge in ConceptNet. ConceptNet relates real-world entities and abstract concepts with lexical as well as commonsense knowledge relations. Adopted from [1].

- A system that represents commonsense knowledge in predicate logic - OpenCyc provided by Cyc [132].
- A network of facts extracted from Wikipedia information boxes - a subset of DBPedia [133].

Combining these sources, ConceptNet contains over 21 million edges and over 8 million nodes. The total number of unique relations in ConceptNet 5.5 is 47. The relations can be symmetric, such as *Antonym* and *DistinctFrom* where the directionality of edges is not important, as well as Asymmetric, such as *AtLocation* and *CreatedBy*, where the directionality of relationships is crucial.

Aristo Tuple v4

Aristo Tuple [65] is a science domain-targeted KG containing facts in (*head relation tail*) triple form relevant to elementary science. Some of the science related facts in Aristo Tuple are *air Have refraction, amino acid MakeUp protein, blood glucose LeadTo disease* and *zygote HasPart chromatin.*

Data Collection Aristo Tuple is collected from the Web following a unique data extraction pipeline that includes text filtering, open information extraction, Amazon Mechanical Turk annotations, and precision prediction to generate high precision triples. The elementary science domain-targeted vocabulary of a 4^{th} grader ($\approx$ 10 year old child) augmented with additional science terms from 4^{th} grade science text is used to search Web via the Bing[12] search engine to provide sentences for knowledge extraction. OpenIE [134, 135], an open information extraction system, is applied to the extracted sentences to generate an initial set of tuples. The tuples are then processed to generate a single head word, refined and scored using workers from Amazon Mechanical Turk, and are used to generate phrasal head and tail entities. The tuples are then applied with the schema mapping rules to make generalisations among seemingly disparate tuples explicit in the KB. The final version of Aristo Tuple v4 contains 294,000 domain-targeted tuples connected with 955 unique relations.

On the Use of ConceptNet and Aristo Tuple KGs

Our aim in this thesis is to develop the NLI models that are robust, generalisable and grounded in real-world knowledge. One way to achieve this aim is to augment the natural language understanding and reasoning capabilities of NLI models via the incorporation of external knowledge. Further, as discussed in the context of NLI Definition 1.1 in the Chapter 1, that for an NLI system to succeed, it must address the full complexity of linguistic as well as real-world commonsense knowledge.

As ConceptNet contains linguistic relations such as *Synonym, Antonym, IsA (hyponym), Partof (meronym)* as well as commonsense relations such as *UsedFor, CapableOf, MotivatedByGoal* (refer Section 2.3.2), it is an ideal external knowledge source to be utilised to incorporate external knowledge. The ConceptNet KG in addition to fulfilling the linguistic and commonsense knowledge requirement of NLI also suits our aim of developing NLI models that are grounded in real-world knowledge. The other external knowledge sources highlighted in Table 2.4 are either focussed

[12] `https://www.bing.com/` – As on August 9, 2020.

on lexical knowledge (WordNet) or concentrates more on the named entities records (DBPedia) [64].

For the SciTail dataset, in addition to ConceptNet, we employ science domain-specific KG Aristo Tuple to enrich the NLI model with domain-specific knowledge. To the best of our knowledge, Aristo Tuple is the only domain-targeted KG that contains science domain-specific facts relevant to the premise-hypothesis of the SciTail dataset; and hence is the preferred option for domain-specific external knowledge.

In the next section, we discuss the performance metric used to evaluate the deep neural NLI models.

2.4 Evaluation Criteria

Accuracy (ACC) As the datasets in the NLI domain are nearly balanced, accuracy is the performance metric used to evaluate the deep-learning-based NLI models. The accuracy is defined as the number of correctly predicted samples over the total number of predictions, which can be calculated in terms of positives and negatives as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP are True Positives, TN are True Negatives, FP are False Positives and FN are False Negatives samples in the in the model predictions. In the next section, we present a taxonomy of deep neural NLI models and comprehensively review the existing literature under the presented taxonomy.

2.5 Deep Learning Models for NLI

This section reviews the current deep learning-based NLI literature by taxonomising the field into sentence encoding-based and joint sentence encoding-based models. The taxonomy is presented in the Figure 2.3. The models are further classified according to the encoder architectures. The model architectures of sentence encoding-based and joint sentence encoding-based models can be derived from the generic

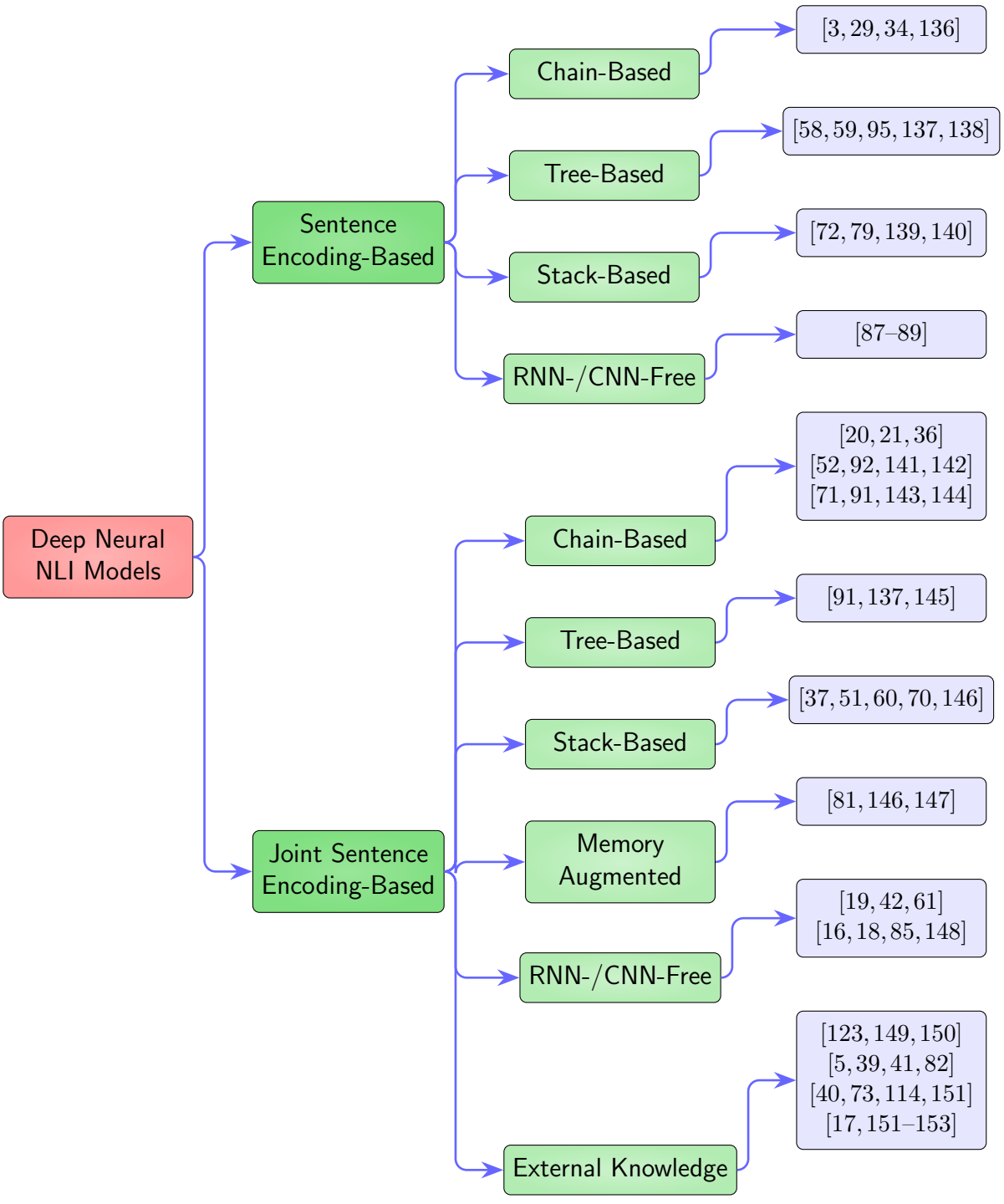


Figure 2.3: Taxonomy of Deep Neural NLI Models.

NLI model discussed in Section 2.2. Individual layers perform the designated function as detailed in the generic NLI model architecture (Section 2.2), however, the layers differ in the manner the specific function is performed. For example, in each of the NLI model, the encoding layer (Section 2.2.2) learns the word representation with respect to other words in context, however to learn the representations various encoders such as LSTMs [154], BiLSTMs [54], GRUs [75], or CNNs [76] are used.

We further categorise the literature based on the structural organisation of encoders (refer Figure 2.3). The encoders in NLI literature can be organised in different structures, namely chain, tree, stack, RNN-/CNN-free (attention based), memory augmented and encoders incorporating external knowledge. Note that the deep neural models share a high degree of commonality, it is not feasible to outline a mutually exclusive classification. We place and discuss the models based on their most salient features. For example, the shortcut-stacked encoder [79] proposed by Nie and Bansal, uses chain-structured BiLSTMs encoders stacked on top of each other with shortcut connections. We place and discuss this model under stack-based encoders as stacking the BiLSTMs with shortcut connection is the distinguishing feature of the model. Following, we discuss sentence encoding-based (Section 2.5.1) and joint sentence encoding-based models (Section 2.5.2) sub-categorised according to their encoder structures.

2.5.1 Sentence Encoding-based Models

The sentence encoding-based models follow the "Siamese" [155] architecture of neural networks. A Siamese neural network shares the layer weights while training on different input vectors to compute comparable output vectors. Each of the corresponding output feature vectors captures the important characteristics of the input and can be used to classify the relationship between the inputs.

Figure 2.4 illustrates the layered architecture of the sentence encoding-based model. The layers in this category of the model share the weights while processing the premise and hypothesis. The individually generated sentence embeddings are matched (refer Section 2.2.7), and input to the classifier layer for relationship identification. We discuss the various sentence encoding-based models below.

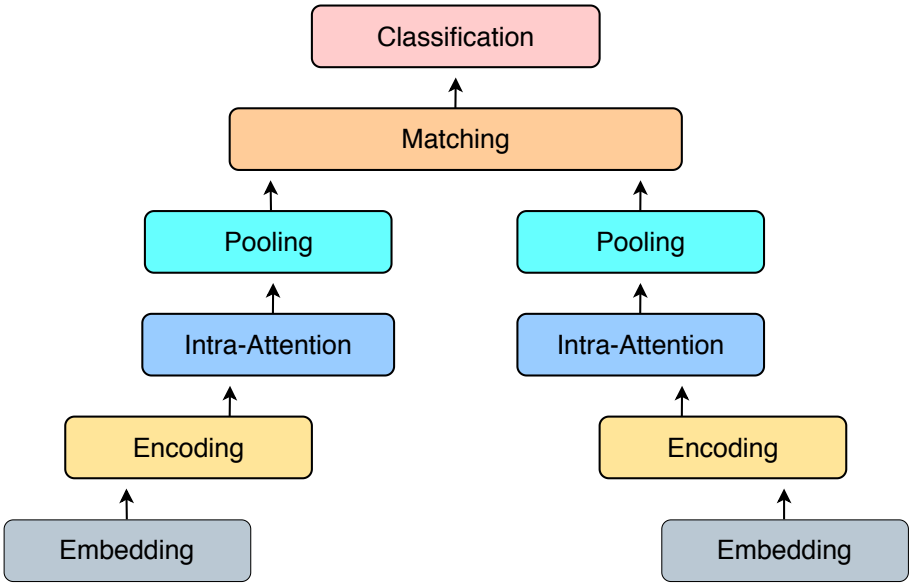


Figure 2.4: The layered architecture of the sentence encoding-based NLI models.

Chain-based Encoder Architectures

The LSTM RNN based sequential deep neural network model for machine translation in [156] has successfully demonstrated the ability of sequential architectures to learn effective sentence embeddings. The sentence encoding-based NLI models widely employ RNNs (LSTMs, BiLSTMs and GRUs) for generating sentence embedding because of their ability to maintain an internal memory to remember information from previous time steps [157].

Bowman et.al. [3] were the first to apply the LSTM RNN encoders for embedding premise and hypothesis and have achieved comparable results with the state-of-the-art feature-based classifier model (ACC: LSTM RNN: 77.6%, Feature-based Classifier: 78.2%). A smaller difference (7.2%) between the model's training and testing accuracy indicates the ability of LSTMs to learn the semantic meaning of the input sequence rather than memorising the training examples. This pioneering work led the way for RNNs to be used as encoders in NLI models and set the baseline for NLI models.

LSTMs process the input sequence left to right (i.e. forward direction), meaning that they consider only the previous context and completely avoid the future context

[158]. Bidirectional LSTMs [54] use the previous and future contexts by processing the sequence in the forward as well as reverse (right to left) direction.

To exploit this advantage of BiLSTMs, authors in [34] have used BiLSTMs to generate sentence embeddings for premise and hypothesis. A two-stage process was used to generate the sentence embedding. First, the sequence is input to the BiLSTM encoding layer. The output of the BiLSTM layer is averaged to generate a sentence embedding. Second, the sentence embedding, generated in the first stage is used to self-attend (Section 2.2.3) the words of the same input sequence to generate a more representative final sentence embedding. The authors referred to the self-attention mechanisms as "inner-attention" as it is applied within the sentence. The visualizations of inner-attention weights show that the attention mechanism helped the model in generating accurate sentence representation by placing more emphasis on the content words (nouns, verbs and adjectives) than the function words (prepositions, auxiliary verbs, conjunctions, grammatical articles). The model outperformed the previous [3, 58, 95] sentence encoding-based models by achieving an accuracy of 84.2% on the SNLI dataset.

A universal/general-purpose sentence encoder is trained on a large text corpus and subsequently used as an encoder for other related tasks with no or minimal task-specific fine-tuning. Most universal sentence encoders, such as BERT [16], SkipThought [32] or FastSent [159] consider learning in an unsupervised manner which is computationally expensive and parameter intensive. Conneau et al. [29] explore the use of the supervised NLI task to learn universal sentence embeddings trained on the SNLI dataset [3]. The authors empirically evaluate multiple different sentence embedding architectures ranging from standard LSTMs and GRUs with mean and max pooling, self-attention based networks to hierarchical CNNs. The results show that among these models, the BiLSTM architecture with max pooling, trained on the SNLI dataset (ACC: 84.5%) outperforms the SkipThought and FastSent on a group of complex transfer tasks such as binary and multi-class classification, NLI and semantic relatedness, and semantic textual similarity.

Kiela et al. [136] explore the supervised learning of task-specific, Dynamic Meta-Embeddings (DME), and apply the technique to the BiLSTM-Max sentence encoder

proposed by Conneau et al. [29]. Meta-embeddings aim to combine diverse pre-trained embeddings each trained using different methods and resources, to yield an embedding set with improved overall quality [160, 161].

In DME, the network learns which embedding to prefer from the available multiple types of embeddings by learning a weight for each type of the embedding. DME achieves this by projecting different embeddings for a word into a common d^l-dimensional embedding space with a liner projection layer. The projected embeddings are combined as a weighted sum of their attention weights. The attention weights are learned as a dot product of a learnable parameter ($\mathbf{a}$) and projected embeddings. The dot product is normalised with softmax function. Authors train the BiLSTM-Max encoder [29] with DME using two embedding types: FastText [162] [68] and GloVe [67]. The results on the SNLI dataset demonstrate that the BiLSTM-Max encoder with DME (ACC: 86.7%) outperforms encoders that have only FastText (ACC: 85.4%) or GloVe embedding (ACC: 85.5%). Further, the experiment with six different embedding types has shown to improve performance.

Tree-based Encoder Architectures

Tree-structured neural networks (also known as recursive neural networks) learn the sentence representation incrementally following the hierarchical structure (for example parse tree) of the input sentence. The learned representation captures the syntactic and compositional-semantic information of the input sentence [163].

Chain-structured RNNs, when compared to tree-structured neural networks, have poor generalisation capabilities over the unseen natural language texts [164]. Further, the meaning of linguistic expressions is known to be constructed recursively according to a tree structure [165]. Hence, the tree-structured neural network is a natural candidate for generating the sentence embeddings as it can exploit the hierarchical structure of linguistic expressions [166, 167].

Mou et al. [95] explore a Tree-Based CNN (TBCNN) [168] sentence encoder to leverage the benefits of tree-structured neural networks. The authors further propose various matching heuristics (refer Section 2.2.7) such as concatenation and element-wise product/difference. The basic idea of the TBCNN sentence encoder is

to learn feature maps by sliding subtree feature detectors individually over the dependency parse tree of the premise and hypothesis. A max-pooling layer generates the sentence embedding by aggregating the information from different feature maps learned along the tree. Finally, the sentence embeddings of the premise and hypothesis are combined by the matching layer employing the proposed heuristics, and the matched vector is input to an MLP classifier. The ablation analysis of matching heuristics on the SNLI dataset suggests that TBCNN achieves the maximum accuracy of 82.1%, when the concatenation of all the proposed matching heuristics is input to the classification layer.

A disadvantage of tree-structured neural networks is that they require a unique parsing structure for each sentence, and hence are not suitable for batched computation [58]. Moreover, the tree-structured neural network relies on external syntactic parsers to produce a parse tree to operate upon. The inability to be batch processed and dependency on external parsers slows and complicates the processing of tree-structured models during training and testing [58].

Bowman et al. [58] proposed a novel Stack-augmented Parser-Interpreter Neural Network (SPINN) architecture to overcome the aforementioned limitations of tree-structured neural networks. The SPINN model linearises the shift-reduce parsing formalism of a tree-structured model. Reading the input sequence left to right, the shift-reduce parsing builds the parse tree of the input sequence in a bottom-up fashion. In contrast to the standard shift-reduce parsing which outputs the parse tree of the input sequence, the SPINN model generates the sentence embedding of the input sentence.

The evaluation of SPINN [58] on the SNLI dataset shows that the model outperforms the LSTM RNN model [3] by 2.6% (ACC: 83.2% vs 80.6%). The authors attributed the high accuracy of tree-structured SPINN model to the better a generalisation to the natural language expressions achieved by leveraging the syntactic and semantic structure of natural language in the parse tree.

Although the SPINN model has linearised the shift-reduce parsing formalism, it still requires a syntactic parse tree as its input which increases the complexity of processing and reduces its practical applicability. With all the added complexity

in linearising the parsing mechanism, the SPINN model performed only marginally better (+1.1%) than the previous TBCNN model (ACC: 82.1%) [95].

To mitigate the limitations of SPINN model [58], Munkhdalai and Yu [137] proposed Neural Tree Indexers (NTI). Unlike SPINN [58], that generates the representation of the input sequences with the help of a syntactic parse tree, NTI creates the input sequence representation by constructing a full n-ary tree of the input sequence in a bottom-up fashion. To effectively capture the long-term dependencies in premise-hypothesis pairs, the NTI model introduced a tree attention mechanism. This attention mechanism allows a parent node in the tree to visit over its children representations, assign importance weights to those representations and create an attention-weighted representation to be passed towards the root. The NTI model is evaluated for binary trees. Experimental results on the SNLI dataset demonstrate that NTI models achieve competitive performance (ACC: 83.4%) to SPINN model (ACC: 83.2%) without needing to parse input sequence.

Another work, Gumbel Tree-LSTM [59] focuses on dispensing with the parse-tree requirement of SPINN [58]. Gumbel Tree-LSTM is a tree-structured LSTM that learns to compose task-specific tree structure from the plain text. The model uses a composition query vector that measures the validity of a composition (set of words in the input sequence). The model recursively selects compositions until a single composition remains that represents the sentence embedding of the input sequence. Gumbel Tree-LSTM utilises the Gumbel-Softmax estimator [169] to sample compositions in the training phase. The Gumbel-Softmax estimator transforms the discrete sampling operation to be continuous and thus the model can be trained via the standard backpropagation algorithm. Evaluation results on the SNLI dataset shows that the Gumbel Tree-LSTM has fewer model parameters, trains in considerably less time and achieves (+2.8%) higher accuracy than SPINN model [58] (ACC: 86.0% vs 83.2%).

Yoon et al. [138] propose a new Dynamic Self-Attention mechanism (DSA) which in contrast to traditional self-attention mechanisms, allows the attention weights to be generated with a dynamic weight vector during inference. The dynamic attention weights allow the network to adapt to each input sentence and hence is more flexible

in adapting to inputs even after training. The DSA mechanism stacked on the CNN-based DenseNet [170] encoder, at the time of writing[13], holds the current state-of-the-art results[14] on the SNLI dataset for sentence encoding-based models with 87.4% accuracy.

Stack-based Encoder Architectures

Stack-based encoder architectures use multiple encoding layers stacked vertically, and generally with connections from the preceding encoding layers with the aim of preserving learned features from all the encoding layers.

Chen et al. [139] proposed a Gated-Attention based BiLSTM model (Gated-Att BiLSTM) with several improvements - First, they augmented the GloVe word embeddings [67] with the character composition embedding. The character composition embedding of a word is learned by feeding all characters of the word to a CNN with max-pooling [171]. Second, the authors used the stacked BiLSTMs with shortcut connections as an encoder. The shortcut connections concatenate word embeddings and input hidden states at each layer in the stacked BiLSTM except for the bottom layer. Third, they introduced the intra-sentence gated-attention, which at each time step learns the attention weight to enhance the hidden state of the top BiLSTM layer. The attention weight is learned from the l^2-norm normalised output of input, forget or output gates, hence the attention is called the gated-attention. Experimental results on the SNLI dataset shows that the model achieves the maximum accuracy of 85.5%, when the input gate is used for attention weight calculation.

Nie and Bansal [79] also utilise the stacked BiLSTMs and shortcut connections in their shortcut-stacked encoder. In contrast to the Gated-Att BiLSTM [139] which concatenates word embeddings to the hidden sates of each stacked BiLSTM, the shortcut-stacked encoder concatenates outputs of all previous layers, plus the original word embedding. The model uses pre-trained GloVe vectors [67] to initialize the word embeddings which are fine-tuned end-to-end via the NLI supervision. Similar to the BiLSTM architecture with max-pooling of [29], the model applies max-pooling

[13] As on August 9, 2020.

[14] SNLI leaderboard: `https://nlp.stanford.edu/projects/snli/` – As on August 9, 2020.

to the output of the final BiLSTM layer to generate the fixed-length sentence embedding of the premise and hypothesis. Experimental results with the SNLI dataset shows that the shortcut-stacked encoder model performs better than gated-attention BiLSTM [139] (ACC: 86.1% vs 85.5%).

Talman et al. [140] extend the BiLSTM-Max architecture introduced by Conneau et al. [29] with three BiLSTM layers. The BiLSTM layers are separate (i.e. do not share parameters) and are initialised with the hidden and memory states of the previous BiLSTM layer. To improve the BiLSTM layer's ability to remember input words, each layer is also fed with the input word embeddings. Due to such initialisation, the model acts as an iterative refinement architecture that reconsiders the input in each BiLSTM layer while being cognizant of the features learned from the previous layer. A max-pooling layer generates a fixed-length vector from the hidden state of each BiLSTM layer. The final sentence embedding is the concatenation of the vectors from each BiLSTM layer. The model on the SNLI dataset improves the test accuracy of BiLSTM-Max architecture of Conneau et al. [29] by 1.1% (ACC: 86.6% vs 84.5%).

Chen and Ling [72] highlight that pooling is a crucial component of a wide variety of sentence encoding-based models. The authors combine all the established high-performance yielding techniques from the previous literature to build a sentence encoder. First, similar to [139], concatenating the pre-trained word embeddings with the character composition embedding. Second, stacking BiLSTMs with shortcut connections to upper layers as is done in previous studies [139, 140]. Third, the multi-head self-attention similar to [78], and finally the fourth, creating the final input to classifier as using the matching heuristics of [95].

The salient feature of Chen and Ling's proposed model is the generalized pooling to transform the output of stacked BiLSTMs to a fixed-length sentence embedding. Specifically, the idea of generalised pooling is to generate the sentence embedding as a weighted sum of the BiLSTM hidden states and attention weight vectors rather than a single attention weight scalar. The attention weight vectors allow the model to control each feature point of the BiLSTM hidden states. To capture different aspects of the input sentence multiple sentence embeddings are learned. The final

sentence embedding is the concatenation of all the learned sentence embeddings. To encourage diversity across different learned aspects of the sentence embedding, the model use different penalization terms at every stage of generalised pooling. The model outperforms all the sentence encoding-based models it adopts the various techniques from by attaining an accuracy of 86.6%.

RNN-/CNN-Free Encoder Architectures

Although RNN and CNN architectures are firmly established as state of the art approaches for sequence modelling tasks, Vaswani et al. [78] argue that attention mechanisms are sufficient to model the long- and short-term dependencies of the sequential data. They argue that the inherent sequential nature of RNNs precludes parallelisation within training examples. The problem compounds at longer sequence lengths, as memory constraints limit batching across training examples. CNNs relax the sequential computation requirement however they struggle to model long-term dependencies in the sequential data [172].

To alleviate these shortcomings of RNN/CNN architectures, Vaswani et al. [78] proposed the Transformer model based solely on attention mechanisms, dispensing the recurrence and convolutions entirely. The Transformer model established new state-of-the-art[15] results for the WMT 2014 English-to-German and WMT 2014 English-to-French machine translation tasks [78].

Motivated by the success of the Transformer model, Shen et al. [87] designed a Directional Self-Attention Network (DiSAN), based solely on the proposed directional self-attention and multi-dimensional attention mechanisms. The light-weight RNN-/CNN-free network has more flexibility in modelling sequence lengths than RNNs/CNNs, and its computation is easily and significantly more parallelisable on existing GPU hardware frameworks. DiSAN applies the proposed, forward- and backward-directional self-attention mechanisms to the word embeddings of the input sequence, and concatenate the two outputs. The multi-dimensional attention generates the sentence embedding from the concatenated output of directional self-

[15] As on December 6, 2017.

attention mechanisms. Visualizations of attention weights of the input sequence show that the semantically important words such as nouns and verbs get high attention but stop words (am, is, are, etc.) do not, and, the words important to the semantics of the whole sequence globally receive high attention. DiSAN attains an accuracy of 85.6% on the SNLI dataset [87].

In the follow-up work, Shen et al. [88] study soft and hard attention mechanisms. Soft attention learns a probability distribution over all the input words of the sequence. The resulting probabilities reflect the importance of each word and are used as weights to generate the context-aware embeddings of the input sequence. Different to soft attention, hard attention focuses on selecting some of the most important input words, and entirely discarding others. The authors proposed a Reinforced Self-Attention (ReSA) [88] integrating soft and hard attention mechanisms. ReSA improved the accuracy on SNLI dataset by 0.7% compared to DiSAN model [87] (ACC: 86.3% vs 85.6%).

Im and Cho [89] argue that when learning the local dependencies, the distance between the words is an important feature, to help understand the context of the input sequence and; the DiSAN model [87] only considers directional information ignoring the distance between the words. Hence, the local dependency in DiSAN is not properly modelled, and the model fails to capture contextual information in long sentences. Inspired by this limitation, the authors incorporate a distance mask along with a directional mask (introduced by Shen et al. [87]) to the multi-head dot-product attention of Transformer model [78]. Visualization of the distance mask matrix on the longest sequence (length 57 words) of the SNLI dataset demonstrates that the distance mask helps the model to concentrate on local words around the reference word and hence learn the local dependency without losing the ability to capture global dependency. Evaluation result on the SNLI dataset shows that the model outperforms DiSAN [87] by 0.7% (ACC: 86.3% vs 85.6%).

2.5.2 Joint Sentence Encoding-Based Models

The generic architecture of joint sentence encoding-based models is illustrated in Figure 2.5. Unlike the sentence encoding-based models, the joint sentence encoding-

based models do not learn the sentence embedding in isolation from the other input sentence. There are interactions while learning the sentence embedding. Various attention mechanisms are proposed for interactions between the premise and hypothesis. The main idea is to reason over the individual words and phrases of the input premise-hypothesis pairs. These models benefit from not squeezing the whole semantics into a single vector which is inefficient due to loss of information during the encoding process [33]. This category of models is also know as "matching encoding-based", "matching-aggregation framework", "co-attention based", and "inter-sentence attention-based" models.

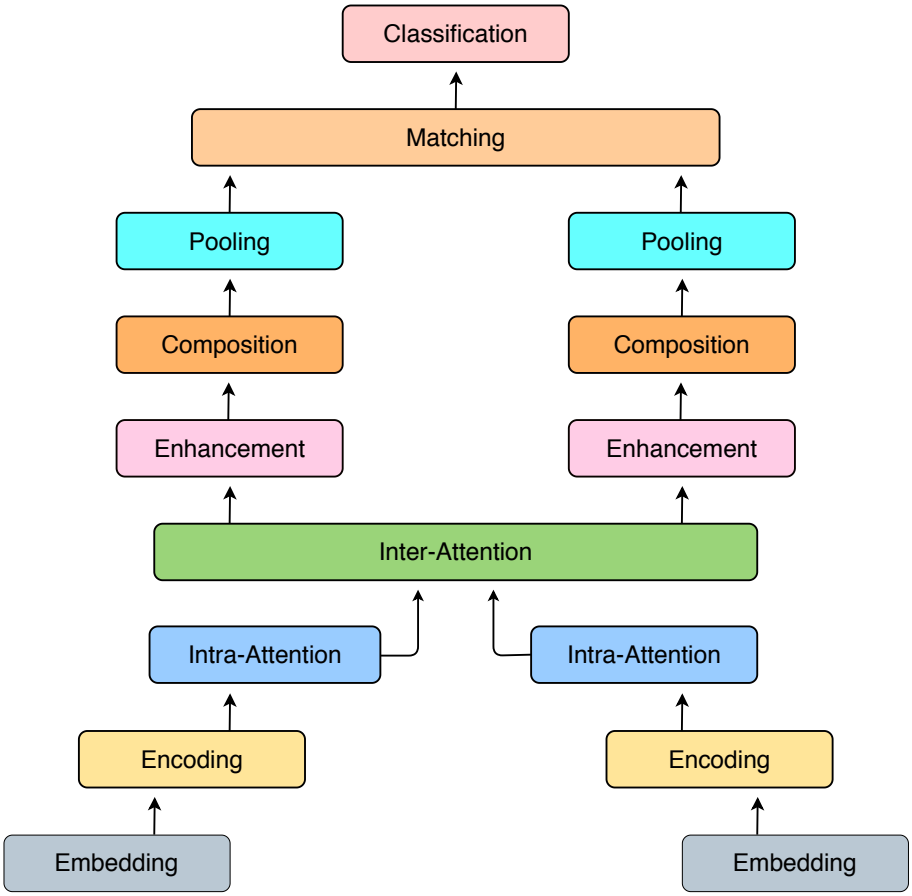


Figure 2.5: The layered architecture of joint sentence encoding-based NLI models.

We refer to them as joint sentence encoding-based models to emphasise that the sentence encoding is created by jointly considering the information from the premise and hypothesis. In the following sections, we discuss the joint sentence encoding-

based models in accordance with the proposed taxonomic structure of Section 2.5.

Chain-based Encoder Architectures

The pioneering work of Rocktäschel et al. [36] established the usefulness of attention mechanisms for NLI. Motivated by the success of attention mechanisms in machine translation [33] and reading comprehension [173] tasks, the authors propose to utilise word-by-word attention mechanisms in NLI. The model consists of two LSTMs which takes as input the premise and hypothesis. At each time step, the word-by-word attention mechanism allows the LSTM processing the hypothesis to attend to the hidden states of the premise LSTM to learn a sentence-pair representation. This word-by-word attention model is the first end-to-end neural model that outperformed the state-of-the-art hand-engineered feature-based model [3] without requiring any feature engineering on the input premise and hypothesis (ACC: 83.5% vs 78.2%).

Liu et al. [143] introduced Deep Fusion LSTMs (DF-LSTMs) to explore attention mechanism at word, phrase and sentence level to model the interactions between the premise and hypothesis pair. The interactions among the premise and hypothesis at the word, phrase and sentence level are referred to as the strong interaction. The strong interactions facilitated by the attention mechanism are employed to produce a highly intermingled sentence encoding by conditionally encoding the subsequences of different premise and hypothesis lengths. To cater to the need of remembering the subsequence interactions of different lengths, an external memory is used. Experiments demonstrate that DF-LSTMs yield an accuracy of 84.6% on the SNLI dataset.

Inspired by the word-by-word attention model of Rocktäschel et al. [36], Wang and Jiang [142] design a match-LSTM (mLSTM) to create a tightly coupled representation of premise-hypothesis. Similar to the word-by-word attention model [36], the model first creates an attention-weighted vector representation of the premise while processing the hypothesis word by word. Unlike [36], which considers only the attention-weighted representation of the premise, mLSTM creates the matching representation by also inputting the current hidden state of the hypothesis and

the attention-weighted representation of the premise. With this slight modification in producing the matching vector, mLSTM improves the performance of word-by-word attention model of Rocktäschel et al. [36] by 2.6% (ACC: 86.1% vs 83.5%). The qualitative analysis of word alignment of the premise and hypothesis suggests that the model can identify contradictions by remembering the mismatch in the content words of the premise and hypothesis.

Sha et al. [144] proposed a new variant of LSTM called re-read LSTM(rLSTM), which takes attention vector of first sentence as an rLSTM inner state, at the time of processing the second sentence. When applied to NLI, the rLSTM model uses the standard BiLSTM to read the premise and the proposed bidirectional rLSTM layer to read the hypothesis. To maintain the interaction between the premise and hypothesis, the rLSTM takes as its input the full hidden state vectors of the premise BiLSTM, the attention-weighted representation of the premise for each word of the hypothesis, and the hidden state and the memory cell state of the previous rLSTM time step. The average of bidirectional rLSTM outputs is used to predict the premise-hypothesis relationship. Experimental results show that rLSTM improves the performance on the SNLI dataset by an absolute improvement of 1.4% over the mLSTM model [142] (ACC: 87.5% vs 86.1%).

Wang et al. [52] introduced a Bilateral Multi-Perspective Matching (BiMPM) mechanism to match every time-step of the premise against all the time-steps of the hypothesis and vice versa. The authors design four matching strategies, namely full, max-pooling, attentive, and max-attentive strategies, to compare each time-step of one sentence against all the time-steps of the other sentence. Similarly, Ghaeini et al. [21] propose a Dependent Reading BiLSTM (DR-BiLSM) network to dependently read the premise and hypothesis at the time of BiLSTM encoding. The dependent reading mechanism, for example, when dependently encoding the premise, first encodes the hypothesis using a BiLSTM and then encodes the premise through a BiLSTM that is initialised with the hidden and memory cell state of the BiLSTM encoding the hypothesis. DR-BiLSTM improves the performance on the SNLI dataset by 1% when compared to BiMPM model (ACC: 88.5% vs 87.5%).

Tay et al. [141] also highlight the importance of attention directionality in the

designed Hermitian Co-Attention Recurrent Network (HCRN), which exploits the property of complex vector space where the complex-valued dot product is non-commutative and hence maintains the directionality in the attention dot product. Experimental result on the SciTail dataset shows that the model achieves an accuracy of 80.0%.

To utilise the effectiveness of different word-level attention mechanisms, Tan et al [92] introduced a Multiway Attention Network (MwAN) which employs multiple attention mechanisms to model premise-hypothesis pairs. Specifically, they use the word-by-word attention mechanism of Rocktäschel et al. [36], bilinear attention from Chen et al. [174], element-wise dot product attention, and element-wise difference attention mechanisms [142, 147]. The model achieves an accuracy of 88.3% demonstrating a significant improvement over the models employing single attention mechanism. Ablation results further show that removing any attention mechanism from MwAN decreases the model accuracy.

Tay et al. [20] presents a new ComProp Alignment-Factorised Encoder (CAFE) neural model by introducing the compare, compress, and propagate (ComProp) architecture for NLI. The key idea of the ComProp architecture is to learn a compressed alignment feature vector (i.e. an attention vector) for each word in the premise and hypothesis, concatenate it with the word embedding and propagate it to the upper encoding layers such as an RNN. Concatenation of the alignment feature vector enhances the representation ability of the words in the premise and hypothesis. The model yields an accuracy of 88.5% on the SNLI dataset.

Pan et al. [71] suggest that discourse markers such as "but" and "and" have deep connections with the intrinsic relation between the two sentences. These markers can be utilised to improve the performance of NLI models. They study a Discourse Marker Augmented Network (DMAN) which employs a pre-trained BiLSTM encoder. The BiLSTM encoders are first trained from scratch on the Discourse Marker Prediction (DMP) task [175]. The idea is to transfer knowledge learned from the DMP task to NLI to benefit from discourse markers. Transferring discourse markers knowledge to NLI yields 89.6% accuracy on the SNLI dataset.

Chen et al. [91] argue that the sequential inference models based on chain LSTM

architectures can achieve higher performance as compared to the previous top-performing models with complex architectures [137, 165]. They empirically demonstrated that the chain LSTMs with careful design can achieve high performance. In the first step, the model uses BiLSTM to encode the premise and hypothesis. In the second step, the network applies dot-product attention over the bidirectional sequential encoding of the premise and hypothesis. The model further enhances the local inference learned via attention mechanism by difference and element-wise product of corresponding hidden states learned in the first and second step. Finally, in the third step, the composition layer sequentially composes the local inference information using a BiLSTM layer. The model is called Enhanced Sequential Inference Model (ESIM). ESIM with an accuracy score of 88.0% is one of the established model in NLI research domain. ESIM is used as an underlying model in a number of subsequent NLI researches [38, 43, 60, 109]. We also utilise the ESIM as one of the underlying model for the models proposed in Chapter 4.

Tree-based Encoder Architectures

Chen et al. [91] investigate the effect of incorporating syntax over the syntactic parse tree structure of the premise and hypothesis. The Hybrid Inference Model (HIM) applies the Tree LSTMs [176] recursively to encode the premise and hypothesis over the parse tree produced by the Stanford PCFG parser [177]. Empirical results on the SNLI dataset suggest that incorporating syntactic information contributes to model performance. The model improved the performance of the ESIM [91] model by 0.6%. (ACC: 88.6% vs 88.0%).

Neural Tree Indexers (NTI) [137] (discussed in Section 2.5.1) proposed a global and tree attention mechanisms over the full n-ary tree of the premise and hypothesis. The global attention mechanisms, at every time step of encoding the hypothesis, attends over all the premise tree nodes whereas the tree attention mechanism attends only to the final learned representation of the premise. The authors study various models with different combination of tree composition function and attention mechanism. The model using the S-LSTM composition function with the global attention achieved the maximum accuracy (87.3%). In comparison to the to the well-known

tree-based SPINN model [58], the main advantage of the NTI model is that it does not require a syntactic parse tree as input. This reduces the model's complexity of operation and increases the practical applicability in natural language applications.

Yin et al. [145] proposed a DEISTE (Deep Explorations of Inter-Sentence interactions for Textual Entailment) model for the SciTail dataset, that learns to assign higher attention weights to the differentiating words in the output representation of a CNN encoder. The model also encode the position of the best aligned words in premise and hypothesis via a learned positional embedding. DEISTE achieves an accuracy of 84.7% on th SciTail dataset outperforming the ESIM [91], decomposable attention [42], and DGEM [5] models.

Stack-based Encoder Architectures

Extending the idea of strong interaction of Deep Fusion LSTMs [143] (refer Section 2.5.2), Liu et al. [37] proposed two deep neural network architecture with parallel but interdependent LSTMs: Loosely coupled-LSTMs (LC-LSTMs) and Tightly coupled-LSTMs (TC-LSTMs). LC-LSTM model explores the idea of creating a sentence representation by conditionally encoding sentences over one another. The TC-LSTM model further combines the hidden states and the memory cell states of two LSTMs for a stronger interaction. TC-LSTM model has achieved a better performance than LC-LSTMs (ACC: 85.1% vs 84.3%), suggesting that greater interaction among sentence pairs and more memory for understanding long-term dependencies can augment the reasoning power of deep neural networks.

Liu et al. [146] state that NLI is challenging since it requires the model to fully understand the lexical and compositional semantics and that the previous models suffer from using only a single-step inference process. To address the limitation, the authors present a Stochastic Answer Network (SAN) with stacked BiLSTM encoding layers to iteratively refine predictions over multiple reading passes of the premise and hypothesis. SAN utilises the scaled dot-product attention [78] at the inter-attention layer and maintains a memory state of the premise and hypothesis information via a BiLSTM composition layer. The answer module predicts the NLI class over the memory states of the premise and hypothesis. The prediction is refined iteratively

considering the previous time step prediction and the memory states. The iterative refining strategy of SAN achieves an accuracy of 88.5% on the SNLI dataset.

McCann et al. [70] train MT-LSTM (Machine Translation LSTM), a two-layer standard BiLSTM network with attention mechanism on the machine translation task with the aim of transferring the learned knowledge to downstream NLP tasks. Context Vectors (CoVe), the output of pre-trained MT-LSTM, transfers the learned knowledge to downstream NLP tasks. The authors further design a Biattentive Classification Network (BCN) to test the efficacy of CoVe to transfer. The input to the BCN is the concatenation of GloVe embeddings and CoVe vectors of each word in the input sequence. A BiLSTM layer with biattention [178] encodes the sequence. Biattention conditions each representation of the input sequences on the another to compute interdependent representations. The evaluation result of BCN with Cove on the SNLI development set shows that CoVe with GloVe achieves (ACC: 88.1%) higher performance than models that use only GloVe (ACC: 87.7%). The model attains 88.1% accuracy on the SNLI test set.

Peters et al. [60] further explore the use of pre-trained models for downstream NLP tasks by introducing deep contextualised word vectors. The deep contextualised word vectors are a linear combination of the intermediate layer representations of a BiLSTM trained with a bidirectional language model (biLM) objective on approximately 30 million sentences [179]. The learned word vectors are called Embeddings from Language Model (ELMo). Contextualised ELMo, representations when incorporated with the ESIM model of Chen et al. [91], improved the accuracy of the ESIM model by 0.7% on SNLI dataset (ACC: 88.7% vs 88.0%).

Kim et al. [51] study a Densely-connected Recurrent Co-attentive neural Network (DRCN), which consists of 5 recurrent layers to utilise the increased representational power of deep recurrent layers. DRCN utilizes shortcut-connections [170] to propagate the concatenated hidden states and the learned attentive features from all the preceding recurrent layers. To alleviate the problem of increasing feature vector dimensions, due to concatenation operation, an autoencoder is used to propagate only a fixed-length feature vector to the higher recurrent layers. Testing the model on the SNLI dataset yields 88.9% accuracy.

Memory Augmented Encoder Architectures

Memory augmented neural networks are an interesting prospect for developing models for natural language inference task because unlike RNNs (LSTMs, BiLSTMs, and GRUs) they are not limited for storing long-term dependencies of the input sequence and hence can generalise well on natural language understanding tasks. Memory Augmented Neural Turing Machines (NTM) [180], the deep neural network with a controller and a fixed-sized random-access memory have shown promising performance on copying and sorting the sequential data.

Neural Semantic Encoders [81] are an extension of NTM [180] with variable memory. NSE uses an attention mechanism to access readable and writable external shared memory. It can also address multiple shared memories simultaneously. NSE transforms the memory through the read, write and compose operations. The compose operation is a composition function which takes as input the memory slot read by the read operation. The write operation writes back the result of composition function to the appropriate memory location. Read, compose and write operations are neural networks and are fine-tuned during training.

NSE is evaluated on a wide variety of challenging natural language understanding tasks, for example, NLI, question answering, machine translation, document sentiment analysis and sentence classification. The NSE encoder model outperformed the previous sentence encoding-based models [3, 58, 95] by achieving an accuracy of 85.4% on the SNLI dataset.

Long Short-Term Memory-Networks (LSTMN) [147] model augments the LSTM unit with an internal memory tape. The memory tape stores the memory cell output for all the previous words of the input sequence read by the LSTMN. Like [36], two LSTMNs are used to read the premise and hypothesis separately. The matching vector is created by employing the word-by-word attention mechanism [36] over the contents of the memory tape of the LSTMN reading the premise. The memory tape improved the generalisation power of LSTMN while modelling long sequences. The experimental result demonstrates that LSTMN marginally outperforms NSE (ACC: 86.3% vs 85.4%).

Similar to the Stochastic Answer Network (SAN) [146], Liu et al. [181] propose to

use multi-turn inference over premise and hypothesis. To match the sentences from various perspectives, the designed Multi-turn Inference Matching Network (MIMN), matches the premise and hypothesis contextual representations to the corresponding aligned vectors using three neural matching functions - concatenation, subtraction, and multiplication. MIMN uses a memory component to store the inference information of the previous turns. At each turn, MIMN matches one matching feature with the historical inference information stored in the memory component. Ablation experiments on the SNLI dataset demonstrate that the memory component is crucial to MIMN performance and the maximum attained accuracy of 88.3% degrades when memory component is removed. The model achieves an accuracy of 84.0% on the SciTail dataset.

RNN-/CNN-Free Encoder Architectures

Before the advent of the Transformer [78] model (refer Section 2.5.1), the decompsable attention model introduced by Parikh et al. [42] demonstrated that attention mechanism can be effectively employed to overcome the disadvantages of sequential processing of RNNs.

Parikh et al. [42] introduced a decomposable attention model which relies solely on the attention mechanism [33], preventing the need of RNN/CNN encoding layers. The model is a simple three-step model consisting of the attend, compare and aggregate steps. The attend step aligns premise-hypothesis words by means of the attention mechanism and decomposes the aligned word-pairs into smaller problems for comparison. The compare step compares each aligned word-pairs identified in the attend step. The comparison is done by running the concatenated vector representations of the aligned word-pairs through a feed-forward neural network. The network outputs comparison vectors which are a non-linear combination of aligned word-pairs. The final aggregate step sums up the comparison vectors and produces the final classification label with another feed-forward neural network.

The high performance (ACC: 86.8%) of the decomposable attention model in comparison to previous complex recurrence based sentence encoding-based [34, 58, 137] and joint sentence encoding-based models [36,143] demonstrated that attention

mechanism can be effectively employed to overcome the disadvantages of sequential processing of RNNs.

Gong et al. [19] introduce a general framework, Interactive Inference Network (IIN) of modelling the sentence pair by hierarchically extracting the semantic features from interaction space (joint representation of the premise and hypothesis). The authors study a Densely Interactive Inference Network (DIIN), an instantiation of IIN, which contains highway network [77] with self-attention as sentence encoder, a dot product cross-attention mechanism to create the joint representation of the premise and hypothesis (interaction space), and DenseNet [170] as a feature extractor from the joint representation. Experimental result shows that DINN achieves an accuracy of 88.0% on the SNLI dataset.

Guo et al. [148] argue that the self attention mechanism of the Transformer model treats the words at various distances to a central word equally which hinders the capacity of the Transformer model to capture local structures in input sequence. To mitigate this shortcoming of the Transformer model, the authors proposed a Gaussian Transformer which introduces a Gaussian prior to self attention mechanism of the Transformer model which emphasises words adjacent to a central word more than the distant words. The Gaussian Transformer achieved an accuracy of 89.2% on the SNLI dataset.

Radford et al. [61] pre-trains the Transformer model [78] on BooksCorpus [182] with the language modelling task in an unsupervised manner. The pre-trained Transformer is then fine-tuned on different downstream supervised tasks such as NLI, question answering, commonsense reasoning, and semantic similarity, improving the state-of-the-art on 9 of the 12 datasets the authors study. On the SNLI and SciTail datasets the model achieved the accuracies of 89.9% 88.3% respectively. This model developed by OpenAI[16] utilises the generative pre-training of the language model is popularly known as OpenAI GPT[17] model for Generative Pre-trained Transformer. OpenAI further released the GPT-2[18] [84] and GPT-3 [85] by scaling-

[16] `https://openai.com/` – As on August 9, 2020.
[17] `https://openai.com/blog/language-unsupervised/` – As on August 9, 2020.
[18] `https://openai.com/blog/better-language-models/` – As on August 9, 2020.

up the Transformer encoding layers in the original GPT model.

Liu et al. [18] present a Multi-Task Deep Neural Network (MT-DNN) for modelling sequences across multiple natural language understanding tasks. MT-DNN incorporates bidirectional Transformer model as a shared encoder in a multi-task learning framework [183]. Experimental results show that MT-DNN achieve the accuracies of 91.5% and 98.7% on the SNLI and SciTail datasets. The representations learned by MT-DNN have exceptional generalisation capability attributed to multi-task learning.

Devlin et al. [16] introduced the state-of-the-art Bidirectional Encoder Representations from Transformers (BERT), which unlike Peters et al. [60] and Radford et al. [61], pre-train deep bidirectional representations from unlabelled text by jointly conditioning on both left and right context in all layers. During pre-training the underlying Transformer model [78] is trained on two unsupervised tasks, masked language model and next sentence prediction with BooksCorpus (800M words) [182] and English Wikipedia (2,500M words).

Recently, BERT has become the model of choice for complex natural language understanding tasks such as, question answering [184], NLI [17, 18, 152], sentiment classification [185], and the tasks in GLUE [186] and Super GLUE [187] benchmarks. The pre-trained BERT models are publicly available[19] in various sizes, including the original $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$ to be use as encoders or to be fine-tuned for downstream NLP tasks.

External Knowledge Augumented Encoder Architectures

Leveraging external knowledge in natural language inference systems has long been proposed [188], however, neural NLI models have only recently started utilising external knowledge to augment the generalisation and reasoning capabilities of the NLI models. The pioneering work of Annervaz et al. [82] infuse real-world knowledge into the deep neural NLI model. The CNN-based model learns the features from the entities and relations of the WordNet [115] and Freebase [117] databases. The

[19] `https://github.com/google-research/bert` – As on August 9, 2020.

learned features are concatenated with the context representations of the input sequences obtained using an LSTM layer. Experimental results demonstrate that the external knowledge augmented LSTM model is able to outperform the vanilla LSTM trained on the full data, with training on only 70% of the data. This suggests that the NLI model gains information valuable for reasoning inference from the external knowledge. The NLI model achieved the maximum accuracy of 73.10% on the SNLI dataset when the external knowledge is incorporated from the WordNet KG.

Chen et al. [41], introduce a Knowledge-based Inference Model (KIM), that incorporates lexical-level semantic knowledge into the attention and composition components of the model. Specifically, small number of external lexical features (such as synonym and antonym) extracted from the lexical database, WordNet [115] is used to form relation embeddings between the premise and hypothesis words. The KIM model requires substantial NLI model adaptations to incorporate external knowledge with marginal performance improvements 0.1% (ACC: 88.6%) over state-of-the-art CAFE model (ACC: 88.5%) [20]. Further, the model is inflexible to incorporate different external knowledge sources and NLI models.

Kang et al. [39] design AdvEntuRe, a framework to train the decomposable attention model [42] with adversarial training examples generated by incorporating knowledge from linguistic resources such as WordNet, and with a sequence-to-sequence neural generator. The experimental result on the SNLI (ACC: 84.7%) and SciTail (ACC: 79.0%) shows that the decomposable attention model is more robust and achieves better performance when trained with knowledge-guided adversarial examples.

The follow-up work, NSnet [40] is a neural-symbolic NLI model, that integrates deep learning approach with the symbolic approach. The model decomposes each of the hypotheses into various facts and verifies each sub-fact against the premises using the decomposable attention model [42] and against the Aristo Tuple KB using a structured scorer. An aggregator network then combines the predictions from the two models to get the final inference class. The qualitative analysis of the SNLI test set examples suggest that the symbolic model assists the neural model in identifying the correct inference class. NSnet achieves an overall accuracy of

77.9% on the SciTail dataset. Weissenborn et al. [73] refine word embeddings by dynamically incorporating relevant background knowledge from the ConceptNet KG [1] and Wikipedia abstracts[20]. For NLI, when the word embeddings are refined from the additional ConceptNet background knowledge the BiLSTM-based NLI model showed an improved accuracy of 86.5% when compared to unrefined BiLSTM model (ACC: 84.4%).

Wang et al. [114] present ConSeqNet, a system of a text-based and graph-based models for the SciTail dataset. The outputs of the two models are concatenated and are fed to the MLP layer for classification. For the text-based model, the authors employ mLSTM model proposed by Wang and Jiang [142] (refer Section 2.5.2). The graph-based model is constructed by mapping the premise-hypothesis phrases to the KGs. The sentence embeddings generated by the two models are concatenated, and input to the MLP layer (refer Section 2.2.8) for the inference class predictions. ConSeqNet achieves an accuracy of 85.2% on the SciTail dataset in conjunction with ConceptNet. Further, experiments on the development set of the SciTail dataset with the WordNet, DBpedia, and ConceptNet KGs achieved the accuracies of 87.6%, 87.3%, and 88.6% respectively. The results suggest the effectiveness of ConceptNet knowledge for the inference task. However, the model does not evaluate the model efficacy on the science domain-specific external knowledge from the science knowledge-based KG Aristo Tuple (refer Section 2.3.2). Further, the model do not generalise well to SNLI dataset and demonstrates a low accuracy of 83.3% with the ConceptNet KG [189].

KG-Augmented Entailment System (KES) [189] augment the NLI models with external knowledge encoded using graph convolutional networks. The knowledge subgraph is selected by mapping the premise and hypothesis words to the ConceptNet KG by max-substring match. KES achieved an accuracy of 85.56% on the SNLI dataset with decomposable attention model [42]. Khot et al. [5] proposed a Decomposed Graph Entailment Model (DGEM) for the SciTail dataset. The model validates the hypothesis graph with the tokens in the premise. Authors use the open

[20]Downloaded from `https://wiki.dbpedia.org/downloads-2016-10` – As on August 9, 2020.

information extraction tuples [113] for hypothesis graph representations. DGEM calculates the hypothesis graph node and edge probability of being supported by the words of premises. The final NLI class is predicted by the average of node and edge probabilities. On the SciTail dataset, DGEM attained an accuracy of 77.3%, significantly outperforming the ESIM [91] (ACC: 70.6%) and decomposable attention models [42] (ACC: 72.3%).

Li and Sethy [123] incorporate the external lexical knowledge from the WordNet [115] into the multi-head attention function of the BERT. The WordNnet's lexical knowledge is shown to improve the robustness of the BERT model on NLI, achieving an accuracy of 90.1% on the SNLI dataset. Yang et al. [150] also utilise the WordNet lexical knowledge to enhance the BERT representation. The external knowledge (hypernymy, hyponymy, co-hyponyms, antonymy, and synonymy) feature vector learned via a CNN and concatenated to the hidden BERT representations, improved the BERT performance.

Pang et al. [149] incorporate the syntactic information from the neural dependency parser in [190] to decomposable attention [42], ESIM [91], BERT [16] and MT-DNN [18] models. For both the SNLI and SciTail datasets, the decomposable attention (ACC: SNLI - 84.8%, SciTail - 78.2%), the ESIM (ACC: SNLI - 88.1%, SciTail - 81.3%), BERT (ACC: SNLI - 90.5%, SciTail - 92.8%), MT-DNN (ACC: SNLI - 91.1%, SciTail - 94.3) have shown performance improvements. Similarly, Wang et al. [153] proposed StructBERT, which extends the pre-training of the BERT model by introducing two auxiliary training objectives to leverage the language structure in contextualised representations. The first, word structural objective, which demands the model to reconstruct the right order of a certain number of intentionally shuffled words. Second, sentence structural objective, that requires the model to predict the correct order of two sentences. StructBERT marginally improved (by 0.2%) the performance of the model proposed by Pang et al. [149] on the SNLI dataset (ACC: 91.7% vs 91.5%).

Li et al. [151] conduct experiments on several state-of-the-art NLI models including BERT [16], OpenAI GPT [61] and have demonstrated that unsupervised pre-training and incorporating knowledge from external sources such as WordNet [115]

are complementary and enhance the performance of pre-trained NLI models. Zhang et al. [152] utilise the explicit semantic cues from the semantic role labelling tasks to help the NLI model better understand natural language. The learned semantic role (*who* did *what* to *whome, when* and *why*) embeddings are concatenated to the embedding of the each input word of the downstream task. For NLI, the added semantic role labels improve the performance of the ESIM model from Peters et al. [60] (ACC: 89.1% vs 88.4%), $\text{BERT}_{\text{BASE}}$ [16] (ACC: 89.6% vs 89.2%) and $\text{BERT}_{\text{LARGE}}$ [16] (ACC: 91.3% vs 90.4%) models. Further, the authors extend the idea to BERT model and study a Semantics-aware BERT (SemBERT) [17] model. SemBERT appends the semantic role label embeddings learned from out-of-shelf semantic role labeler [60] to each input word to generate a semantic aware sentence embedding. Incorporating semantic role labels into the contextual word representations improved the performance of the $\text{BERT}_{\text{BASE}}$ model from 90.7% to 91.0% and $\text{BERT}_{\text{LARGE}}$ model performance from 91.1% to 91.6% on the SNLI dataset.

2.6 Conclusions

In this chapter, we introduced a generic neural NLI architecture. The layered architecture consists of the embedding, encoding, intra-attention, inter-attention, enhancement, composition, pooling, matching, and output MLP layers and an external knowledge source component. Different deep neural NLI models proposed in the research literature can be derived from the presented generic architecture. We highlighted the different datasets available for NLI model evaluation and elaborate on the SNLI and SciTail datasets that we utilise to evaluate the NLI models we develop in this thesis. Further, we tabulate the different external knowledge sources utilised in the NLI domain, and, elaborated on the the ConceptNet and Aristo Tuple KGs. We employ ConceptNet and Aristo Tuple KGs to incorporate linguistic, general commonsense knowledge and science domain-specific knowledge into the deep neural NLI models that we propose in this thesis.

We presented a taxonomy of the existing NLI literature by categorising the field into sentence encoding-based models and joint sentence encoding-based models. The

models are further categorised based on the architecture of the encoders they employ. Consequently, we comprehensively reviewed the NLI literature under the presented taxonomic structure.

In the following chapters, we present several neural NLI models to address the limitations of the existing literature discussed in Section 1.2. The next chapter focuses on combining the intra-attention and inter-attention mechanisms in a combined attention model to maximally utilise the two attention mechanisms.

CHAPTER 3

CAM: A Combined Attention Model for Natural Language Inference

3.1 Introduction

Traditional approaches to NLI range from machine learning-based [191], lexical and semantic similarity-based [192, 193], to the methods that extracts structured information such as discourse commitments [194] and predicate-argument [195]. Formal reasoning [196] and natural logic [24] methods are also applied to NLI. However, traditional approaches require extensive feature engineering. Moreover, these approaches do not generalise well because of the complexity and domain dependence nature of the feature engineering task [95, 144].

Machine learning has been a dominant approach to NLI [13]. However, the machine learning research for NLI is severely limited in performance by the lack of gold-standard premise-hypothesis pairs [3]. The field has renewed prosperity by the recent introduction of big datasets such as SNLI [3] and SciTail [5]. The public availability of these big datasets has made it feasible to train complex neural network models for NLI. Recurrent Neural Networks (RNNs), particularly bidirectional LSTMs (BiLSTMs) [53] in combination with attention mechanisms [33] have shown

state-of-the-art results on the SNLI dataset [91].

Attention mechanisms have shown promising performance for complex natural language understanding sequence modelling tasks such as machine translation [78, 197], dialogue generation [198], machine comprehension [199] and natural language inference [200]. Attention mechanisms allow the RNNs to automatically search for the most relevant parts of an input sequence and assign importance weights to those parts. These weights are used for creating the attention-weighted representation of the input sequence [33].

As mentioned in the context of Interaction Layer (Section 2.2.3), the two broad categories of attention mechanisms in the research literature are intra-attention and inter-attention.

The intra-attention mechanism, also known as self-attention [35], involves applying attention to the input sentence itself. During training, the model learns to assign a higher weight to those parts of the input sentence which are important to its semantics. The attention-weighted sentence representations thus generated also capture the global context of the sentence [34].

In inter-attention mechanism, attention is applied between the input sentences. The attention-weighted sentence representation of one sentence is generated based on the contents of another sentence. In the sentence representation, the information that is important with respect to other sentences is assigned higher weights.

As discussed in Chapter 2, the attention mechanism is an essential component of the models achieving state-of-the-art performance on the NLI task [200]. However, the current models that employ intra-attention [34, 87] do not utilize information from another sentence. The models utilizing inter-attention [36, 37] do not exploit the contexts in the individual sentences.

This chapter presents a Combined Attention Model (CAM), which employs intra-attention in conjunction with inter-attention to utilise the benefits of both the mechanisms. The model first captures the semantics of the individual input premise and hypothesis with intra-attention and then aligns the premise and hypothesis with inter-sentence attention to learn cross sentence dependencies.

The rest of this chapter is organised as follows. In Section 3.2, we present the

layer-by-layer description of the proposed, CAM model. In Section 3.3, we explore different experiments on CAM and discuss the results. In Sections 3.3.3 and 3.3.4, we discuss the quantitative results on the SNLI and SciTail datasets respectively. In Section 3.3.5, we investigate the ablation analysis on CAM, followed by the fine-grained accuracy analysis of the CAM and the ablated models in Section 3.3.6. In Section 3.3.7, we discuss the efficacy of the model on the varying lengths of premise-hypothesis pair. Qualitative analysis is studied in Section 3.4. Finally, the conclusions of the chapter is presented in Section 3.5.

3.2 Proposed Model: CAM

The proposed model combines intra-attention and inter-attention for modelling the interactions between the premise and hypothesis. Figure 3.1 illustrate the high-level view of the proposed CAM model. The layered architecture is composed of the following main layers: input embedding, encoding, intra-attention, projection, inter-attention, enhancement, pooling, matching and classification.

Given a sequence of premise $P^{emb} = (\mathbf{p}_1, \ldots, \mathbf{p}_n)$ and hypothesis $H^{emb} = (\mathbf{h}_1, \ldots, b_m)$ with lengths n and m respectively. Each $\mathbf{p}_i$, $\mathbf{h}_j \in \mathbb{R}^d$, is a word embedding of d-dimensional, which can be initialized with pre-trained embedding vectors, such as Glove [67] or Word2Vec [201] (refer Section 2.2.1).

3.2.1 Input Encoding Layer

We utilize BiLSTMs [54] to encode the input premise and hypothesis sentences. The BiLSTM processes the input sequence in the forward and backward directions to incorporate contextual information at each time step of processing the input sequence. The hidden state output at any time step is the concatenation of forward and backward hidden states. The representations $\bar{\mathbf{p}} \in \mathbb{R}^{n \times k}$ and $\bar{\mathbf{h}} \in \mathbb{R}^{m \times k}$ in the Equations (3.1) and (3.2) respectively, represents the k-dimensional encoded representation for each word in the premise and hypothesis respectively. Where k

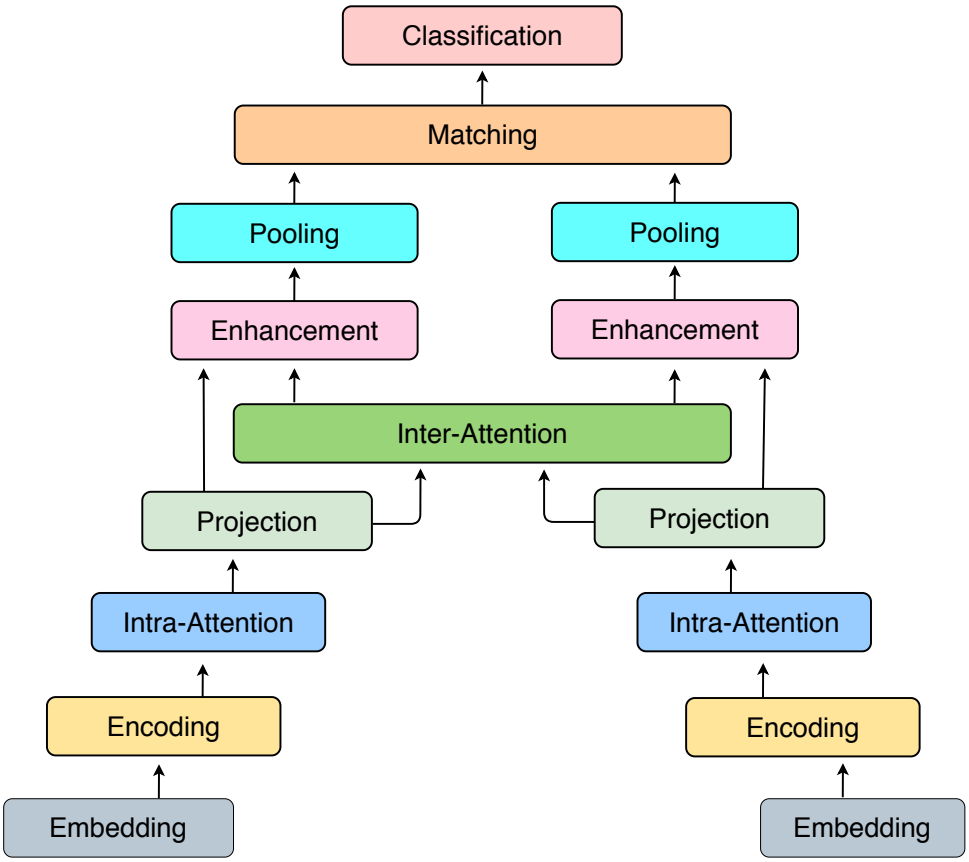


Figure 3.1: A high level view of our Combined Attention Model (CAM).

is the dimension of hidden states of the BiLSTM layer.

$$\bar{\mathbf{p}}_i = \text{BiLSTM}(\mathbf{p}, i), \quad \forall i \in [1, \ldots, n] \tag{3.1}$$

$$\bar{\mathbf{h}}_j = \text{BiLSTM}(\mathbf{h}, j), \quad \forall j \in [1, \ldots, m] \tag{3.2}$$

3.2.2 Intra-Attention Layer

This layer applies intra-attention [34] to the premise and hypothesis sentences individually. Through attention weights, the intra-attention layer emphasizes the words important to the semantics of the input sentence. The attention-weighted sentence representation thus generated represents a more accurate and focused sentence representation of the input sentence. The attention-weighted sentence representation

is generated as follows:

$$M = \tanh(W^y Y + W^h \mathbf{r}^{avg} \otimes \mathbf{e}^L) \tag{3.3}$$

$$\alpha = \text{softmax}(\mathbf{w}^T M) \tag{3.4}$$

$$\mathbf{r} = Y\alpha^T \tag{3.5}$$

where $W^y, W^h \in \mathbb{R}^{k \times k}$ are trained projection matrices, $Y \in \mathbb{R}^{k \times L}$ is the matrix of hidden output vectors of the BiLSTM layer, $\mathbf{r}^{avg} \in \mathbb{R}^k$ is obtained from the average pooling of Y, $\mathbf{e}^L \in \mathbb{R}^L$ is a vector of 1s, $\mathbf{w} \in \mathbb{R}^k$ is a learned parameter vector and $\mathbf{w}^T$ is its transpose, $\alpha \in \mathbb{R}^L$ is a vector of attention weights and $\mathbf{r}$ is the attention-weighted sentence representation. The attention-weighted sentence representation is generated for the premise and hypothesis and is projected with a standard projection layer with ReLU activation to generate P^{intra_atten} and H^{intra_atten} matrices respectively.

3.2.3 Inter-Attention Layer

The inter-attention layer uses the soft attention [88, 91] mechanism to associate the relevant sub-components between the attention-weighted representations of the premise and hypothesis. The inter-attention layer, first, computes the unnormalized attention weights as the similarity of hidden states of the intra-attention-weighted representations premise and hypothesis following the Equation (3.6).

$$e_{ij} = \tilde{\mathbf{p}}_i^T \tilde{\mathbf{h}}_j \tag{3.6}$$

where $e_{ij} \in E \in \mathbb{R}^{n \times m}$ and $\tilde{\mathbf{p}}_i \in P^{intra\_atten}$ and $\tilde{\mathbf{h}}_j \in H^{intra\_atten}$ are the intra-attention-weighted representation of the i-th and j-th word of the premise and hypothesis respectively.

Next, for each word in the intra-attention-weighted representation of the premise, P^{intra_atten}, the relevant semantics in intra-attention-weighted representation of the hypothesis is identified and composed using e_{ij}, more specifically the Equation (3.7) details this procedure.

$$\hat{\mathbf{p}}_i = \sum_{j=1}^{m} \frac{\exp(e_{ij})}{\sum_{k=1}^{m} \exp(e_{ik})} \tilde{\mathbf{h}}_j, \quad \forall i \in [1, \ldots, n] \tag{3.7}$$

$$\hat{\mathbf{h}}_j = \sum_{i=1}^{n} \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \tilde{\mathbf{p}}_i, \quad \forall j \in [1, \ldots, m] \tag{3.8}$$

where $\hat{\mathbf{p}}_i$ is a weighted summation of the representations in H^{intra_atten}. Intuitively, the representations in H^{intra_atten} that is relevant to $\tilde{\mathbf{p}}_i$ is selected and represented as $\hat{\mathbf{p}}_i$. Similarly, the same is carried out for each of the intra-attention-weighted representation, $(\tilde{\mathbf{h}}_j)$, of the hypothesis following the Equation (3.8). We further enhance the similarities [93] between the intra-attention-weighted representation of the premise and hypothesis and the local inference information learned by inter-attention, through the element-wise multiplication of the corresponding representations of intra-attention and inter-attention layer as

$$\mathbf{f}_p = \tilde{\mathbf{p}} \odot \hat{\mathbf{p}} \qquad \mathbf{f}_h = \tilde{\mathbf{h}} \odot \hat{\mathbf{h}} \tag{3.9}$$

For the enhancement, we also considered the element-wise difference [91, 93], however that did not further improved the model performance.

3.2.4 Pooling Layer

To facilitate the classification of the relationship between the premise and hypothesis, a relation vector is formed from the average and max pooling of the encoding of the premise and hypothesis representations generated previously by inter-attention layer in the Equations (3.9). Pooling is performed following the Equations (3.10) and (3.11).

$$\mathbf{v}^{p,avg} = \text{average}\{f_p, i\}_{i=1}^{n} \qquad \mathbf{v}^{p,max} = \max \{f_p, i\}_{i=1}^{n} \tag{3.10}$$

$$\mathbf{v}^{h,avg} = \text{average}\{f_h, j\}_{j=1}^{m} \qquad \mathbf{v}^{h,max} = \max\{f_h, j\}_{j=1}^{m} \tag{3.11}$$

where $\mathbf{v}^{p,avg}$ and $\mathbf{v}^{p,max}$ represents the fixed length vector for premise sentences resulting from the average and max pooling over $\{f_p, i\}_{i=1}^{n}$. Similarly, the fixed

length representations is generated for hypothesis according to Equation (3.11).

3.2.5 Matching and Classification Layer

To classify the relationship between the premise and hypothesis, we create a matching vector by the concatenation of max- and average-pooled vectors obtained from the Equations (3.10) and (3.11). Specifically, the matching vector is composed as in Equation (3.12).

$$\mathbf{f}^{relation} = [\mathbf{v}^{p,avg}; \mathbf{v}^{p,max}; \mathbf{v}^{h,avg}; \mathbf{v}^{h,max}] \tag{3.12}$$

The matching vector is input to the a Multilayer Perceptron (MLP) classifier. The MLP classifier consists of a hidden layer with tanh activation and a softmax output layer. The network is then trained in an end-to-end manner with the standard cross-entropy loss (refer Section 2.2.8).

3.3 Results and Discussion

3.3.1 Data

The datasets used for evaluating the CAM model are SNLI [3] and SciTail [5]. For both the datasets, we used the standard train/development/test splits (refer Section 2.3.1).

3.3.2 Parameters

We use pre-trained 300-D Glove 840B vectors [67] to initialize the word embeddings. The out-of-vocabulary (OOV) words are initialized with the uniform distribution in $[-0.05, 0.05]$. The hidden states of all the layers for the SciTail and SNLI datasets are set to 100 and 300 (refer Chapter 5, Section 5.4.1). We use Adam optimizer [202] for optimisation with first momentum coefficient of 0.9 and second momentum coefficient of 0.999. To find best hyper-parameter for the model, we use grid search over the combination of L2 regularisation in [1e-4, 1e-5, 1e-6], batch size in $[32, 64, 128, 256]$, learning rate in $[0.001, 0.0003, 0.0004]$ and dropout

rate in $[0.2, 0.3, 0.4, 0.5]$. Each model is optimized on the development set for the best performance.

3.3.3 Results on SNLI

Table 3.1 shows the performances of the different models on the SNLI dataset. The first row presents the lexical classifier by Bowman et al. [3]. Sentence encoding based models are shown in the second group of Table 3.1. Bowman et al. [3] used LSTMs to generate sentence encoding of the premise and hypothesis. The sentence encodings are then fed to an MLP to identify the relationship between the premise and hypothesis. As discussed in Section 2.5.1, following this strategy various sentence encoding-based models are proposed in the NLI literature. These models are shown in the second group in Table 3.1.

Models	**Accuracy**	
	Train	Test
Lexical Classifier [3]	99.7	78.2
100D LSTM [3]	84.8	77.6
300D LSTM [58]	83.9	80.6
1024D GRU [203]	98.8	81.4
300D Tree-based CNN [95]	83.3	82.1
600D BiLSTM (intra-attention) [34]	84.5	84.2
300D Directional self-attention network [87]	91.1	85.6
600D Gumbel TreeLSTM [59]	93.1	86.0
600D Residual stacked encoders [79]	91.0	86.0
100D LSTMs word-by-word attention [36]	85.3	83.5
100D Deep Fusion LSTM [143]	85.2	84.6
600D BiLSTM (intra-attention with diversing input) [34]	85.9	85.0
50D Stacked TC-LSTMs [37]	86.7	85.1
300D MMA-NSE (attention) [137]	86.9	85.4
300D LSTMN (deep attention fusion) [147]	87.3	85.7
200D Decomposable attention (intra-attention) [42]	90.5	86.8
600D ESIM + 300D TreeLSTM [91]	93.5	88.6
ESIM + ELMo [60]	91.6	88.7
300D Combined attention mechanism (CAM, our approach)	90.5	**86.1**

Table 3.1: Accuracies of the sentence encoding- and joint sentence encoding-based models compared to the proposed CAM model on the SNLI dataset.

The third group of models are the joint sentence encoding-based models described in Section 2.5.2, utilising the inter-attention mechanism to align the sub-phrases between the premise and hypothesis. Peters et al. [60] holds the current state-of-the-art[1] performance on the SNLI dataset among the inter-attention, non-ensemble models. Embeddings from Language Models (ELMo) word embeddings of Peters et al. [60], when used with ESIM model of Chen et al. [91] improved the accuracy from 88.0% to 88.7%.

Among the models employing inter-sentence attention, our model, CAM, achieves a competitive accuracy of 86.1% on the SNLI dataset. Our model outperforms the previous models proposed by Rocktäschel et al. [36], Liu et al. [143], Liu et al. [34], Liu et al. [37], Munkhdalai and Yu [137], and Cheng et al. [147]. In CAM, we augment the intra-attention mechnaism of Liu et al. [34] by the inter-attention mechanism and our model significantly outperforms both the sentence encoding-based (in Table 3.1 model – 600D BiLSTM (intra-attention) [34], by 1.9) and joint sentence encoding-based (in Table 3.1 model – 600D BiLSTM (intra-attention with diversing input) [34], 1.1) variants. This evaluation result highlights the benefits of utilising the two attention mechanisms in a combined manner.

3.3.4 Results on SciTail

The SciTail dataset contains labelled data for the NLI classes of – neutral and entailment. NLI thus transforms into a binary classification task. Table 3.2 shows the results on the SciTail dataset. The low accuracies of the state-of-the-art ESIM [91] and decomposable attention model [42] suggest that SciTail is a difficult dataset to model. Our model considerably outperforms the ESIM and decomposable attention model, improving the accuracy by 6.6% and 4.9% respectively. The high accuracy of CAM on the SciTail dataset demonstrates that it can effectively model long and complex sentences.

[1] As on March 24, 2018

Models	Test Accuracy
Majority class	60.3
NGram	70.6
ESIM	70.6
DGEM w/o edges	70.8
Decomposable attention	72.3
DGEM	77.3
CAM (our approach)	**77.23**

Table 3.2: Accuracies of the NLI models [5] compared to the proposed CAM model on the SciTail dataset.

3.3.5 Ablation Analysis

We evaluate the effectiveness of the individual components of our model on the SciTail and SNLI datasets. Table 3.3 depicts the results.

Models	Test Accuracy(%)	
	SciTail	**SNLI**
Combined Attention	**77.23**	**86.14**
Intra-attention-only	75.49	80.27
Inter-attention-only	76.06	85.04

Table 3.3: Ablation analysis results for the SciTail and SNLI datasets.

For both the SciTail and SNLI datasets, none of the attention mechanisms individually achieved the accuracy higher than their combination. The results of the ablation analysis further demonstrate that the intra-attention and inter-attention mechanisms work constructively and achieve high accuracy when they are combined. We discuss the results for each of the datasets as follows.

For the SciTail dataset, both of our intra-attention-only and inter-attention-only models outperform the models of Parikh et al. [42] and Chen et al. [91] by a large margin. When we remove inter-attention mechanism from CAM, the intra-attention-only model has an accuracy of 75.49% and outperforms the decomposable attention model of Parikh et al. [42] and ESIM model of Chen et al. [91] (please refer Table 3.2 for the model accuracy of Parikh et al. [42] and Chen et al. [91]) by 3.1% and 4.9% respectively.

When we remove the intra-attention mechanism from CAM, the inter-attention-only model achieves an accuracy of 76.06%. The inter-attention-only model improves over the accuracy of decomposable attention of Parikh et al. [42] by 3.76% and by 5.46% over the ESIM model of Chen et al. [91].

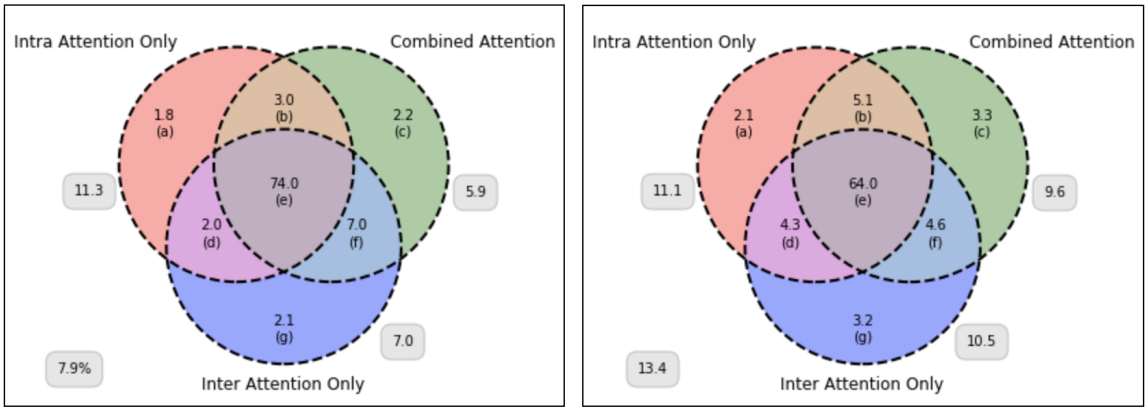
For the SNLI dataset, the intra-attention-only model does not perform well and it achieves an accuracy of 80.27%. However, the inter-attention-only model achieves an accuracy of 85.04%, which is higher than the word-by-word attention model of Rocktäschel et al. [36] by 1.5% and deep fusion LSTM model of Liu et al. [143] by 0.4%. The inter-attention-only model performs competitively with the intra-attention with diversing input model of Liu et al. [34].

It is worth noting that the SciTail dataset contains longer premises and hypotheses than the SNLI dataset [5]. The results of the ablation analysis for the SciTail dataset suggest that for long sentences, it is crucial to first capture the semantics of the input sentence by the intra-attention mechanism.

3.3.6 Fine-grained Accuracy Analysis

To investigate the effectiveness of each attention mechanism individually and in combination with each other, we further analyse the performance of each model in Table 3.3. The result of the analysis is shown in Fig. 3.2.

For the SNLI dataset, the results are shown in the Venn diagram of Figure 3.2(a). The three models, that is, the intra-attention-only, the inter-attention-only, and the combined attention model, correctly classified 74% the test samples (central region (e)). Combined attention model outperforms each of the individual attention mechanisms by correctly classifying 2.2% of test cases individually (region(c)) as compared to 1.8% of intra-attention-only (shown in region (a)) and 2.1% of inter-attention-only model (shown in region (g)). The inter-attention model and the combined attention model correctly classify 7.0% of test samples (shown in region (f)) whereas intra-attention and combined attention correctly classify 3.0% of test samples (shown in region (b)). This suggests that inter-attention is crucial for the high performance on the SNLI dataset. The intra-attention and inter-attention correctly classifies 2.0% of test samples. There are 7.9% test samples which cannot



(a) SNLI (b) SciTail

Figure 3.2: Venn diagram showing the percent of test samples correctly classified by each model in Table 3.3. The central overlapped region depicts the percent of correctly classified test samples by all the three models. The label adjoining each attention model shows the percent of test cases incorrectly classified by the individual model. The label at the left bottom shows the percent of test samples incorrectly classified by all the models. For instance, for the SNLI dataset (Fig. (a)) the three models classified 74.0% of test cases correctly. The combined attention model individually misclassified 5.9% of test cases and all the three models misclassified 7.9% of test cases.

be classified correctly by any of the three models.

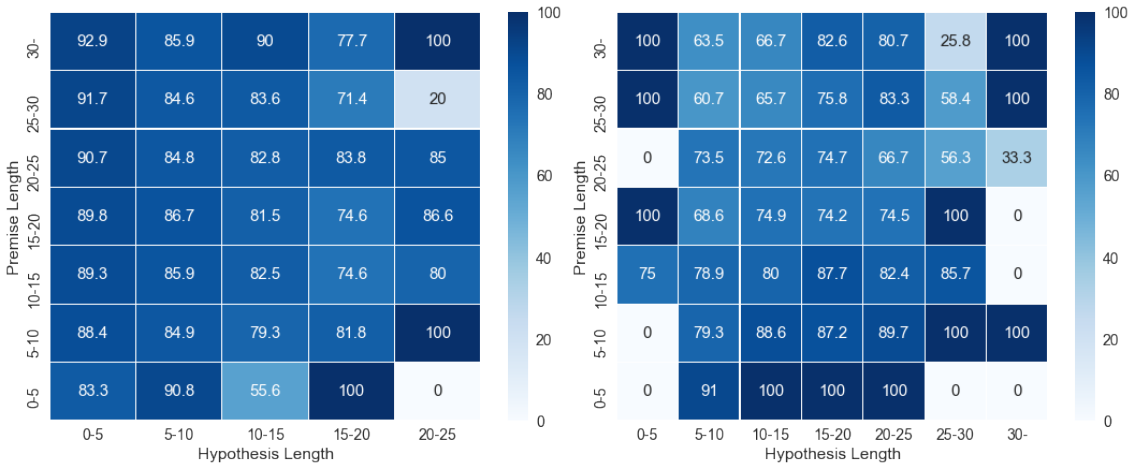
For the SciTail dataset, the results are depicted in Figure 3.2(b). The three models correctly classified 64% of the test cases (central region (e)). Similar to the SNLI dataset, the combined attention model gets the highest (3.3%) percent of test samples classified correctly. Unlike for the SNLI dataset, the intra-attention-only and combined attention models agree on a larger number of test cases (5.1%, region (b)) than the inter-attention-only and combined attention model, which agree on 4.6% (region (f)) of the test cases. Given the fact that the SciTail dataset is difficult to model [200], the result suggest that capturing the semantics of individual sequence first with intra-sentence attention is crucial for modelling complex datasets. Moreover, a significant number of test samples (13.4%) are not classified correctly by any of the model. This further indicates the high complexity of the SciTail dataset.

Linguistic analysis of the test samples in each region of Figure 3.2 is an interesting investigation to understand the behaviour of each model. Particularly, it is interesting to analyze syntax and semantics of the premise-hypothesis pairs, which

are incorrectly classified by the intra-attention-only and inter-attention-only models but correctly classified by combined attention model. Region (c) in Fig. 3.2 depicts these test cases. A preliminary linguistic observation on the syntactic structure of the premise-hypothesis pairs in this region suggest that for longer premises (word count > 20) the combined attention model predicts the test classes correctly more often than the intra-attention-only and inter-attention-only models.

3.3.7 Length Analysis

To understand the effectiveness of CAM for the premise and hypothesis sentences of varying lengths (word count), we evaluate the model accuracy when the hypothesis and premise lengths vary in the intervals 0-5, 10-15, 15-20, 20-25, 25-30 and greater than 30 words. The results are reported in Fig. 3.3. For both the SNLI and SciTail datasets, the result suggests that for all the premise length intervals, the model is very effective for hypothesis lengths greater than 10 words. The accuracy of 0% shows that no test case exists in that interval of premise-hypothesis length.



(a) SNLI (b) SciTail

Figure 3.3: The CAM model accuracies for the varying premise and hypothesis lengths of the SNLI and SciTail datasets.

3.4 Qualitative Analysis

We semantically and syntactically analysed the premise-hypothesis pairs of the SciTail test set that are correctly classified and the pairs that are misclassified by our model. The semantic analysis suggests that our model effectively learns to reason between the premise and hypothesis and do not depend on the word overlap between them. Table 3.4 and Table 3.5 illustrates some of the correctly and misclassified examples from the SciTail dataset.

S.No	Premise\Hypothesis Pair	Correct Test Label
1.	Helium is the second most abundant element in the known universe, after hydrogen.\The element hydrogen is the most abundant in the universe.	Entailment
2.	The reality is that plasmas make up over 98% of the matter in the universe.\Plasma matter makes up most of the universe.	Entailment
3.	A convex lens is a lens that is thicker in the middle than at its edges.\A concave lens is thicker at the edges than it is in the middle.	Neutral

Table 3.4: Correctly classified test cases by the CAM model from the SciTail test set.

S.No	Premise\Hypothesis Pair	Correct Test Label
1.	In the terminology of engineering mechanics, statics is the study of forces on structures, and dynamics is the study of forces on structures in motion.\Dynamics is the study of how forces affect the motion of objects.	Entailment
2.	Our digestive system requires that our food is chewed by teeth, go through the esophagus, stomach, intestine and many associate organs.\Esophagus, stomach, intestines are the structures that make up the digestive system in the human body.	Entailment

Table 3.5: Misclassified test cases by the CAM model from the SciTail test set.

The test case 1 in Table 3.4, suggests that the model correctly learns to reason

that hydrogen is the most abundant element in the universe without this being explicitly stated in the premise sentence. Similarly, for the text case 2, for the premise-hypothesis pair to be correctly classified, the model must learn the numerical reasoning by which it can conclude that - "98 percent of the matter" is – the "most" of the universe. Text case 3 is an interesting example where our model excels. The test case has a high degree of word overlap, however, the model does not get confused and correctly identifies that hypothesis is neutral to the premise. For the misclassified text cases of Table 3.5, we observed that premise-hypothesis pairs are generally syntactically and semantically intricate and contain ambiguous words. We believe that it is essential to embed external linguistic and real-world knowledge in the NLI model to correctly classify these text cases. Chapter 4 and Chapter 6 present the linguistic and real-world knowledge enhanced NLI models that demonstrates superior performances on such premise-hypothesis pairs.

3.5 Conclusions

In this chapter, we proposed a natural language inference model called the Combined Attention Model (CAM), that leverage the intra-attention and inter-attention mechanisms to learn the accurate semantic representations of the premise and hypothesis. The model first captures the semantics of the individual premise and hypothesis inputs with intra-attention and then aligns the premise and hypothesis with the inter-sentence attention mechanism to learn cross sentence dependencies.

Qualitative and quantitative evaluations on two datasets – SNLI and SciTail, demonstrate that the proposed Combined Attention Model is capable of modelling the semantics of long and complex sentences. CAM performs particularly effectively on the hard to model SciTail dataset, achieving 77.23% accuracy and outperforming the state-of-the-art ESIM by 6.6% and decomposable attention models by 4.9%. Further, the results of ablation analysis show that the intra-attention and inter-attention mechanisms work constructively and achieve higher accuracy when they are combined together in the same model than when they are used individually.

Despite the superior performance of CAM and the other state-of-the-art NLI

models on the SNLI and SciTails datasets, the NLI models suffer from the lack of lexical and commonsense knowledge that is present and learnable from these datasets. As discussed in the NLI task definition (Section 1.1), NLI relies on common human understanding of language and the real-world commonsense knowledge on which the (human) entailment judgement relies. Thus, it is crucial to investigate the effect of incorporating external linguistic and commonsense knowledge into the NLI models. In the next chapter, we explore the effect of incorporating external knowledge into the NLI models.

CHAPTER 4

Bilinear Fusion of Commonsense Knowledge with Attention-Based NLI Models

"In order for an intelligent creature to act sensibly in the real world, it must know about that world and be able to use its knowledge effectively. The common knowledge about the world that is possessed by every schoolchild and the methods for making obvious inferences from this knowledge are called commonsense . . . How to endow a computer program with commonsense has been recognized as one of the central problems of artificial intelligence since the inception of the field."

— Ernest Davis, Representations of Commonsense Knowledge, Ch. 1, 2014

One of the major limitations of the contemporary neural NLI models is the sole reliance on the training data to learn the linguistic knowledge (word meaning, syntactic structure and semantic interpretation) and also the commonsense knowledge about real-world. Given the challenging nature of human judgement centric NLI task, the models can not depend solely on training data to acquire all this knowledge. Primarily, because as discussed in Section 1.2, humans do not express the implicit knowledge and a vast majority of real-world commonsense knowledge is not

mentioned in the texts. Hence, we consider the task of incorporating real-world commonsense knowledge into the deep neural NLI models.

In the context of artificial intelligence, commonsense knowledge is the set of background information about the everyday world, that an individual is expected to know or assume, and the ability to use it when appropriate [188]. The importance of commonsense or factual background knowledge in natural language understanding applications has long been recognised [188]. Many complex NLU applications such as question-answering [204] and machine reading [205] achieved improved performance when supplied with commonsense knowledge.

Premise: Two young girls hang tinsel on a Christmas tree in a room with blue curtains. tinsel IsA christmas tree decoration
Hypothesis: Two girls are decorating their Christmas tree. tree RelatedTo christmas
Premise: People sit and watch as a street performer is singing. people Antonym person
Hypothesis: A person is performing on the street. performing HasSubevent singing, person Antonym people

Table 4.1: The SNLI datset examples with commonsense triples (in red) from the ConceptNet KG. Commonsense knowledge helps the NLI model to reason over the premises and hypotheses.

Thus far, NLI research has not fully leveraged the additional information available via the use of commonsense knowledge. For example, state-of-the-art NLI models [39,41] are limited to incorporating only lexical-level external knowledge, such as synonym and hypernymy. However, NLI is a complex reasoning task, in addition to lexical-level external knowledge, the task requires real-world commonsense knowledge to reason about inference. Table 4.1 shows examples from the SNLI dataset [3], where the commonsense knowledge is retrieved from the ConceptNet KG [206]. The common knowledge that, *tinsel IsA Christmas tree decoration* and *tree RelatedTo christmas* is useful to ascertain the inference relationship of entailment. Similarly, for the second pair, the information that *performing HasSubevent singing* and *person Antonym people* enrich the contexts of the premise and hypothesis respectively which is crucial to reason the relationship of premise-hypothesis.

Due to the lack of such common knowledge, state-of-the-art NLI models perform substantially worse for the premise-hypothesis pairs that require real-world commonsense knowledge for reasoning inference [38].

Effectively incorporating the external commonsense knowledge in deep neural NLI models is challenging. The main challenges are:

- **Structured Knowledge Retrieval:** Given a premise-hypothesis pair, how to effectively retrieve the specific and relevant commonsense knowledge from the massive amounts of data in KGs.
- **Encoding Retrieved Knowledge:** Learning the representations of the retrieved external knowledge amenable to be fused with representations of premise-hypothesis is challenging.
- **Feature Fusion:** How to fuse the learned external knowledge encoding with the premise-hypothesis. This feature fusion requires substantial NLI model adaptations with marginal performance gains [40, 41, 114].

In this chapter, we propose a novel framework, BiCAM (acronym for **Bi**linear fusion of **C**ommonsense knowledge with **A**ttention-based NLI **M**odels), to address the abovementioned challenges. The BiCAM framework approach the stated challenges in the follwing unique ways.

- **Structured Knowledge Retrieval:** We formulate an effective set of heuristics to retrieve commonsense knowledge from the KGs (refer Section 4.1.1 for knowledge retrieval heuristics and Section 4.3.5 for the quality analysis of the retrieved knowledge).
- **Encoding Retrieved Knowledge:** We first embed the retrieved knowledge with Holographic Embeddings (HolE) [207], a KG embedding method to learn the embeddings of entities and relations in the KG and then encode the retrieved knowledge over the HolE embeddings via a CNN-based encoder.

  The HolE embedding technique learns expressive KG triple representations and is simple and efficient to train [50]. CNN-based commonsense knowledge

encoder learns the features from the input in a bag-of-words manner providing the accuracy and convenience in feature learning from the part sequential, triple-based KG data [171] (refer Section 4.1.2 for embedding and encoding of retrieved knowledge).

- **Feature Fusion:** Finally, we use a state-of-the-art feature fusion technique, factorized bilinear pooling [208], to learn the fused feature representation of the learned commonsense encoding and the sentence encoding from the NLI model. Bilinear pooling allows each feature point in the fusing feature vectors to interact and captures complex associations between them. The joint representations created in such a manner are more expressive than the representations created through the concatenation or element-wise summation or multiplication of fusing vectors (refer Section 4.1.2 for details on bilinear pooling and the ablation analysis Section 4.3.3 for the effectiveness of bilinear pooling).

The salient feature of the proposed, BiCAM, framework is that it is an NLI model-independent framework that generalises across NLI models, datasets and commonsense knowledge sources and does so without any architectural changes to the underlying NLI model. In summary, the main contributions of this chapter are:

- We introduce an NLI model-independent neural framework, BiCAM, to incorporate external commonsense knowledge into the NLI models. The experimental result demonstrates that BiCAM generalizes across NLI models, datasets, and commonsense knowledge sources.

- To the best of our knowledge, we are the first to use the nonlinear feature fusion technique – factorized bilinear pooling, to fuse premise-hypothesis and commonsense knowledge features in the NLI models.

- An extensive evaluation of the proposed approach with two established NLI baselines, ESIM [91] and decomposable attention model [42] in combination with a general commonsense KG, ConceptNet and a (science) domain-specific KG, Aristo Tuple on two NLI datasets, SNLI and SciTail.

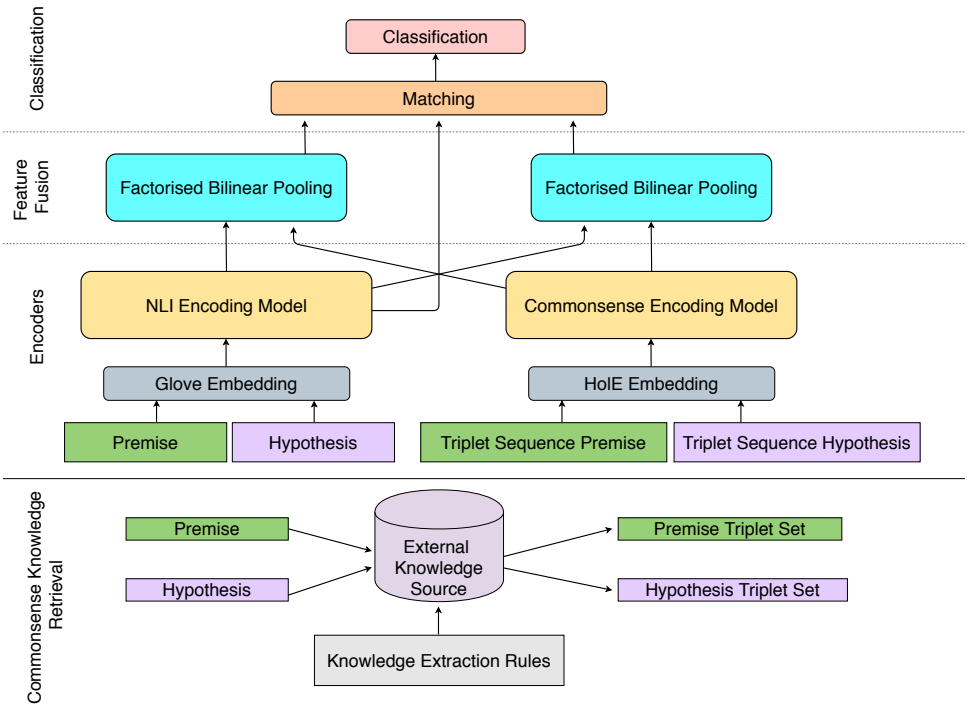


Figure 4.1: A high-level view of the proposed BiCAM architecture. The data (premise, hypothesis and the corresponding commonsense triples) flows from bottom to top. Premise and the corresponding triples are depicted in green, hypothesis and the corresponding triples are shown in purple.

4.1 Methods

A high-level view of our proposed BiCAM framework is illustrated in Figure 4.1. In this section, we discuss the individual BiCAM components and the uniquely structured framework. We catalogue the heuristics to retrieve relevant commonsense knowledge from KGs in the "Commonsense Knowledge Retrieval" section (4.1.1). Embedding and encoding of commonsense knowledge and NLI premise-hypothesis are discussed in the "Encoders" section (4.1.2). The "Feature Fusion" section (4.1.2) discusses factorized bilinear pooling.

4.1.1 Commonsense Knowledge Retrieval

To extract external commonsense knowledge, we consider two KGs: ConceptNet, for general real-world commonsense knowledge and Aristo Tuple, for (science) domain-

specific knowledge (refer Sections 2.3.2). To reiterate, the knowledge in these KGs is represented as a triple *(head, relation, tail)*, where *head* and *tail* are the real-world entities and the *relation*, is a specific set of associations, describing the relationship between the entities. Examples of triples in ConceptNet KG are *(employees AtLocation work)*, *(shirt UsedFor wearing)*.

Retrieval and preparation of contextually specific and relevant information from the KGs are complex and challenging tasks and are the crucial steps in our model. Several heuristics, statistical and neural approaches have been proposed in the field [209]. For this research work, we use a heuristic retrieval mechanism. The retrieved triples are ranked as per the order of retrieval. To retrieve contextually specific and relevant KG triples, we successively formulate the proposed heuristics. Specifically, we started with the individual heuristic, qualitatively analysed the retrieved KG knowledge for relevance to the context of premise and hypothesis and alter the current heuristic or formulate a new heuristic.

Further, we find empirically that non-specific commonsense knowledge from the KGs degrades the model performance. Below we catalogue the heuristics and illustrate the triples retrieved by the application of each heuristic in Table 4.2.

1. Stop words are removed from the premise and hypothesis.
2. To identify the relations between the words within the premise or hypothesis, we retrieve all triples involving each pair of words as head and tail.
3. To identify the relations from premise words to hypothesis words, we retrieve the triples with premise words as head and the words of the hypothesis as the tail. For the hypothesis, we extract the relations from the hypothesis words to premise words.
4. The relation *RelatedTo* has the largest number of triples in the ConceptNet KG. Although the relation communicates that the head and tail are related, it does not specify the specific relationship between them. To eschew the extracted commonsense knowledge from non-specific information and a higher number of triples with *RelatedTo* relation, we randomly select one triplet with *RelatedTo* relation, if multiple such triples are extracted. Additionally, we removed any duplicated triples from the final set of retrieved triples.

Step	Premise	Hypothesis
Input	A **white horse** is pulling a **cart** while a **man** stands and watches.	An **animal** is walking **outside**.
1.	'white', 'horse', 'pulling', 'cart', 'man', 'stands', 'watches'	'animal', 'walking', 'outside'
2.	horse HasProperty white, cart RelatedTo horse	animal AtLocation outside
3.	horse IsA animal, horse RelatedTo animal, horse AtLocation outside	animal RelatedTo horse animal antonym man animal DistinctFrom man
4.	horse HasProperty white, cart RelatedTo horse horse IsA animal, horse AtLocation outside	animal AtLocation outside, animal RelatedTo horse, animal antonym man

Table 4.2: A step by step illustration of commonsense knowledge retrieval for a SNLI premise-hypothesis pair from the ConceptNet KG. Each step in the table corresponds to the heuristic detailed in the Section 4.1.1 – Commonsense Knowledge Retrieval. Step 4 shows the final set of retrieved triples for the premise and hypothesis.

5. Finally, if the words of the premise and the hypothesis do not extract any commonsense knowledge by the application of above heuristics, we randomly select a word from them and extract a triple from one of the relations in *entails, synonym, antonym.*

4.1.2 Model Architecture

Commonsense Encoding Model. The commonsense encoding model learns the features from the retrieved commonsense triples. We provide a layer-by-layer description (refer Figure 4.1).

Embedding Layer. To represent the retrieved commonsense triples, we learn the Holographic Embeddings (HolE) [207] of KG triples. HolE embedding technique learns expressive representations and is simple and efficient to train [50].

Given a commonsense triple $(h,\ r,\ t)$, HolE represents both the entities and relations as vectors in $\mathbb{R}^d$. First, HolE compose the head and tail into $\mathbf{h} \star \mathbf{t} \in \mathbb{R}^d$

using the circular correlation:

$$[\mathbf{h} \star \mathbf{t}]_i = \sum_{k=0}^{d-1} [\mathbf{h}]_k \odot [\mathbf{t}_{(k+i) \bmod d}] \tag{4.1}$$

where $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$ are the *head* and *tail* entities embedding, $\star$ and $\odot$ respectively denotes circular correlation and the Hadamard product. The compositional vector obtained is then matched with the continuous representation of relation to score the commonsense triple using the scoring function defined as:

$$f_r(h,t) = \mathbf{r}^{\mathrm{T}}(\mathbf{h} \star \mathbf{t}) = \sum_{i=0}^{d-1} [\mathbf{r}]_i \sum_{k=0}^{d-1} [\mathbf{h}]_k \odot [\mathbf{t}]_{(k+i) \bmod i} \tag{4.2}$$

where $\mathbf{r} \in \mathbb{R}^d$ is the relation embedding. The score measures the plausibility of the commonsense triple. We train the HolE embeddings (Θ) using the pairwise ranking loss computed as:

$$\min_{\Theta} \sum_{i \in \Gamma_+} \sum_{j \in \Gamma_-} \max(0,\ \gamma + \sigma(\eta_j) - \sigma(\eta_i)) \tag{4.3}$$

where Γ_+ denotes the set of triples in the KG, Γ_- denotes the "negative" triples that are not observed in KG and $\gamma > 0$ specifies the width of margin, $\sigma(.)$ denotes the logistic function and η is the value of the scoring function.

For ConceptNet and Aristo Tuple, we train the HolE embeddings for the triples retrieved from the SNLI and SciTail vocabulary. Table 4.3 shows the *train/development /test* splits used to learn the embeddings.

NLI	**KG (#rel)**	**Train Set**	**Dev Set**	**Test Set**
SNLI	ConceptNet (47)	560,718	62,302	69,224
SciTail	ConceptNet (47)	467,362	51,929	57,699
	Aristo Tuple (1605)	232, 109	27,346	30,385

Table 4.3: Number of data triples from the ConceptNet and Aristo Tuple KGs for learning the HolE Embeddings. (#rel) is the number of relations in the KG.

For all the three pairs of NLI/KG data in Table 4.3, we use AdaGrad [210]

to optimize the objective in Equation (4.3), via an extensive grid search over an initial learning rate of $(0.001, 0.01, 0.1)$, a margin of $(0.2, 1, 2, 10)$, mini-batch size $(50, 100, 150, 200)$ and entity embedding dimensions of $(50, 100, 150, 200)$. At each gradient step, we randomly generate 5 negative *tail* entities with respect to a positive triple.

We evaluate the learned HolE embeddings on the triplet classification task. For SNLI/ConceptNet pair, the model achieves the highest accuracy of 64.0% with an embedding dimension of 150. For SciTail/ConceptNet and SciTail/Aristo Tuple pairs, HolE reported the top accuracy of 62.8% and 69.4% respectively at embedding dimension 100.

Encoding Layer. To learn the features over the pre-trained HolE embeddings, we employ a CNN-based neural model [171].

For each premise/hypothesis, let $\mathrm{T} = (\tau_1, \tau_2, \ldots, \tau_m)$ be a sequence of length n created by joining the m retrieved triples from the KG. Each τ is of the form (h, r, t) and, hence, $n = 3m$. The sequence T, padded where necessary, and represented as:

$$\mathrm{T} = (x_1, x_2, , x_3), (x_4, x_5, , x_6), \ldots, (x_{n-2}, x_{n-1}, x_n) \tag{4.4}$$

where x_i is the i-th word in the sequence and the $(x_1, x_2, , x_3)$ are the words from the triple τ_1. Let $\mathbf{x}_i \in \mathbb{R}^d$ be the d-dimensional HolE embedding corresponding to the i-th word. A sequence of length n is represented as a matrix $X \in \mathbb{R}^{d \times n}$, by concatenating its word's HolE embedding as columns, *i.e.*, $\mathbf{x}_i$ is the $i-$th column of X. We apply a convolution operation with filter $W \in \mathbb{R}^{d \times h}$, to a window of h words. The convolution operation learns a new feature map from the set of h words with the operation:

$$\mathbf{c} = f(X * W + b) \in \mathbb{R}^{(\frac{n-h}{s})+1} \tag{4.5}$$

where $\mathbf{c}$ is a feature map, $b \in \mathbb{R}^{(\frac{n-h}{s})+1}$ is the bias term, s is the stride of convolution filter, and $f(\cdot)$ is the activation function, rectified linear unit in our experiments and $*$ denote convolution operation. The filter convolve over each window $(\mathbf{x}_{ih+1:\,(i+1)h})$ where $0 \leq i \leq n-1$ in X to produce a feature map $(\mathbf{c})$. We

set the h and s to 3 for the commonsense triples. Convolving the same filter with the 3-gram beginning at every 3$^{\text{rd}}$ position in the triple sequence allows the features to be extracted from every triplet in the sequence. We then apply a max-over-time pooling operation over the feature map and take the maximum value $\hat{c} = \max\{\mathbf{c}\}$ as a feature corresponding to this filter. Max pooling operation captures the most important feature for each feature map.

Above we detailed the process of extracting one feature from one filter. Multiple filters (with fixed window size and stride of 3) are employed to obtain multiple features. Each filter is considered as a linguistic feature detector that learns to recognize a specific feature from the commonsense triple. The output of the commonsense encoder is a l-dimensional vector to represent commonsense.

NLI Encoders. We incorporate BiCAM with two established NLI baselines: ESIM [91] (refer Section 2.5.2) and decomposable attention model [42] (refer Section 2.5.2).

Feature Fusion. We apply factorized bilinear pooling [208] to fuse the commonsense features and NLI sentence features. Let $\mathbf{p}, \mathbf{h} \in \mathbb{R}^{d^{enc}}$ be the NLI model generated encoding of premise and hypothesis respectively. Also, let $\mathbf{p}^{cs}, \mathbf{h}^{cs} \in \mathbb{R}^{l^{enc}}$ denote the corresponding commonsense encoding generated by commonsense encoder for premise and hypothesis respectively. We apply the factorized bilinear pooling defined as:

$$\mathbf{z}^p = \text{SumPooling}(U^T\,\mathbf{p} \odot V^T\,\mathbf{p}^{cs}, k) \tag{4.6}$$

$$\mathbf{z}^h = \text{SumPooling}(U^T\,\mathbf{h} \odot V^T\,\mathbf{h}^{cs}, k) \tag{4.7}$$

where $\mathbf{z}^p$, $\mathbf{z}^h \in \mathbf{R}^o$ are the NLI and the corresponding commonsense feature-fused representations of the premise and hypothesis respectively and $U \in \mathbb{R}^{d^{enc} \times ko}$, $V \in \mathbb{R}^{l^{enc} \times ko}$ are projection matrices learned during training. The k and o, the factor and the output dimensionality, respectively, are the hyper-parameters of the factorised bilinear pooling method. SumPooling(x, k) denote a sum pooling over x with a one-dimensional non-overlapped window of size k and $\odot$ represents the Hadamard product.

To prevent overfitting, we also added a dropout layer [63] after the element-wise

multiplication of the projection matrices. Further, to allow the model to converge to a satisfactory local minimum, we append power normalization ($\mathbf{z} \leftarrow \text{sign}(\mathbf{z})|\mathbf{z}|^{0.5}$) and l_2 normalization layers ($\mathbf{z} \leftarrow \mathbf{z}/\|\mathbf{z}\|$) after SumPooling layer [208].

The factorized bilinear pooling captures the complex association between the features from premise-hypothesis and the corresponding commonsense features. The pooling method is implemented as a feed-forward neural network.

Classification Layer. We classify the relationship between the premise and hypothesis using a Multilayer Perceptron (MLP) classifier. The input to the MLP is the concatenation of sentence embeddings ($\mathbf{p}$ and $\mathbf{h}$) obtained from the NLI model and their corresponding commonsense feature-fused representations ($\mathbf{z^p}$ and $\mathbf{z^h}$) obtained from the factorised bilinear pooling layer. The input is represented as

$$\mathbf{f}^{final} = [\mathbf{p}; \mathbf{z}^p; \mathbf{h}; \mathbf{z}^h] \tag{4.8}$$

The MLP consists of two hidden layers with tanh activation and a softmax output layer to obtain the probability distribution for each class. The network is trained in an end-to-end manner using multi-class cross-entropy loss.

4.2 Experiments and Results

Our aim is to incorporate commonsense knowledge into NLI models in order to augment its reasoning capabilities. The method to do so should generalise across different NLI models, datasets, and KGs. To test BiCAM's efficacy and investigate its generalisability, we evaluate the BiCAM framework using two attention-based NLI baselines on two benchmark datasets in combination with two KGs. We compare our models with both external knowledge-based and attention-based NLI models. We refer to BiCAM as Bi**D**CAM, when the decomposable attention model is used as NLI baseline and Bi**E**CAM, when ESIM is used (see Figure 4.1). We next introduce the general evaluation setting of our BiCAMs (BiDCAM and BiECAM).

Experimental Settings For the NLI encoders, in our BiECAM and BiDCAM models, we follow the experimental settings such as dropout locations, word embedding initialisation with 300 dimensional Glove $840B$ embeddings [67] and the hidden layer sizes as suggested originally in the ESIM [91] and decomposable attention [42] models respectively.

For the commonsense encoding model, ConceptNet and Aristo Tuple KG triples are initialised with 150 and 100 dimensional pre-trained HolE embeddings [207]. These embeddings are trained and selected as discussed in Section 4.1.2 for the embedding layer of commonsense encoding model. The number of filter are fine-tuned from $[50, 100, 120, 150]$. The max pooling layer of the commonsense encoding model is regularised with dropout regularisation.

For the factorised bilinear pooling, dropout is applied after the element-wise multiplication of the projection matrices [208] (refer Section 4.1.2) and to the output of the layer. The factor number and output dimensions are fine-tuned from $[5, 6, 7]$ and $[500, 600, 700]$ respectively.

For the overall BiCAM framework, to find the best hyper-parameter, along with the specific hyper-parameters for the individual BiCAM components discussed above, we use a grid search over the combination of L2 regularisation in [$1e$-4, $1e$-5, $1e$-6], batch size in $[32, 64, 128, 256]$, initial learning rate in $[0.001, 0.0003, 0.0004]$ and dropout rate in $[0.2, 0.3, 0.4, 0.5]$. Each model is optimized on the development set for the best performance.

Datasets. We assess BiCAMs on the SNLI [3] and SciTail [5] benchmark datasets (refer Section 2.3.1). We consider ConceptNet KG for general commonsense knowledge and Aristo Tuple KG for domain-specific knowledge (refer Section 2.3.2).

Results on SNLI. Table 6.1 shows the results of the state-of-the-art external knowledge-based and attention-based NLI models in comparison to BiCAMs. We evaluate ConceptNet KG for commonsense knowledge for the SNLI dataset. The models, BiDCAM and BiECAM, improve the performance of their respective attention-based baselines (decomposable attention and ESIM models) by +0.4% and +0.8%.

BiCAMs also perform consistently better among the external knowledge-based

NLI Model	Test Acc(%)
Attention-based Baselines	
CAM [62]	86.1
Decomposable Attention [42]	**86.3**
ESIM [91]	**88.0**
External Knowledge-based Baselines	
CNN-based KG [82]	73.1
AdvEntuRe [39]	84.6
BiLSTM (E_3) [73]	86.5
ESIM (E_3) [73]	87.3
Char+CoVe-L [70]	88.1
ESIM + Syntactic TreeLSTM [91]	**88.6**
KIM [41]	**88.6**
Our Models	
BiDCAM + ConceptNet	**86.7**
BiECAM + ConceptNet	**88.8**

Table 4.4: Accuracies of the state-of-the-art attention-based and external knowledge-based NLI models as compared to BiCAMs on the SNLI dataset. BiCAMs enhance the NLI models with the external knowledge retrieved from the ConceptNet KG.

and attention-based NLI models. BiECAM model achieves an accuracy of 88.8% outperforming (+0.2% accuracy improvement)[1] the state-of-art external knowledge-based NLI models, ESIM+Syntactic Tree LSTM [91] and KIM [41].

Results on SciTail. The test accuracy of different NLI models on the SciTail benchmark dataset is summarised in Table 6.2. For the SciTail dataset, we study the performance of BiCAMs on the general commonsense ConceptNet KG as well as the (science) domain-targeted Aristo Tuple KG.

All our models significantly outperform the incorporated baselines across both the KGs, achieving absolute improvements of up to 4.5% (BiDCAM + ConceptNet), 5% (BiDCAM + Aristo Tuple) on decomposable attention baseline and 7% (BiECAM + ConceptNet), 8% (BiECAM + Aristo Tuple) on ESIM baseline. This demonstrates our framework's ability to generalize well across a number of NLI models and different KGs.

The BiECAM + Aristo Tuple model achieves an improvement of 0.7% over the

[1]SNLI is a highly competitive dataset. Models differ very slightly in accuracy. Leaderboard as on July 18, 2020 – https://nlp.stanford.edu/projects/snli/

NLI Model	Test Acc(%)
Attention-based Baseline	
ESIM [91]	**70.6**
Decomposable Attention [42]	**72.3**
CAM [62]	77.0
External Knowledge-based Baseline	
Majority classifier [40]	60.3
AdvEntuRe(seq2seq generator) [39]	76.9
DGEM [5]	77.3
NSnet [40]	77.9
AdvEntuRe(seq2seq + rule generator) [39]	**78.6**
AdvEntuRe (rule generator) [39]	**79.0**
Our Models	
BiDCAM + ConceptNet	**76.8**
BiDCAM + Aristo Tuple	**77.3**
BiECAM + ConceptNet	**77.6**
BiECAM + Aristo Tuple	**78.6**

Table 4.5: Accuracies of the state-of-the-art attention-based and external knowledge-based NLI models as compared to BiCAMs on the SciTail dataset. BiCAMs enhance the NLI models with the external knowledge retrieved from the ConceptNet and Aristo Tuple KGs.

NSnet model, which is an established baseline developed for the SciTail dataset, demonstrating the effectiveness of BiCAM. All our models perform competitively on attention-based baselines, CAM and DGEM. BiECAM + Aristo Tuple observes an accuracy improvement of 1.3% over previous state-of-the-art DGEM model.

4.3 Analysis

4.3.1 Number of Commonsense Features

To investigate the effect of incorporating various numbers of commonsense features, we vary the number of triples input to the commonsense encoding model. Particularly, we are interested in answering the question: How many commonsense features are required for the optimal model performance? For the experiment we follow the order of retrieval of triples and do not rank them. The results are depicted in Figure 6.2.

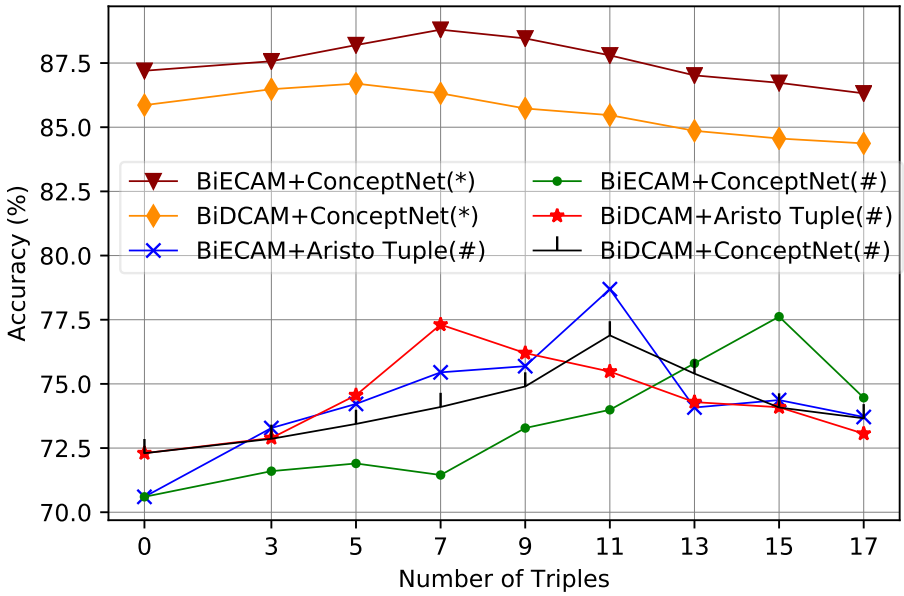


Figure 4.2: Accuracy of the BiCAMs with the varying number of commonsense triples. (*) denotes the SNLI and (#) the SciTail datasets.

For SNLI The model BiECAM + ConceptNet achieves the highest accuracy (88.8%) using 7 triples. We observe a decrease in accuracy with increasing the number of triples. BiDCAM + ConceptNet follow the same trend, however, it attains the highest accuracy (86.7%) with the fewer number (5) of triples. The fewer number of triples required for BiCAMs to achieve their maximum accuracies on the SNLI dataset, is attributed to the limited linguistic variation and short average length of stop-word filtered premise (7.35 for entails and neutral class) and hypothesis (3.61 for entails and 4.45 for neutral class) [5] of the SNLI dataset, which limit its ability to fully extract and exploit KG knowledge.

For SciTail The BiCAMs, when evaluated using the general commonsense knowledge source ConceptNet, require a relatively high number of triplets (11 and 15 resp.) to achieve their maximum accuracy. This is due to the higher syntactic and semantic complexity of the SciTail dataset, that needs more knowledge to reason about inference. However, when evaluated with the domain-specific Aristo Tuple KG, the models achieve the highest accuracies with fewer (BiDCAM at 7 and BiECAM at

11) triples. The specialised scientific knowledge in Aristo Tuple improves the model performance with less external knowledge.

We observe that the BiCAMs, when trained on the SciTail dataset, require a higher number of triples to attain maximum accuracy relative to when trained on the SNLI dataset. This can be attributed to the small training size of the SciTail dataset, which thus requires a higher number of triples to compensate for missing knowledge. We conclude that:

- The commonsense features, when incorporated in the correct number, help reason the relationship between premise and hypothesis.
- The number of commonsense features required depends on the syntax, semantics and size of the target dataset, as well as the domain of source KG.

4.3.2 Fine-grained Accuracy Analysis

To investigate the effectiveness of BiCAMs, we perform a fine-grained analysis of the BiECAM performance in conjunction with ESIM baseline. Figure 4.3 shows the Venn diagram of the analysis on the SNLI and SciTail datasets. For the SNLI dataset, both the baseline ESIM and BiECAM models classify 83.44% of test cases accurately. The models incorrectly classified 7.65% (label f) of test cases. Label c depicts the region for the test case percentage when only BiECAM correctly classifies. This region label shows that BiECAM significantly outperforms ESIM on the number of test cases (shown by label a) that it classifies correctly. Label e depicts the percentages of test case BiECAM, classifies incorrectly which is consistently lower than ESIM (label d) across both the datasets. We further utilise the test cases in different regions of the Venn diagram in the model error analysis in Section 4.3.5.

4.3.3 Ablation Study

To evaluate the impact of factorized bilinear feature fusion, we perform an ablation study on BiECAM + Aristo Tuple, our best performing model on the SciTail dataset. Table 4.6 demonstrates the performance of various non-bilinear and bilinear pooling methods. We observe that factorized bilinear pooling significantly outperforms all

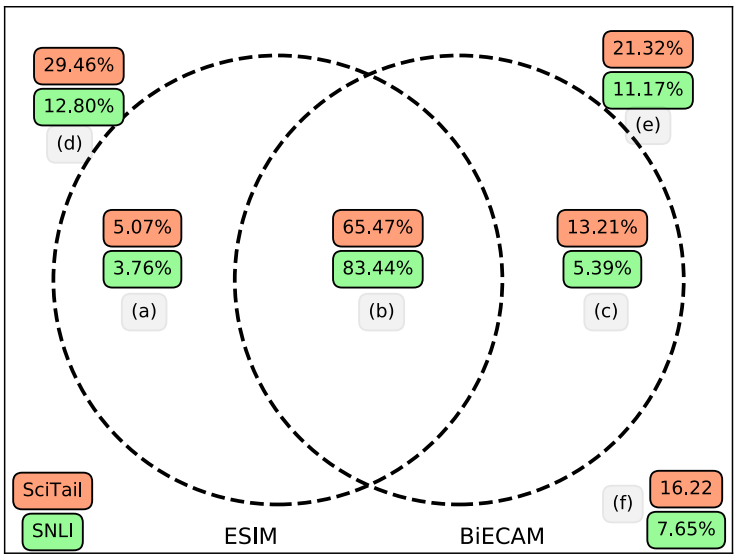


Figure 4.3: Fine-gain accuracy analysis on the ESIM and BiECAM models. Different labels (such as, a and b), orange for SciTail and green for SNLI, depicts model accuracy. For example, region marked (b) depicts the percent of correctly classified test cases by both the models. Labels (d) and (e) show the percent of test cases incorrectly classified by individual models. The label f, shows the percent of test samples incorrectly classified by both the models.

the non-bilinear pooling methods. To ascertain that the performance gain is not due to the higher number of parameters in bilinear method, we stack fully connected layers (with 1200 units per layer, ReLU activation and dropout) to increase the parameters in non-bilinear methods. We observe that increasing the number of parameters does not increase the model accuracy. The high accuracy of factorized bilinear pooling may be attributed to the outer product between the NLI sentence and the commonsense feature vectors. Outer product allows each feature point in the two feature vectors to interact and capture associations between them. The joint representations created in such manner are more expressive than the representations created through concatenation or element-wise summation or multiplication.

4.3.4 On the use of CNNs for Commonsense Encoder

For the commonsense encoder, our experiments with RNNs (LSTMs and BiLSTMs), considerably degraded the performance of the BiCAMs. This may be attributed to the inherent nature of RNNs, which learns the representations of words in the

Feature Fusion Method	Acc(%)
Concatenation	74.6
FC + Concatenation	75.5
FC + FC + Concatenation	74.3
FC + Element-wise Sum	72.5
FC + FC + Element-wise Sum	73.3
FC + Element-wise Product	76.4
FC + FC + Element-wise Product	76.8
FC + Element-wise Difference $\oplus$ FC + Element-wise Product	77.6
Factorized Bilinear Pooling	**78.6**

Table 4.6: Comparison of different pooling methods for the BiECAM + Aristo Tuple model on the SciTail dataset. FC is a fully connected layer with 1200 neural units and ReLU activation.

context of all previous words in the sequence. However, the set of triples input to the commonsense encoder is sequential within an individual triple. For example, in the set of triples – *outside Antonym inside* and *table RelatedTo eating*, the word *inside* is associated with the words in its own triple, *outside* and *Antonym*, but not with the words *table*, *RelatedTo*, and *eating* of the second triple. RNNs, due to their inherent recurrent nature, learn the incorrect features from the part-sequential input of set of triples. In contrast, CNNs learns features independently of the position of words in the sequence. In the commonsense encoder, learning the features over the window of three words with a stride of three, allows the correct features to be learnt from the part-sequential set of input triples.

4.3.5 Qualitative Analysis

Retrieved Commonsense Knowledge

To investigate the quality of commonsense knowledge retrieved from the proposed knowledge retrieval heuristics (refer Section 4.1.1), we inspect the retrieved commonsense knowledge for the premise and hypothesis. Table 4.7 presents some SNLI premise-hypothesis pairs with the retrieved ConceptNet triples. The retrieved commonsense knowledge shows that the heuristics are effective in retrieving the knowledge beneficial to reason inference.

p: A group of people are walking through a city street. **people IsA group, people AtLocation city, street UsedFor walking** **h:** People are walking the street. **street UsedFor walking, people IsA group, people AtLocation city**
p: A woman ironing a delicate blue fabric. **ironing RelatedTo clothes** **h:** A woman ironing clothes. **clothes RelatedTo fabric**
p: A group of people sitting at a conference table. **people IsA group, people AtLocation conference, table AtLocation conference, conference IsA meeting, table AtLocation meeting** **h:** Coworkers are having a meeting. **meeting RelatedTo conference**
p: A woman in shorts and sandals is being pulled by a small child as a subway train goes by. **small RelatedTo child** **h:** The train excites the toddler. **toddler IsA child**
p: A dog leaping to catch a Frisbee in the yard. **dog AtLocation outside, yard AtLocation outside** **h:**The dog is outside. **dog AtLocation outside**
p: A girl catches a baseball. **baseball UsedFor catching** **h:** The girl is catching something. **catching HasContext baseball**
p: People sit and watch as a street performer is singing. **people Antonym person** **h:** A person is performing on the street. **performing HasSubevent singing, person Antonym people**
p: A little girl and boy sit while reading books. **girl Antonym boy, books UsedFor reading** **h:** Two people sit while looking at books. **books UsedFor reading**

Table 4.7: The SNLI dataset premise-hypothesis pairs with the corresponding commonsense knowledge from the ConceptNet KG (**in bold)** retrieved with the proposed retrieval mechanism in Section 4.1.1. The retrieved commonsense knowledge enriches the contexts of the premise and hypothesis and helps the NLI model to reason over premise and hypothesis.

Error Analysis

Table 4.8 highlights selected sentences from the SNLI test set showing correct and incorrect inference prediction example for both BiECAM and the baseline ESIM. For the first example, BiECAM has additional context for premise and hypothesis from the knowledge that *wave RelatedTo crash* and *crash IsA hit*, which helps the model to correctly predict the inference class. However, the specific knowledge, about the *wave* and the *crash* is not available to the baseline ESIM model and hence, it incorrectly predicts the inference class. Similarly, for the second example,

the implicit common knowledge *performing HasPrerequisite skill* and *skill RelatedTo good* available to BiECAM helps identify the correct inference class, whereas ESIM fails.

BiECAM Correct ESIM Incorrect	
P/G	**Sentence with Retrieved Commonsense Knowledge**
n/e	**p:** Four boys are about to be **hit** by an approaching **wave**. **wave RelatedTo crash** **h:** A giant **wave** is about to **crash** on some boys. **crash IsA hit**
n/e	**p:** Young man **performing** a skateboard trick on a sidewalk in a city. **performing HasPrerequisite skill** **h:** A young man is **performing** a **good** skill on a skateboard on the sidewalk in a metropolitan area. **skill RelatedTo good**
BiECAM Incorrect ESIM Correct	
n/c	**p:** A **red** truck is parked next to a burning **blue** building while a man in a **green** vest runs toward it. **red Antonym blue, blue Antonym green, green Antonym red** **h:** The burning **blue** building smells of smoke. **blue Antonym red, blue Antonym green**

Table 4.8: Accurately and inaccurately predicted test cases from the SNLI test set. Retrieved commonsense knowledge is shown in bold. **P** is the predicted and **G** is the gold label. n: neutral, e: entailment, c: contradiction are the three inference classes of the SNLI dataset.

We observe that BiECAM fails to predict the correct inference class when noisy and irrelevant knowledge is retrieved from the KGs. For example, the last test case in Table 4.8, only retrieves the information that colors (such as red and blue) are antonyms of each other. The retrieved knowledge is irrelevant and is not completely correct, which does not help BiECAM.

4.4 Conclusions

In this chapter, we introduced an NLI model-independent neural framework, BiCAM, that incorporates commonsense knowledge to augment the reasoning capabilities of the NLI models. Combined with convolutional feature detectors and bilinear feature fusion, BiCAM provides a conceptually simple mechanism that generalises across NLI models, datasets and KGs. Moreover, BiCAM can be easily applied to different NLI model and KG combinations. Evaluation results show that our BiCAM

considerably improves the performance of all the NLI baselines it incorporates, across all the NLI datasets and the KGs and does so without any architectural change to the incorporated NLI model. Particularly for the smaller, syntactically and semantically complex SciTail dataset, commonsense knowledge incorporation via BiCAM achieves performance improvements of 7.0% with ConceptNet and 8.0% with Aristo Tuple KG. In addition, the fine-grained accuracy analysis of our BiECAM model in combination with the established ESIM baseline demonstrates the effectiveness of our model on both the SNLI and SciTail datasets.

Experimentation to investigate the effect of incorporating various numbers of commonsense features demonstrates that commonsense features when incorporated in the correct number, help reason the relationship between the premise and hypothesis. Further, the number of commonsense features required depends on the syntactic and semantic complexities, as well as the domain of external knowledge sources. Ablation analysis on BiECAM in combination with the SciTail dataset and Aristo Tuple KG shows that fusing NLI and commonsense features by the factorized bilinear pooling method is more effective than the traditional techniques such as concatenation, element-wise product and summation.

We observe that retrieval and selection of commonsense knowledge relevant for reasoning over the premise and hypothesis are challenging. Although heuristics are effective in retrieving the external knowledge, the challenge remains to retrieve contextually relevant external knowledge from the massive amounts of data in the KGs. Moreover, the high performance of the BiCAMs on SciTail with the contextually relevant external knowledge from the Aristo Tuple KG, emphasises that NLI models benefit more from the contextually relevant external knowledge. Further, although KG embeddings perform well they might not be expressive enough for the complex NLI task. Finally, the feature fusion also requires substantial effort to learn an expressive joint representation.

In Chapter 6, in light of the state-of-the-art developments in the field of contextual word representations and PTLMs, we address the abovementioned limitations of the proposed framework by exploiting the pre-trained language model, BERT [16] for the utilisation of external knowledge in the NLI task. However, in the next chap-

ter, we explore an area of high empirical significance for RNN-based NLI models — the application of regularisation technique dropout.

During the evaluations of the model, CAM, proposed in Chapter 3, we observed that the RNN-based neural networks are highly sensitive to model hyper-parameters, especially the dropout locations in the model and dropout rates notably affects model's performance. This observation led us to investigate an unaddressed challenge in the RNN-based NLI models — the lack of a coherent set of dropout application guidelines in the RNN-based NLI models.

In order to address this challenge, we exhaustively evaluated the proposed, CAM model for various locations of the dropout application with the varying dropout rates and analysed the results to gain insights for the application of dropout in RNN-based NLI models. Further, we validate the insights on our another RNN-based NLI model, BiECAM, proposed in this chapter.

In the next chapter, we present the study on dropout applications in the RNN-based NLI models, before returning to improve the utilisation of the external knowledge for the NLI task in Chapter 6.

CHAPTER 5

An Exploration of Dropout with RNNs for Natural Language Inference

As reviewed in Chapter 2, neural networks based on RNN encoders are the dominant class of neural NLI models. Hyper-parameter optimisation is highly significant to the performance of these RNN-based NLI models [51, 52], however, it is surprisingly overlooked in the NLI literature. RNNs (LSTMs [53] and BiLSTMs [54]) owing to a large number of parameters are susceptible to overfitting [55] — the case when neural networks learn the exact patterns present in the training data but fails to generalise to unseen data [56].

In RNN-based NLI models, regularisation techniques such as early stopping [41], L2 regularisation [37] and dropout [200] are used to prevent overfitting.

Among these techniques, dropout is the most effective regularisation technique [55, 211]. The idea of dropout is to randomly omit the computing units (neurons) in a neural network during training but keeping all of them for testing. This is achieved by the element-wise multiplication of the neural network layer activations with a zero-one mask (r_j) during training. Each element of this zero-one mask is drawn independently from the Bernoulli(p) distribution, where p is the probability with which the units are retained in the network. During testing, activations of the

layer are multiplied by p [56]. Dropout injected uncertainty for the presence of the nodes in the neural network during training enforces the network not to depend on some specific nodes to learn and produce the desired final result.

Due to the effectiveness of dropout in regularising the RNNs [55,212,213], nearly all the RNN-based NLI models discussed in the literature review of Chapter 2, apply dropout to one or more layers. However, the location of dropout applications in these RNN-based NLI models varies considerably and is based on the trial-and-error experiments. For example, Bowman et al. [3] apply dropout only to the input and output of the sentence embedding models. Ghaeini et al. [21] for their DR-BiLSTM, Tay et al. [200] in the CAFE and Chen et al. [91] in the ESIM models apply dropout to each feed-forward layer in the model whereas others, for example, Liu et al. [34] in the inner-attention model and Nie and Bansal [79] in their shortcut-stacked model use dropout only in the final MLP classifier (refer Section 2.2).

The SPINN model proposed by Bowman et al. [58] and the Gumbel Tree-LSTM model of Choi et al. [59] applied dropout to the output of the embedding layer and to the inputs and outputs of the MLP classifier (refer Section 2.2.8). In the BiMPM model, Wang et al. [52] applied dropout at every model layer.

The dropout rates are also crucial to the use of dropout regularisation [57]. However, even the models [58,59] that apply the dropout at the same layers, for example at the embedding and at the final MLP classifier layers (refer Section 2.2), use different dropout rates. Munkhdalai and Yu proposed NTI model [137] (refer Section 2.5.1) and explore nine different NTI variants with marginally varying complexities, however, use a different dropout rate for each model variant.

Therefore, there is a considerable ambiguity with regards to the application of the crucial regularisation parameter dropout in RNN-based NLI models and the research field lacks a set of coherent set of guidelines for the dropout application.

Motivated by the lack of consensus and a clear set of guidelines for the application of dropout, in this chapter, we investigate the effect of applying dropout at different locations in RNN-based NLI models. We first comprehensively investigate the CAM model (Chapter 3) with several locations of dropout application. At each dropout location, we also investigate the effect of varying the dropout rates. Based on the

CAM's empirical evaluations and analysis of the evaluation results, we draw certain findings for regularising RNN-based NLI models. To recommend the findings as the set of guidelines for regularising RNN-based NLI models, we further validate the findings on another RNN-based NLI model, BiECAM, proposed in Chapter 4. Finally, the validated set of findings are recommended as the guidelines for regularising RNN-based NLI models.

The CAM model utilising the embedding, encoding (recurrent), intra-attention, inter-attention, enhancement, pooling, matching, and the MLP classifier layers represents the generic architecture of neural NLI models introduced in Chapter 2, very closely (refer Section 2.2). Consequently, we believe that the guidelines empirically evaluated on the CAM model and validated on the BiECAM model will be relevant for other RNN-based NLI models.

To the best of our knowledge, this research is the first exploratory analysis of the dropout for RNN-based NLI models. The main contributions of this research work are:

- An exhaustive evaluation and comparative analysis of different locations of dropout application with varying dropout rates in the RNN-based NLI model, CAM.
- A comprehensive validation of the findings of the CAM model evaluations on another RNN-based NLI model, BiECAM.
- An empirically evaluated and validated set of guidelines for the application of dropout and dropout rates in the RNN-based NLI models.
- An investigation into the effectiveness of dropout in preventing the overfitting with an analysis of dropout rates on the performance of the RNN-based NLI model and on the dropout locations.

5.1 Related Research

Regularisation via dropout in RNNs lacks coherency and hence a number of studies on the application of dropout in the RNNs have been conducted in different fields

such as handwriting recognition [213] and speech recognition [55].

Dropout research by Pachitariu and Sahani [212] on language models, Pham et.al [213] on handwriting recognition and Zaremba et al. [55] on a number of tasks such as machine translation and image caption generation via RNNs have established that the recurrent connection dropout should not be applied as it affects the long-term dependencies in the sequential data.

Bluche et al. [57] studied dropout at different locations with respect to the LSTM layer in the handwriting recognition network of Pham et.al [213]. The results show that significant performance difference is observed when dropout is applied to distinct places. They concluded that applying dropout only after recurrent layers as applied by Pham et al. [213] or between every feed-forward layer, as done by Zaremba et al. [55], does not always yield good results. Cheng et al. [214], investigated the effect of applying dropout in LSTMs. They randomly switch off the outputs of various gates of LSTM, achieving an optimal word error rate on speech recognition task, when dropout is applied to output, forget and input gates of the LSTM.

Our work differs from previous studies in several aspects. First, by the size of the datasets used to evaluate the RNN models in previous studies. Evaluations in the previous research were conducted on datasets with fewer samples. We evaluate the RNN model on a larger, SNLI dataset (570,000 data samples) as well as on a smaller, SciTail dataset (27,000 data samples) (refer Section 2.3.1). Second, the previous studies concentrated only on the locations of dropout in the model with a fixed dropout rate. We further investigate the effect of varying dropout rates at each dropout application location. Third, we validate our findings on another RNN-based NLI model, which to the best of our knowledge, is lacking from all the previous studies.

In this work, we focus on the application of widely used conventional dropout [56] to the non-recurrent connections in the RNN-based NLI models.

5.2 Methodology

To evaluate the efficacy of dropout at different locations in the RNN-based NLI model, we selected the locations of the dropout application prevalent in the NLI literature. In addition, we consider other probable locations in the CAM model. Figure 5.1 depicts these locations. Table 5.1 illustrates the different combinations of these locations where the dropout is evaluated. At each location, we evaluate the model varying the dropout rate ranging from 0.1 to 0.5 with a granularity of 0.1. For each of the models in Table 5.1, the hyper-parameters are fine-tuned via grid search on the development set of the evaluation datasets (Section 5.3.1) from the parameters specified in Section 5.3.2. We report the best performance of each model from the grid search.

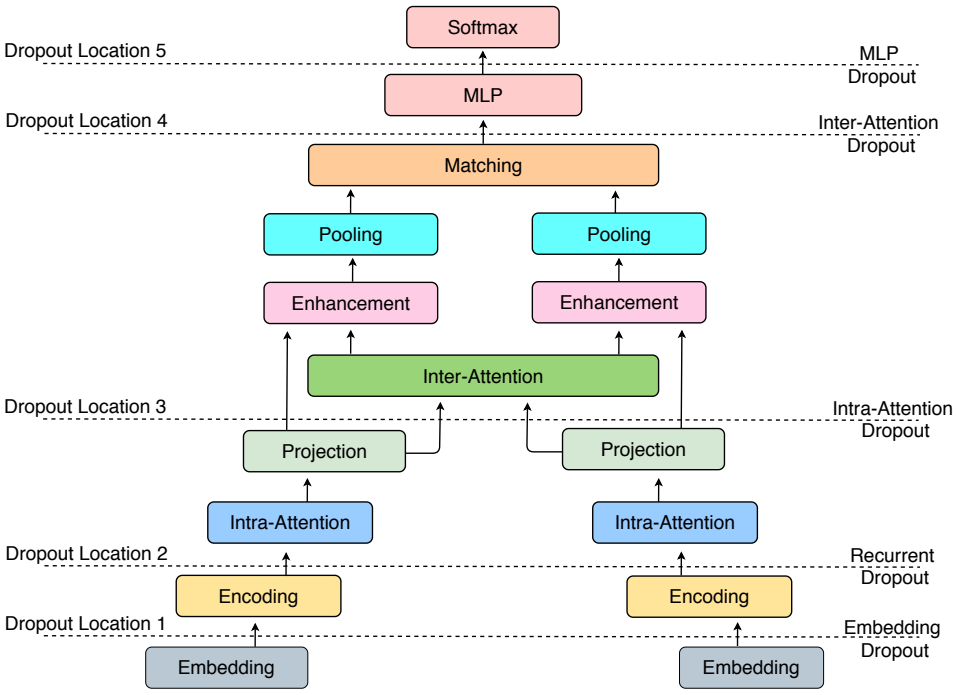


Figure 5.1: The Combined Attention Model (CAM) with the identifed dropout locations for the evaluation.

Model	Layer
Model 1	No Dropout (Baseline)
Model 2	Embedding
Model 3	Recurrent
Model 4	Embedding and Recurrent
Model 5	Recurrent and Intra-Attention
Model 6	Inter-Attention and MLP
Model 7	Recurrent, Inter-Attention and MLP
Model 8	Embedding, Inter-Attention and MLP
Model 9	Embedding, Recurrent, Inter-Attention and MLP
Model 10	Recurrent, Intra-Attention, Inter-Attention and MLP
Model 11	Embedding, Intra-Attention, Inter-Attention and MLP
Model 12	Embedding, Recurrent, Intra-Attention, Inter-Attention and MLP
Model 13	Embedding, Recurrent, Inter-Attention and MLP

Table 5.1: CAM model with different combination of layers to the output of which the dropout is applied.

5.3 Experimental Setup

5.3.1 Datasets

In order to investigate the effect of dropout on dataset sizes in combination with dropout locations and dropout rates, we evaluate all the models depicted in Table 5.1 on two diverse datasets (refer Section 2.3.1) — SNLI and SciTail. We utilise the standard train/development/test splits of the datasets to train the models.

5.3.2 Hyper-parameters

The hyper-parameters for the model are selected separately for the SNLI and SciTail datasets by a grid search from the combination of L2 regularisation in [$1e$-4, $1e$-5, $1e$-6], batch size in $[32, 64, 128, 256]$ and learning rate in $[0.001, 0.0003, 0.0004]$. The Adam [202] method is used as an optimizer. The first momentum is set to 0.9 and the second to 0.999. The word embeddings are initialized with pre-trained 300-D Glove $840B$ vectors [67] and are not fine-tuned with the model. The model is trained

with early stopping based on the development set accuracy with the patience of 15. These hyper-parameter settings allow the efficacy of dropout to be evaluated in the realistic scenarios as employed by the NLI models in the literature. We report the best accuracy for each of the model in Table 5.1.

5.4 Results and Discussion

The evaluation results of dropout application for each model in Table 5.1 are presented in Table 5.2. The non-regularised model, Model 1, is our baseline model. We discuss the results for individual and multiple layers in the following sections.

Dropout at Individual Layers We first applied dropout at each layer including the embedding layer. Although the embedding layer is highly parameter intensive (and, hence susceptible to overfitting), it is often not regularised for many language applications [215]. However, we observe the benefit of regularising it. For the SNLI dataset, the highest accuracy is achieved when the embedding layer is regularised (Model 2, DR 0.4).

For the SciTail dataset, the highest accuracy is attained when the recurrent layer is regularised (Model 3, DR 0.1). The dropout injected noise at lower layers prevents the higher fully connected layers from overfitting. We further experimented regularising higher layers (Intra-Attention, Inter-Attention, MLP) individually, however, no significant performance gains were observed[1].

Dropout at Multiple Layers We next explore the effect of applying dropout at multiple layers. For SNLI and SciTail, the models achieve higher performance when dropout is applied to embedding and recurrent layer (Model 4, DR 0.2). This supports the importance of regularising embedding and recurrent layer as seen for the individual layers.

It is interesting to note that regularising the recurrent layer helps the SciTail dataset (Model 7, DR 0.2), whereas regularising the embedding layer helps the

[1]results are not shown in Table 5.2

Models	Dataset	Dropout Rate (DR)				
		0.1	0.2	0.3	0.4	0.5
Model 1	SNLI			84.45		
	SciTail			74.18		
Model 2	SNLI	84.56	84.59	84.42	**86.14**	84.85
	SciTail	**75.45**	75.12	74.22	73.10	74.08
Model 3	SNLI	84.12	**84.21**	83.76	81.04	79.63
	SciTail	**76.15**	75.78	73.50	73.19	75.26
Model 4	SNLI	83.83	**85.22**	84.34	80.82	79.92
	SciTail	74.65	**76.08**	74.22	74.46	73.19
Model 5	SNLI	**84.72**	83.43	72.89	70.49	62.13
	SciTail	**75.87**	75.13	75.26	73.71	72.25
Model 6	SNLI	84.17	**84.32**	83.71	82.79	81.68
	SciTail	73.85	**75.68**	75.26	73.95	73.28
Model 7	SNLI	**84.33**	82.97	82.00	81.15	79.25
	SciTail	73.75	**75.02**	74.37	73.37	73.42
Model 8	SNLI	84.67	**85.82**	84.60	84.14	83.94
	SciTail	73.80	73.52	69.29	**75.82**	73.89
Model 9	SNLI	**84.44**	83.05	82.09	81.64	79.62
	SciTail	75.68	**76.11**	75.96	70.84	74.55
Model 10	SNLI	**84.45**	80.95	75.31	70.81	69.34
	SciTail	73.30	**75.21**	74.98	74.65	71.59
Model 11	SNLI	**84.31**	82.43	78.94	74.93	70.54
	SciTail	**75.63**	73.47	74.93	74.93	70.32
Model 12	SNLI	**84.32**	82.60	73.36	71.53	66.67
	SciTail	73.47	**75.63**	74.74	73.42	74.40

Table 5.2: CAM model accuracies on different dropout locations with varying dropout rates for the SNLI and SciTail datasets. Bold numbers shows the highest accuracy for the model within the dropout range.

SNLI dataset (Model 8, DR 0.2). A possible explanation to this is that for the smaller SciTail dataset, the model can not afford to lose information in the input, whereas for the larger SNLI dataset, the model has a chance to learn even with the loss of information in the input. Further, the bigger datasets like SNLI have a larger vocabulary, and hence a large number of parameters. Due to high parametrisation,

the embedding layer is extremely susceptible to overfitting and is a suitable layer to be regularised with dropout. The results from the models 7 and 8 suggests that applying dropout at a single lower layer (Embedding or Recurrent; depending on the amount of training data) and to the inputs and outputs of MLP layer improves performance.

We can infer from models 9, 10, 11, 12 that applying dropout to each feed-forward connection helps to prevent the model overfitting for the SciTail dataset (DR 0.1 and 0.2). However, for both the datasets with different dropout locations the performance of the model decreases as the dropout rate increases (refer Section 5.4.2).

5.4.1 The Effectiveness of Dropout for Overfitting

We investigate the efficacy of dropout on overfitting. The main results are shown in Figure 5.2. For SNLI, Figure 5.2 (a) - (b), shows the convergence curves for the baseline model and the model achieving the highest accuracy (Model 2, DR 0.4).

The convergence curve shows that dropout is very effective in preventing overfitting. However, for the smaller SciTail dataset when regularising multiple layers, we observe that the highest accuracy achieving model (Model 9, DP 0.2), overfits significantly (Figure 5.2(d)). This overfitting is due to the large model size. With limited training data of the SciTail dataset, our model with a higher number of hidden units learns the relationship between the premise and the hypothesis most accurately (Figure 5.2(d)). However, these relationships are not representative of the validation set data and thus the model does not generalize well.

When we reduced the model size (50, 100 and 200 hidden units), we achieved the best accuracy for SciTail at 100 hidden units (Table 5.3). The convergence curve (Figure 5.2(c)) shows that dropout effectively prevents overfitting in the model with 100 hidden units in comparison to 300 units.

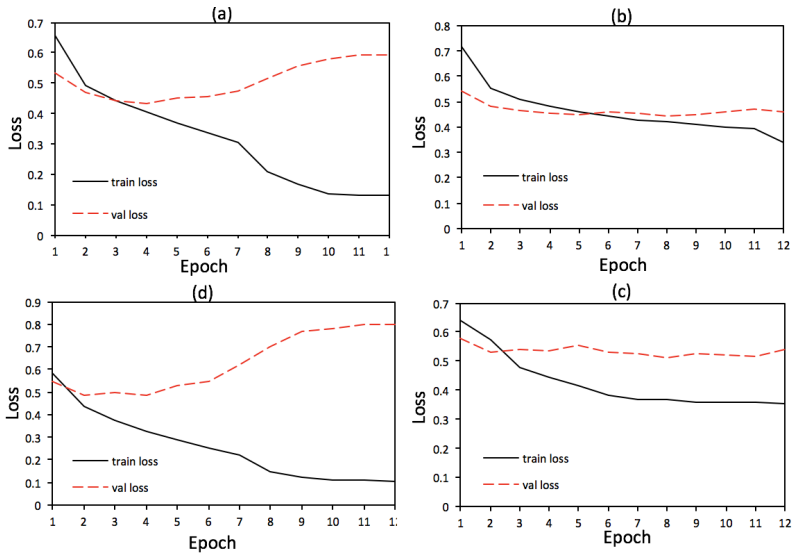


Figure 5.2: Convergence Curves: (a) Baseline Model for SNLI, (b) Best Model for SNLI, (c) 100 Unit Model for SciTail, (d) 300 Unit Model for SciTail.

Models	**Dataset**	**Dropout Rate (DR)**				
		0.1	0.2	0.3	0.4	0.5
Model 13	SciTail	76.72	76.25	72.58	**77.05**	74.22

Table 5.3: Accuracy for 100 unit model for the SciTail dataset.

5.4.2 Dropout Rate Effect on Accuracy and Dropout Location

We next explore the effect of varying dropout rates on the accuracy of the models and on different dropout locations identified in Table 5.1. Figure 5.3 illustrates varying dropout rates and the corresponding test accuracy for the SNLI dataset. We observe some distinct trends from the plot. First, the dropout rate and location do not significantly affect the accuracy of the Model 2, over the baseline model (Model 1), for the dropout rates of $0.1, 0.2, 0.3$ and 0.5. Model 8 shows the same pattern for the dropout rates of $0.1, 0.3, 0.4$ and 0.5. Second, in the dropout range [0.2 - 0.5], the dropout locations affect the accuracy of the models significantly. Increasing the dropout rate from 0.2 to 0.5 the accuracy of the Models 5 and 12 decreases significantly by 21.3% and 15.9% respectively. For most of the models, $(3, 4, 6, 7, 9$ and $10)$ the dropout rate of 0.5 decreases accuracy.

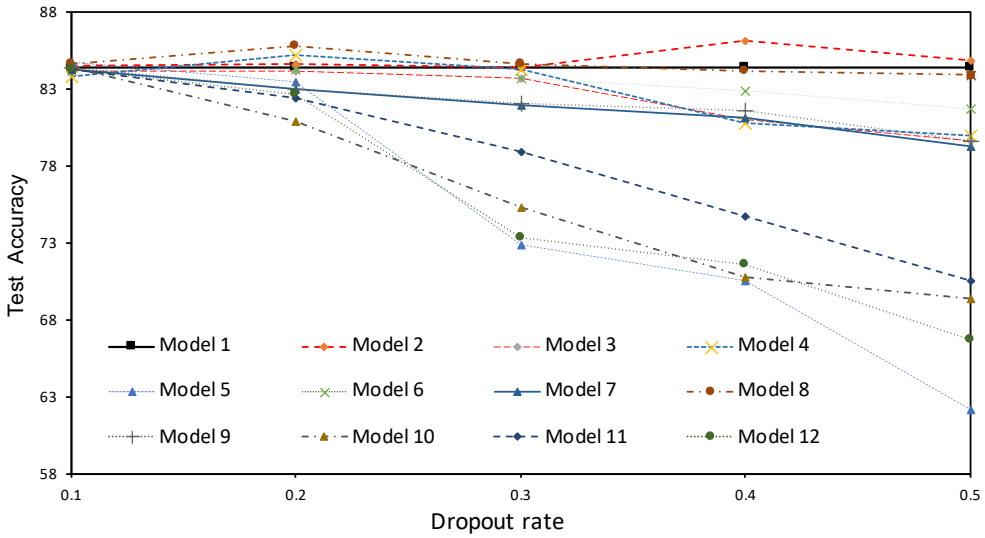


Figure 5.3: Plot showing the variation of accuracies for the CAM models identified in Table 5.1 across the dropout range for the SNLI dataset.

From the experiments on the SciTail dataset (Figure 5.4), we can see that model performance does not vary significantly by the location of dropout application and the variation in dropout rates, with the exception of Models 8 and 9.

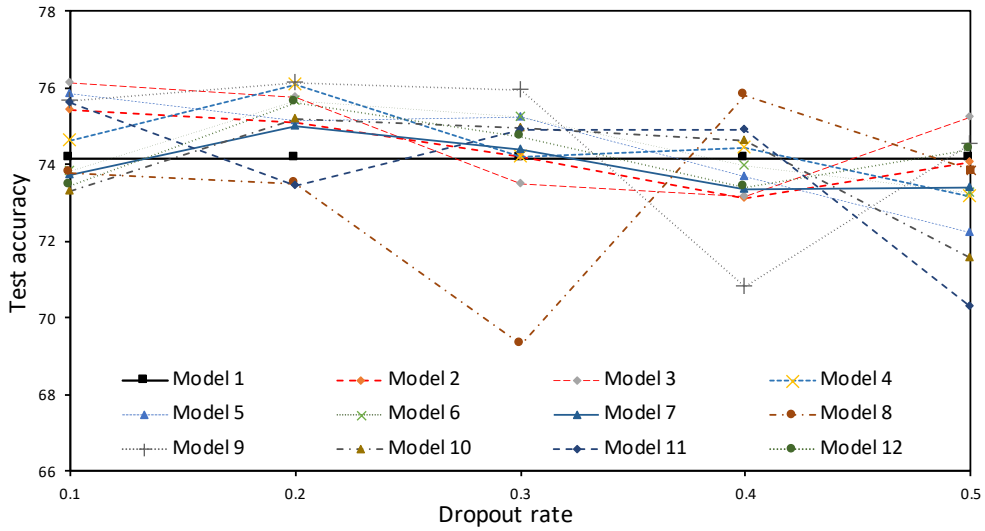


Figure 5.4: Plot showing the variation of accuracies for the CAM models identified in Table 5.1 across the dropout range for the SciTail dataset.

Finally, for almost all the experiments a large dropout rate (0.5) decreases the accuracy of the model. The dropout rate of 0.5 works for a wide rang of neural

networks and tasks [56]. However, our results show that this is not desirable for RNN-based NLI models. Based on our evaluations a dropout rate ranging from $[0.2 - 0.4]$ is advised.

5.5 Findings

Based on the CAM's empirical evaluations and analysis of the evaluation results, we draw the following observations for regularising RNN-based NLI models:

1. Embedding layer should be regularised for large datasets such as SNLI. For smaller datasets like SciTail regularising recurrent layer is an efficient option. The dropout injected noise at these layers prevents the higher fully connected layers from overfitting.

2. When regularising multiple layers, regularising a lower layer (embedding or recurrent; depending on the amount of data) with the inputs and outputs of the MLP layer should be considered. Regularising intermediate projection layers with a large number of parameters helps to prevent overfitting.

3. When dropout is applied at multiple feed-forward connections, it is almost always better to apply it at a lower rate within the range — $[0.2 - 0.4]$.

4. Given the high learning capacity of RNNs, an appropriate model size selection according to the amount of training data is essential. Dropout may independently be insufficient to prevent overfitting in the scenarios otherwise.

5.6 Finding's Validation

In this section, we validate the regularisation findings on another RNN-based NLI model, BiECAM (refer Chapter 4), by evaluating it with different dropout locations with varying dropout rates on the SNLI and SciTail datasets. We utilise our best performing BiECAM variant i.e. BiECAM + Aristo Tuple (refer Section 4.2), for the validation.

The BiECAM model utilises the ESIM model [91] (refer Section 2.5.2) as an underlying NLI model in the BiCAM framework. The BiECAM model consists of embedding, encoding, inter-attention, projection, composition, pooling^{nli}, pooling^{cs}, Factorised Bilinear Pooling (FBP), matching and the MLP layers (refer Section 2.2). The underlying ESIM model utilises the projection layer to prevent the model overfitting that may arise due to the increased number of parameters at the inference enhancement stage [91]. The pooling^{nli} represents the standard max and mean pooling layers of the ESIM model and the pooling^{cs} represents the standard max pooling layer of the commonsense encoding model.

To validate the regularisation findings on the BiECAM model in a simplified and effective manner, we identify the locations of dropout application in the BiECAM as illustrated in Table 5.4. To achieve the maximum accuracy for each of the regularisation settings depicted in the Table 5.4, we perform a grid search over the hyper-parameter combinations of the BiECAM model as detailed in Section 4.2. Note that BiECAM is a complex model with multiple layers and there are a large number of permutations of dropout application locations, however, exploring all these permutations is computationally infeasible. We believe the locations identified in Table 5.4 represent the best combinations of layers to validate the findings from the CAM model.

Model	**Layer**
Model 1	No Dropout (Baseline)
Model 2	Embedding
Model 3	Recurrent
Model 4	Embedding and MLP
Model 5	Recurrent and MLP
Model 6	Embedding, Projection, Matching and MLP
Model 7	Embedding, Projection, Pooling^{cs}, FBP, Matching and MLP
Model 8	Recurrent, Projection, Pooling^{cs}, FBP, Matching and MLP

Table 5.4: BiECAM model variants with the corresponding layers to the outputs of which dropout is applied.

5.7 Results and Discussion

Evaluation results of the different dropout application locations for the SNLI and SciTal datasets are shown in Table 5.5. Each model identified in Table 5.4 is evaluated with the dropout rates ranging from 0.1 to 0.5 with a granularity of 0.1. We consider the Model 1, with no dropout regularisation, as the baseline model.

Models	**Dataset**	**Dropout Rate (DR)**				
		0.1	0.2	0.3	0.4	0.5
Model 1	SNLI			87.10		
	SciTail			74.23		
Model 2	SNLI	87.14	87.30	**87.36**	87.20	86.90
	SciTail	74.25	74.39	**74.54**	74.10	74.26
Model 3	SNLI	**86.90**	86.75	86.23	85.88	83.14
	SciTail	74.90	**75.91**	75.45	74.62	74.26
Model 4	SNLI	87.20	87.34	**87.48**	87.21	86.61
	SciTail	74.98	75.42	**75.55**	74.11	73.76
Model 5	SNLI	86.21	86.33	**86.88**	86.29	85.66
	SciTail	74.80	75.68	**75.75**	75.43	74.73
Model 6	SNLI	87.48	87.54	**87.88**	86.14	85.23
	SciTail	75.33	75.76	**76.11**	75.65	75.02
Model 7	SNLI	87.88	88.23	**88.80**	87.77	86.10
	SciTail	75.26	75.90	76.41	**78.64**	76.23
Model 8	SNLI	**86.76**	86.29	85.84	84.20	83.12
	SciTail	75.21	75.47	75.87	**77.21**	75.21

Table 5.5: BiECAM model accuracies for different dropout locations with varying dropout rates for the SNLI and SciTail datasets. Bold numbers shows the highest accuracy for the model within the dropout range.

Dropout at Individual Layers Model 2 and 3 shows the results of the dropout application at the individual embedding and recurrent layers respectively. Although the performance of Model 2 does not significantly improve over the base model (Model 1), regularising the embedding layer for the SNLI dataset in Model 2 demonstrates a clear benefit when compared to regularising the recurrent layer in the Model

3. The performance of the model consistently degraded across the dropout rate range when the recurrent layer in Model 3 is regularised.

For the SciTail dataset, on the contrary, the model accuracies across the dropout rate range demonstrate that regularising the recurrent layer in the Model 3 is advantageous (especially for the lower dropout rates of 0.2 and 0.3) as compared to regularising the embedding layer in the Model 2. Further, among the Models, 2 and 3, the highest accuracy of 75.91% is attained when the recurrent layer is regularised at the dropout rate of 0.2.

As in the case of the CAM model, regularising individual embedding and recurrent layers did not achieve the overall highest accuracy, however, the evaluation results of Model 2 and Model 3 demonstrates the Finding 1 that regularising embedding layer is favourable for the larger datasets such as SNLI and regularising the recurrent layer is beneficial for the smaller dataset such as SciTail.

Dropout at Multiple Layers Models 4 through 8 in Table 5.5 shows the results when multiple layers are regularised. Figure 5.5 and 5.6 depicts the accuracies of the models in Table 5.5 against the dropout rates for the SNLI and SciTail datasets respectively.

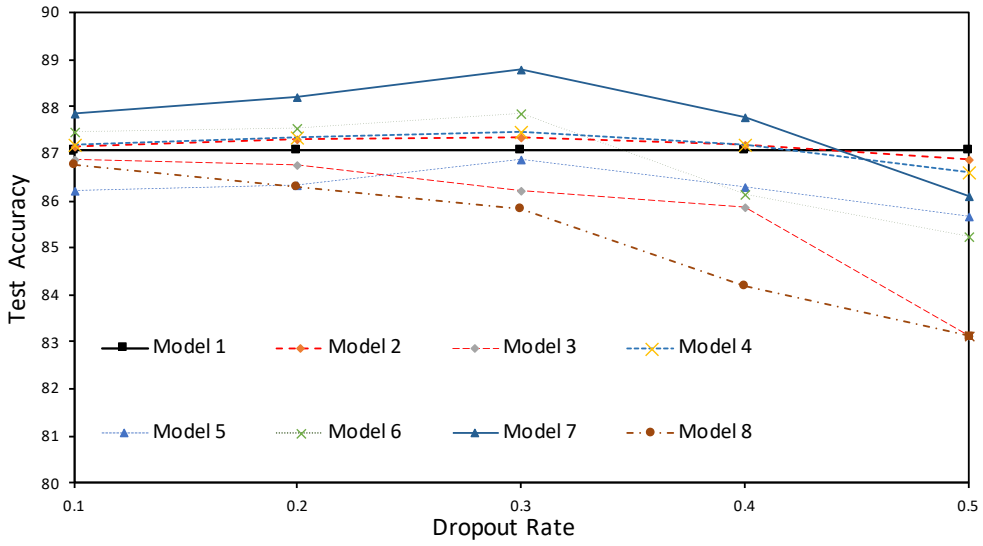


Figure 5.5: Plot showing the variation of accuracies for the BiECAM models identified in Table 5.4 across the dropout range for the SNLI dataset.

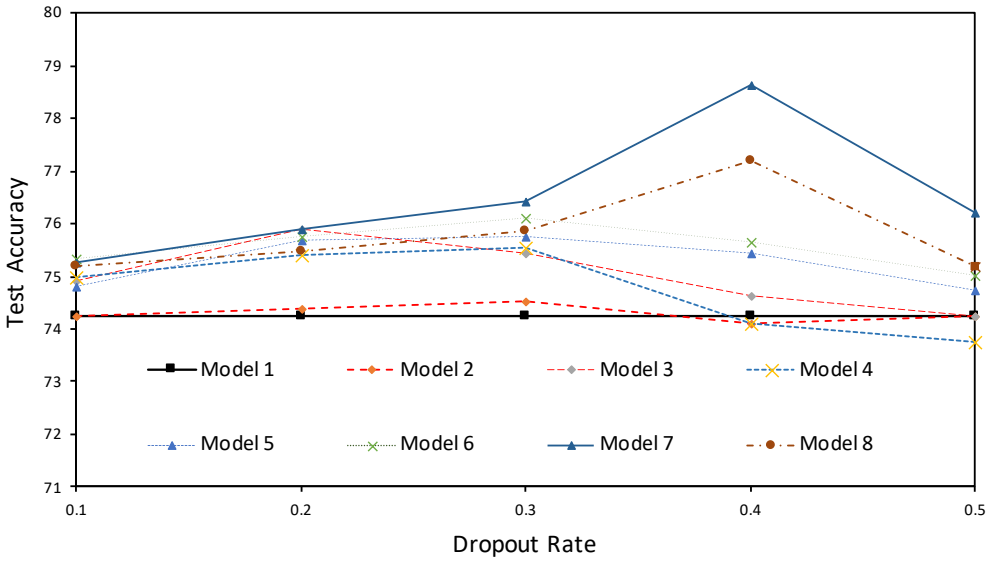


Figure 5.6: Plot showing the variation of accuracies for the BiECAM models identified in Table 5.4 across the dropout range for the SciTail dataset.

For both the SNLI and SciTail datasets, our model achieved the highest accuracy when regularising multiple layers in the regularisation setting of Model 7. The SNLI datasets attained the highest accuracy of 88.80% at the dropout rate 0.3 whereas the SciTail dataset attained the highest accuracy of 78.64% at the dropout rate of 0.4. Note that the Model 6, the dropout setting of the original ESIM model did not achieve the highest accuracy for our model. However, the highest accuracy attaining regularisation setting (Model 7), in addition to Pooling^{cs} and FBP layers, consists of all the layers originally regularised in the ESIM model (Model 6). Given that the Pooling^{cs} and FBP layers consists of a large number of parameters (refer Section 4.2), the achievement of the highest model accuracy in the regularisation setting of Model 7, demonstrates the significance of regularising the intermediate parameter intensive layers. This empirical result is consistent with the Finding 2 observed for the CAM.

For the smaller SciTail dataset, it is worth noting that all the studied regularisation settings improve the model performance. The model achieves higher accuracy than the baseline model, Model 1, in all of these settings. This highlights the significance of dropout regularisation in smaller datasets.

Regarding to the Finding 3, it can be viewed in the Figures 5.5 and 5.6 that the

performance of all the models on both the datasets did not remarkably improve with the dropout rate of 0.1 and the performance of all the models degraded significantly with the dropout rate of 0.5. Especially, in Models 3, when only the recurrent layer is regularised at the higher drop rate of 0.5, the model on the SNLI dataset achieved one of lowest accuracy of 83.14% among all the studied regularisation settings. The inadequacy of dropout rate 0.1 to prevent overfitting and the degradation of performance with the high dropout rate of 0.5, indeed suggests that the dropout rate range of [0.2 - 0.4] is advisable for the RNN-based NLI models.

Models	**Dataset**	**Dropout Rate (DR)**				
		0.1	0.2	0.3	0.4	0.5
Model 9	SciTail	74.41	74.29	75.38	**76.19**	73.48

Table 5.6: Accuracy for 200 unit model for the SciTail dataset.

To validate Finding 4, we evaluate our model, BiECAM, with a reduced hidden dimension size of 200 with the best performing regularisation setting of Model 7 on the SciTail dataset. The results are presented in the Table 5.6. Different to the observation for the CAM model, the performance of the BiECAM model did not improve due to the reduction in the model hidden states. We conjecture that the already optimised regularisation setting of the underlying ESIM model with the projection layer to reduce the model overfitting, the BiECAM model does not depend on the model size to prevent overfitting. Nonetheless, the ESIM model also employs different means (for example, the projection layer) to prevent model overfitting and does not solely depend on the dropout.

5.8 Guidelines for Dropout Application

We validated the findings observed from the analysis of the evaluation results of the CAM model on the BiECAM model. The evaluation results on the BiECAM model validate all the observed findings. Finally, we recommend the following guidelines for regularising the RNN-based NLI models via dropout:

- Embedding layer should be regularised for the big datasets like SNLI. For

the smaller datasets such as SciTail, regularising recurrent layer is an efficient option. The large number of parameters at the embedding layer owing to the large vocabulary of big datasets, causes the embedding layer to overfit. The dropout injected noise at the embedding or recurrent layers prevent the higher fully connected layers from overfitting.

- In the complex RNN-based NLI models, regularising the single embedding or recurrent layer is insufficient to prevent overfitting. When regularising multiple layers, regularising a lower layer (embedding or recurrent; depending on the amount of data) with the inputs and outputs of the MLP layer should be considered. Regularising intermediate projection layers with a large number of parameters helps to prevent overfitting.

- The higher dropout rates are not advisable for RNN-based NLI models, especially for the recurrent layer in the model. The dropout when applied to multiple feed-forward connections, it is almost always better to apply it at a lower rate within the range – $[0.2 - 0.4]$.

- RNNs have high learning capacity [216]. Dropout may independently be insufficient to prevent overfitting of the RNN-based NLI models. Different crucial factors such as reducing the dimensionality of the hidden layers or employing intermediate projection layers should be considered to prevent overfitting.

5.9 Conclusions

In this chapter, we explored an understudied area of high empirical significance — the application of the dropout regularisation in the RNN-based NLI models. We exhaustively evaluated the different locations for the dropout application in our RNN-based NLI model, CAM. Further at each location, we evaluated the dropout rate in the range of $[0.1 - 0.5]$ with a granularity of 0.1.

Based on the analysis of the empirical evaluations, we highlighted the findings for the suitable dropout locations and an appropriate range of the dropout rates in the model. Additionally, we validated the findings on our another RNN-based NLI

model, BiECAM. Finally, the validated findings are recommended for the application of the dropout in RNN-based NLI models.

Our guidelines highlight the significance of regularising the parameter intensive embedding layer for larger datasets such as SNLI and regularising the recurrent layer for the smaller datasets such as SciTail. Further, in complex models, when regularising multiple layers, regularising a lower layer (embedding or recurrent; depending on the amount of data) with the inputs and outputs of the MLP layer should be considered. Also, regularising intermediate projection layers with a large number of parameters helps to prevent overfitting.

The dropout rates are also crucial to the use of dropout regularisation. Our empirical evaluations suggest that higher dropout rates are not suitable for the high performance of RNN-based NLI models. The dropout rates in the range of $[0.1 - 0.4]$ are advisable for RNN-based NLI models. Further, owing to the high learning capacities of the RNNs, the sole reliance on the dropout regularisation is not recommended to prevent the overfitting in the RNN-based NLI models.

As the CAM model is a close representation of the RNN-based models in the NLI literature, we believe that the empirically evaluated and validated set of guidelines proposed in this chapter will also benefit other RNN-based NLI models.

In the next chapter, we focus on utilising the external knowledge to augment the grounding of NLI models in real-world knowledge. Specifically, we address the shortcomings of the BiCAM framework, proposed in Chapter 4, by the use of contextual word representations from the state-of-the-art RNN-/CNN-free, BERT model [16].

CHAPTER 6

ExBERT: An External Knowledge Enhanced BERT for Natural Language Inference

The BiCAM framework proposed in Chapter 4 demonstrated an improved reasoning and inferencing capabilities as a result of external knowledge incorporation. Further, the experimental results demonstrated that when the SciTail dataset is supplied with the contextually relevant external knowledge from the Aristo Tuple KG, the models attained superior performance. Motivated by the results and the recent developments in the field of learning the contextual word representations [16, 61, 84, 85], in this chapter, we propose several improvements to the BiCAM framework as well as to the existing models [5, 39–41, 82] utilising the external knowledge. Further, we apply the proposed improvements to the state-of-the-art BERT model to address its limitation of lack of grounding in real-world knowledge.

As discussed in Section 2.2.2, recently, PTLMs such as ELMO [60], OpenAI GPTs [61, 84, 85] and BERT [16] has achieved impressive performance improvements on a wide range of NLP tasks. These models are trained on large amounts of raw texts using a self-supervised language modelling objective. However, they lack grounding in real-world knowledge and are often unable to remember real-world facts when required [217, 218]. Investigations into the learning capabilities

of PTLMs reveal that the models fail to recall facts learned at training time, and do not generalise to rare/unseen entities [219]. Knowledge probing tests [218] on the commonsense knowledge of ConceptNet reveals that PTLMs such as BERT have critical limitations when solving problems involving commonsense knowledge. Hence, infusing the external real-world commonsense knowledge can enhance the language understanding capabilities of PTLMs and subsequently the performance on the complex reasoning tasks such as NLI.

In Chapter 4, we highlighted three main challenges for the incorporation of external knowledge into the NLI models. We reiterate the challenges for the ease of reading and highlight the shortcomings of the current literature in addressing these challenges.

- **Structured Knowledge Retrieval:** Given a premise-hypothesis pair how to effectively retrieve specific and relevant external knowledge from the massive amounts of data in KGs [220]. Existing models [5, 39–41], including our BiCAMs, use heuristics and word surface forms of the premises and hypothesis which may be biased and the retrieved knowledge may not always be contextually relevant for reasoning over premise-hypothesis pair.

- **Encoding Retrieved Knowledge:** Learning the representations of the retrieved external knowledge amenable to be fused with the representations of premise-hypothesis is challenging. Various KG embedding techniques [50], such as HolE [207] in BiCAMs, TransE [221] in KIM [41] and DKRL [222] in Convolution-based KG [82] models are employed to learn these representations. However, while learning these embeddings, the embeddings are only required to be valid within the individual KG fact and hence might not be predictive enough for the downstream tasks [50]. Moreover, the inexpressive KG embeddings may produce a cascading error effect during the training of the downstream task.

- **Feature Fusion:** How to fuse the learned external knowledge features with the premise-hypothesis embeddings. This feature fusion requires considerable efforts in learning the fused representation via special techniques such fac-

torised bilinear pooling in BiCAMs or requires substantial NLI model adaptations as in NSnet [40], ConSeqNet [114] and KIM [41] models with marginal performance gains (see Section 2.5.2).

To overcome the abovementioned shortcomings in addressing the challenges of external knowledge incorporation in NLI models and to improve the grounding of BERT model in the real-world knowledge, we propose, **ExBERT** – an External knowledge enhanced BERT model.

The ExBERT utilises the contextual word representations from the BERT model to retrieve external knowledge that is contextually relevant to the premise and hypothesis. Further, it incorporates the retrieved external knowledge to the BERT model to improve BERT's grounding in the real-world knowledge and reinforce its reasoning and inference capabilities for NLI. Thus, ExBERT utilises BERT for retrieving the contextually relevant external knowledge as well as to reason over the premise and hypothesis. The aim here is to take full advantage of the contextual word representations obtained from the PTLMs, the state-of-the-art pre-trained BERT encoder and the real-world commonsense knowledge from KGs.

Our approach has several benefits.

- First, for **structured knowledge retrieval**, we utilise contextual word representations from the BERT model to retrieve the most contextually similar external knowledge from the KGs. Further, we retrieve the external knowledge based on the bigrams, trigrams, fourgrams and the average of the whole of contextual BERT representation of the premise and hypothesis. Different from word-based knowledge retrieval this approach retrieves fine-grained contextually similar external, knowledge, avoids any biases of heuristic knowledge retrieval and requires no feature engineering.

- Second, in contrast to previous approaches, for **encoding the retrieved knowledge**, we again employ the BERT encoder to learn the contextual representations of the external knowledge. This BERT encoder shares parameters with premise-hypothesis BERT encoder and learns the contextual embeddings of external knowledge in the same embedding space.

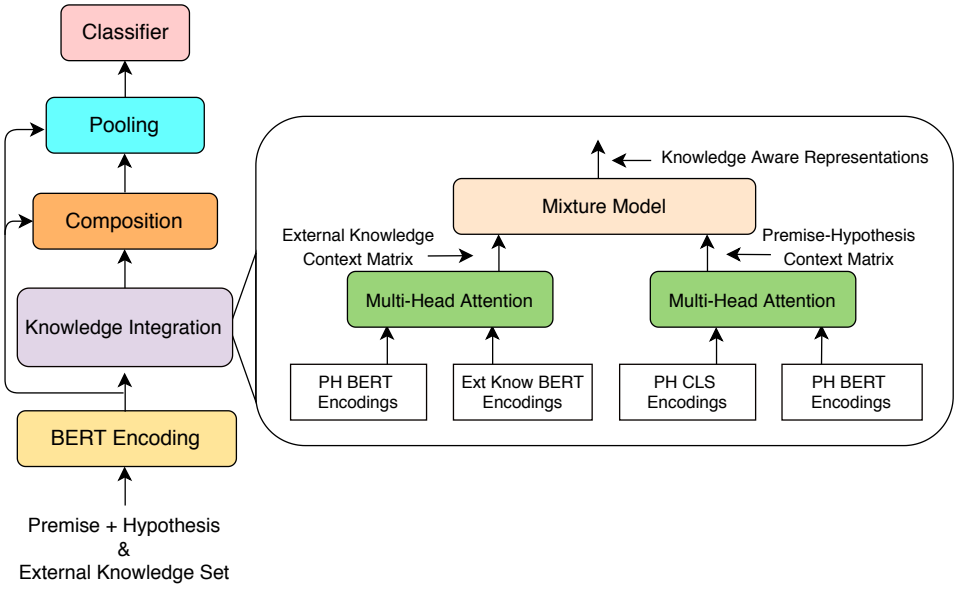


Figure 6.1: A high-level view of the ExBERT architecture.

- Third, the parameter sharing facilitates the **feature fusion** as the premise-hypothesis and the external knowledge representations are in the same embedding space, the representations can be fused via simple techniques such as summation or concatenation, eschewing the need for complex feature fusion techniques.

As depicted in Figure 6.1, given a premise-hypothesis pair and the set of retrieved external knowledge, first the "*BERT Encoding layer*" learns the deep contextual representations of the premise-hypothesis and each retrieved external knowledge using the BERT encoder. The "*Knowledge Integration Layer*" then adaptively learns to incorporate the external knowledge into the learned premise-hypothesis representations via a mixture model. The "*Composition Layer*" fuses the output of knowledge integration layer with the original premise-hypothesis contextual representations to create knowledge enhanced representations. The "*Pooling Layer*" creates fixed-length representations from the original premise-hypothesis contextual representations and knowledge enhanced representations. Finally, the "*Classifier Layer*" predicts the final inference class.

The main contributions of this chapter are:

- We propose a new approach, ExBERT, to incorporate external knowledge in contextual word representations. ExBERT outperforms the state-of-the-art NLI models.
- We investigate and demonstrate the feasibility of using contextual word representations for encoding external knowledge obviating learning specialised KG embeddings such as TransE or HolE. To the best of our knowledge, this is the first study of its kind, indicating a potential future research direction.
- We introduce a new external knowledge retrieval mechanism capable of retrieving fine-grained contextually relevant external knowledge from KGs.

6.1 Methodology

ExBERT architecture is depicted in Figure 6.1. In this section, we describe the key components of ExBERT and their detailed implementation including the model architecture in Section 6.1.2. We start by describing the contextual representation based external knowledge retrieval procedure in Section 6.1.1.

6.1.1 External Knowledge Retrieval: Selection and Ranking

Retrieval and preparation of contextually specific and relevant information from knowledge graphs are complex and challenging tasks. Different from the previous approaches that use word surface forms to retrieve external knowledge, we use the cosine similarity between the contextual representations of the premise-hypothesis words and external knowledge. The external knowledge for the premise and hypothesis is retrieved individually and is merged to create a final set of retrieved knowledge at the end of ranking step as described below. Below we explain the procedure for the premise. The same procedure is applied to the hypothesis. The output of external knowledge retrieval is the set of contextually relevant external knowledge sentences for the premise and hypothesis. We divide the external knowledge retrieval process into two parts: Selection and Ranking.

Selection We first filter the stop words from the premise. Then we retrieve all the KB triples that contain the tokens of the premise as one of the words in the head entity of KG triples. For example, for the token "*speaking*" one of the retrieved KG fact is "*public_speaking IsA speaking*". The retrieved triples are converted to external knowledge sentences. For example, the previous triple is transformed into the sentence – "*public speaking is a speaking*". During the conversion some of the triples may produce incomplete sentences. For example, the triple "*hammer UsedFor flatten_metal_on_anvil*", produces "*hammer used for flatten metal on anvil*" instead of "*hammer is used for flattening metal on anvil*". Recent research [223, 224] on assessing the syntactic abilities of pre-trained language models such as BERT, suggests that these models are robust to ungrammatical sentences due to pre-training on large text corpus. Moreover, in general, the best practices of stemming, lemmatisation and stop-word removal further systematically turns the input text ungrammatical without any performance degradation. Hence, the conversion of triples to ungrammatical sentences does not significantly impact the model performance.

The selection process retrieves a large number of external knowledge sentences, which are not all relevant to the context of the premise. We filter the selected external knowledge sentences in the ranking step.

Ranking The ranking step ranks the selected external knowledge sentences according to the contextual similarity to the fine-grained context of the premise. Specifically, given the BERT generated context-aware representation of the premise tokens, we group all the bigrams of the representations. Each group of the bigram representation is averaged, and the cosine similarity is calculated with the average of the BERT representation of each of the selected external knowledge sentence (retrieved in selection step). For each bigram, we choose the external knowledge sentence with the highest cosine similarity score.

To capture the fine-grained context of the premise, we repeat the ranking step with the trigrams, fourgrams, and the average of the whole premise contextual BERT representations and retrieve the external knowledge sentence with the highest cosine similarity for each of the grams. The final set of retrieved external knowledge is

created by merging the external knowledge sentences retrieved for the premise and hypothesis by removing duplicates.

6.1.2 Model Architecture

BERT Encoding Layer This layer uses the BERT encoder to learn the context-aware representations of the premise-hypothesis pair and the set of retrieved external knowledge.

Specifically, given the sentences, premise $\acute{P} = \{p_i\}_{i=1}^{n}$, hypothesis $\acute{H} = \{h_j\}_{j=1}^{m}$, and the set of external knowledge $EXT = \{\{e_r\}_{r=1}^{l}\}_{v=1}^{t}$, where r is the number of tokens in the external knowledge sentence and t is the number of retrieved external knowledge sentences. For encoding the premise and hypothesis, we input $\acute{P}$ and $\acute{H}$ to BERT in the following form

$$S^{ph} = [\langle \text{CLS} \rangle,\ \acute{P},\ \langle \text{SEP} \rangle,\ \acute{H},\ \langle \text{SEP} \rangle] \tag{6.1}$$

$$H = \text{BERT}(S^{ph}) \in \mathbb{R}^{(n+m+3)\times h} \tag{6.2}$$

where $\langle \text{SEP} \rangle$ is the token separating $\acute{P}$ and $\acute{H}$, $\langle \text{CLS} \rangle$ is the classification token, and h is the dimension of the hidden states (768 for the BERT model we employ). When the BERT model is fine-tuned for the downstream task, the fine-tuned hidden state vector ($\mathbf{h}^{cls}$) corresponding to the classification token is used as the aggregate representation for the sequence.

For each of the external knowledge sentence in the set EXT, we generate the context-aware representations using the same BERT encoder as used for premise-hypothesis above as follows

$$S^{ext\_know} = [\langle \text{CLS} \rangle, e_1, \ldots, e_l, \langle \text{SEP} \rangle] \tag{6.3}$$

$$E^{ext\_know} = \text{BERT}(S^{ext\_know}) \tag{6.4}$$

$$\mathbf{e} = \text{MeanPooling}(E^{ext\_know}) \tag{6.5}$$

where S^{ext_know} is the sequence created by inserting external knowledge sentence between the $\langle \text{CLS} \rangle$ and $\langle \text{SEP} \rangle$ tokens as required by the BERT model, E^{ext_know} is

the matrix of contextual word representations for the external knowledge sequence S^{ext_know}. The averaged contextual word representation ($\mathbf{e} \in \mathbb{R}^h$) generated for each of the (t) retrieved external knowledge sentence are stacked to create the context-aware matrix, $E \in \mathbb{R}^{t \times h}$.

Knowledge Integration Layer This layer integrates external knowledge into the premise-hypothesis contextual representations by means of multi-head dot product attention. The layer uses a mixture model [205] to allow a better trade-off between the context from external knowledge and the premise-hypothesis context. The mixture model learns two parameter matrices, A and B, that weigh the importance of premise-hypothesis context and the context from external knowledge.

Muliti-head Attentions To measure the importance of external knowledge to each context-aware premise-hypothesis representation, we apply multi-head dot product attention [78] between the context-aware representations of external knowledge and that of premise-hypothesis.

In multi-head dot product attention, the context-aware representations are projected linearly to generate the queries, keys and values. As we use the multi-head attention to highlight the external knowledge important to premise-hypothesis context, premise-hypothesis representation (H) generates the query matrix (H^q) via linear projection and the two linear projections of external knowledge representation (E) generate the keys (E^k) and values (E^v). The attention function is defined as

$$\text{Attention}(H^q, E^k, E^v) = \text{softmax}(\frac{H^q {E^k}^T}{\sqrt{h_k}})E^v \tag{6.6}$$

Then the multi-head attention is

$$C_{ph}^{ext} = \text{MH}(H^q, E^k, E^v) = \text{Concat}(\text{head}_1, \ldots, \text{head}_\text{h})W^o \tag{6.7}$$

where $\text{head}_\text{i} = \text{Attention}(H^q W_i^q, E^k W_i^k, E^v W_i^v)$ and W_i^q, W_i^k, W_i^v, and W^o are projection matrices and i is the number of attention heads (12 in our case). The output of multi-head attention, $C_{ph}^{ext} \in \mathbb{R}^{(m+n+3)\times h}$ is an attention-weighted context matrix measuring the importance of the external knowledge context to each of the

context-aware premise-hypothesis representation.

Similarly, to measure the importance of each premise-hypothesis BERT representation (H) to the aggregate premise-hypothesis representation (hidden representation $\mathbf{h}^{cls}$ corresponding to CLS token), we apply the multi-head attention between $\mathbf{h}^{cls}$ token representation and H as

$$\text{Attention}(C^q_{cls}, H^k, H^v) = \text{softmax}(\frac{C^q_{cls}{H^k}^T}{\sqrt{h_k}})H^v \tag{6.8}$$

where $C^q_{cls} \in \mathbb{R}^{(m+n+3)\times h}$ is a matrix obtained by repeating $\mathbf{h}_{cls}$ hidden state $(n+m+3)$ number of times. The multi-head attention is calculated similar to (Eq. 6.7) that outputs a context matrix $C^{cls}_{ph} \in \mathbb{R}^{(m+n+3)\times h}$. The output matrix C^{cls}_{ph} is an attention-weighted context matrix measuring the importance of each of the premise-hypothesis representation to the aggregate premise-hypothesis representation.

Mixture Model The mixture model learns a trade-off between the premise-hypothesis context and the context from external knowledge and is defined as

$$M = AC^{ext}_{ph} + BC^{cls}_{ph} \tag{6.9}$$

where A and B are the parameter metrices, learned with a single layer neural network and $A + B = J \in \mathbb{R}^{(n+m+3)\times 1}$, J is a matrix of all ones. The parameters A and B learn to balance the proportion of incorporating the premise-hypothesis context and the context from external knowledge. Each of the representations in $M \in \mathbb{R}^{(m+n+3)\times h}$ can be regarded as a knowledge aware state representation that encodes external knowledge context information with respect to the context of each of the premise-hypothesis representation.

Composition Layer We compose the knowledge state representation (M) to the corresponding premise-hypothesis representation to obtain knowledge-aware matrix $\widehat{H}$ as

$$\widehat{H} = H + M \tag{6.10}$$

Pooling Layer The pooling layer creates a fixed-length representation from premise-hypothesis representations H and the knowledge-aware representations $\widehat{H}$ (refer Section 2.2.6). We apply the standard mean and max polling mechanisms as

$$\mathbf{h}^{mean} = \text{MeanPooling}(H) \qquad \mathbf{h}^{max} = \text{MaxPooling}(H) \tag{6.11}$$

$$\hat{\mathbf{h}}^{mean} = \text{MeanPooling}(\widehat{H}) \qquad \hat{\mathbf{h}}^{max} = \text{MaxPoolling}(\widehat{H}) \tag{6.12}$$

Classification Layer We classify the relationship between premise and hypothesis using a MLP classifier (refer Section 2.2.8). The matching input (refer Section 2.2.7) to the MLP is the concatenation of pooled representations as

$$\mathbf{f}^{final} = [\mathbf{h}^{mean}; \hat{\mathbf{h}}^{mean}; \mathbf{h}^{max}; \hat{\mathbf{h}}^{max}] \tag{6.13}$$

The MLP consists of two hidden layers with tanh activation and a softmax output layer to obtain the probability distribution for each class. The network is trained in an end-to-end manner using multi-class cross-entropy loss.

6.2 Experiments

6.2.1 Datasets

NLI & KGs The key contribution of this chapter is the unique method to incorporate external knowledge into the pre-trained BERT representations. ExBERT is capable of incorporating knowledge from any external knowledge source that allows the knowledge to be retrieved, given an entity as input. This includes KBs with *(head, relation, tail)* graph structure, KBs that contain only entity metadata without a graph structure and those that combine both a graph and entity metadata.

In this work, we retrieve external commonsense knowledge from ConceptNet [206] for evaluating ExBERT on the SNLI [3] and SciTail [5] benchmark datasets, and from the science domain-targeted KG, Aristo Tuple [65] for evaluation on science domain SciTail dataset (refer Section 2.3).

ConceptNet is a multilingual KG comprising of 83 languages. We pre-process

the ConceptNet data to retrieve the facts with head and tail entities in the English language. The final pre-processed ConceptNet that we retrieve the external knowledge from contains $3,098,816$ ($\approx$ 3M) commonsense facts connected by 47 relations. Aristo Tuple is an English language KG that contains $294,000$ science domain facts connected with 955 unique relations. We search the whole Aristo Tuple KG to retrieve relevant external knowledge.

6.2.2 Experimental Setup

Following our external knowledge retrieval mechanism discussed in Section 6.1.1, we first retrieve the external knowledge from ConceptNet and Aristo Tuple KGs for the SNLI and SciTail datasets via selection and ranking steps. In the ranking step, the English uncased $\text{BERT}_{\text{BASE}}$ [16] model is employed in feature extraction mode (i.e. without fine-tuning) to learn the contextual representations of the premise, the hypothesis and to each of the selected KG triple sentences. We then use the retrieved external knowledge to train the following three variants of ExBERT.

Models We used the English uncased $\text{BERT}_{\text{BASE}}$ to train three variants of ExBERT: Two ExBERT+ConceptNet models on SNLI and SciTail respectively and one ExBERT+AristoTuple model on SciTail. The models utilise the external knowledge from the KG their name is suffixed.

Comparison Setting ExBERT is compared with the state-of-the-art pre-trained models on the leaderboards of the SNLI and SciTail datasets that utilise external knowledge. The baselines vary from the original $\text{BERT}_{\text{BASE}}$ model to the models exploiting any external supervision. For example, the baseline SemBERT [17] enhances the BERT contextual representations with explicit contextual semantics clues from an external pre-trained semantic role labeler [60] (refer Section 2.5.2).

Training Details ExBERT is implemented in PyTorch using the base implementation of BERT[1]. The underlying BERT is initialised with the pre-trained BERT parameters and follows the same fine-tuning procedure as the original BERT. During training, the pre-trained BERT parameters are fine-tuned with the other ExBERT

[1] `https://github.com/huggingface/transformers` – As on July 1, 2020.

parameters. We use the Adam optimiser [202] with the initial learning rate fine-tuned from {8e-6, 2e-5, 3e-5, 5e-5} and warm-up rate of 0.1. The batch size is selected from $\{16, 24, 32\}$. The maximum number of epochs is chosen from $\{2, 3, 4, 5\}$. Texts are tokenised using word pieces, with a maximum length of 40 for SNLI, 60 for SciTail, and 15 for external knowledge. The hyper-parameters are fine-tuned on the development set of each NLI dataset.

6.3 Results

The results of top-performing models on the SNLI[2] and SciTail[3] dataset leaderboards are summarised in Table 6.1 and Table 6.2 respectively.

Models with $\mathbf{BERT_{BASE}}$ as Base Model	
NLI Model	**Test Acc(%)**
$\text{BERT}_{\text{BASE}}$ + SRL [152]	89.6
OpenAI GPT [61]	89.9
$\text{BERT}_{\text{BASE}}$ [123]	90.5
$\text{BERT}_{\text{BASE}}$ [18]	90.8
BERT+LF [149]	90.5
$\text{SemBERT}_{\text{BASE}}$ [17]	91.0
$\text{MT} - \text{DNN}_{\text{BASE}}$ [18]	91.1
MT-DNN+LF [149]	91.1
Models with $\mathbf{BERT_{LARGE}}$ as Base Model	
$\text{BERT}_{\text{LARGE}}$ [18]	91.0
$\text{BERT}_{\text{LARGE}}$ + SRL [152]	91.3
$\text{SemBERT}_{\text{LARGE}}$ [17]	91.6
$\text{MT} - \text{DNN}_{\text{LARGE}}$ [18]	91.6
ExBERT+ConceptNet (Ours)	**91.5**

Table 6.1: Results on SNLI dataset. State-of-the-art NLI models accuracy compared to the proposed ExBERT model. ExBERT utilises ConceptNet KG for external knowledge.

[2] `https://nlp.stanford.edu/projects/snli/` – As on July 1, 2020.
[3] `https://leaderboard.allenai.org/scitail/submissions/public` – As on July 1, 2020.

Models with $\text{BERT}_{\text{BASE}}$ as Base Model	
NLI Model	**Test Acc(%)**
OpenAI GPT [61]	88.3
$\text{BERT}_{\text{BASE}}$ [18]	92.5
BERT+LF [149]	92.8
$\text{MT} - \text{DNN}_{\text{BASE}}$ [18]	94.1
MT-DNN+LF [149]	94.3
Models with $\text{BERT}_{\text{LARGE}}$ as Base Model	
$\text{BERT}_{\text{LARGE}}$ [18].	94.4
$\text{MT} - \text{DNN}_{\text{LARGE}}$ [18]	95.0
ExBERT+ConceptNet (Ours)	**95.2**
ExBERT+AristoTuple (Ours)	**95.9**

Table 6.2: Results on SciTail dataset: State-of-the-art NLI models accuracy compared to the proposed ExBERT model. ExBERT uses ConceptNet and AristoTuple KGs for external knowledge.

Results on SNLI On the SNLI dataset, as shown in Table 6.1, the performance of the state-of-the-art models is highly competitive. We observe that ExBERT outperforms all the existing baselines creating a new state-of-the-art result on the SNLI dataset and pushing the benchmark to 91.5% within the models using $\text{BERT}_{\text{BASE}}$ as the base model. ExBERT achieves a maximum performance improvement of +1.9% over the previous state-of-the-art $\text{BERT}_{\text{BASE}}$ + SRL [152] baseline.

Among the models built on $\text{BERT}_{\text{LARGE}}$ with more than 340M million parameters [16], our ExBERT with $\text{BERT}_{\text{BASE}}$ (110M parameter) remarkably outperforms the $\text{BERT}_{\text{LARGE}}$ and $\text{BERT}_{\text{LARGE}}$ + SRL [152] models with the absolute improvements of 0.5% and 0.2% respectively, and is able to match the performance of $\text{SemBERT}_{\text{LARGE}}$ [17] and $\text{MT} - \text{DNN}_{\text{LARGE}}$ [18] models.

Results on SciTail On the SciTail dataset (Table 6.2), ExBERT outperforms all the existing models including the models built on $\text{BERT}_{\text{LARGE}}$ model. Our best performing model, ExBERT+AristoTuple demonstrate an absolute improvement of 7.6% over the established baseline of OpenAI GPT [61]. Moreover, using only $\text{BERT}_{\text{BASE}}$ as the underlying model, our ExBERT+AristoTuple outperforms

$\text{BERT}_{\text{LARGE}}$ based $\text{MT} - \text{DNN}_{\text{LARGE}}$ [18] model by 1.9%.

We observe higher performance improvements on the smaller SciTail dataset which demonstrates that incorporating external knowledge helps the model with small training data. Further, we observe that ExBERT attains higher accuracy when external knowledge is incorporated from the science domain-specific KG, Aristo Tuple as compared to when external knowledge is added from the commonsense KG, ConceptNet. The specialised scientific knowledge in Aristo Tuple is more beneficial to the SciTail dataset.

6.4 Analysis

6.4.1 Number of External Features

To investigate the effect of incorporating various numbers of external knowledge features, we vary the number of external knowledge sentences input to ExBERT. Particularly, we are interested in answering the question: How many commonsense features are required for the optimal model performance? Figure 6.2 illustrates the results of the experiment.

For SNLI ExBERT achieves the highest accuracy (91.5%) using 11 external knowledge sentences. We observe a decrease in accuracy when increasing the number of external knowledge sentences after 11. The fewer number of external knowledge sentences required, compared to the SciTail dataset, to achieve the maximum accuracy on the SNLI dataset, is attributed to the limited linguistic and semantic variation and the short average length of stop-word filtered premise (7.35 for entailment and neutral class) and hypothesis (3.61 for entailment and 4.45 for neutral class) [5] of the SNLI dataset, which limits its ability to fully extract and exploit external KG knowledge.

For SciTail ExBERT when evaluated using the general commonsense knowledge source ConceptNet, requires a relatively high number of external knowledge sentences (13) to achieve the maximum accuracy. This is due to the higher syntactic

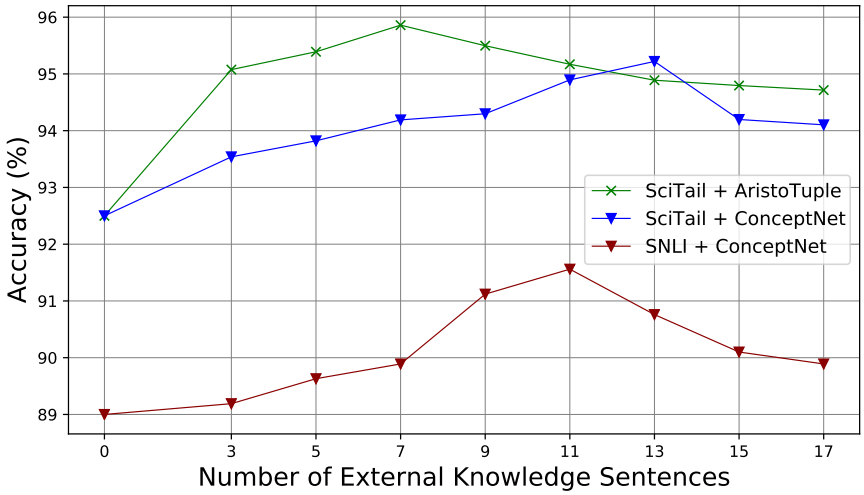


Figure 6.2: ExBERT accuracy with the varying number of external knowledge sentences from the ConceptNet and Aristo Tuple KGs.

and semantic complexity of the SciTail dataset, that needs more knowledge to reason. However, when evaluated with the domain-specific Aristo Tuple KG, the model achieve the highest accuracy with fewer (7) external knowledge sentences. To reiterate, domain specific knowledge in Aristo Tuple improves the model performance with less external knowledge.

6.4.2 Qualitative Analysis

Case Study

This section provides the case study of different premise-hypothesis pairs and the corresponding external knowledge, to vividly show the effectiveness of ExBERT in adaptively identifying the relevant features from the supplied external knowledge. Recall that given a context-aware representation of premise-hypothesis token, the relevance of the retrieved external knowledge in E is measured by the multi-head attention defined in Equation (6.6). We average the attention weights of all the heads and plot a heat map, visualising these attention weights.

Figure 6.3 presents the heat map showing the attention of premise-hypothesis tokens to the retrieved external knowledge sentences from ConceptNet. In Figure 6.3,

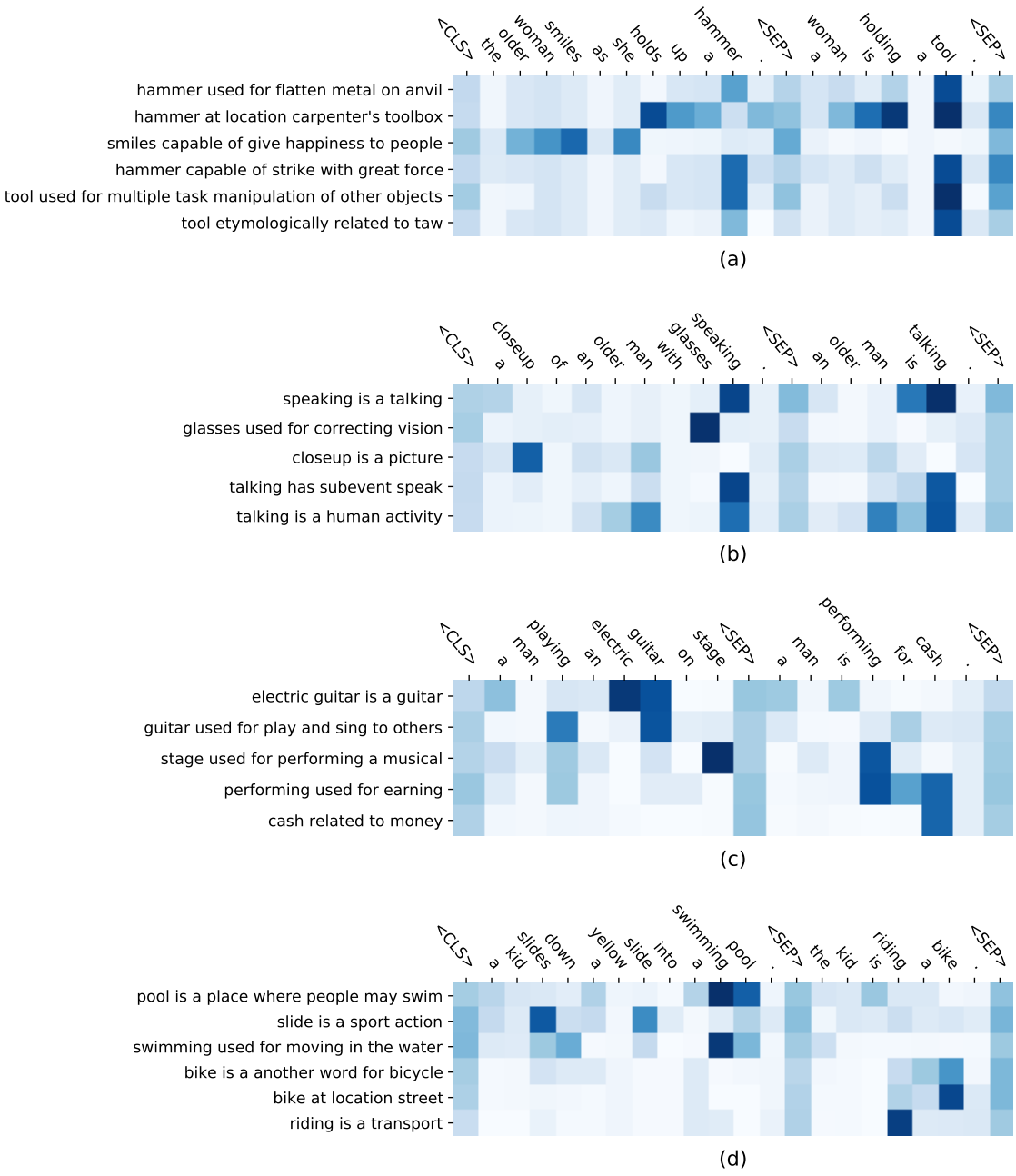


Figure 6.3: Case Study. Visualisation of ExBERT's attention between external knowledge from ConceptNet (y axis) and SNLI premise-hypothesis pair tokens (x axis).

we can see, these attention distribution is quite meaningful. For example, in attention heat map (a), the external knowledge *smiles capable of give happiness to people* for the phrase *older woman smiles* in the premise. Similarly, the external knowledge with the word *hammer* is attended to whenever the word *tool* appears in the premise and hypothesis. In Figure 6.3(b), the attention distribution is also explanatory, with

the "*speaking*" and "*talking*" attending mainly to the retrieved external knowledge "*speaking is talking*". Similarly, the tokens "*speaking*" "*talking*" and "*man*" attends to "*talking is a human activity*". In Figure 6.3(c) among the other attentions, the most prominent can be observed between the tokens "*performing for cash*" and the external knowledge sentence "*performing used for earning*".

Attending to the relevant external knowledge demonstrates the ExBERT's ability to effectively utilise the retrieved external knowledge based on the context from the premise and hypothesis. In the next section, we study the efficacy and quality of retrieved external knowledge.

Retrieved External Knowledge: Efficacy and Quality

We investigate the effectiveness and quality of the retrieved external knowledge. Table 6.3 presents the SNLI and SciTail test set premise-hypothesis pairs which the baseline $\text{BERT}_{\text{BASE}}$ model predicted incorrectly. In ExBERT, these premise-hypothesis pairs when supplied with the external knowledge (EXT) retrieved from our contextual similarity-based knowledge retrieval mechanism (Section 6.1.1), the pairs were predicted correctly. ExBERT enriches the premise-hypothesis contexts with retrieved external knowledge and augments the reasoning capabilities of the baseline $\text{BERT}_{\text{BASE}}$ model. Further, Table 6.3 illustrates the retrieved external knowledge (EXT) for the premise-hypothesis pair, we observe that most of the retrieved external knowledge sentences are contextually relevant to the premise-hypothesis and our knowledge retrieval mechanism is effective in retrieving the external knowledge beneficial to reason about inference.

6.5 Conclusions

In this chapter, we introduced ExBERT to enrich the contextual representation of BERT with real-world commonsense knowledge from external knowledge sources and to enhance its language understanding and reasoning capabilities for NLI. Overcoming the shortcomings of – biased and non-contextual knowledge retrieval, inadequate external knowledge representation and complexities of feature fusion, ExBERT

True Label	SNLI + ConceptNet
Entailment	P: six dogs swimming in a river. H: six dogs are outdoors. EXT: dogs is a pets, swimming is a outdoor activity, dogs desires chase frisbees in fields.
Contradiction	P: a man in an army uniform speaks into a microphone. H: a woman soldier speaks into the microphone. EXT: man antonym woman, army receives action made up of many troops, woman is a female, microphone used for turn sounds into electrical signals, soldier used for protect citizens of country.
True Label	**SciTail + Aristo Tuple**
Entailment	P: the duodenum is the first part of the small intestine and most of the chemical digestion occurs here. H: in the body, chemical digestion mainly takes place in the small intestine. EXT: digestion occur in small intestine, duodenum is a intestine, intestine become large intestine, digestion lead to lower calorie intake
Neutral	P: Hydrocarbon a compound containing only the elements carbon and hydrogen. H: Compounds containing the element carbon are the basis of all known life. EXT: carbon has property essential to all known biological life, hydrocarbon is a inanimate object, hydrogen defined as cleanest fuel for fuel cells.

Table 6.3: SNLI and SciTail Test Set Premise (P), Hypothesis (H) and the retrieved External Knowledge (EXT). The retrieved external knowledge augments the reasoning capability of $\text{BERT}_{\text{BASE}}$ model.

presents an elegant solution to augment the reasoning capabilities of BERT model via the incorporation of external knowledge. The objective of the overall approach is to take the full advantage of the expressive contextual word representations, the state-of-the-art pre-trained BERT encoder and the external knowledge from the KGs.

Utilizing the contextual word representations, ExBERT can incorporate external knowledge from any external knowledge source that allows the knowledge to be retrieved, given an entity. Further, we demonstrated the feasibility of utilising contextual representations for encoding the external knowledge from KGs, which indicates a potential direction for future research. The independence of ExBERT from the KG embedding techniques makes the overall framework simple, robust and efficient.

Quantitative and qualitative evaluations on the SNLI and SciTail datasets in conjunction with ConceptNet and Aristo Tuple KGs demonstrate that ExBERT outperforms the competing contemporary NLI models [18,61,123,149,152], including those which are enhanced by $\text{BERT}_{\text{LARGE}}$. Among the models presented previously within this thesis, ExBERT achieving the accuracy of 91.5% on the SNLI dataset in combination of the external knowledge source ConceptNet KG and the accuracies of 95.2% and 95.9% on the SciTail dataset, respectively, with the ConceptNet and

Aristo Tuple KGs is also our best-performing NLI model in terms of accuracy. In the next chapter, we summarise the research work presented in the thesis, highlight our contributions and outline the direction of future research.

CHAPTER 7

Conclusion

NLI is a crucial task in the domain of natural language understanding. The task relies on common human understanding of language and the real-world commonsense knowledge on which the (human) entailment judgement relies. It encapsulates natural language understanding capabilities within a very simple formulation — determining whether a natural language hypothesis can be inferred from a given premise. For an NLI system to succeed, it must address the full complexity of lexical and compositional semantics at all levels of language analysis (lexical, syntactic, semantic, discourse, and pragmatic) as well as real-world commonsense knowledge. Consequently, developing such systems considerably advances the developments towards true natural language understanding in NLP. Attributed to its significance to natural language understanding, NLI has received considerable recent attention from both academia and industry.

Despite the considerable literature that has arisen, the contemporary deep neural NLI models face the challenges arising from the sole reliance on the training data to comprehend all the linguistic and real-world commonsense knowledge and the underutilisation of the crucial attention mechanism. Further, the field lacks a coherent set of guidelines for the application of one of the most crucial regularisation

hyper-parameter — dropout in the RNN-based NLI models.

To address the aforementioned limitations and challenges, the central aim of this thesis has been to propose and implement robust, generalisable and knowledge-grounded neural architectures for NLI via the incorporation of external knowledge and maximising the utilisation of attention mechanisms. Towards achieving this aim, Chapter 3 present a combined attention model by integrating intra-attention and inter-attention mechanisms to maximally utilise the benefits of both the mechanisms. Chapters 4 and 6 introduces BiCAM and ExBERT frameworks respectively, to address the limitation of the inadequate knowledge learning form the training data for the complex reasoning required for NLI. Chapter 5 formulates a set of guidelines for the application of the crucial regularisation hyper-parameter — dropout for the RNN-based NLI models.

The following sections outline the main contributions of the thesis and identify future work.

7.1 Contributions

In Chapter 2, we introduce a generic neural architecture that encompasses the contemporary layered neural NLI architectures and presented a comprehensive review of the existing literature in the field of deep learning for NLI.

In Chapter 3, we focus on leveraging the attention mechanisms to learn the accurate and focussed semantic representations of the premise and hypothesis. We propose [62] a natural language inference model that uniquely utilises the intra-attention and inter-attention mechanisms. The model first captures the semantics of the individual premise and hypothesis inputs with intra-attention and then aligns the premise and hypothesis with the inter-sentence attention mechanism to learn cross sentence dependencies.

The unique combination of intra-attention and inter-attention mechanisms demonstrates the superior capabilities of modelling the semantics of the long and complex sentences. The detailed qualitative and quantitative evaluations on the SNLI and SciTail datasets, shows that in the proposed model the intra-attention and inter-

attention mechanisms work constructively and achieve higher accuracy when they are combined together in the same model than when they are used individually. Our model also outperformes the contemporary competing models [3, 58, 87, 95, 143, 203] on the SNLI dataset and the models [5, 42, 91, 200] on the SciTail dataset. The proposed model performs particularly effectively on the hard to model SciTail dataset, achieving an accuracy of 77.23% and outperforming the state-of-the-art ESIM by 6.6% and decomposable attention models by 4.9%.

Addressing the difficulties of learning the required linguistic and commonsense knowledge solely from the training data, in Chapter 4, we consider the task of incorporating real-world commonsense knowledge into deep neural NLI models. We introduce an NLI model-independent framework, which unlike the state-of-the-art models [39, 41], incorporates both external linguistic and commonsense knowledge into the NLI model and does so without any architectural changes to the underlying NLI model. Combined with convolutional feature detectors and bilinear feature fusion, the framework provides a conceptually simple mechanism that generalises across NLI models, datasets and KGs. Moreover, the framework can be easily applied to different NLI model and KG combinations.

Evaluation results of the proposed model demonstrates that the framework considerably improves the performance of the incorporated NLI baselines [42, 91] as well as the state-of-the-art models [39, 41, 70, 82] on the SNLI and the SciTail [5, 39, 40] datasets. Particularly for the smaller, syntactically and semantically complex SciTail dataset, the framework (BiECAM) achieves performance improvements of 7.0% (BiECAM accuracy 77.6%) with ConceptNet and 8.0% (BiECAM accuracy 78.6%) with Aristo Tuple KG. However, despite the superior performance of the proposed approach, the utilisation of the external knowledge can be improved by the use of recently proposed pre-trained language models such as BERT [16]. Based on the state-of-the-art developments in the NLP field, we introduce a novel approach exploiting the pre-trained language models for the utilisation of external knowledge in NLI task in Chapter 6 of the thesis.

During the evaluations of the models proposed in Chapter 3 and Chapter 4, we observed that the RNN-based neural networks are highly sensitive to model

hyper-parameters, especially the dropout locations in the model and dropout rates notably affects its performance. This observation led us to an unaddressed challenge in the RNN-based NLI models — the lack of a coherent set of dropout application guidelines in the RNN-based NLI models. Our exhaustive empirical evaluations and analysis in Chapter 5 [63], result in a set of validated guidelines applicable to a broad range of RNN-based NLI models. Among the other findings, the study establishes that the higher dropout rates are not conducive for the high performance of the RNN-based NLI models and regularising embedding layer for larger datasets and regularising recurrent layers for the smaller dataset is productive. After the excursion to dropout applications in RNNs, we return to improve the utilisation of the external knowledge in NLI following the state-of-the-art developments in the field of contextual word representations.

Although the models proposed in Chapter 4 are effective, the external knowledge is retrieved using engineered heuristics which can be biased and the retrieved knowledge may not be contextually relevant to the reasoning of the premise and hypothesis. Further, learning the representation of external knowledge and feature fusion with premise-hypothesis representation requires specialised techniques which may not be predictive for the NLI task and may produce a cascading error effect in the whole model.

Based on the state-of-the-art developments in the field of contextual word representations and PTLMs, in Chapter 6, we propose a novel model to overcome the abovementioned shortcomings. The proposed model overcomes the challenges of external knowledge incorporation at the crucial steps of external knowledge retrieval, the encoding of the retrieved knowledge and the fusion of the encoded knowledge with the premise-hypothesis representation in novel ways. The model utilises the contextual word representations to retrieve contextually relevant external knowledge and also to encode the retrieved knowledge. Further, we enhance the contextual representations of the BERT model with the retrieved external knowledge to improve its grounding in real-world knowledge and reinforce the reasoning and inference capabilities for NLI.

Quantitative and qualitative evaluations on the SNLI and SciTail datasets in con-

junction with the ConceptNet and Aristo Tuple KGs demonstrate that the proposed model outperforms the competing contemporary NLI models [18, 61, 123, 149, 152], including those which are enhanced by $\text{BERT}_{\text{LARGE}}$. The proposed model, achieving the accuracies of 91.5% on the SNLI dataset with ConceptNet KG and 95.2% and 95.9% on the SciTail dataset, respectively, with ConceptNet and Aristo Tuple KGs, is also our best-performing NLI model (in terms of accuracy) presented previously within this thesis.

7.2 Future Work

The field of NLP has been fast progressing, especially in the last few years groundbreaking research such as Transformer model [78] and the Transformer based PTLMs [16, 61, 84, 85] have enjoyed increased popularity. In light of the recent research developments, the models and the findings presented in this thesis suggest a number of possible improvements and the directions of interesting future research. In particular, we aim to pursue the following research in the future.

7.2.1 Experiments with Latest Attention Mechanisms

A number of novel attention mechanisms [87–89, 225] are introduced in the NLP research. In Chapter 3, we introduced a combined attention model benefiting from the intra-attention and inter-attention mechanisms. While this approach is efficient, further studies evaluating the efficacy of the overall framework with the recently proposed attention mechanisms [87–89, 225] will need to be undertaken.

7.2.2 Enhancing Models on Specialised Datasets

As discussed in Chapter 2, recently there has been a number of new specialised datasets focussing on evaluating the particular abilities such as cross-lingual language understanding [96] and scalar implicatures and presuppositions [111] have been proposed. Table 2.1 illustrates these datasets. As a part of future work, models presented in this thesis, especially external knowledge enhanced models proposed in Chapters 4 and 6 can be evaluated and enhanced on these datasets. We believe

that the incorporated external knowledge will also benefit the specialised reasoning required for these datasets.

7.2.3 Deploying models in natural language understanding tasks

As discussed in Chapter 1, the inferencing and reasoning abilities of NLI systems are employed in other complex neural natural language understanding tasks such as abstractive summarisation and machine comprehension. As further research work, the efficacy of the CAM, BiCAM and ExBERT models proposed in Chapters 3, 4 and 6 can be evaluated in association with other complex natural language understanding tasks such as summarisation and question answering. We believe that deploying these models in complex natural language understanding systems will increase the overall effectiveness of the system.

7.2.4 Experiments with knowledge reterival mechanisms

For external knowledge retrieval, we proposed retrieval heuristics (Section 4.1.1) in association with the BiCAM models proposed in Chapter 4 and the enhanced contextual similarity-based, selection and ranking mechanisms (Section 6.1.1) for the ExBERT model proposed in Chapter 6. Another interesting direction of future research is to explore the use of external knowledge retrieved via heuristics with ExBERT model and the external knowledge retrieved via contextual similarity-based mechanism with BiCAM models. The switching of external knowledge retrieval mechanisms between the BiCAM and ExBERT models will further enhance the understanding of the efficacies of the knowledge retrieval mechanisms and the proposed models.

7.2.5 Exploring PTLMs and External Knowledge Sources

As discussed in Chapter 6, although PTLMs have significantly improved the state-of-the-art on many complex natural language understanding tasks, they lack grounding to real-world knowledge and are often unable to remember facts when re-

quired [217, 218]. Knowledge probing tests [218] on the commonsense knowledge of ConceptNet [206] reveal that PTLMs have critical limitations when solving problems involving commonsense knowledge. In Chapter 6, we have demonstrated that external knowledge is beneficial to one of the PTLM, BERT [16], further work is required to investigate other PTLMs [61, 84, 85].

External knowledge sources are crucial for grounding the NLI models in real-world knowledge. We have explored ConceptNet for general commonsense knowledge and Aristo Tuple for domain-specific external knowledge. However, as KGs are noisy and incomplete [220], hence incorporating external knowledge from the combination of different KGs might further improve the reasoning capabilities of NLI models. In Chapter 2, Table 2.4, we illustrated different external knowledge sources. Further work is required to investigate ways to combine these external knowledge sources and utilise the knowledge from them.

7.2.6 Training Dataset - Indic Languages

English is currently the dominant language for the NLI task mainly due to the availability of large datasets in the English language. The XNLI dataset [96] extends the development and test sets of the MultiNLI [105] dataset to different languages, however, there are no dedicated NLI datasets for Hindi and other Indic languages such as Marathi and Punjabi. As part of future work, we are highly interested in gathering a dataset and releasing it publicly to advance the case of NLI in these languages and multilingual NLI.

Bibliography

[1] R. Speer and C. Havasi, "Representing general relational knowledge in ConceptNet 5," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, (Istanbul, Turkey), pp. 3679–3686, European Language Resources Association (ELRA), May 2012.

[2] I. Dagan, O. Glickman, and B. Magnini, "The pascal recognising textual entailment challenge," in *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment* (J. Quiñonero-Candela, I. Dagan, B. Magnini, and F. d'Alché Buc, eds.), (Berlin, Heidelberg), pp. 177–190, Springer Berlin Heidelberg, 2006.

[3] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, Association for Computational Linguistics, 2015.

[4] P. Clark, O. Etzioni, T. Khot, A. Sabharwal, O. Tafjord, P. D. Turney, and D. Khashabi, "Combining retrieval, statistics, and inference to answer elementary science questions," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA* (D. Schuurmans and M. P. Wellman, eds.), pp. 2580–2586, AAAI Press, 2016.

[5] T. Khot, A. Sabharwal, and P. Clark, "Scitail: A textual entailment dataset from science question answering," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018* (S. A. McIlraith and K. Q. Weinberger, eds.), pp. 5189–5197, AAAI Press, 2018.

[6] D. G. Bobrow, "Natural language input for a computer problem solving system," tech. rep., USA, 1964.

[7] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, p. 36–45, Jan. 1966.

[8] E. D. Liddy, "Natural language processing," 2001.

[9] E. Cambria and B. White, "Jumping NLP curves: A review of natural language processing research [review article]," *IEEE Comput. Intell. Mag.*, vol. 9, no. 2, pp. 48–57, 2014.

[10] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, 2015.

[11] I. Dagan, D. Roth, M. Sammons, and F. M. Zanzotto, "Recognizing textual entailment: Models and applications," *Synthesis Lectures on Human Language Technologies*, vol. 6, no. 4, pp. 1–220, 2013.

[12] I. Dagan and O. Glickman, "Probabilistic textual entailment: Generic applied modeling of language variability," *Learning Methods for Text Understanding and Mining*, vol. 2004, pp. 26–29, 2004.

[13] I. Androutsopoulos and P. Malakasiotis, "A survey of paraphrasing and textual entailment methods," *J. Artif. Int. Res.*, vol. 38, pp. 135–187, May 2010.

[14] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nat.*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] G. Marcus, "Deep learning: A critical appraisal," *CoRR*, vol. abs/1801.00631, 2018.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[17] Z. Zhang, Y. Wu, H. Zhao, Z. Li, S. Zhang, X. Zhou, and X. Zhou, "Semantics-aware bert for language understanding," *arXiv preprint arXiv:1909.02209*, 2019.

[18] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 4487–4496, Association for Computational Linguistics, July 2019.

[19] Y. Gong, H. Luo, and J. Zhang, "Natural language inference over interaction space," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, 2018.

[20] Y. Tay, A. T. Luu, and S. C. Hui, "Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 1565–1575, Association for Computational Linguistics, Nov. 2018.

[21] R. Ghaeini, S. A. Hasan, V. Datla, J. Liu, K. Lee, A. Qadir, Y. Ling, A. Prakash, X. Fern, and O. Farri, "DR-BiLSTM: Dependent reading bidirectional LSTM for natural language inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 1460–1469, Association for Computational Linguistics, June 2018.

[22] S. R. Bowman, "Modeling natural language semantics in learned representations," 2016.

[23] R. Cooper, D. Crouch, J. Van Eijck, C. Fox, J. Van Genabith, J. Jaspars, H. Kamp, D. Milward, M. Pinkal, M. Poesio, *et al.*, "Using the framework," tech. rep., 1996.

[24] B. MacCartney, *Natural language inference*. Stanford University, 2009.

[25] R. Pasunuru, H. Guo, and M. Bansal, "Towards improving abstractive summarization via entailment generation," in *Proceedings of the Workshop on New Frontiers in Summarization*, (Copenhagen, Denmark), pp. 27–32, Association for Computational Linguistics, Sept. 2017.

[26] R. Pasunuru and M. Bansal, "Multi-reward reinforced summarization with saliency and entailment," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, (New Orleans, Louisiana), pp. 646–653, Association for Computational Linguistics, June 2018.

[27] A. Trischler, Z. Ye, X. Yuan, P. Bachman, A. Sordoni, and K. Suleman, "Natural language comprehension with the EpiReader," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 128–137, Association for Computational Linguistics, Nov. 2016.

[28] A. Poliak, Y. Belinkov, J. Glass, and B. Van Durme, "On the evaluation of semantic phenomena in neural machine translation using natural language inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, (New Orleans, Louisiana), pp. 513–523, Association for Computational Linguistics, June 2018.

[29] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proceedings of the 2017 Conference on Empirical Methods in Natural*

Language Processing, (Copenhagen, Denmark), pp. 670–680, Association for Computational Linguistics, Sept. 2017.

[30] J. Kiros and W. Chan, "InferLite: Simple universal sentence representations from natural language inference data," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 4868–4874, Association for Computational Linguistics, Oct. 2018.

[31] S. Subramanian, A. Trischler, Y. Bengio, and C. J. Pal, "Learning general purpose distributed sentence representations via large scale multi-task learning," in *International Conference on Learning Representations*, 2018.

[32] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 3294–3302, Curran Associates, Inc., 2015.

[33] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[34] Y. Liu, C. Sun, L. Lin, and X. Wang, "Learning natural language inference using bidirectional lstm model and inner-attention," *arXiv preprint arXiv:1605.09090*, 2016.

[35] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[36] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kocisky, and P. Blunsom, "Reasoning about entailment with neural attention," in *International Conference on Learning Representations (ICLR)*, 2016.

[37] P. Liu, X. Qiu, Y. Zhou, J. Chen, and X. Huang, "Modelling interaction of sentence pair with coupled-LSTMs," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 1703–1712, Association for Computational Linguistics, Nov. 2016.

[38] M. Glockner, V. Shwartz, and Y. Goldberg, "Breaking NLI systems with sentences that require simple lexical inferences," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Melbourne, Australia), pp. 650–655, Association for Computational Linguistics, July 2018.

[39] D. Kang, T. Khot, A. Sabharwal, and E. Hovy, "AdvEntuRe: Adversarial training for textual entailment with knowledge-guided examples," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 2418–2428, Association for Computational Linguistics, July 2018.

[40] D. Kang, T. Khot, A. Sabharwal, and P. Clark, "Bridging knowledge gaps in neural entailment via symbolic models," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 4940–4945, Association for Computational Linguistics, Nov. 2018.

[41] Q. Chen, X. Zhu, Z.-H. Ling, D. Inkpen, and S. Wei, "Neural natural language inference models enhanced with external knowledge," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 2406–2417, Association for Computational Linguistics, July 2018.

[42] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 2249–2255, Association for Computational Linguistics, Nov. 2016.

[43] A. Romanov and C. Shivade, "Lessons from natural language inference in the clinical domain," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018* (E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, eds.), pp. 1586–1596, Association for Computational Linguistics, 2018.

[44] M. Davies, "Knowledge (explicit, implicit and tacit): Philosophical aspects," in *International Encyclopedia of the Social & Behavioral Sciences: Second Edition*, pp. 74–90, Academic Press, 2015.

[45] E. Davis and G. Marcus, "Commonsense reasoning and commonsense knowledge in artificial intelligence.," *Commun. ACM*, vol. 58, no. 9, pp. 92–103, 2015.

[46] A. C. Graesser, *Prose comprehension beyond the word*. Springer Science & Business Media, 2013.

[47] Z. Neverilová, "Paraphrase and textual entailment generation in czech," *Computación y Sistemas*, vol. 18, no. 3, 2014.

[48] T. Mihaylov and A. Frank, "Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 821–832, Association for Computational Linguistics, July 2018.

[49] A. Yang, Q. Wang, J. Liu, K. Liu, Y. Lyu, H. Wu, Q. She, and S. Li, "Enhancing pre-trained language representations with rich knowledge for machine reading comprehension," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 2346–2357, Association for Computational Linguistics, July 2019.

[50] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, pp. 2724–2743, Dec 2017.

[51] S. Kim, I. Kang, and N. Kwak, "Semantic sentence matching with densely-connected recurrent and co-attentive information," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 6586–6593, AAAI Press, 2019.

[52] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 4144–4150, 2017.

[53] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[54] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[55] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[56] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[57] T. Bluche, C. Kermorvant, and J. Louradour, "Where to apply dropout in recurrent neural networks for handwriting recognition?," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pp. 681–685, IEEE, 2015.

[58] S. R. Bowman, J. Gauthier, A. Rastogi, R. Gupta, C. D. Manning, and C. Potts, "A fast unified model for parsing and sentence understanding," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 1466–1477, 2016.

[59] J. Choi, K. M. Yoo, and S. Lee, "Learning to compose task-specific tree structures," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018* (S. A. McIlraith and K. Q. Weinberger, eds.), pp. 5094–5101, AAAI Press, 2018.

[60] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 2227–2237, Association for Computational Linguistics, June 2018.

[61] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[62] A. Gajbhiye, S. Jaf, N. A. Moubayed, S. Bradley, and A. S. McGough, "Cam: A combined attention model for natural language inference," in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 1009–1014, Dec 2018.

[63] A. Gajbhiye, S. Jaf, N. A. Moubayed, A. S. McGough, and S. Bradley, "An exploration of dropout with rnns for natural language inference," in *Artificial Neural Networks and Machine Learning – ICANN 2018* (V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, eds.), (Cham), pp. 157–167, Springer International Publishing, 2018.

[64] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA* (S. P. Singh and S. Markovitch, eds.), pp. 4444–4451, AAAI Press, 2017.

[65] M. B. Dalvi, N. Tandon, and P. Clark, "Domain-targeted, high precision knowledge extraction," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 233–246, 2017.

[66] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States* (C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 3111–3119, 2013.

[67] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, Oct. 2014.

[68] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, (Miyazaki, Japan), European Language Resources Association (ELRA), May 2018.

[69] Q. Xipeng, S. TianXiang, X. Yige, S. Yunfan, D. Ning, and H. Xuanjing, "Pre-trained models for natural language processing: A survey," *SCIENCE CHINA Technological Sciences*, 2020.

[70] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus,

S. Vishwanathan, and R. Garnett, eds.), pp. 6294–6305, Curran Associates, Inc., 2017.

[71] B. Pan, Y. Yang, Z. Zhao, Y. Zhuang, D. Cai, and X. He, "Discourse marker augmented network with reinforcement learning for natural language inference," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 989–999, Association for Computational Linguistics, July 2018.

[72] Q. Chen, Z.-H. Ling, and X. Zhu, "Enhancing sentence embedding with generalized pooling," in *Proceedings of the 27th International Conference on Computational Linguistics*, (Santa Fe, New Mexico, USA), pp. 1815–1826, Association for Computational Linguistics, Aug. 2018.

[73] D. Weissenborn, T. Kočiskỳ, and C. Dyer, "Dynamic integration of background knowledge in neural nlu systems," *arXiv preprint arXiv:1706.02596*, 2017.

[74] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," *arXiv preprint arXiv:1801.01078*, 2017.

[75] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1724–1734, Association for Computational Linguistics, Oct. 2014.

[76] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[77] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 2377–2385, Curran Associates, Inc., 2015.

[78] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 5998–6008, Curran Associates, Inc., 2017.

[79] Y. Nie and M. Bansal, "Shortcut-stacked sentence encoders for multi-domain inference," in *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, (Copenhagen, Denmark), pp. 41–45, Association for Computational Linguistics, Sept. 2017.

[80] A. Talman, A. Yli-Jyrä, and J. Tiedemann, "Sentence embeddings in nli with iterative refinement encoders," *Natural Language Engineering*, vol. 25, no. 4, pp. 467–482, 2019.

[81] T. Munkhdalai and H. Yu, "Neural semantic encoders," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, (Valencia, Spain), pp. 397–407, Association for Computational Linguistics, Apr. 2017.

[82] A. K M, R. C. S. Basu, and A. Dukkipati, "Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 313–322, Association for Computational Linguistics, June 2018.

[83] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers* (I. Gurevych and Y. Miyao, eds.), pp. 328–339, Association for Computational Linguistics, 2018.

[84] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[85] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020.

[86] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen, "Recurrent neural network-based sentence encoder with gated attention for natural language inference," in *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, (Copenhagen, Denmark), pp. 36–40, Association for Computational Linguistics, sep 2017.

[87] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "Disan: Directional self-attention network for rnn/cnn-free language understanding," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018* (S. A. McIlraith and K. Q. Weinberger, eds.), pp. 5446–5455, AAAI Press, 2018.

[88] T. Shen, T. Zhou, G. Long, J. Jiang, S. Wang, and C. Zhang, "Reinforced self-attention network: A hybrid of hard and soft attention for sequence modeling," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, p. 4345–4352, AAAI Press, 2018.

[89] J. Im and S. Cho, "Distance-based self-attention network for natural language inference," *arXiv preprint arXiv:1712.02047*, 2017.

[90] M. Guo, Y. Zhang, and T. Liu, "Gaussian transformer: a lightweight approach for natural language inference," in *The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[91] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced LSTM for natural language inference," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Vancouver, Canada), pp. 1657–1668, Association for Computational Linguistics, July 2017.

[92] C. Tan, F. Wei, W. Wang, W. Lv, and M. Zhou, "Multiway attention networks for modeling sentence pairs," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden* (J. Lang, ed.), pp. 4411–4417, ijcai.org, 2018.

[93] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016* (M. Balcan and K. Q. Weinberger, eds.), vol. 48 of *JMLR Workshop and Conference Proceedings*, pp. 1378–1387, JMLR.org, 2016.

[94] S. Kim, I. Kang, and N. Kwak, "Semantic sentence matching with densely-connected recurrent and co-attentive information," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6586–6593, 2019.

[95] L. Mou, R. Men, G. Li, Y. Xu, L. Zhang, R. Yan, and Z. Jin, "Natural language inference by tree-based convolution and heuristic matching," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Berlin, Germany), pp. 130–136, Association for Computational Linguistics, Aug. 2016.

[96] A. Conneau, R. Rinott, G. Lample, A. Williams, S. Bowman, H. Schwenk, and V. Stoyanov, "XNLI: Evaluating cross-lingual sentence representations," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 2475–2485, Association for Computational Linguistics, Nov. 2018.

[97] "EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference", author = "ravichander, abhilasha and naik, aakanksha and rose, carolyn and hovy, eduard," in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, (Hong Kong, China), pp. 349–361, Association for Computational Linguistics, Nov. 2019.

[98] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli, "A SICK cure for the evaluation of compositional distributional semantic models," in *Proceedings of the Ninth International Conference on Language*

Resources and Evaluation (LREC'14), (Reykjavik, Iceland), pp. 216–223, European Language Resources Association (ELRA), May 2014.

[99] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.

[100] A. Lai, Y. Bisk, and J. Hockenmaier, "Natural language inference from multiple premises," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers* (G. Kondrak and T. Watanabe, eds.), pp. 100–109, Asian Federation of Natural Language Processing, 2017.

[101] A. S. White, P. Rastogi, K. Duh, and B. Van Durme, "Inference is everything: Recasting semantic resources into a unified evaluation framework," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Taipei, Taiwan), pp. 996–1005, Asian Federation of Natural Language Processing, Nov. 2017.

[102] D. Reisinger, R. Rudinger, F. Ferraro, C. Harman, K. Rawlins, and B. Van Durme, "Semantic proto-roles," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 475–488, 2015.

[103] C. J. Fillmore and C. F. Baker, "Frame semantics for text understanding," in *In Proceedings of WordNet and Other Lexical Resources Workshop*, 2001.

[104] A. Rahman and V. Ng, "Resolving complex cases of definite pronouns: The winograd schema challenge," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea* (J. Tsujii, J. Henderson, and M. Pasca, eds.), pp. 777–789, ACL, 2012.

[105] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, Association for Computational Linguistics, 2018.

[106] A. Poliak, A. Haldar, R. Rudinger, J. E. Hu, E. Pavlick, A. S. White, and B. Van Durme, "Collecting diverse natural language inference problems for sentence representation evaluation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 67–81, Association for Computational Linguistics, Nov. 2018.

[107] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, 2019.

[108] O.-M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom, "e-snli: Natural language inference with natural language explanations," in *Advances in Neural Information Processing Systems 31* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), pp. 9539–9549, Curran Associates, Inc., 2018.

[109] T. McCoy, E. Pavlick, and T. Linzen, "Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 3428–3448, Association for Computational Linguistics, July 2019.

[110] M. Schmitt and H. Schütze, "SherLIiC: A typed event-focused lexical inference benchmark for evaluating natural language inference," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 902–914, Association for Computational Linguistics, July 2019.

[111] P. Jeretic, A. Warstadt, S. Bhooshan, and A. Williams, "Are natural language inference models imppressive? learning implicature and presupposition," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020* (D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, eds.), pp. 8690–8705, Association for Computational Linguistics, 2020.

[112] J. Welbl, N. F. Liu, and M. Gardner, "Crowdsourcing multiple choice science questions," in *Proceedings of the 3rd Workshop on Noisy User-generated Text*, (Copenhagen, Denmark), pp. 94–106, Association for Computational Linguistics, Sept. 2017.

[113] T. Khot, A. Sabharwal, and P. Clark, "Answering complex questions using open information extraction," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Vancouver, Canada), pp. 311–316, Association for Computational Linguistics, July 2017.

[114] X. Wang, P. Kapanipathi, R. Musa, M. Yu, K. Talamadupula, I. Abdelaziz, M. Chang, A. Fokoue, B. Makni, N. Mattei, *et al.*, "Improving natural language inference using external knowledge in the science questions domain," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 7208–7215, 2019.

[115] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, p. 39–41, Nov. 1995.

[116] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives, "Dbpedia: A nucleus for a web of open data," in *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007* (K. Aberer,

K. Choi, N. F. Noy, D. Allemang, K. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, eds.), vol. 4825 of *Lecture Notes in Computer Science*, pp. 722–735, Springer, 2007.

[117] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, (New York, NY, USA), p. 1247–1250, Association for Computing Machinery, 2008.

[118] O. Bodenreider, "The unified medical language system (umls): integrating biomedical terminology," *Nucleic acids research*, vol. 32, no. suppl_1, pp. D267–D270, 2004.

[119] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, *et al.*, "Knowledge graphs," *arXiv preprint arXiv:2003.02320*, 2020.

[120] M. Ghazvininejad, C. Brockett, M. Chang, B. Dolan, J. Gao, W. Yih, and M. Galley, "A knowledge-grounded neural conversation model," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018* (S. A. McIlraith and K. Q. Weinberger, eds.), pp. 5110–5117, AAAI Press, 2018.

[121] T. Young, E. Cambria, I. Chaturvedi, H. Zhou, S. Biswas, and M. Huang, "Augmenting end-to-end dialogue systems with commonsense knowledge," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018* (S. A. McIlraith and K. Q. Weinberger, eds.), pp. 4970–4977, AAAI Press, 2018.

[122] X. Pan, K. Sun, D. Yu, J. Chen, H. Ji, C. Cardie, and D. Yu, "Improving question answering with external knowledge," in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, (Hong Kong, China), pp. 27–37, Association for Computational Linguistics, Nov. 2019.

[123] A. H. Li and A. Sethy, "Knowledge enhanced attention for robust natural language inference," *arXiv preprint arXiv:1909.00102*, 2019.

[124] T. Young, E. Cambria, I. Chaturvedi, H. Zhou, S. Biswas, and M. Huang, "Augmenting end-to-end dialogue systems with commonsense knowledge," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018* (S. A. McIlraith and K. Q. Weinberger, eds.), pp. 4970–4977, AAAI Press, 2018.

[125] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "Commonsenseqa: A question answering challenge targeting commonsense knowledge," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), pp. 4149–4158, Association for Computational Linguistics, 2019.

[126] P. Singh *et al.*, "The public acquisition of commonsense knowledge," in *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, 2002.

[127] J. C. A. Silva, H. Lieberman, M. Tsutsumi, V. P. de Almeida Néris, A. A. de Carvalho, J. H. Espinosa, M. de Souza Godoi, and S. Zem-Mascarenhas, "Can common sense uncover cultural differences in computer applications?," in *Artificial Intelligence in Theory and Practice, IFIP 19th World Computer Congress, TC 12: IFIP AI 2006 Stream, August 21-24, 2006, Santiago, Chile* (M. Bramer, ed.), vol. 217 of *IFIP*, pp. 1–10, Springer, 2006.

[128] L. von Ahn, M. Kedia, and M. Blum, "Verbosity: A game for collecting common-sense facts," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, (New York, NY, USA), p. 75–78, Association for Computing Machinery, 2006.

[129] Y.-l. Kuo, J.-C. Lee, K.-y. Chiang, R. Wang, E. Shen, C.-w. Chan, and J. Y.-j. Hsu, "Community-based game design: Experiments on social games for commonsense data collection," in *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '09, (New York, NY, USA), p. 15–22, Association for Computing Machinery, 2009.

[130] F. Bond and R. Foster, "Linking and extending an open multilingual wordnet," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pp. 1352–1362, The Association for Computer Linguistics, 2013.

[131] J. Breen, "JMdict: a Japanese-multilingual dictionary," in *Proceedings of the Workshop on Multilingual Linguistic Resources*, (Geneva, Switzerland), pp. 65–72, COLING, Aug. 28 2004.

[132] C. Elkan and R. Greiner, "D. b. lenat and r. v. guha, building large knowledge-based systems: Representation and inference in the cyc project," *Artif. Intell.*, vol. 61, no. 1, pp. 41–52, 1993.

[133] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives, "Dbpedia: A nucleus for a web of open data," in *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007* (K. Aberer, K. Choi, N. F. Noy, D. Allemang, K. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux,

eds.), vol. 4825 of *Lecture Notes in Computer Science*, pp. 722–735, Springer, 2007.

[134] S. Soderland, J. Gilmer, R. Bart, O. Etzioni, and D. S. Weld, "Open information extraction to KBP relations in 3 hours," in *Proceedings of the Sixth Text Analysis Conference, TAC 2013, Gaithersburg, Maryland, USA, November 18-19, 2013*, NIST, 2013.

[135] Mausam, M. Schmitz, S. Soderland, R. Bart, and O. Etzioni, "Open language learning for information extraction," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, (Jeju Island, Korea), pp. 523–534, Association for Computational Linguistics, July 2012.

[136] D. Kiela, C. Wang, and K. Cho, "Dynamic meta-embeddings for improved sentence representations," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 1466–1477, Association for Computational Linguistics, Nov. 2018.

[137] T. Munkhdalai and H. Yu, "Neural tree indexers for text understanding," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, (Valencia, Spain), pp. 11–21, Association for Computational Linguistics, Apr. 2017.

[138] D. Yoon, D. Lee, and S. Lee, "Dynamic self-attention: Computing attention over words dynamically for sentence embedding," *arXiv preprint arXiv:1808.07383*, 2018.

[139] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen, "Recurrent neural network-based sentence encoder with gated attention for natural language inference," in *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, (Copenhagen, Denmark), pp. 36–40, Association for Computational Linguistics, Sept. 2017.

[140] A. Talman, A. Yli-Jyrä, and J. Tiedemann, "Sentence embeddings in NLI with iterative refinement encoders," *Nat. Lang. Eng.*, vol. 25, no. 4, pp. 467–482, 2019.

[141] Y. Tay, A. T. Luu, and S. C. Hui, "Hermitian co-attention networks for text matching in asymmetrical domains," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden* (J. Lang, ed.), pp. 4425–4431, ijcai.org, 2018.

[142] S. Wang and J. Jiang, "Learning natural language inference with LSTM," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (San Diego, California), pp. 1442–1451, Association for Computational Linguistics, June 2016.

[143] P. Liu, X. Qiu, J. Chen, and X. Huang, "Deep fusion LSTMs for text semantic matching," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Berlin, Germany), pp. 1034–1043, Association for Computational Linguistics, Aug. 2016.

[144] L. Sha, B. Chang, Z. Sui, and S. Li, "Reading and thinking: Re-read LSTM unit for textual entailment recognition," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, (Osaka, Japan), pp. 2870–2879, The COLING 2016 Organizing Committee, Dec. 2016.

[145] W. Yin, D. Roth, and H. Schütze, "End-task oriented textual entailment via deep explorations of inter-sentence interactions," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Melbourne, Australia), pp. 540–545, Association for Computational Linguistics, July 2018.

[146] X. Liu, Y. Shen, K. Duh, and J. Gao, "Stochastic answer networks for machine reading comprehension," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 1694–1704, Association for Computational Linguistics, July 2018.

[147] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 551–561, Association for Computational Linguistics, Nov. 2016.

[148] M. Guo, Y. Zhang, and T. Liu, "Gaussian transformer: A lightweight approach for natural language inference," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 6489–6496, AAAI Press, 2019.

[149] D. Pang, L. H. Lin, and N. A. Smith, "Improving natural language inference with a pretrained parser," *CoRR*, vol. abs/1909.08217, 2019.

[150] X. Yang, X. Zhu, H. Zhao, Q. Zhang, and Y. Feng, "Enhancing unsupervised pretraining with external knowledge for natural language inference," in *Advances in Artificial Intelligence* (M.-J. Meurs and F. Rudzicz, eds.), (Cham), pp. 413–419, Springer International Publishing, 2019.

[151] T. Li, X. Zhu, Q. Liu, Q. Chen, Z. Chen, and S. Wei, "Several experiments on investigating pretraining and knowledge-enhanced models for natural language inference," *arXiv preprint arXiv:1904.12104*, 2019.

[152] Z. Zhang, Y. Wu, Z. Li, and H. Zhao, "Explicit contextual semantics for text comprehension," 2019.

[153] W. Wang, B. Bi, M. Yan, C. Wu, J. Xia, Z. Bao, L. Peng, and L. Si, "Structbert: Incorporating language structures into pre-training for deep language understanding," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020.

[154] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, p. 1735–1780, Nov. 1997.

[155] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in neural information processing systems*, pp. 737–744, 1994.

[156] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 3104–3112, Curran Associates, Inc., 2014.

[157] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[158] M. Tan, C. d. Santos, B. Xiang, and B. Zhou, "Lstm-based deep learning models for non-factoid answer selection," *arXiv preprint arXiv:1511.04108*, 2015.

[159] F. Hill, K. Cho, and A. Korhonen, "Learning distributed representations of sentences from unlabelled data," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (San Diego, California), pp. 1367–1377, Association for Computational Linguistics, June 2016.

[160] W. Yin and H. Schütze, "Learning word meta-embeddings," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Berlin, Germany), pp. 1351–1360, Association for Computational Linguistics, Aug. 2016.

[161] J. Coates and D. Bollegala, "Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, (New Orleans, Louisiana), pp. 194–198, Association for Computational Linguistics, June 2018.

[162] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[163] R. Socher, C. C.-Y. Lin, A. Y. Ng, and C. D. Manning, "Parsing natural scenes and natural language with recursive neural networks," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, (Madison, WI, USA), p. 129–136, Omnipress, 2011.

[164] S. R. Bowman, C. D. Manning, and C. Potts, "Tree-structured composition in neural networks without tree-structured architectures," in *Proceedings of the 2015th International Conference on Cognitive Computation: Integrating Neural and Symbolic Approaches - Volume 1583*, COCO'15, (Aachen, DEU), p. 37–42, CEUR-WS.org, 2015.

[165] T. Munkhdalai and H. Yu, "Neural semantic encoders," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, (Valencia, Spain), pp. 397–407, Association for Computational Linguistics, Apr. 2017.

[166] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Beijing, China), pp. 1556–1566, Association for Computational Linguistics, July 2015.

[167] J. Li, T. Luong, D. Jurafsky, and E. Hovy, "When are tree structures necessary for deep learning of representations?," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Lisbon, Portugal), pp. 2304–2314, Association for Computational Linguistics, Sept. 2015.

[168] L. Mou, H. Peng, G. Li, Y. Xu, L. Zhang, and Z. Jin, "Discriminative neural sentence modeling by tree-based convolution," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Lisbon, Portugal), pp. 2315–2325, Association for Computational Linguistics, Sept. 2015.

[169] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[170] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 2261–2269, IEEE Computer Society, 2017.

[171] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1746–1751, Association for Computational Linguistics, Oct. 2014.

[172] R. Zhang, H. Lee, and D. R. Radev, "Dependency sensitive convolutional neural networks for modeling sentences and documents," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (San Diego, California), pp. 1512–1521, Association for Computational Linguistics, June 2016.

[173] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, (Cambridge, MA, USA), p. 1693–1701, MIT Press, 2015.

[174] D. Chen, J. Bolton, and C. D. Manning, "A thorough examination of the CNN/daily mail reading comprehension task," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Berlin, Germany), pp. 2358–2367, Association for Computational Linguistics, Aug. 2016.

[175] A. Nie, E. Bennett, and N. Goodman, "DisSent: Learning sentence representations from explicit discourse relations," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 4497–4510, Association for Computational Linguistics, July 2019.

[176] X. Zhu, P. Sobhani, and H. Guo, "Long short-term memory over recursive structures," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, p. 1604–1612, JMLR.org, 2015.

[177] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 7-12 July 2003, Sapporo Convention Center, Sapporo, Japan* (E. W. Hinrichs and D. Roth, eds.), pp. 423–430, ACL, 2003.

[178] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[179] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson, "One billion word benchmark for measuring progress in statistical language modeling," in *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014* (H. Li, H. M. Meng, B. Ma, E. Chng, and L. Xie, eds.), pp. 2635–2639, ISCA, 2014.

[180] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[181] C. Liu, S. Jiang, H. Yu, and D. Yu, "Multi-turn inference matching network for natural language inference," in *Natural Language Processing and Chinese Computing - 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26-30, 2018, Proceedings, Part II* (M. Zhang, V. Ng, D. Zhao, S. Li, and H. Zan, eds.), vol. 11109 of *Lecture Notes in Computer Science*, pp. 131–143, Springer, 2018.

[182] Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 19–27, IEEE Computer Society, 2015.

[183] X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y.-y. Wang, "Representation learning using multi-task deep neural networks for semantic classification and information retrieval," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (Denver, Colorado), pp. 912–921, Association for Computational Linguistics, June 2015.

[184] C. Alberti, K. Lee, and M. Collins, "A bert baseline for the natural questions," *arXiv preprint arXiv:1901.08634*, 2019.

[185] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune BERT for text classification?," in *Chinese Computational Linguistics - 18th China National Conference, CCL 2019, Kunming, China, October 18-20, 2019, Proceedings* (M. Sun, X. Huang, H. Ji, Z. Liu, and Y. Liu, eds.), vol. 11856 of *Lecture Notes in Computer Science*, pp. 194–206, Springer, 2019.

[186] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, (Brussels, Belgium), pp. 353–355, Association for Computational Linguistics, Nov. 2018.

[187] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Superglue: A stickier benchmark for general-purpose language understanding systems," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada* (H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, eds.), pp. 3261–3275, 2019.

[188] M. Minsky, *The Society of Mind*. New York, NY, USA: Simon & Schuster, Inc., 1986.

[189] P. Kapanipathi, V. Thost, S. S. Patel, S. Whitehead, I. Abdelaziz, A. Balakrishnan, M. Chang, K. Fadnis, C. Gunasekara, B. Makni, N. Mattei, K. Talamadupula, and A. Fokoue, "Infusing knowledge into the textual entailment task using graph convolutional networks," in *Proceedings of the Thirty-Fourth Conference on Association for the Advancement of Artificial Intelligence (AAAI)*, 2020.

[190] T. Dozat and C. D. Manning, "Deep biaffine attention for neural dependency parsing," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[191] B. MacCartney, T. Grenager, M.-C. de Marneffe, D. Cer, and C. D. Manning, "Learning to recognize features of valid textual entailments," in *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, (Stroudsburg, PA, USA), pp. 41–48, Association for Computational Linguistics, 2006.

[192] V. Jijkoun, M. de Rijke, *et al.*, "Recognizing textual entailment using lexical similarity," in *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pp. 73–76, Citeseer, 2005.

[193] M. Heilman and N. A. Smith, "Tree edit models for recognizing textual entailments, paraphrases, and answers to questions," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1011–1019, Association for Computational Linguistics, 2010.

[194] A. Hickl, "Using discourse commitments to recognize textual entailment," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pp. 337–344, Association for Computational Linguistics, 2008.

[195] R. Wang and Y. Zhang, "Recognizing textual relatedness with predicate-argument structures," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pp. 784–792, Association for Computational Linguistics, 2009.

[196] J. Bos and K. Markert, "Recognising textual entailment with logical inference," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 628–635, Association for Computational Linguistics, 2005.

[197] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Lisbon, Portugal), pp. 1412–1421, Association for Computational Linguistics, Sept. 2015.

[198] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," *arXiv preprint arXiv:1503.02364*, 2015.

[199] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," *arXiv preprint arXiv:1611.01603*, 2016.

[200] Y. Tay, L. A. Tuan, and S. C. Hui, "A compare-propagate architecture with alignment factorization for natural language inference," *arXiv preprint arXiv:1801.00102*, 2017.

[201] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[202] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[203] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun, "Order-embeddings of images and language," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2016.

[204] H. Sun, B. Dhingra, M. Zaheer, K. Mazaitis, R. Salakhutdinov, and W. Cohen, "Open domain question answering using early fusion of knowledge bases and text," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 4231–4242, Association for Computational Linguistics, Nov. 2018.

[205] B. Yang and T. Mitchell, "Leveraging knowledge bases in LSTMs for improving machine reading," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Vancouver, Canada), pp. 1436–1446, Association for Computational Linguistics, July 2017.

[206] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," 2017.

[207] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 1955–1961, AAAI Press, 2016.

[208] Z. Yu, J. Yu, J. Fan, and D. Tao, "Multi-modal factorized bilinear pooling with co-attention learning for visual question answering," in *Proceedings of the IEEE international conference on computer vision*, pp. 1821–1830, 2017.

[209] B. Mitra and N. Craswell, "Neural models for information retrieval," *ArXiv*, vol. abs/1705.01509, 2017.

[210] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[211] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. C. Courville, Y. Bengio, and S. Lacoste-Julien, "A closer look at memorization in deep networks," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, pp. 233–242, PMLR, 2017.

[212] M. Pachitariu and M. Sahani, "Regularization and nonlinearities for neural language models: when are they needed?," *arXiv preprint arXiv:1301.5650*, 2013.

[213] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pp. 285–290, IEEE, 2014.

[214] G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur, and Y. Yan, "An exploration of dropout with lstms," in *Proceedings of Interspeech*, 2017.

[215] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, (Red Hook, NY, USA), p. 1027–1035, Curran Associates Inc., 2016.

[216] R. Józefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015* (F. R. Bach and D. M. Blei, eds.), vol. 37 of *JMLR Workshop and Conference Proceedings*, pp. 2342–2350, JMLR.org, 2015.

[217] M. E. Peters, M. Neumann, R. L. L. IV, R. Schwartz, V. Joshi, S. Singh, and N. A. Smith, "Knowledge enhanced contextual word representations," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019* (K. Inui, J. Jiang, V. Ng, and X. Wan, eds.), pp. 43–54, Association for Computational Linguistics, 2019.

[218] S. Kwon, C. Kang, J. Han, and J. Choi, "Why do masked neural language models still need common sense knowledge?," *CoRR*, vol. abs/1911.03024, 2019.

[219] R. L. L. IV, N. F. Liu, M. E. Peters, M. Gardner, and S. Singh, "Barack's wife hillary: Using knowledge graphs for fact-aware language modeling," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers* (A. Korhonen, D. R. Traum, and L. Màrquez, eds.), pp. 5962–5971, Association for Computational Linguistics, 2019.

[220] H. Bast, B. Buchhold, and E. Haussmann, "Semantic search on text and knowledge bases," *Found. Trends Inf. Retr.*, vol. 10, no. 2-3, pp. 119–271, 2016.

[221] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 2787–2795, Curran Associates, Inc., 2013.

[222] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," in *Proceedings of the Thirtieth*

AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA (D. Schuurmans and M. P. Wellman, eds.), pp. 2659–2665, AAAI Press, 2016.

[223] Y. Goldberg, "Assessing bert's syntactic abilities," *CoRR*, vol. abs/1901.05287, 2019.

[224] R. Rosa and D. Marecek, "Inducing syntactic trees from BERT representations," *CoRR*, vol. abs/1906.11511, 2019.

[225] A. Galassi, M. Lippi, and P. Torroni, "Attention, please! A critical review of neural attention models in natural language processing," *CoRR*, vol. abs/1902.02181, 2019.