# STRUCTURAL EMBEDDING OF CONSTITUENCY TREES IN THE ATTENTION-BASED MODEL FOR MACHINE COMPREHENSION

by

Mayank Anand

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
August 2023

*This thesis is dedicated to my family, friends and Professors who have supported me in every situation, and to the divine presence of the goddess Radha, whose blessings have guided me throughout this journey.*

## Table of Contents

## List of Tables

# List of Figures

## Abstract

Incorporating hierarchical structures for various Natural Language Processing (NLP) tasks, which involves training the model with syntactic information of constituency trees, has been shown to be very effective. Constituency trees in the simplest form are graph representations of sentences that capture and illustrate syntactic hierarchical structure of a sentence by showing how words are grouped into constituents. However, the majority of research in NLP using Deep Learning to incorporate structural information has been conducted on recurrent models, which are effective but operate sequentially. To the best of our knowledge, no research has been done on attention-based models for the reading comprehension task. In this work, we aim to include syntactic information of constituency trees in the model QAnet which is based on self-attention and specifically designed for Machine Reading Comprehension task. The proposed solution involves the use of “Hierarchical Accumulation” to encode constituency trees in self-attention in parallel time complexity. Our model, QATnet, achieved competitive results compared to the baseline QAnet model. Furthermore, we demonstrated by analyzing context-question pair examples that using a hierarchical structure model exhibited a remarkable ability to retain contextual information over longer distances and enhanced attention towards punctuation and other grammatical intricacies.

## List of Abbreviations and Symbols Used

| | |
|---|---|
| BIDAF | Bidirectional Attention Flow for Machine Comprehension |
| BPE | Byte-Pair Encoding |
| DCN | Dynamic Co-attention Networks For Question Answering |
| dev | development |
| EM | Exact Match |
| JSON | JavaScript Object Notation |
| len | length |
| MRC | Machine Reading Comprehension |
| NLP | Natural Language Processing |
| POS | Part-Of-Speech |
| SEST | Structural Embedding of Syntactic Trees |
| SQuAD | Stanford Question Answering Dataset |
| SQuAD 2.0 | Stanford Question Answering Dataset 2.0 |

## Acknowledgements

I want to extend my heartfelt appreciation to my family, especially my sister Dr. Kamia Punia, for their constant support and encouragement throughout my thesis. The unwavering dedication they have shown has been instrumental in my journey. Additionally, I would like to express sincere gratitude to the DNLP lab for their friendship, support, and invaluable feedback, all of which have played a pivotal role in shaping my work. Finally, I wish to sincerely thank my supervisor, Dr. Vlado Keselj, for being an exceptional mentor and providing me with invaluable guidance every step of the way.

# Chapter 1

# Introduction

In recent years, attention-based models such as Transformers have revolutionized the field of Natural Language Processing (NLP) by achieving the state of the art performance in many language-related tasks as observed in papers by Vaswani et al. [23], Devlin et al. [5], and Wei et al. [25]. Despite their success, there is yet no evidence, to the best of our knowledge, that constituency trees are learned implicitly by these models. Constituency trees, also known as parse trees, represent the hierarchical structure of sentences by breaking them into constituent parts, such as phrases and clauses.

Constituency trees play a critical role in representing the structure of the sentence. They are graphical representations of how words in a sentence are related. By incorporating syntactic information into NLP models, we can capture the syntactic structure of the sentence, which is vital for understanding its meaning. For example, the sentences "Ram hit the ball with a bat," and "Ram hit the bat with a ball." have the same words, but their meanings are entirely different. Without the knowledge of constituency trees, a model may not be able to distinguish between two sentences, which can result in inaccurate predictions.

Much work has been proposed to leverage constituency trees in deep neural networks, however most of the research by Tai et al. [22], Liu and Hu. [9], and Shen et al. [20] operates on a recurrent or recursive mechanism, which is not parallelizable, and is not suitable for training longer sequences. Bai et al. [3] and Nguyen et al. [12], both in their research showcased very novel techniques for incorporating constituency trees but did not test on longer sequences such as the task of Machine Reading Comprehensions. In this work, we specifically worked on machine reading comprehension because utilizing knowledge of constituency trees can reduce the size of candidate space to help the model identify the correct answer.

For example, Fig. 1.1 shows constituency tree of the sentence "Beyoncé would perform alongside Coldplay at Super Bowl 50 in February.". "Coldplay" and "Beyoncé" are labelled as noun phrases ("NP"), which is critical for answering the question *"Beyonce would perform with who at Superbowl 50?"*. The question asks for the name of another singer that can be best answered using a noun phrase.

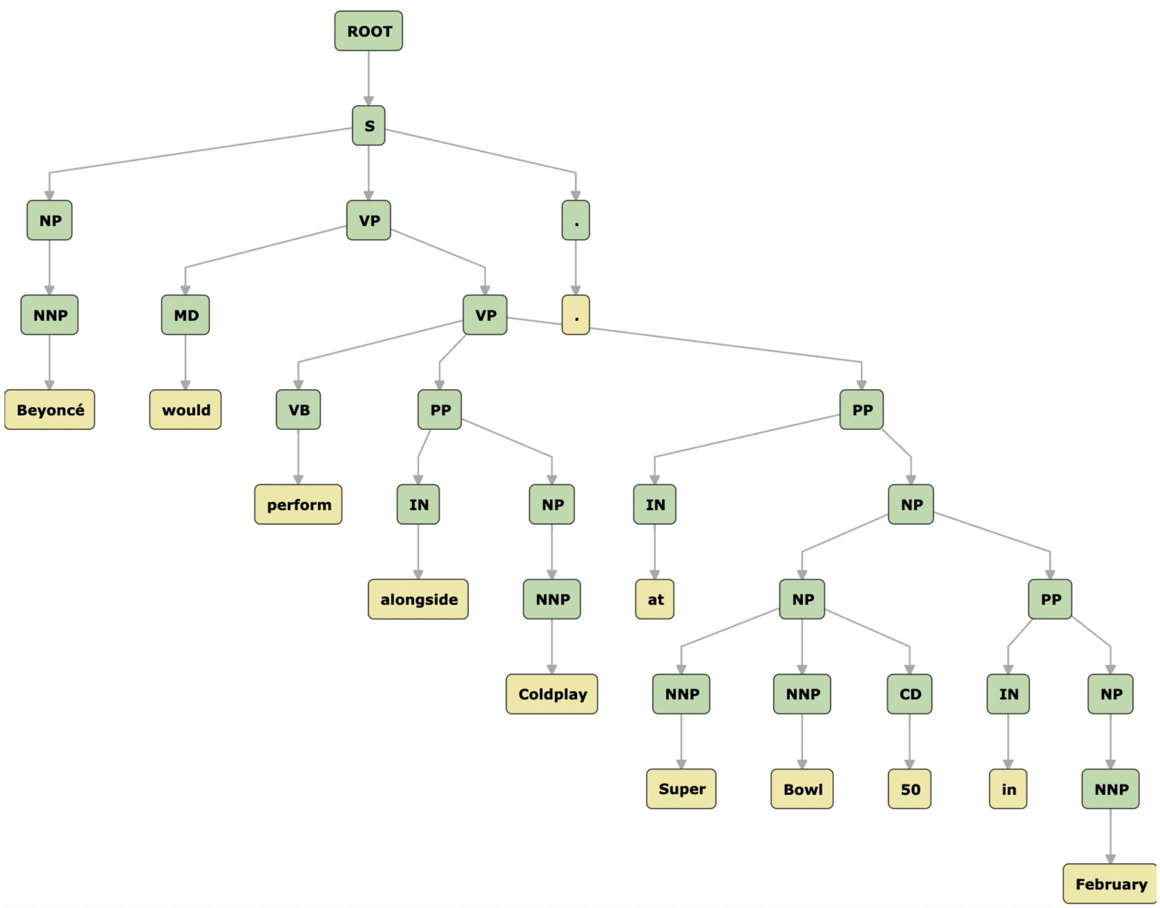


Figure 1.1: The constituency tree for context "Beyoncé would perform alongside Coldplay at Super Bowl 50 in February."

In this work, we attempt to find out whether incorporating constituency trees could help the model identify the right answer. We attempted to incorporate multi-head self-attention with hierarchical accumulation inspired by Nguyen et al. [12], tuned it for multi-sentences (multi-trees) and combined it with convolutions in a model, which is novel and has not been researched before to the best of our knowledge. The main contribution of this thesis is the novel adaptation of the hierarchical neural network model to capture the constituency structure of sentences, which we evaluate on the task of question answering. Our detailed analysis demonstrates the types of questions where this approach provides better performance in question answering.

The subsequent chapters of this thesis provide a comprehensive overview of our research, beginning with Chapter 2, where we delve into the different neural layers used in the construction of our model and discuss other techniques employed to train such a large-scale model. Additionally, we explore previous research that has focused on incorporating constituency trees for various language modeling tasks, laying the foundation for our proposed approach. Chapter 2 serves as a critical building block for our research, as it explores the neural layers and techniques that form the backbone of our model. We delve into the intricacies of these layers, including their design, functionality, and the motivations behind their integration. By thoroughly investigating these aspects, we ensure a comprehensive understanding of the underlying principles guiding our model's development. Furthermore, this chapter delves into the training of large-scale models, which poses unique challenges. We discuss various techniques and strategies employed to train such models effectively. These encompass approaches like distributed training, gradient accumulation, and regularization techniques that mitigate overfitting. By thoroughly examining the intricacies of training large-scale models, we establish a solid foundation for the subsequent chapters of our research.

In Chapter 3, we present a comprehensive methodology that underpins our research. We focus on the Stanford Question Answering Dataset 2.0 dataset and its context, which is a widely-used benchmark in the field of question answering. We provide a detailed exploration of the dataset, including its characteristics, structure, and challenges it poses. Moreover, we discuss the necessary preprocessing steps and techniques employed to obtain constituency trees from the dataset, resulting in the creation of a binarized dataset. An integral part of our methodology involves the incorporation of hierarchical accumulation, a technique first introduced by Nguyen et al. [12] in their research. In Chapter 3, we provide a comprehensive explanation of this technique and its adaptation to our model. By incorporating constituency trees into self-attention-based models, we aim to capture the hierarchical structure of sentences more effectively, leveraging syntactic information for improved question answering performance.

Chapter 4 delves into the detailed architecture of our proposed model. We meticulously describe each layer and its role within the overall framework. Additionally, we showcase how the hierarchical accumulation technique is integrated into the model, emphasizing its impact on capturing constituency structures. By presenting a comprehensive model architecture, we aim to provide a clear understanding of how each component contributes to the overall performance of our approach. In Chapter 5, we present the results of extensive experimentation and analysis. We meticulously evaluate our model's performance on Exact Match (EM) and F1 measures, comparing it against the baseline model and identifying instances where our proposed method exhibits superiority. Furthermore, we conduct in-depth analysis to gain insights into the strengths and limitations of our approach, shedding light on the specific scenarios where our model excels.

Finally, in the concluding chapter, we summarize our findings, discuss the implications of our research, and provide recommendations for future work. We reflect on the contributions made by our thesis, emphasizing the novel integration of constituency trees into self-attention-based models for question answering tasks. Additionally, we outline potential directions for further exploration, highlighting areas that warrant future investigation. In summary, this thesis explores the integration of constituency trees into self-attention-based models for question answering tasks. Through a comprehensive examination of neural layers, model architecture, methodology, and extensive experimentation, we aim to advance the understanding of how syntactic information can be effectively leveraged to improve the performance of question answering systems.

# Chapter 2

# Background and Related Work

Chapter 2, serves as a comprehensive foundation for the research presented in this thesis. This chapter begins with an exploration of normalization techniques, specifically Batch Normalization and Layer Normalization, which play a crucial role in training deep neural networks. The concept of Internal Covariance Shift is introduced to highlight the challenges faced during network training and the importance of addressing them. The subsequent sections delve into the significance of residual connections, which enable the successful training of deep networks by mitigating the vanishing gradient problem. Word embeddings and character embeddings are examined as essential tools for representing textual data in a high-dimensional space, capturing semantic and syntactic relationships between words. Additionally, the concept of highway networks is introduced, emphasizing their ability to control information flow and facilitate complex transformations. The chapter concludes with an overview of related work, including Tree-LSTM, SEST, ON-LSTM, self-attention mechanisms, Syntax-Bert, and hierarchical accumulation, providing a comprehensive understanding of existing research in the field. By establishing this solid background and reviewing relevant literature, this chapter sets the stage for the subsequent chapters and contributes to the overall knowledge and context surrounding the research presented in this thesis.

## 2.1 Background

In this section, we focus on network normalization techniques, specifically Batch Normalization, Layer Normalization, and Residual Connections for better flow of gradients. These techniques play a crucial role in addressing challenges such as internal covariance shift, vanishing gradients, and training deep neural networks. By understanding these techniques, we gain valuable insights into improving the training process.

### 2.1.1 Parse Trees

Parse trees [8] are graphical representations of the syntactic structure of sentences in natural language. They are used in Natural Language Processing (NLP) to analyze and understand the grammatical relationships between words in a sentence. Parse trees are constructed by breaking down a sentence into its constituent parts, such as phrases and clauses, and then representing these parts as nodes in a tree structure. The tree structure consists of nodes, which represent the different parts of the sentence, and edges, which represent the relationships between these parts.

### 2.1.2 Normalization

Normalization mentioned in this subsection refers to a technique used in neural networks to standardize the inputs or activations of a layer. It aims to alleviate the issue of vanishing or exploding gradients during training. By normalizing the inputs, the network becomes more stable and can learn more efficiently. In the context of neural networks, normalization techniques such as batch normalization and layer normalization are commonly used. Batch normalization computes the mean and variance of the inputs within a mini-batch, while layer normalization calculates the mean and variance of the summed inputs to the neurons in a layer on a single training case.

Training deep neural networks is very difficult given the fact that the distribution of each layer's input changes during training because the parameters of the previous layer change on which inputs depend. This can lead to slow convergence or even to the divergence of the training process, as the later layers must constantly adapt to the changing input distribution. Ioffe et al. [7] named this phenomenon in their research the Internal Covariance Shift.

#### Batch Normalization

To mitigate this phenomenon, Ioffe et al. [7] in their research introduced a concept of Batch Normalization. Batch normalization involves normalizing the inputs to each layer of the network so that they have zero mean and unit variance. This helps to reduce the internal covariance shift and can improve the training process by making the optimization of the network parameters more stable. Below are the advantages of normalizing the inputs while training deep neural nets.

- By normalizing inputs, it reduces number of steps needed to train the model.
- It helps tackle the problem of vanishing and exploding gradients.
- Every epoch takes a little longer because of extra computation but the total number of epochs required for training is lower; i.e., it achieves the same accuracy faster.

$$\begin{aligned} \mu &= \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \sigma = \frac{1}{m}\sum_{i=1}^{m}(x_i-\mu)^2 \\ \hat{X}_i &= \frac{x_i-\mu}{\sqrt{\sigma+\epsilon}} \qquad y_i = \gamma\hat{X}_i + \beta \end{aligned} \tag{2.1}$$

where $m$ is the size of the mini-batch and $\gamma$ and $\beta$ are the learnable parameters.

### Layer Normalization

The effect of batch normalization is very effective, however, it is dependent on mini-batch size and it is not obvious how to apply it to recurrent neural networks. To tackle this drawback, Lei Ba et al. [1] transposed bath normalization into layer normalization by computing the mean and variance used for normalization from all of the summed inputs to the neurons in a layer on a single training case. The below equation shows how it is calculated, where $H$ denotes the number of hidden units in a layer which is represent by l in Eq. 2.2.

$$\mu^l = \frac{1}{H}\sum_{i=1}^{H} a_i^l \qquad \sigma^l = \sqrt{\frac{1}{H}\sum_{i=1}^{H}\left(a_i^l-\mu^l\right)^2} \tag{2.2}$$

The difference between both is that under layer normalization, all the hidden units in a layer share the same normalization terms $\mu$ and $\sigma$, but different training cases have different normalization terms. Therefore, unlike batch normalization, layer normalization doesn't impose any constraint on batch size and can be applied to even training with batch size 1.

### 2.1.3 Residual Connections

As neural networks have become deeper and more complex, they have become more difficult to train because of vanishing gradients. Residual connections, first introduced by He and Zhang et al. [6] in their research, represented new architecture ResNet, which was able to train much deeper neural networks than previously possible. A

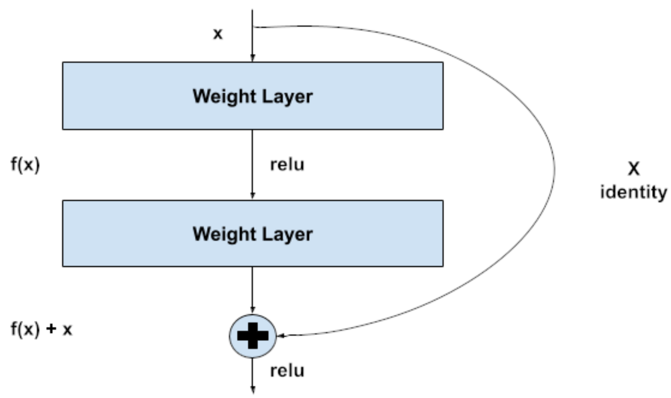


Figure 2.1: Residual Connection

residual connection as shown in Fig. 2.1, also known as a shortcut network, allows information to bypass one or more layers in a neural network. Instead of being passed directly from one layer to the next, the output of the layer is added to the input of a later layer. Due to this, gradients can easily flow from earlier layers of a deep neural network during backpropagation.

### 2.1.4 Word Embeddings

Word embeddings are vector representations of words in a high-dimensional space, where each dimension represents a feature of the word. These embeddings are learned from large corpora of text using unsupervised learning algorithms. The main idea behind word embeddings is that words with similar meanings will have similar vector representations. The GloVe embedding was first introduced by Pennington et al. [15], which uses a co-occurrence matrix-based approach which considers the global co-occurrence statistics of words. The co-occurrence matrix can be thought of as a way of measuring the similarity between words based on their co-occurrence patterns. The GloVe algorithm takes the co-occurrence matrix as input and learns word embeddings by factorizing the matrix into a product of two low-rank matrices. The first matrix captures the global context information, while the second matrix captures the local context information. The global context information refers to the overall distribution of words in the corpus, while the local context information refers to the co-occurrence patterns of words in the context of a specific word. By combining both global and local context information, GloVe can learn word embeddings that capture both semantic and syntactic relationships between words.

### 2.1.5 Character Embeddings

As we used pretrained GloVe [15] embeddings, but there will be some words in our training that will be out of vocabulary words. For those words, Seo et al.[19] in their research used 1-D convolution as Kim [26] used in his research, to get character

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.4 | 0.1 | -0.2 | 0.6 | 0.4 | 0.1 | -0.2 | 0.5 | -0.4 | 0.3 |
| -0.2 | 0.8 | 0.2 | 0.4 | 0.1 | -0.2 | 1.5 | 0.6 | 0.1 | 0.1 |
| 0.7 | -0.4 | 0.3 | 1.2 | 0.7 | 1.6 | 0.6 | 0.1 | -0.2 | 0.5 |
| -0.3 | 0.4 | 0.1 | -0.2 | -0.3 | 1.8 | 0.4 | 0.5 | 0.9 | 0.4 |
| 0.4 | 0.1 | -0.2 | 0.7 | -0.4 | 0.3 | 0.5 | 0.1 | -0.2 | 0.8 |
| **o** | **b** | **f** | **u** | **s** | **c** | **a** | **t** | **o** | **r** |

Figure 2.2: Matrix created after assigning vectors to all characters for word 'obfuscator'

After that, we created a convolution filter $\mathbb{H}$, which is also known as the kernel is a matrix that is used to scan the word. The height of $\mathbb{H}$ is the same as the dimensionality of character vectors $d$ and the width is always kept shorter than the length $l$ as shown in Fig. 2.3. $\mathbb{H}$ is also randomly initialized and trainable during model training.

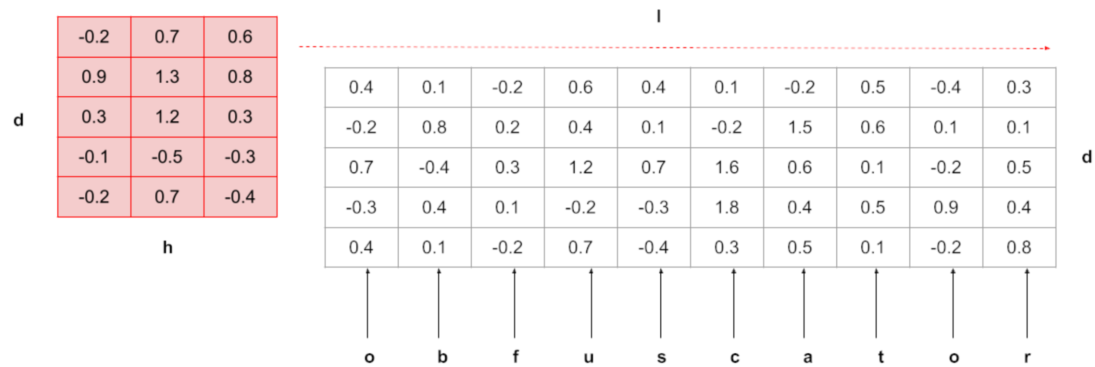


Figure 2.3: Kernel $\mathbb{H}$ scanning over the matrix $\mathbb{C}$

Then we overlay $\mathbb{H}$ over matrix $\mathbb{C}$ and take element-wise product of $\mathbb{H}$ and its projection on the matrix $\mathbb{C}$, which outputs a matrix with same dimensionality as $\mathbb{H}$ as shown in Fig. 2.4. Then we sum up all the numbers in the matrix obtained to get a scalar value. This scalar value is the first element of the vector f.


Figure 2.4: Kernel $\mathbb{H}$ is overlayed on the matrix $\mathbb{C}$

We repeat this step until we scanned over full-length $l$ and got a vector $f$. Then we max-pool the value from the vector $f$. This process is repeated with different convolution filters of different widths, resulting in summary scalars. Finally, the summary scalars from all these scanning processes are collected to form a character embedding of the given word.

### 2.1.6 Highway Networks

A highway network introduced by Srivastava et al. [21] consists of multiple layers, each equipped with gating mechanisms to control the flow of information. Let's consider a single highway layer for illustration purposes. The input to the layer is denoted as $\mathbf{x}$, and the output is denoted as $\mathbf{y}$. Highway networks employ two types of gating mechanisms the carry gate and the transform gate. The carry gate controls the direct flow of the input $\mathbf{x}$ to the output $\mathbf{y}$, while the transform gate controls the transformation applied to the input. These gates are implemented as sigmoid functions, ensuring that their values lie between 0 and 1. The output of a highway layer can be computed as follows:

$$\mathbf{y} = \mathbf{T}(\mathbf{x}) \times \mathbf{H}(\mathbf{x}) + \mathbf{x} \times (1 - \mathbf{T}(\mathbf{x})) \tag{2.3}$$

Here, $\mathbf{H}(\mathbf{x})$ represents the transformed input $\mathbf{x}$, while $\mathbf{T}(\mathbf{x})$ represents the transformation gate that controls the transformation. The term $\mathbf{x} \times (1 - \mathbf{T}(\mathbf{x}))$ corresponds to the carry gate, allowing the input to bypass the transformation process if deemed necessary. The transformation function $\mathbf{T}(\mathbf{x})$ takes the input $\mathbf{x}$ and applies a set of learned transformations to capture the complex relationships within the data. This function can be implemented using any neural network architecture, such as a feed-forward network or a convolutional neural network. In our research we used 1-D CNN as transformation function. By allowing the network to learn the transformation function, highway networks enable the adaptation of the transformation process to the specific task at hand.

### 2.1.7 Attention

The Attention function is defined as when we map queries and key-value pairs to an output where all queries, keys, values, and output are vectors. The output is computed as the weighted sum of values, where the weights assigned to each value is computed by a compatibility function of the query with corresponding key.

#### Scaled Dot-Product Attention

In the original research by Vaswani et al. [23], the defined inputs are queries and keys of dimension $d_k$, and values of dimension $d_v$. The authors computed the dot products of the queries with all keys, divided the result by $\sqrt{d_k}$, and applied a softmax function to obtain the weights on the values as shown in Eq. 2.4 where $W^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ are the trainable matrices projections.

$$Attention(Q, K, V) = \text{softmax}((\mathbf{QW}^Q)(\mathbf{KW}^K)^T/\sqrt{d}))(\mathbf{VW}^V) \tag{2.4}$$

$$MultiHead(Q, K, V) = Concat(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}^O \tag{2.5}$$

### Multi-Head Attention

In multi-head self-attention, we use multiple heads instead of just one, which allows the model to jointly attend to the information from different representations and represented as shown in below Eq. 2.5 where each head is calculated as shown in Eq. 2.4 where $\mathbf{W}^O$ is also a trainable weight matrix.

## 2.2 Related Work

The field of Natural Language Processing has seen a surge in interest in tasks like Machine Reading Comprehension and Question Answering. In MRC, the goal is to locate the exact answer within a given context, while Question Answering involves answering questions using common-sense reasoning. Unlike Question Answering, MRC doesn't require as much common-sense reasoning, making it simpler to evaluate during the testing phase. This has made it a favourable task for researchers in the NLP community. One significant advancement in this area has been the introduction of attention mechanisms. These mechanisms enable systems to focus on specific parts of a passage that are relevant to the task. An example of this progress is the Bidirectional Attention Flow for Machine Comprehension (BIDAF) model, developed by Seo et al. [19]. This model employs a multi-stage hierarchical approach that captures context at different levels of detail. By using bidirectional attention flow, it creates a context representation that takes into account the queries, all without prematurely summarizing the information. This approach achieved state-of-the-art performance in the MRC task.

One weakness of this model is its sluggishness in both training and inference due to its recurrent nature. In order to enhance the speed of machine comprehension, Yu et al. [25] introduced the QAnet model. Unlike its predecessor BIDAF, QAnet relies exclusively on convolutions and self-attention as the foundational components. These components are used to separately process the context and query. Then, the interactions between the context and query are learned using standard attention mechanisms as outlined by Bahdanau et al. [2]. The resulting information is once again encoded using encoders that don't rely on recurrence. This architectural approach not only accelerates the training process but also speeds up inference, making it a practical solution for deployment.

In the quest to enhance the efficiency and effectiveness of machine comprehension models, researchers have explored innovative architectural approaches. One notable advancement in this direction is the technique of incorporating constituency and dependency trees, which has interested many researchers. In their research, Tai et al. [22] improved the task performance by incorporating parse trees using an LSTM structure to encode parse trees recurrently. In their work, they demonstrated the effectiveness of Tree-LSTM on semantic relatedness and sentiment classification tasks. Structural Embedding of Syntactic Trees (SEST) proposed by Liu and Hu [9] encodes syntactic information of constituency and dependency trees using Bi-directional LSTM and showcased better results than the baseline model BIDAF Seo et al. [19] which was the state of the art model during that time.

One of the very novel approaches different from both Tree-LSTM and SEST called Ordered Neurons, was introduced by Shen et al. [20]. In their research, they proposed a novel recurrent unit called ON-LSTM, which included a new gating mechanism and a new activation function called cumax($\cdot$). With this, they brought RNNs closer to performing tree-like composition operations by separately allocating hidden state neurons with long and short-term information.

Tree-LSTM and SEST encoding approaches, respectively demonstrated by Tai et al. [22], and Li and Hu [9] and ON-LSTM by Shen et al. [20], while very effective, operate sequentially because of the sequential nature of LSTMs and lack behind current models incorporating self-attention as demonstrated by Vaswani et al. [23] in his research. Though the self-attention method has shown very promising results in Natural Language Processing, there is no evidence to the best of our knowledge that parse trees are implicitly encoded in self-attention based models such as Transformers.

Syntax-Bert, introduced by Bai et al. [3], worked on incorporating constituency trees effectively and efficiently into pre-trained Transformer models. In their research, they proposed Syntax-BERT, unlikely BERT which is based on complete self-attention topology, decomposed the self-attention mechanism into multiple subnetworks according to the tree structure. Each sub-network encapsulates one relationship from constituency trees, including ancestor, offspring and sibling relationships from different hops. They mentioned comparable results which verify the effectiveness of incorporating constituency trees in transformer-based models.

Tree-structured attention using hierarchical accumulation has been proposed by Nguyen et al. [12]. Although the approach we proposed in our work draws inspiration from this paper but to the best of our knowledge, we are not aware of any work on multi-sentence (multi-tree) incorporating hierarchical accumulation with convolution neural networks in a model. Hierarchical accumulation introduced by researchers used to encode the value component of each non-terminal node by aggregating the hidden states of all of its descendants. The accumulation process is in three stages, explained in detail in section 3.3.1. This approach incorporates constituency parse trees as an architecture bias to the self-attention mechanism of the Transformers network introduced by Vaswani et al. [23].

# Chapter 3

# Methodology

The methodology chapter of this thesis focuses on the approaches and techniques employed to conduct the research on Machine Reading Comprehension (MRC). This chapter begins by discussing the dataset used for the task of MRC, highlighting the limitations of previous datasets and the introduction of the Stanford Question Answering Dataset (SQuAD). SQuAD consists of a large collection of question-answer pairs extracted from Wikipedia articles and serves as a benchmark dataset for evaluating the performance of QA systems. The chapter then provides an overview of the updated version of SQuAD, SQuAD 2.0, which includes unanswerable questions to enhance the challenge for QA systems. Furthermore, the chapter covers data preprocessing steps, including the creation of parse trees and the binarization of the dataset. The methodology approach section describes the hierarchical accumulation technique used to incorporate the structural embedding of constituency trees in the question-answering model. This approach enables parallel processing and encoding of hierarchical structures, overcoming the limitations of sequential models. Overall, this chapter provides a comprehensive overview of the dataset, data preprocessing, and the proposed methodology approach for the research on MRC.

## 3.1 Dataset

The Machine Reading Comprehension (MRC) is a well-known NLP task that has gained significant attention and popularity over the years. Reading Comprehension, or the ability to read the text and then give answers about the context, is a challenging task for machines. Researchers have worked on this task for a very long on various datasets such as MCTest introduced by Richardson et al [18]. One of the main limitations that researchers found with MCTest was that the questions require commonsense reasoning, which hindered any noticeable progress in the task of MRC.

As dataset like MCTest for the task of MRC remains quite challenging, researchers from Stanford University created a new reading comprehension dataset consisting of more than 80,000 questions on a set of Wikipedia articles. Stanford Question Answering Dataset [17] was introduced in 2016 as a benchmark dataset for QA systems. The dataset consists of more than 80,000 question-answer pairs collected from Wikipedia articles. Each question-answer pair is associated with a specific paragraph of text from the corresponding article. The dataset evaluates a system's ability to answer questions based on context.

The dataset is split into a training set, a development set, and a test set. The training set contains 80,000 question-answer pairs, while the development and test sets contain 10,000 pairs. The dataset is balanced with respect to the type of questions asked, with approximately 50% being "who" questions, 30% being "what" questions, and the remaining 20% being "where," "when," "why" and "how" questions.

### 3.1.1 SQuAD 2.0

Stanford Question Answering Dataset 2.0 (SQuAD 2.0) [16] is an updated version of the original SQuAD dataset, introduced in 2018. The dataset is designed to be more challenging than the original dataset by introducing a new type of question, called the unanswerable question. In addition to the original dataset, SQuAD 2.0 includes more than 50,000 unanswerable questions, making it more difficult for QA systems to achieve high accuracy scores.

SQuAD 2.0 is also split into a training set, a development set, and a test set. The training set contains more than 130,000 question-answer pairs, while the development and test sets contain approximately 12,000 pairs. The distribution of question types in SQuAD 2.0 is similar to that of the original dataset.

Moreover, the dataset has been extensively used for the training of deep learning models, specifically in the area of neural network-based QA systems. The availability of large amounts of high-quality training data in SQuAD 2.0 has enabled researchers to develop more accurate and effective QA systems.

The key difference between SQuAD and SQuAD 2.0 is the inclusion of unanswerable questions in the latter dataset. While SQuAD only contains answerable questions, SQuAD 2.0 includes answerable and unanswerable questions, making it more challenging for QA systems to achieve high accuracy scores.

Including unanswerable questions in SQuAD 2.0 is a crucial feature that makes the dataset more challenging and representative of real-world situations. In many cases, providing a definitive answer to a question is difficult or impossible, and models that can identify when a question is unanswerable are more valuable in practical applications.

```
1 datasets["train"][1]
```


```
{'id': '56be85543aeaaa14008c9065',
 'title': 'Beyoncé',
 'context': 'Beyoncé Giselle Knowles-Carter (/biːˈjɒnseɪ/ bee-YON-say) (born September 4, 1981) is an American singer,
songwriter, record producer and actress. Born and raised in Houston, Texas, she performed in various singing and
dancing competitions as a child, and rose to fame in the late 1990s as lead singer of R&B girl-group Destiny\'s Child.
Managed by her father, Mathew Knowles, the group became one of the world\'s best-selling girl groups of all time.
Their hiatus saw the release of Beyoncé\'s debut album, Dangerously in Love (2003), which established her as a solo
artist worldwide, earned five Grammy Awards and featured the Billboard Hot 100 number-one singles "Crazy in Love" and
"Baby Boy".',
 'question': 'What areas did Beyonce compete in when she was growing up?',
 'answers': {'text': ['singing and dancing'], 'answer_start': [207]}}
```


Figure 3.1: One SQuAD 2.0 training example in the JSON format

In our research, we used the SQuAD 2.0 dataset considering all the features that it supports for deep neural network training. The dataset is available on the original website and can be downloaded easily. The dataset is in the popular JavaScript Object Notation (JSON) data format, as shown in Fig. 3.1, which can easily be converted into classes for further training purposes.

## 3.2 Data Prepossessing

In our research, we mainly focus on how to include structure embedding of syntactic trees in question-answering models. Therefore, prepossessing of the SQuAD data was the fundamental step for the research. In this research, we basically focused on incorporating the structural embedding of parse trees.

### 3.2.1 Creating Parse Trees on SQuAD 2.0

The first step in creating parse trees on SQuAD 2.0 is to preprocess the data. This involves tokenizing the text and assigning part-of-speech tags to each token. The Stanford Core NLP toolkit by Manning et al. [11] provides functions for both of these tasks.

Tokenization involves breaking up the text into individual tokens, which are typically words but can also include punctuation marks and other special characters. The tokenizer in the Stanford Core NLP toolkit uses a set of rules to determine how to split the text into tokens.

Part-of-speech tagging involves assigning a grammatical category to each token, such as a noun, verb, or adjective. The part-of-speech tagger in the Stanford Core NLP toolkit uses a statistical model to assign tags based on the context of the token within the sentence.

Once the text has been tokenized and tagged, the next step is to perform dependency parsing to create parse trees. The dependency parser in the Stanford Core NLP toolkit uses a transition-based algorithm, which means that it processes the sentence incrementally and builds the parse tree as it goes along.

```
(ROOT (S (PP (IN During) (NP (DT the) (JJ last) (JJ interglacial) (NN period))) (, ,) (NP (NP (DT the) (NML (NNP Red) (NNP Sea)) (NN coast)) (PP (IN of) (NP (NNP Eritrea)))) (
VP (VBD was) (VP (VBN occupied) (PP (IN by) (NP (JJ early) (ADJP (RB anatomically) (JJ modern)) (NNS humans))))) (. .)))#####------#####(ROOT (S (NP (PRP It)) (VP (VBZ is) (VP
(VBN believed) (SBAR (IN that) (S (NP (DT the) (NN area)) (VP (VBD was) (PP (IN on) (NP (DT the) (NN route))) (PP (IN out) (PP (IN of) (NP (NNP Africa)))) (SBAR (IN that) (S (
NP (DT some) (NNS scholars)) (VP (VBP suggest) (SBAR (S (VP (VBD was) (VP (VBN used) (PP (IN by) (NP (JJ early) (NNS humans))) (PP (IN to) (VP (VB colonize) (NP (NP (DT the) (
NN rest)) (PP (IN of) (NP (DT the) (NNP Old) (NNP World)))))))))))))))))) (. .)))#####------#####(ROOT (S (PP (IN In) (NP (CD 1999))) (, ,) (NP (NP (DT the) (JJ Eritrean) (NNP
Research) (NNP Project) (NNP Team)) (VP (VBN composed) (PP (IN of) (NP (ADJP (JJ Eritrean) (, ,) (JJ Canadian) (, ,) (JJ American) (, ,) (JJ Dutch) (CC and) (JJ French)) (NNS
scientists))))) (VP (VBD discovered) (NP (DT a) (JJ Paleolithic) (NN site)) (PP (IN with) (NP (NP (NML (NN stone) (CC and) (NN obsidian)) (NNS tools)) (VP (VBN dated) (ADJP (
NP (QP (IN to) (IN over) (CD 125,000)) (NNS years)) (JJ old)) (PP (IN near) (NP (NP (DT the) (NML (NML (NNP Bay)) (PP (IN of) (NP (NNP Zula)))) (NN south)) (PP (IN of) (NP (
NNP Massawa))))) (, ,) (PP (IN along) (NP (DT the) (NNP Red) (NNP Sea) (NNP littoral))))))) (. .)))#####------#####(ROOT (S (NP (DT The) (NNS tools)) (VP (VBP are) (VP (VBN
believed) (S (VP (TO to) (VP (VB have) (VP (VBN been) (VP (VBN used) (PP (IN by) (NP (JJ early) (NNS humans))) (S (VP (TO to) (VP (VB harvest) (NP (JJ marine) (NNS resources))
(PP (IN like) (NP (NNS clams) (CC and) (NNS oysters)))))))))))) (. .)))
```

```
(ROOT
    (SBARQ
        (WHNP
            (WDT What)
            (NN Team)
        )
        (SQ
            (VBD was)
            (VP
                (VBN composed)
                (PP
                    (IN of)
                    (NP
                        (ADJP
                            (JJ Eritrean)
                            (, ,)
                            (JJ Canadian)
                            (, ,)
                            (JJ American)
                            (, ,)
                            (JJ Dutch)
                            (CC and)
                            (JJ French)
                        )
                        (NNS scientists)
                    )
                )
            )
        )
        (. ?)
    )
)
```

Figure 3.2: Example parse trees for a context and a question

In our research, we created a utility program in Python `parseSQUAD.py`, which reads the context and question data local file system. It then processes the data and creates parse trees using the Stanford Core NLP by listening on port 9000. Once the parse trees are created, they are dumped using a Python 'pickle' dump file on the local file system. To separate multiple parse trees in context, we use the separator string '`#####------#####`' as in Fig. 3.2, which shows parse tree generated for one example including context and question.

We have used the separator to separate multiple parse trees for generating context, and when we read these parse trees, it made it easy to get all the parse trees for context as list items. We also used Byte-Pair Encoding (BPE) for generating these parse trees, which has been proven very effective for good results in various NLP research.

### 3.2.2 Creating Binarized Dataset

This section focuses on the creation of a binarized dataset for our research. We start by converting saved tree strings into nodes, leaves, spans, and Part-Of-Speech (POS) tags. Additionally, we build a vocabulary of tokens from our corpus. Inspired by Nguyen's work on Machine Translation, we utilize the Fairseq library [13] and its data utils class for vocabulary creation. Nodes represent the internal structure of sentences, leaves correspond to individual words with POS tags, and spans aid in identifying linguistic units. NLTK's `Tree.fromstring(str)` method is employed to generate trees, and we apply binarization techniques based on Nyugen's research. These preparations set the stage for further model training and analysis.

#### Creating Vocabulary of Tokens

After creating tree strings as described in Section 3.2.1, these saved tree strings have to be converted to nodes, leaves, spans, and Part-Of-Speech (POS) tags. Along with that we created a vocabulary or dictionary of all the tokens present in our corpus. To extract vocabulary we borrowed some classes from open source implementation of Nguyen [12] research. In their research, as they worked on Machine Translation task, they used Fairseq [13] library by Facebook for model training. In our research we used fairseq.data_utils class for vocabulary creation. We build the token list from all the trees and made a vocabulary out of it as shown in Fig. 3.3 and kept the same vocabulary for both contexts and questions.

#### Creating Nodes, Leaves, Spans, POS Tags from Parse Trees

In our research, in order to achieve hierarchical accumulation in parallelizable time we have to generate Nodes, Leaves, Spans and Part-Of-Speech (POS) tags before feeding data to our model. Nodes are the internal nodes of the parse tree, which represent the grammatical structure of the sentence. Each node has a label, which describes its role in the sentence. For example, the root node of the parse tree represents the entire

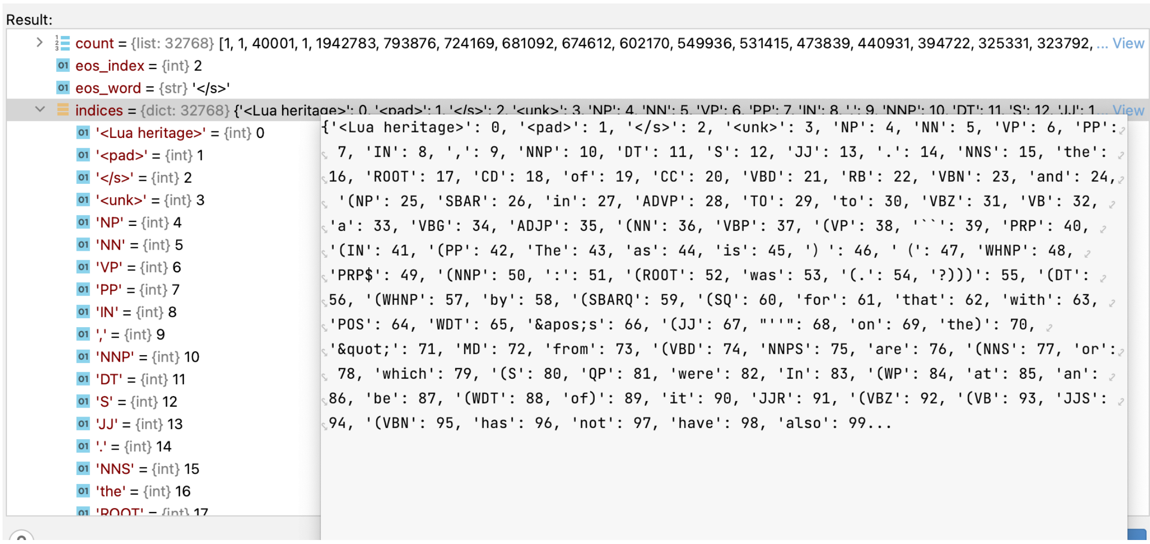


Figure 3.3: Snapshot of vocabulary obtained from tree strings

sentence, while other nodes represent phrases such as noun phrases or verb phrases. Leaves are the terminal nodes of the parse tree, which represent the individual words in the sentence. Each leaf has a label, which is the POS tag of the word. Spans are a sequence of nodes or leaves that share a common ancestor in the parse tree. Spans are useful for identifying phrases or other linguistic units in the sentence. For example, a span might correspond to a noun phrase or a verb phrase. The POS tags are labels that indicate the grammatical function of a word. POS tags are typically assigned to the leaves of the parse tree, and can be used for tasks such as named entity recognition or sentiment analysis. To generate nodes, leaves, spans, and POS tags from parse tree strings, we used NLTK's `Tree.fromstring(str)` as shown in Fig. 3.4 method to generate the tree out of saved strings. And, for binarizing these values we used some classes (e.g., Binarizer.py) from Nguyen [12] research, to binarize the way he binarized data before feeding it to the model for further training.

## 3.3 Methodology Approach

In this section, we address the challenge of incorporating hierarchical structures of language. Previous studies have attempted to utilize parse and dependency trees with recurrent models, but these models are sequential and difficult to parallelize and train. In our research, we propose a similar approach to Nguyen [12], aiming

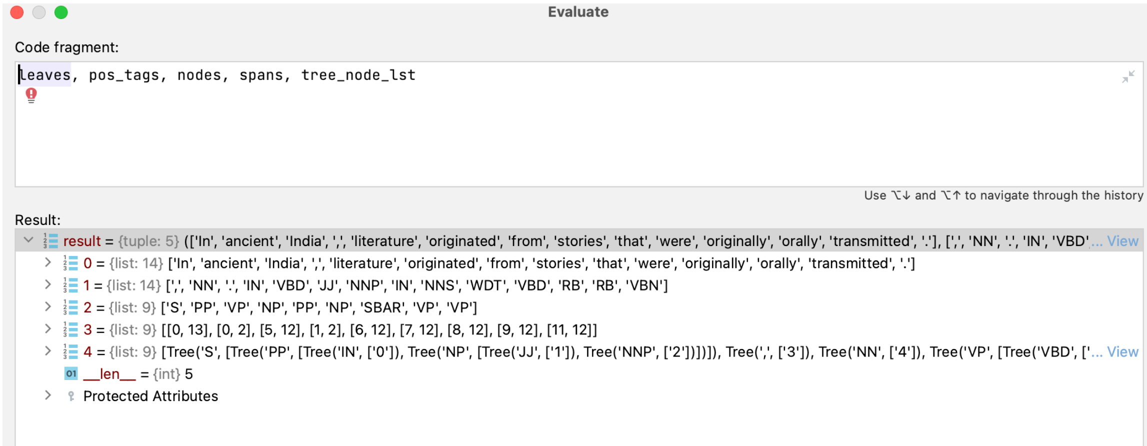


Figure 3.4: Snapshot of extracting Nodes, Leaves, Spans and POS tags from tree strings

to represent hierarchical structures in a parallelizable data structure. We describe the process of tree accumulation, where we assign values to nodes and leaves based on their relationships and rules. We then perform upward cumulative averaging to compose node representations in a bottom-up fashion. By following this methodology, we aim to capture the hierarchical structure of language effectively.

### 3.3.1 Hierarchical Accumulation in Self Attention

Researchers have been trying to incorporate hierarchical structures of language for a very long time. In their research, Liu and Hu [9] incorporated the parse tree and dependency trees using Bidirectional LSTMs. But as we know, recurrent models are sequential by nature and are not parallelizable. Moreover they are very difficult to train because of vanishing and exploding gradients problems. In our research, we did something very similar to what Nguyen [12] did in his research. To encode hierarchical structure in parallel, we have to represent a data structure that can be parallelized. Let us see the hierarchical accumulation of trees with an example. Given a sentence $X$ of length $n$, let $G\{X\}$ be the directed spanning tree which represents the parse tree of $X$, produced by a parser. We define a transformation $\mathcal{F}(X) = (\mathcal{L}, \mathcal{N}, \mathcal{R})$, where $\mathcal{L}$ denotes the ordered sequence of $n$ terminal nodes of the parse tree, and $\mathcal{N}$ denotes the set of $m$ non-terminal nodes such as VP or NP, and $\mathcal{R}$ is a mapping over all non-terminal nodes in $\mathcal{N}$ such that for each node $x \in N$, $R(X)$ denotes a set of all non-terminal and terminal nodes that belong to the subtree rooted at $x$. For

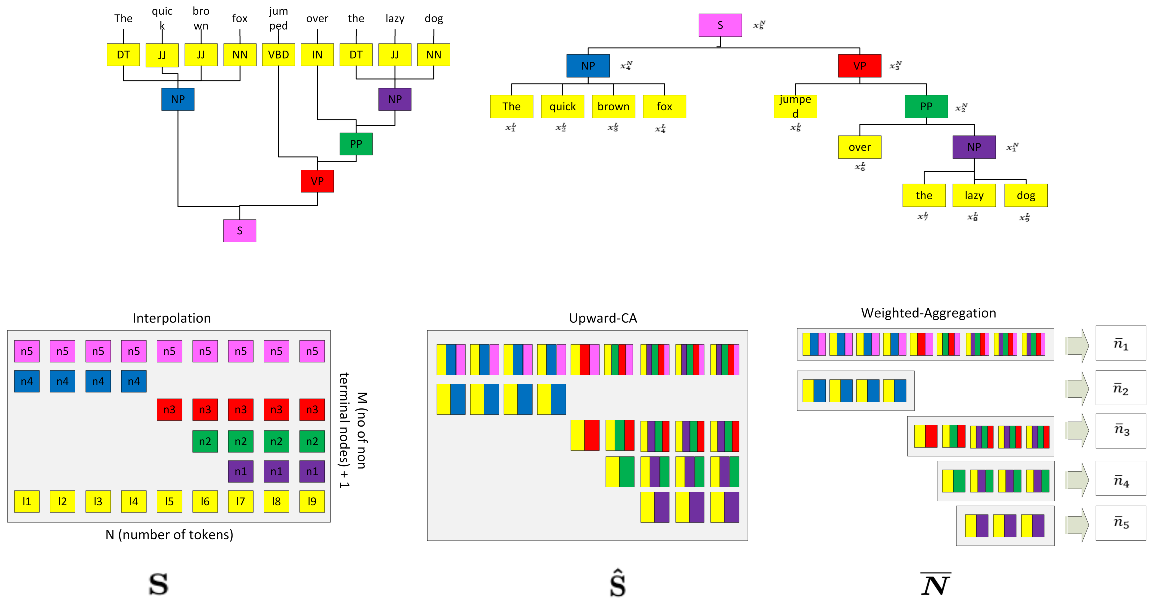


Figure 3.5: The hierarchical accumulation process of tree structures. Given a parse tree, it is interpolated into a tensor $\mathbf{S}$, which is then accumulated vertically from bottom to top to produce $\hat{\mathbf{S}}$. Next, the (branch-level) component representations of the non-terminal nodes are combined into one representation as $\overline{\boldsymbol{N}}$ by weighted aggregation [12]. The figure is based on original work of Nguyen [12], is further modified to accommodate changes specific to our research.

example, for non-terminals $x_1^{\mathcal{N}}$ and $x_2^{\mathcal{N}}$ in Fig. 3.6, $\mathcal{R}(x_1^{\mathcal{N}}) = \{x_1^{\mathcal{N}}, x_7^{\mathcal{L}}, x_8^{\mathcal{L}}, x_9^{\mathcal{L}}\}$ and $\mathcal{R}(x_2^{\mathcal{N}}) = \{x_2^{\mathcal{N}}, x_6^{\mathcal{L}}, x_1^{\mathcal{N}}, x_7^{\mathcal{L}}, x_8^{\mathcal{L}}, x_9^{\mathcal{L}}\}$.

We will now describe the tree accumulation method. The tree accumulation method uses hidden vector representations of leaves and non-terminal nodes of dimension $d$. The leaf representations are actually input coming from the network, and non-terminal node representations are randomly initialized and then trained through the tree accumulation method. Fig. 3.5 shows the overall process, but we will go through one example. Let $L = (l_1, l_2, \ldots, l_n)$ and $N = (n_1, n_2, \ldots, n_m)$ be the hidden representations of the leaves $\mathcal{L} = (x_1^{\mathcal{L}}, \ldots, x_n^{\mathcal{L}})$ and nodes $\mathcal{N} = (x_1^{\mathcal{N}}, \ldots, x_n^{\mathcal{N}})$, respectively. We then apply function $\mathcal{F} : (\mathbb{R}^{n\times d}, \mathbb{R}^{m\times d}) \rightarrow \mathbb{R}^{(m+1)\times n\times d}$, which takes $\mathcal{L}$, $\mathcal{N}$, and $\mathcal{R}$ as input and returns a tensor $\mathbf{S} \in \mathbb{R}^{(m+1)\times n\times d}$, using Eq. 3.1:

$$\mathbf{S}_{i,j} = \mathcal{F}(L, N, R)_{i,j} = \begin{cases} l_j & \text{if } i = 1 \\ n_{i-1} & \text{else if } x_j^{\mathcal{L}} \in \mathcal{R}(x_{i-1}^{\mathcal{N}}) \\ 0 & otherwise, \end{cases} \tag{3.1}$$

where $1 \leq i \leq m+1$, $1 \leq j \leq n$, and $\mathbf{S}_{i,j}$ is a $d$-dimensional vector. Let us take an example to understand it better. Since the matrix $\mathbf{S}$ corresponds to the parse tree in which the leaves are on bottom we use matrix indexing starting from bottom-left corner; i.e., the element $\mathbf{S}_{1,1}$ is the leftmost bottom element and so on. In Fig. 3.6, to form matrix $\mathbf{S}$, let us say we try to fill position $\mathbf{S}_{i=4,j=6}$ (in accumulation rows of $\mathbf{S}$ as counted from the bottom). Since $i > 1$, we check if $x_j^{\mathcal{L}} \in \mathcal{R}(x_{i-1}^{\mathcal{N}})$. $\mathcal{R}(x_3^{\mathcal{N}}) = \{x_5^{\mathcal{L}}, x_3^{\mathcal{N}}, x_2^{\mathcal{N}}, x_6^{\mathcal{L}}, x_2^{\mathcal{N}}, x_1^{\mathcal{N}}, x_7^{\mathcal{L}}, x_8^{\mathcal{L}}, x_9^{\mathcal{L}}\}$. As $x_6^{\mathcal{L}} \in \mathcal{R}(x_3^{\mathcal{N}})$, hence $\mathbf{S}_{4,6} = n_3$. Similarly, we fill the whole matrix $\mathbf{S}$. In matrix $\mathbf{S}$, 0 denotes a zero vector of length

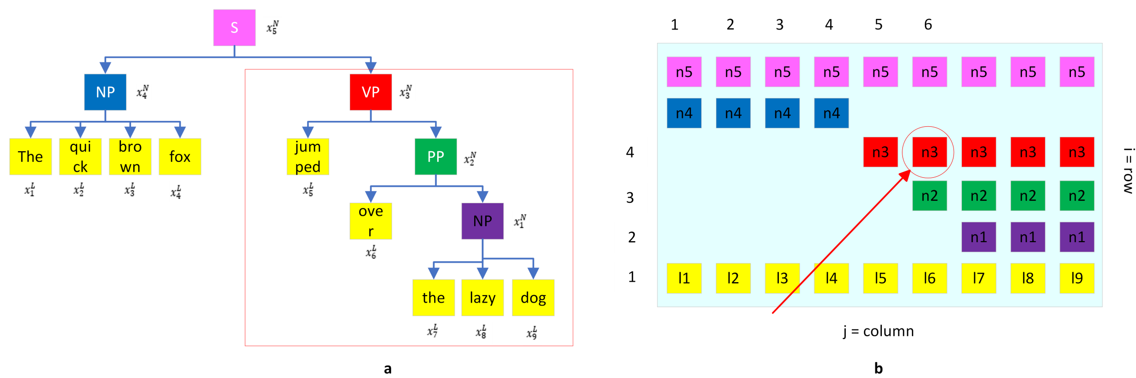


Figure 3.6: Interpolation shows how given a parse tree is interpolated into a tensor $\mathbf{S}$. The figure is based on original work of Nguyen [12], is further modified to accommodate changes specific to our research.

$d$, and Nguyen [12] in his paper omitted POS tags of the words which constitute the preterminal nodes in a constituency tree. In our research, we accommodated these embeddings by simply concatenating them with word embedding. Nguyen [12] in his research takes random embedding for POS Tags. Similarly, in our research, we used random initialized vectors for POS Tags. Next, we will perform an upward cumulative average function $\mathcal{U}$ on $\mathbf{S}$ to compose the node representations in a bottom-up fashion over the induced tree structure. The result of this operation will be a tensor $\hat{\mathbf{S}} \in \mathbb{R}^{m \times n \times d}$, in which each non-terminal node representation is averaged along with all its descendants in a particular branch.

$$\mathcal{U}(\mathbf{S})_{i,j} = \hat{\mathbf{S}}_{i,j} = \begin{cases} 0 & \text{if } \mathbf{S}_{(i+1),j} = 0 \\ \sum_{\mathbf{S}_{t,j} \in C_j^i} \mathbf{S}_{t,j} / |C_j^i| & \text{otherwise.} \end{cases} \tag{3.2}$$

where $C_j^i$ is the set of vectors in $\mathbf{S}$ representing the leaves and nodes in the part of a column of matrix $\mathbf{S}$ that starts with $x_i^{\mathcal{N}}$ and ends with $x_j^{\mathcal{L}}$. Let us take an example how $\hat{\mathbf{S}}$ will be calculated with an example as illustrated in Fig. 3.7. Let us say, we try to fill position $\hat{\mathbf{S}}_{i=3,j=6}$. The case 1 from Eq. 3.2 fails as $\mathbf{S_{4,6}} \neq 0$. Therefore, after applying the case 2 of Eq. 3.2 we will sum all the leaves and nodes in a branch that starts from $x_i^{\mathcal{N}} = x_3^{\mathcal{N}}$ and ends with $x_j^{\mathcal{L}} = x_6^{\mathcal{L}}$ which are $\{n_3, n_2, l_6\}$ as shown in below

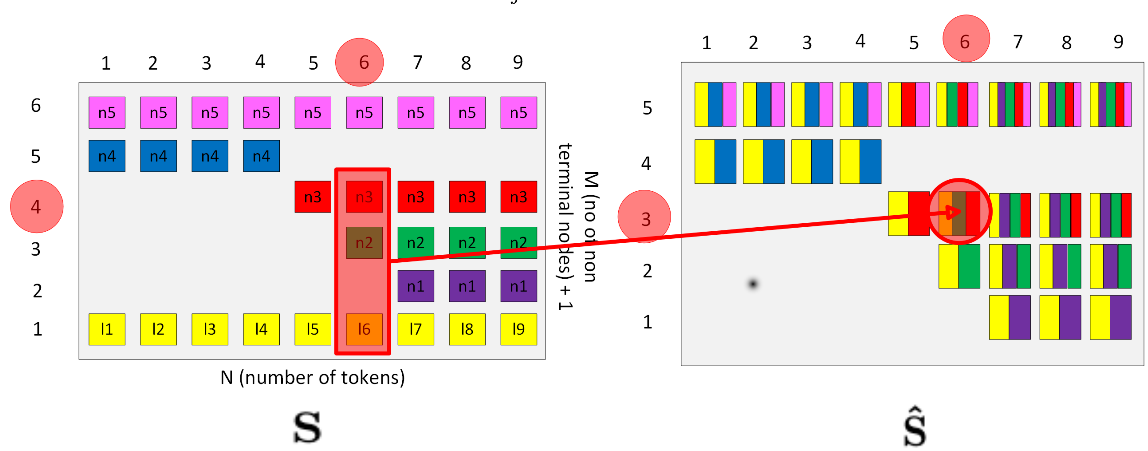


Figure 3.7: Upward Cumulative Average shows how it is accumulated vertically from bottom to top to produce $\hat{\mathbf{S}}$. The figure is based on original work of Nguyen [12], is further modified to accommodate changes specific to our research.

of a non-terminal node $x_i^{\mathcal{N}}$ into single vector $\bar{n}_i$ that encapsulates all the elements in subtree rooted by $x_i^{\mathcal{N}}$. Nguyen in his research did a weighted aggregation operation $\mathcal{V}$ and we will do something very similar to that. We will take $\hat{\mathbf{S}}$ as input and a weighting vector $w \in \mathbb{R}^n$, and computes the final node representation $\bar{N} = (\bar{n}_1, ..., \bar{n}_m) \in \mathbb{R}^{m \times d}$, where each row vector is computed by Eq. 3.3 and can be visualized as shown in Fig. 3.8:

$$\mathcal{V}(\hat{\mathbf{S}}, w)_i = \bar{n}_i = \frac{1}{|\mathcal{L} \cap \mathcal{R}(x_i^{\mathcal{N}})|} \sum_{j: x_j^{\mathcal{L}} \in \mathcal{R}(x_i^{\mathcal{N}})} w_j \odot \hat{\mathbf{S}}_{i,j} \tag{3.3}$$

$$\overline{\boldsymbol{N}}' = \mathcal{V}\left(\mathcal{U}\left(\mathcal{F}\left(\boldsymbol{L}\boldsymbol{W}^V, \boldsymbol{N}\boldsymbol{W}^V, \mathcal{R}\right)\right), \boldsymbol{w}\right) \tag{3.4}$$

The above-explained hierarchical accumulation process is used in the model architecture described in section 4.3.5 where we used self-attention in the encoder block of our model. And we perform hierarchical operation as in Eq. 3.4 where $\boldsymbol{w} = \mathbf{L}\boldsymbol{u}_s$ with $\boldsymbol{u}_s \in \mathbb{R}^d$ and $\boldsymbol{W}^Q$, $\boldsymbol{W}^K$, $\boldsymbol{W}^V$, $\boldsymbol{W}^O \in \mathbb{R}^{d \times d}$ are the trainable weight matrices. The process outlined above illustrates how a parse tree is initially transformed into a matrix, which is a parallelizable data structure. This transformation process is termed

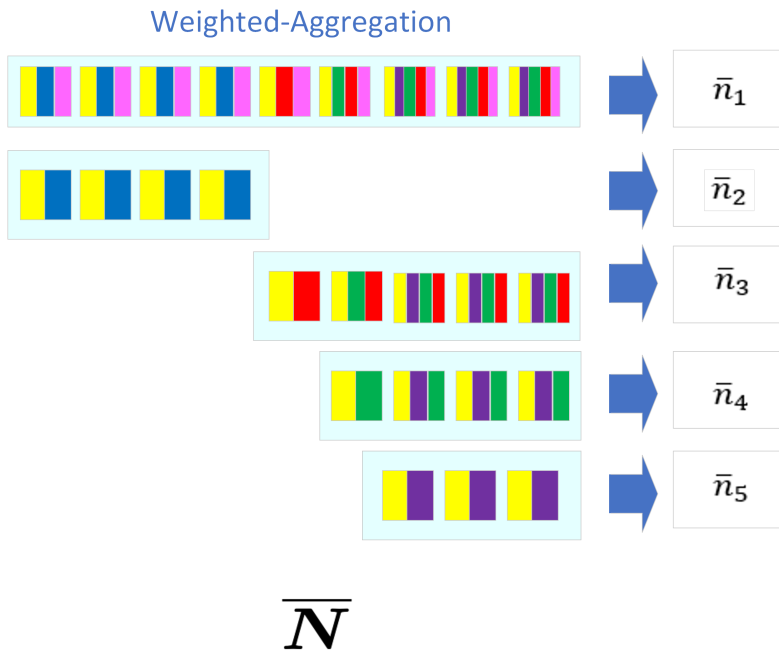


Figure 3.8: Figure illustrates weighted aggregation in which the branch-level component representations of the non-terminal nodes are combined into one representation as $\overline{\boldsymbol{N}}$. The figure is based on original work of Nguyen [12], is further modified to accommodate changes specific to our research.

interpolation, as depicted in Fig. 3.6. This interpolation is carried out to leverage the efficiency of matrix computations in deep learning. Subsequently, the matrix $\mathbf{S}$ undergoes a vertical accumulation process to perform an upward cumulative averaging, resulting in the formation of $\hat{\mathbf{S}}$, as illustrated in Fig. 3.7. Moving forward, the representations of the non-terminal nodes at the branch level are combined into a single representation referred to as $\overline{\boldsymbol{N}}$. This is achieved through a weighted aggregation approach illustrated in Fig. 3.8.

# Chapter 4

# Model Architecture

This chapter of the thesis delves into the model architecture used to address the problem statement of extracting a span from a given context paragraph and query sentence, which we more precisely define in the first section. We introduce two models: QAnet and QATnet, which are designed to tackle this task. The QAnet model consists of five major components, including an embedding layer, embedding encoder layer, context-query attention layer, model encoder layer, and output layer. It stands out by leveraging convolution and self-attention mechanisms for improved performance. On the other hand, the QATnet model follows a similar structure but incorporates multi-head self-attention with hierarchical accumulation. This chapter provides a detailed explanation of the various layers and components of both models, highlighting their unique characteristics and functionalities.

## 4.1 Problem Statement

The problem statement targeted in this thesis research is defined as follows. For a given context paragraph of $n$ words $C = (c_1, c_2, c_3..., c_n)$ and a query sentence $Q = (q_1, q_2, ..., q_m)$, the task is to find a span $S = (c_i, c_{i+1}, ..., c_{i+j})$ as a substring of the original paragraph $C$ that is the most relevant and correct answer to the query $Q$. The relevance is based on the meaning of the query, the paragraph, and the spans; i.e., different substrings of the paragraph, where correctness is based on manual labelling.

## 4.2 QAnet

### 4.2.1 Overview

The high-level structure of QAnet model as illustrated in Fig. 4.1, is similar to most existing reading comprehension models with five major components: an embedding layer, an embedding encoder layer, a context-query attention layer, a model encoder layer, and an output layer. However, the major differences between QAnet model and other models are as follows: For both embedding and modelling encoders the researchers used only convolution and self-attention mechanism as described by Vaswani et al. [23] in his research. As per the researchers of QAnet [25] paper, self-attention combined with depth-wise convolution shows better results than only self-attention encoders. As illustrated in Fig. 4.1, the encoder block in the middle is the unit block that contains multiple convolution, self attention and feed-forward layers. These encoder blocks are stacked together as shown in blue Stacked Model Encoder Blocks in main architecture.

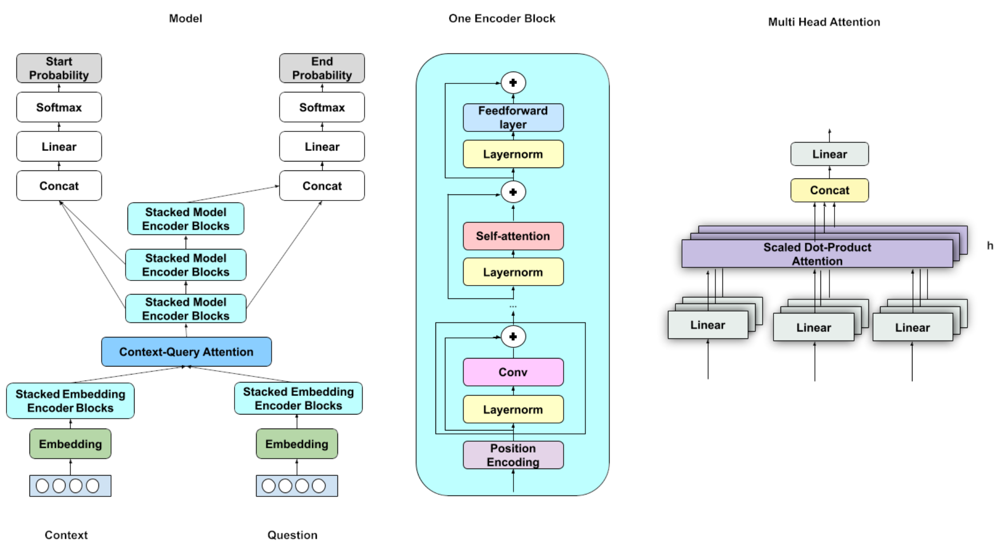


Figure 4.1: QAnet [25] Model with each encoder block further magnified to show Multi-Head Self-Attention. The figure is based on original work of Yu et al. [25], is further modified to accommodate changes specific to our research.

### 4.2.2 Input Embedding Layer

As adopted in the original research, in our thesis research, we also adopted standard embedding of each word $w$ by concatenating the word and its character embedding. We used fixed $p_1 = 300$ dimensional pre-trained Glove [15] embedding word vectors, which are fixed during training. All out-of-vocabulary words are mapped to a trainable `<UNK>` token, which embedding is initialized randomly. The character embedding is obtained as follows: each vector is represented as a trainable vector of dimension

$p_2 = 200$, which means that each word can be viewed as a concatenation of embedding vectors for each of its characters. And then, we take maximum value of each row from this matrix to get fixed size vector representation of each word. Finally, the output of a given word $x$ from embedding layer is $[x_w; x_c] \in \mathbf{R}^{p_1+p_2}$, where $x_w$ is the pre-trained Glove embedding and $x_c$ are the convolution output of character embedding of $x$ respectively. As in the original paper QAnet [25], we also used a two-way highway network on top of this representation.

### 4.2.3 Encoder Embedding Layer

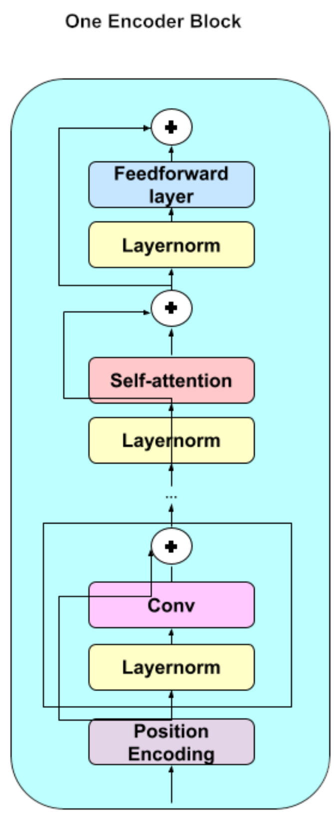


Figure 4.2: Figure illustrates one embedding encoder block consisting of [convolution-layer × 4 + self-attention-layer with hierarchical accumulation + feed-forward-layer]. The figure is based on original work of Yu et al. [25], is further modified to accumulate changes specific to our research.

The encoder embedding layer is a stack of the following building blocks as explained in the original paper QAnet [25]: [ convolution-layer × # + self-attention-layer + feed-forward-layer ] in Fig. 4.2 where we took # = 4 according to original implementation which is number of convolution layers each block has. Similar to the original paper QAnet [25] we also used depth-wise convolution as compared to traditional ones with a kernel size of 7, the number of filters is $d = 128$ and the number of convolution layers within a block is 4 and is represented as #. For the self-attention layer, similar to Vaswani et al. [23] we kept number of heads as 8 in all encoder layers. The basic configuration ( conv + self-attention + ffn ) is placed inside a residual block for better gradient flow during backpropagation. The total number of encoder blocks is 1. Note that the input of this layer is a vector of dimension $p_1 + p_2 = 500$ for each individual word, which is immediately mapped to a lower-dimensional space ($d = 128$) by a one-dimensional convolution. The output of this layer is a also of dimension $d = 128$.

### 4.2.4 Context-Query Attention Layer

We use $C$ and $Q$ to denote the encoded context and query. The context-to-query attention is constructed as follows: We first compute the similarities between each pair of context and query words, rendering a similarity matrix $S \in \mathbb{R}^{n \times m}$. We then normalize each row of $S$ by applying the softmax function, getting a matrix $\bar{S}$. Then the context-to-query attention is computed as $A = \bar{S} \cdot Q^T \in \mathbb{R}^{n \times d}$. The similarity function used here is the trilinear function in the original paper Wei et al [25]: $f(q, c) = W_0[q; c; q \circ c]$, where $\circ$ is the element-wise multiplication and $W_0$ is a trainable variable. we also computed the column normalized matrix $\bar{\bar{S}}$ of S by softmax function, and the query-to-context attention is $B = \bar{S} \cdot \bar{\bar{S}}^T \cdot C^T$. This way of calculating query-to-context attention is first introduced in researches like Bidirectional Attention Flow for Machine Comprehension (BIDAF) [19] and Dynamic Co-attention Networks For Question Answering (DCN) [24] and we followed same in the our implementation of QAnet [25].

### 4.2.5 Model Encoder Layer

Similar to the original paper QAnet [25], the input to this layer at each position is $[c, a, c \odot a, c \odot b]$, where $a$ and $b$ are respectively rows of attention of matrices $A$ and $B$. The layer parameters are the same as the Embedding encoder layer, except that convolution layers are 2 within each block, and the total number of blocks used is 7. All weights are shared between each of 3 repetitions of the model encoder.

### 4.2.6 Output Layer

The last layer is task-specific, as we are solving the SQuAD question-answering problem, and each example in SQuAD is labelled with a span in the context containing the answer. We also adapted the strategy to predict the probability of each position in the context being the start or end of the answer span.

$$p^1 = softmax(W_1[M_0; M_1]),\ p^2 = softmax(W_2[M_0; M_2]), \tag{4.1}$$

where $W_0$ and $W_1$ are the trainable variables and $M_0$, $M_1$, and $M_2$ are outputs of 3 model encoders. The score of a span is the product of its start position and end position probabilities. Finally, the objective function is defined as the negative sum of log probabilities of the predicted distributions indexed by true start and end indices, averaged over all the training examples where $y_i^1$ and $y_i^2$, are respectively the ground-truth starting and ending position of example i.

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\left[\log\left(p^1_{y_i^1}\right) + \log\left(p^2_{y_i^2}\right)\right] \tag{4.2}$$

## 4.3 QATnet

### 4.3.1 Overview

The high-level structure of QATnet as illustrated in Fig. 4.3 is close to original QAnet model. An embedding layer, an embedding encoder, a context-query attention layer, a model encoder layer and an output layer. However, the main difference in the model encoders is that we implemented multi-head self-attention with hierarchical accumulation as Nguyen [12] did in his research. As illustrated in Fig. 4.3, the encoder block in the middle is the unit block that contains multiple convolution, self attention and feed-forward layers. These encoder blocks are stacked together as shown in blue Stacked Model Encoder Blocks in main architecture.

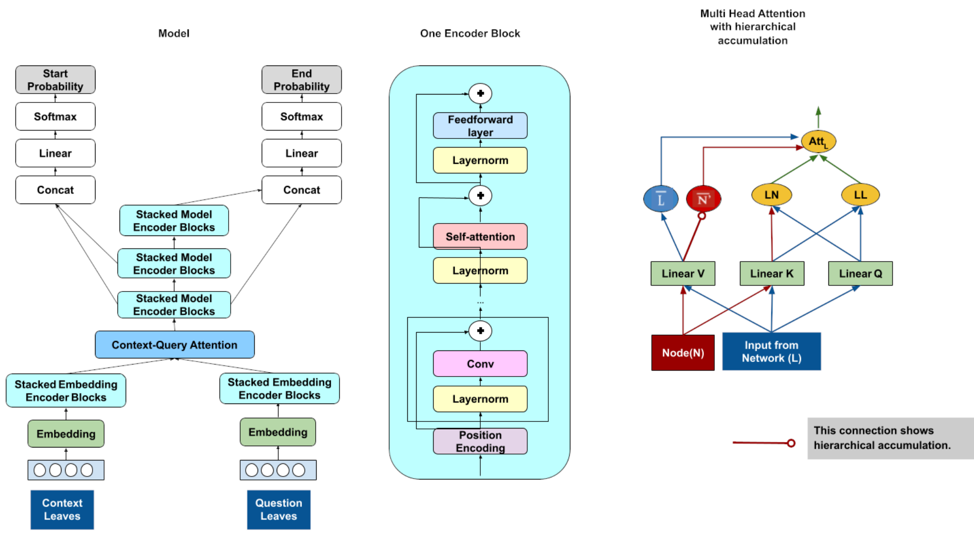


Figure 4.3: QATnet model is shown in the figure, which is further magnified to show each encoder block consisting of Multi-Head Self-Attention using hierarchical accumulation. The figure is based on original work of Yu et al. [25] and Nguyen [12], is further modified to accommodate changes specific to our research.

### 4.3.2 Input Embedding Layer

As adopted in the QAnet model, in our QATnet model, we also adopted standard embedding of each word $w$ by concatenating its word and character embedding. As input to this layer, we sent leaves which are the same as the original context and question in the SQUAD problem statement. We kept everything else the same as mentioned in section 4.2.2.

### 4.3.3 Encoder Embedding Layer

In our QATnet model, we kept the self-attention layer in the encoder block the same as what Vaswani et al. [23] used in their research. We did not use Self-attention with hierarchical accumulation in the Encoder embedding block because in the early stage of testing, we found that doing that changed the meaning of used pre-trained Glove [15] embeddings. So we kept the encoder embedding block very similar to what we have mentioned in section 4.2.3 .

### 4.3.4 Context-Query Attention Layer

In our QATnet model, we used the context query attention layer similar to the QAnet model. We first computed the similarities between each pair of context and query words, rendering a similarity matrix and then normalized each row, and CQ Attention is computed as described in Section 4.2.4.

### 4.3.5 Model Encoder Layer

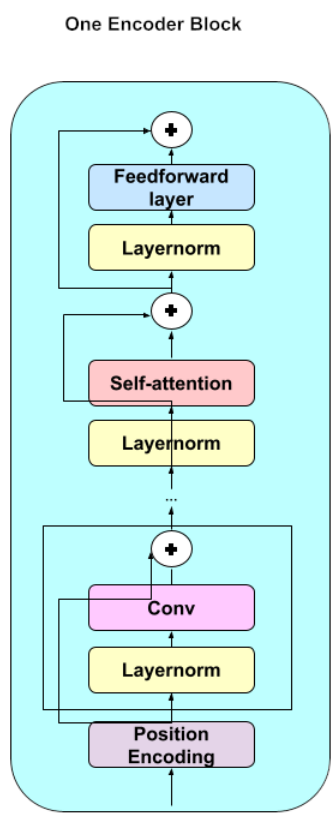


Figure 4.4: Figure illustrates one model encoder block consisting of [convolution-layer $\times$ 2 + self-attention-layer with hierarchical accumulation + feed-forward-layer ]. The figure is based on original work of Yu et al. [25], is further modified to accommodate changes specific to our research.

The Model encoder embedding layer is a stack of the following building block as explained in the original paper: [ convolution-layer × # + self-attention-layer with hierarchical accumulation + feed-forward-layer ] shown in Fig. 4.4. Self attention layer in the model encoder block is similar to what Nguyen [12] used in his research. We also used depth-wise convolution with a kernel size of 7, the number of filters is $d = 128$ and the number of convolution layers within a block is 2. For the self-attention with hierarchical accumulation layer, we kept number of heads as 8 in all encoder layers. The basic ( conv + self-attention with hierarchical accumulation + ffn ) is placed inside a residual block for better gradient flow during backpropagation. The total number of encoder blocks are 7.

### Encoder Self Attention with Hierarchical Accumulation

Let $\mathbf{L} \in \mathbb{R}^{n\times d}$ and $\mathbf{N} \in \mathbb{R}^{m\times d}$ respectively denote output from convolution layer of encoder block and node, along with parse tree represented as $T(X) = (\mathcal{L}, \mathcal{N}, \mathcal{R})$. First, we compute query-key affinity matrices $\mathbf{A}_{LL} \in \mathbb{R}^{n\times n}$ and $\mathbf{A}_{LN} \in \mathbb{R}^{n\times m}$ as follows:

$$\mathbf{A}_{LL} = (\mathbf{L}\mathbf{W}^{Q})(\mathbf{L}\mathbf{W}^{K})^{T}/\sqrt{d} \qquad \mathbf{A}_{LN} = (\mathbf{L}\mathbf{W}^{Q})(\mathbf{N}\mathbf{W}^{K})^{T}/\sqrt{d} \tag{4.3}$$

Then, the value representation $\overline{\mathbf{L}}$ of the output from conv layers $\mathbf{L}$, which have hidden leaf representations in them, is computed by a linear layer. Meanwhile, the value representation $\overline{\mathbf{N}'}$ of the nodes $\mathbf{N}$ is encoded using the tree structure using hierarchical accumulation, as explained in Section 3.3.1.

$$\overline{\boldsymbol{N}}' = \mathcal{V}\left(\mathcal{U}\left(\mathcal{F}\left(\boldsymbol{L}\boldsymbol{W}^{V}, \boldsymbol{N}\boldsymbol{W}^{V}, \mathcal{R}\right)\right), \boldsymbol{w}\right) \tag{4.4}$$

$$\overline{\boldsymbol{L}} = \boldsymbol{L}\boldsymbol{W}^{V} \tag{4.5}$$

where $\boldsymbol{w} = \mathbf{L}\boldsymbol{u}_{s}$ with $\boldsymbol{u}_{s} \in \mathbb{R}^{d}$ and $\boldsymbol{W}^{Q}$, $\boldsymbol{W}^{K}$, $\boldsymbol{W}^{V}$, $\boldsymbol{W}^{O} \in \mathbb{R}^{d\times d}$ are the trainable weight matrices as explained in Section 2.1.7. The final attention of leaves as illustrated in Fig. 4.5 is then computed as weighted averages of value vectors in $\overline{\mathbf{L}}$ and $\overline{\mathbf{N}'}$.

$$\mathrm{Att}_{L} = \mathrm{softmax}\left(\mu([\boldsymbol{A}_{LN}; \boldsymbol{A}_{LL}])\right)\left[\overline{\boldsymbol{N}}'; \overline{\boldsymbol{L}}\right] \tag{4.6}$$

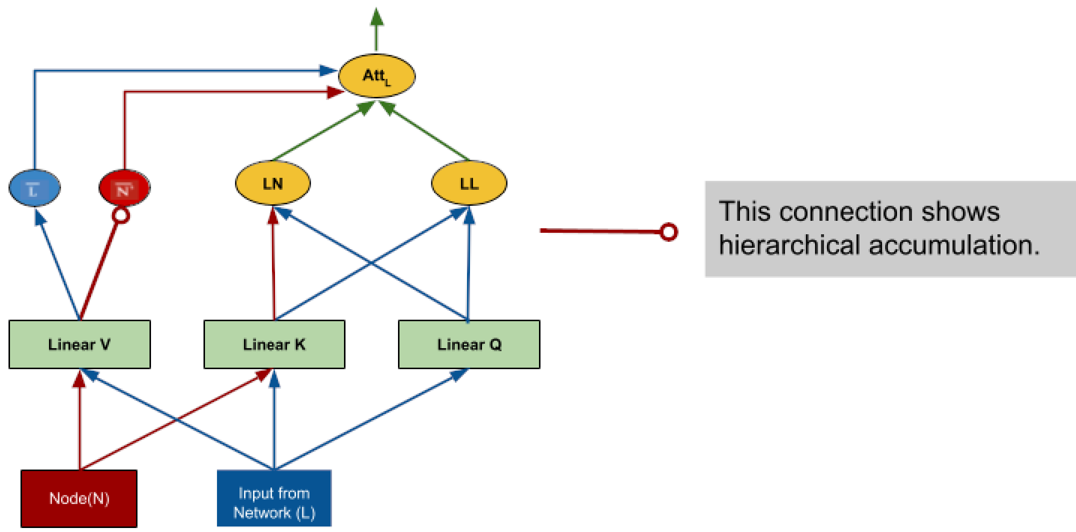


Figure 4.5: Self-Attention with hierarchical accumulation, circle-ended arrows indicate where hierarchical accumulations take place. The figure is based on original work of Nguyen [12], is further modified to accommodate changes specific to our research.

### 4.3.6 Output Layer

The last Output layer is exactly similar to what is explained in Section 4.2.6.

$$p^1 = softmax(W_1[M_0; M_1]),\ p^2 = softmax(W_2[M_0; M_2]), \tag{4.7}$$

where $W_0$ and $W_1$ are the trainable variable and $M_0, M_1, M_2$ are outputs of 3 model encoders. The score of a span is the product of its start position and end position probabilities. Finally, the objective function is defined as the negative sum of log probabilities of the predicted distributions indexed by true start and end indices, averaged over all the training examples where $y_i^1$ and $y_i^2$, are respectively the ground-truth starting and ending position of example i.

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\left[\log\left(p^1_{y_i^1}\right) + \log\left(p^2_{y_i^2}\right)\right] \tag{4.8}$$

# Chapter 5

# Results and Analysis

## 5.1 Experimental Settings

During training, we padded sentences with <PAD>, which were shorter than the maximum context length of 400 and any paragraph longer was discarded. We use pre-trained Glove [15] word embeddings, and all out-of-vocabulary words were replaced by <UNK> and were trained during training. For words, we kept the embedding dimension of 300, and for characters, we kept it to 200. We did not perform any data augmentation; therefore train set had examples of around 128.8k, and development (dev) set had examples of around 12k. We kept only two splits of train and dev because the test set for SQUAD2.0 is not available for download and is hidden. One has to submit the code to a Codelab and work with the authors of SQUAD2.0 [16] to retrieve final results. In our experiments, we only report the performance on dev set or also called validation set. And according to our experiments and previous works such as Seo et al. [19] and Wei et al. [25], the validation score is very correlated with the test score. We use two types of regularization techniques, first L2 regularization with nothing but weight decay on all the trainable parameters, with parameter $\lambda = 3 \times 10^{-7}$. We also used dropout on both embeddings and between layers. For word embeddings we kept dropout of 0.1, and for character embedding, we kept dropout of 0.05. And dropout between every two layers is 0.1. We also use the layer dropout method as shown in the original implementation of QAnet Wei et al. [25]. The size of hidden layer and convolutions filters is 128, and we took the batch size of 16. The total convolution layers in embedding and model encoders are 4 and 2, respectively. For optimizer we used ADAM, with $\beta_1 = 0.8, \beta_2 = 0.999, \epsilon = 10^{-7}$. We used a warm-up learning rate with an increase from 0.0 to 0.001 for the first 100 steps, and then $lr$ is maintained at 0.001. An exponential moving average is applied to all trainable variables with a decay rate of 0.9999. We implemented our model in python using PyTorch [14] and performed all experiments on Compute Canada Beluga server GPU node 2.

## 5.2 Results

The F1 and the Exact Match (EM) are the two evaluation metrics of accuracy for the model performance, where F1 as shown in Eq. 5.4 is the harmonic mean between measures of Precision and Recall, whereas EM as shown in Eq. 5.1 is 1 only if it is the exact match as ground truth else 0.

$$\text{EM} = \begin{cases} 1 & \text{if prediction\_tokens == ground\_truth\_tokens} \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

$$\text{Precision} = \frac{\text{common\_tokens}}{\text{len(prediction\_tokens)}} \tag{5.2}$$

$$\text{Recall} = \frac{\text{common\_tokens}}{\text{len(ground\_truth\_tokens)}} \tag{5.3}$$

$$\text{F1} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{5.4}$$

We show the results in comparison of both models in Table 5.1. We also give the training history as shown in Figure 5.1 of these models and the results shown are on the dev set. Our QATnet model as shown with green in Figure 5.1a and 5.1b is very close to the baseline, which shows that hierarchical accumulation can give competitive results and in the next section, we analyze the test cases where it outperforms the baseline QAnet model. The presented Table 5.2 provides a comparative analysis of the EM and F1 measures for the QAnet and QATnet models, based on 10 observations. Statistical tests were conducted to evaluate the significance of the observed differences between the models. For the EM measure, QAnet achieved a maximum value of 64.11, while QATnet attained a maximum of 60.61. The mean EM score for QAnet was 62.83 ($\pm$ 1.85), whereas QATnet had a mean of 59.80 ($\pm$ 0.60). These differences were found to be statistically significant ($p < 0.05$), indicating a notable variation in performance between the models. Regarding the F1 measure, QAnet demonstrated a maximum score of 75.50, surpassing QATnet's maximum of 72.94. The mean F1 score for QAnet was 74.70 ($\pm$ 1.40), while QATnet achieved a mean of 72.5 ($\pm$ 0.50). Statistical tests revealed significant differences ($p < 0.05$) between the two models in terms of F1 performance. The statistical analyses performed emphasize the distinct performance characteristics of QAnet and QATnet. The observed differences in EM

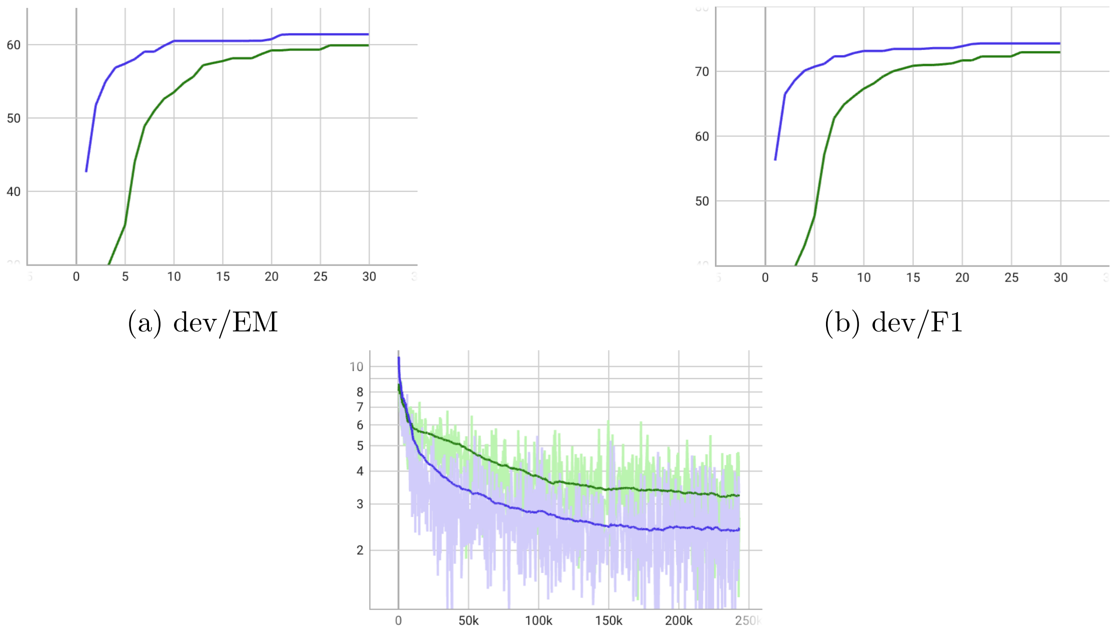


Figure 5.1: Training history of the models QAnet (blue) and QATnet (green)

and F1 scores between the models are statistically significant, suggesting varying levels of effectiveness in question answering tasks.

## 5.3 Analysis

Despite the overall performance of our model QATnet falling short of the baseline model QAnet, our research uncovered numerous instances where our model demonstrated superiority. Detailed analysis of the results highlighted that the integration of structural embedding of constituency trees enabled our model to excel in various aspects. Specifically, it exhibited a remarkable ability to retain contextual information over longer distances and exhibit heightened attention toward punctuation and other grammatical intricacies. These findings solidify our belief in the efficacy of incorporating constituency tree structures to enhance language models.

In Table 5.3, the example shows how our model managed to retain the context from a long distance. As the example was not answerable which means it is hard for any model to spot the right answer because to answer, it had to retain context from long distances. As without asking explicitly about Phase II, the question is

| Epoch | QAnet | | QATnet | |
|---|---|---|---|---|
| | Dev/EM | Dev/F1 | Dev/EM | Dev/F1 |
| Epoch 1 | 42.61250 | 56.22314 | 20.70758 | 31.03000 |
| Epoch 2 | 51.75000 | 66.48670 | 27.25735 | 37.95541 |
| Epoch 3 | 54.97500 | 68.63228 | 28.93410 | 39.75668 |
| Epoch 4 | 56.86250 | 70.12614 | 32.19537 | 43.14000 |
| Epoch 5 | 57.40000 | 70.72369 | 35.41772 | 47.64151 |
| Epoch 6 | 58.03750 | 71.17191 | 44.06580 | 57.19414 |
| Epoch 7 | 59.05000 | 72.31226 | 48.89980 | 62.77154 |
| Epoch 8 | 59.05000 | 72.31226 | 50.98171 | 64.87769 |
| Epoch 9 | 59.85000 | 72.82371 | 52.62009 | 66.13778 |
| Epoch 10 | 60.50000 | 73.14573 | 53.49349 | 67.31842 |
| Epoch 11 | 60.50000 | 73.14573 | 54.74795 | 68.10809 |
| Epoch 12 | 60.50000 | 73.14573 | 55.62532 | 69.23988 |
| Epoch 13 | 60.50000 | 73.45302 | 57.18634 | 70.02139 |
| Epoch 14 | 60.50000 | 73.45302 | 57.49029 | 70.43170 |
| Epoch 15 | 60.50000 | 73.45302 | 57.75689 | 70.85730 |
| Epoch 16 | 60.50000 | 73.47356 | 58.13914 | 70.97303 |
| Epoch 17 | 60.50000 | 73.58324 | 58.13914 | 70.97303 |
| Epoch 18 | 60.52500 | 73.58324 | 58.13914 | 71.08414 |
| Epoch 19 | 60.52500 | 73.58324 | 58.72269 | 71.23402 |
| Epoch 20 | 60.73750 | 73.87633 | 59.21542 | 71.68954 |
| Epoch 21 | 61.35000 | 74.21101 | 59.21542 | 71.68954 |
| Epoch 22 | 61.41250 | 74.31665 | 59.32175 | 72.30894 |
| Epoch 23 | 61.41250 | 74.31665 | 59.32175 | 72.30894 |
| Epoch 24 | 61.41250 | 74.31665 | 59.32175 | 72.30894 |
| Epoch 25 | 61.41250 | 74.31665 | 59.32175 | 72.30894 |
| Epoch 26 | 61.41250 | 74.31665 | 59.89882 | 72.94437 |
| Epoch 27 | 61.41250 | 74.31665 | 59.89882 | 72.94437 |
| Epoch 28 | 61.41250 | 74.31665 | 59.89882 | 72.94437 |
| Epoch 29 | 61.41250 | 74.31665 | 59.89882 | 72.94437 |
| Epoch 30 | 61.41250 | 74.31665 | 59.89882 | 72.94437 |

Table 5.1: EM and F1 scores on dev set for both QAnet-baseline and QATnet-hierarchical.

| | EM | | | F1 | | |
|---|---|---|---|---|---|---|
| **Model (n=10)** | **Max** | **Mean (± SD)** | **p-value** | **Max** | **Mean (± SD)** | **p-value** |
| QAnet | 64.11 | 62.83 (± 1.85) | 0.0014 (< 0.05) | 75.50 | 74.70 (± 1.40) | 0.0016 (< 0.05) |
| QATnet | 60.61 | 59.80 (± 0.60) | | 72.94 | 72.5 (± 0.50) | |

Table 5.2: Comparison of EM and F1 Measures for QAnet and QATnet (10 Observations)

Table 5.3: Case Study 1

| **Context and Question** | **Answer** |
|---|---|
| Context: DECnet is a suite of network protocols created by Digital Equipment Corporation, originally released in 1975 in order to connect two PDP-11 minicomputers. It evolved into one of the first peer-to-peer network architectures, thus transforming DEC into a networking powerhouse in the 1980s. Initially built with three layers, it later (1982) evolved into a seven-layer OSI-compliant networking protocol. The DECnet protocols were designed entirely by Digital Equipment Corporation. However, DECnet Phase II (and later) were open standards with published specifications, and several implementations were developed outside DEC, including one for Linux.<br>Question: What did DECnet Phase I become? | Answerable: 0<br><br>Ground truth: open standards with published specifications<br><br>QATnet: open standards<br><br>QAnet: a networking powerhouse |

"what does Phase I become?" The baseline QAnet model struggled to access what the Phase I is and looked for a similar word to "become," which is "transforming" in the context and gave the wrong answer. Model QAnet miserably failed to access the word Phase I, which is not mentioned in the context. However, our model QATnet managed to retain the gist of context over the longer distance and was also able to access what Phase I would have become later and answered not the exact match but most parts of the ground truth. Our model QATnet artfully managed to gauge what the question is asking about Phase II rather than Phase I, which QAnet model could not and simply answered about Phase I instead. This is one of the many more examples in which our model QATnet was able to remember the context over long distances, which validates that structural embedding of syntactic trees managed to retain context over long distances as compared to the baseline QAnet model with no information of syntactic trees. Many more examples like this while studying results substantiate that due to the incorporation of syntactic trees, the QATnet model was able to capture the context over long distances and outperformed the baseline QAnet model where it was needed to retain information from the whole context to answer.

In Table 5.4, our model showcased an extraordinary resemblance to the behaviour exhibited in the preceding example. Remarkably, this time around, the model demonstrated an exceptional aptitude for preserving contextual coherence across extensive

Table 5.4: Case Study 2

| **Context and Question** | **Answer** |
|---|---|
| Context: The early United States expressed its opposition to Imperialism, at least in a form distinct from its own Manifest Destiny, through policies such as the Monroe Doctrine. However, beginning in the late 19th and early 20th century, policies such as Theodore Roosevelt's interventionism in Central America and Woodrow Wilson's mission to "make the world safe for democrac" changed all this. They were often backed by military force, but were more often affected from behind the scenes. This is consistent with the general notion of hegemony and imperium of historical empires. In 1898, Americans who opposed imperialism created the Anti-Imperialist League to oppose the US annexation of the Philippines and Cuba. One year later, a war erupted in the Philippines causing business, labor and government leaders in the US to condemn America's occupation in the Philippines as they also denounced them for causing the deaths of many Filipinos. American foreign policy was denounced as a "racket" by Smedley Butler, an American general. He said, "Looking back on it, I might have given Al Capone a few hints. The best he could do was to operate his racket in three districts. I operated on three continents".<br>Question: Which country besides the Cuba did the United states try to annex in 1898? | Answerable: 1<br><br>Ground truth: 'the Philippines', 'Philippines', 'Philippines', 'Philippines', 'Philippines'<br><br>QATnet: Philippines<br><br>QAnet: Cuba |

distances, even within the question component. The inquiry explicitly entailed the identification of a country besides Cuba, yet the QAnet model, serving as our baseline, peculiarly fixated on the annexation aspect instead. However, in stark contrast, our revolutionary QATnet model astutely retained and processed the vital information pertaining to the query for an alternative country to Cuba. This demonstrates that our model not only preserves information over extended contextual spans but also does so within the question itself.

In Table 5.5, the example demonstrates how our QATnet model effectively incorporates punctuation cues within the context. Specifically, when asked about the union of Spain and Portugal within the European Union (EU), the baseline QAnet model struggled to recognize the significance of the comma. Conversely, our model,

Table 5.5: Case Study 3

| **Context and Question** | **Answer** |
|---|---|
| Context: The principal Treaties that form the European Union began with common rules for coal and steel, and then atomic energy, but more complete and formal institutions were established through the Treaty of Rome 1957 and the Maastricht Treaty 1992 (now: TFEU). Minor amendments were made during the 1960s and 1970s. Major amending treaties were signed to complete the development of a single, internal market in the Single European Act 1986, to further the development of a more social Europe in the Treaty of Amsterdam 1997, and to make minor amendments to the relative power of member states in the EU institutions in the Treaty of Nice 2001 and the Treaty of Lisbon 2007. Since its establishment, more member states have joined through a series of accession treaties, from the UK, Ireland, Denmark and Norway in 1972 (though Norway did not end up joining), Greece in 1979, Spain and Portugal 1985, Austria, Finland, Norway and Sweden in 1994 (though again Norway failed to join, because of lack of support in the referendum), the Czech Republic, Cyprus, Estonia, Hungary, Latvia, Lithuania, Malta, Poland, Slovakia and Slovenia in 2004, Romania and Bulgaria in 2007 and Croatia in 2013. Greenland signed a Treaty in 1985 giving it a special status.<br>Question: In what years did Spain and Portugal join the European Union? | Answerable: 1<br><br>Ground truth: 1985<br><br>QATnet: Spain and Portugal 1985<br><br>QAnet: 1979 |

benefiting from structural embeddings of syntactic trees, successfully attributed importance to punctuation and provided the correct answer. This observation extends to numerous other instances during the analysis of both models, highlighting how the structural embedding approach enhances the model's ability to assign value to punctuation marks and subsequently derive accurate answers. Considering the broader context and syntactic relationships, our QATnet model showcases improved grammatical accuracy and coherency, empowering it to handle a wider range of linguistic nuances and produce more reliable responses.

In Table 5.6, both models performed well, but our research consistently showed that the QATnet model demonstrates a stronger focus on grammatical nuances when

Table 5.6: Case Study 4

| **Context and Question** | **Answer** |
|---|---|
| Context: Between 1832 and 2002 the currency of Greece was the drachma. After signing the Maastricht Treaty, Greece applied to join the eurozone. The two main convergence criteria were a maximum budget deficit of 3% of GDP and a declining public debt if it stood above 60% of GDP. Greece met the criteria as shown in its 1999 annual public account. On 1 January 2001, Greece joined the eurozone, with the adoption of the euro at the fixed exchange rate 340.75 to €1. However, in 2001 the euro only existed electronically, so the physical exchange from drachma to euro only took place on 1 January 2002. This was followed by a ten-year period for eligible exchange of drachma to euro, which ended on 1 March 2012.<br>Question: What did Greece sign to apply to join the eurozone? | Answerable: 1<br><br>Ground truth: Maastricht Treaty<br><br>QATnet: the Maastricht Treaty<br><br>QAnet: Maastricht Treaty |

answering questions. This is evident in Case Study 4, where the QATnet model paid attention not only to the main components of the sentence but also to determiners. We observed similar behaviour in various other instances where the QATnet model allocated attention to determiners, adjectives, nouns, and pronouns. These findings strongly support our belief that incorporating structural embedding of syntactic trees could significantly enhance a model's ability to attend to the grammatical nuances of languages. By including syntactic tree structures, models can capture the hierarchical relationships between words and phrases, thereby gaining a better understanding of the underlying grammatical structure. This approach empowers the model to assign attention to various grammatical elements, such as determiners, adjectives, nouns, and pronouns, which play crucial roles in constructing accurate and contextually appropriate responses. Overall, our research suggests that incorporating syntactic tree structures holds immense potential for improving models' treatment of grammatical nuances and enhancing their language understanding capabilities.

# Chapter 6

# Conclusion and Future Work

In this final chapter, we conclude our research on combining constituency trees with attention mechanisms in neural network models for machine comprehension tasks. We highlight our key contributions and suggest future research directions. This chapter summarizes our findings and sets the stage for progress in the field. Our investigation into integrating constituency trees aims to enhance neural network models' ability to capture context and improve performance in language understanding tasks.

## 6.1 Conclusion

In this research, we have explored the integration of constituency trees into the attention mechanism of neural network models for machine comprehension tasks. The development of our model, QATnet, involved creating constituency trees and constructing a binarized dataset that facilitated the hierarchical accumulation of constituency trees within the attention mechanism of encoder layers. Through extensive evaluation on the SQuAD 2.0 [16] dataset, we compared the performance of QATnet with the baseline QAnet [25] model. While QATnet slightly lagged behind QAnet [25] in overall performance, it exhibited remarkable strengths in specific areas. One notable advantage was its ability to retain contextual information over longer distances, enabling a better understanding of complex passages. Additionally, QATnet demonstrated an enhanced attention towards punctuation and grammatical intricacies, suggesting its potential in capturing finer linguistic details. The integration of constituency trees into the attention mechanism added an extra layer of understanding and improved attention distribution. By considering the hierarchical structures of sentences, QATnet showcased a more comprehensive understanding of the relationships between words and phrases. This unique feature contributed to its improved performance in certain scenarios, highlighting its potential for capturing long-range dependencies and context-driven variations. In conclusion, our research has provided valuable insights into the implications of incorporating constituency trees into machine comprehension models. The findings highlight the advantages and capabilities of QATnet in capturing contextual information and improving performance.

### 6.2 Future Work

Several avenues can be explored to build upon the findings of this research and further enhance the incorporation of constituency trees into machine comprehension models. The following areas can be considered:

1. **Investigation of Hierarchical Accumulation in Other Models:** While QATnet demonstrated promising results, it is worth exploring the application of hierarchical accumulation techniques with other neural network models for machine comprehension tasks. Different models may have distinct characteristics that can benefit from the integration of constituency trees. Conducting experiments with various models, such as BERT [5], RoBERTa [10], or Transformer-XL [4], can provide insights into the generalizability and effectiveness of hierarchical accumulation across different architectures.

2. **Visualization of Attention Heads:** To gain a deeper understanding of how the attention mechanism interacts with constituency trees, future work can focus on visualizing the attention heads. Visualizations can help analyze which parts of the input are receiving higher attention and provide insights into the model's decision-making process. Examining the attention distributions and identifying patterns or biases can lead to further improvements and interpretability of the model's performance.

3. **Hyperparameter Tuning:** Hyperparameter tuning plays a crucial role in optimizing the performance of neural network models. Future research can explore different hyperparameter settings for QATnet or other models using the hierarchical accumulation technique. Systematic experimentation and tuning of hyperparameters such as learning rate, batch size, and regularization methods can lead to improved results and better convergence.

4. **Comparison with Other Tree Structures:** While this research focused on constituency trees, future work can investigate the integration of other tree structures, such as dependency trees or semantic role labeling trees. Comparing the performance of models using different tree structures can shed light on the influence of tree representations on machine comprehension tasks and provide insights into the benefits and trade-offs of each approach.

# Bibliography

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. Syntax-BERT: Improving pre-trained transformers with syntax trees, 2021.

[4] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[8] Dan Jurafsky and James H. Martin. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.* Pearson Prentice Hall, Upper Saddle River, N.J., 2009.

[9] Rui Liu, Junjie Hu, Wei Wei, Zi Yang, and Eric Nyberg. Structural embedding of syntactic trees for machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 815–824, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[11] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[12] Xuan-Phi Nguyen, Shafiq Joty, Steven CH Hoi, and Richard Socher. Tree-structured attention with hierarchical accumulation. *arXiv preprint arXiv:2002.08046*, 2020.

[13] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[14] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019.

[15] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[16] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *CoRR*, abs/1806.03822, 2018.

[17] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

[18] Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[19] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

[20] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. Ordered neurons: Integrating tree structures into recurrent neural networks, 2019.

[21] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *CoRR*, abs/1505.00387, 2015.

[22] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks, 2015.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. volume abs/1706.03762, 2017.

[24] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*, 2016.

[25] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. *CoRR*, abs/1804.09541, 2018.

[26] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

# Appendix A

## Additional Examples Bolstering our Findings

Table A.1: Additional instances that strengthen our findings, showcasing the QATnet model's superiority over the QAnet model.

| **Context and Question** | **Answer** |
|---|---|
| Context: During the mid-Eocene, it is believed that the drainage basin of the Amazon was split along the middle of the continent by the Purus Arch. Water on the eastern side flowed toward the Atlantic, while to the west water flowed toward the Pacific across the Amazonas Basin. As the Andes Mountains rose, however, a large basin was created that enclosed a lake; now known as the Solimões Basin. Within the last 5–10 million years, this accumulating water broke through the Purus Arch, joining the easterly flow toward the Atlantic.<br>Question: Where did water to the east of the Amazon drainage basin flow towards? | • Answerable: 1<br>• Ground truth: 'the Atlantic', 'the Atlantic', 'Atlantic'<br>• QATnet: the Atlantic<br>• QAnet: the Pacific across the Amazonas Basin |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: Jacques Legardeur de Saint-Pierre, who succeeded Marin as commander of the French forces after the latter died on October 29, invited Washington to dine with him. Over dinner, Washington presented Saint-Pierre with the letter from Dinwiddie demanding an immediate French withdrawal from the Ohio Country. Saint-Pierre said, "As to the Summons you send me to retire, I do not think myself obliged to obey it." He told Washington that France's claim to the region was superior to that of the British, since René-Robert Cavelier, Sieur de La Salle had explored the Ohio Country nearly a century earlier.<br>Question: How did Saint-Pierre respond to Washington? | • Answerable: 1<br>• Ground truth: "As to the Summons you send me to retire, I do not think myself obliged to obey it.", 'said, "As to the Summons you send me to retire, I do not think myself obliged to obey it.", "As to the Summons you send me to retire, I do not think myself obliged to obey it.", "I do not think myself obliged to obey", "As to the Summons you send me to retire, I do not think myself obliged to obey it"<br>• QATnet: As to the Summons you send me to retire"'<br>• QAnet: with the letter from Dinwiddie demanding an immediate French withdrawal from the Ohio Country |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: In October 2010, the open-access scientific journal PLoS Pathogens published a paper by a multinational team who undertook a new investigation into the role of Yersinia pestis in the Black Death following the disputed identification by Drancourt and Raoult in 1998. They assessed the presence of DNA/RNA with Polymerase Chain Reaction (PCR) techniques for Y. pestis from the tooth sockets in human skeletons from mass graves in northern, central and southern Europe that were associated archaeologically with the Black Death and subsequent resurgences. The authors concluded that this new research, together with prior analyses from the south of France and Germany, ". . . ends the debate about the etiology of the Black Death, and unambiguously demonstrates that Y. pestis was the causative agent of the epidemic plague that devastated Europe during the Middle Ages".<br>Question: In what year were Polymerase Chain Reactions first used by researchers? | • Answerable: 0<br>• Ground truth: '1998'<br>• QATnet: 1998<br>• QAnet: 2010, the open-access scientific journal PLoS Pathogens published a paper by a multinational team who undertook a new investigation into the role of Yersinia pestis in the Black Death following the disputed identification by Drancourt and Raoult in 1998 |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: Formed in November 1990 by the equal merger of Sky Television and British Satellite Broadcasting, BSkyB became the UK's largest digital subscription television company. Following BSkyB's 2014 acquisition of Sky Italia and a majority 90.04% interest in Sky Deutschland in November 2014, its holding company British Sky Broadcasting Group plc changed its name to Sky plc. The United Kingdom operations also changed the company name from British Sky Broadcasting Limited to Sky UK Limited, still trading as Sky. Question: What is the name of the United Kingdom operation for BSkyB? | • Answerable: 1<br>• Ground truth: 'Sky UK Limited', 'Sky UK Limited', 'Sky UK Limited'<br>• QATnet: British Sky Broadcasting Limited to Sky UK Limited<br>• QAnet: Sky |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: Prince Louis de Condé, along with his sons Daniel and Osias,[citation needed] arranged with Count Ludwig von Nassau-Saarbrücken to establish a Huguenot community in present-day Saarland in 1604. The Count supported mercantilism and welcomed technically skilled immigrants into his lands, regardless of their religion. The Condés established a thriving glass-making works, which provided wealth to the principality for many years. Other founding families created enterprises based on textiles and such traditional Huguenot occupations in France. The community and its congregation remain active to this day, with descendants of many of the founding families still living in the region. Some members of this community emigrated to the United States in the 1890s. Question: Who was Count Ludwig von Nassau-Saarbucken's father? | • Answerable: 0<br>• Ground truth: 'Prince Louis de Condé'<br>• QATnet: Prince Louis de Condé<br>• QAnet: Daniel and Osias,[citation |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
| --- | --- |
| Context: Former IPCC chairman Robert Watson has said “The mistakes all appear to have gone in the direction of making it seem like climate change is more serious by overstating the impact. That is worrying. The IPCC needs to look at this trend in the errors and ask why it happened”. Martin Parry, a climate expert who had been co-chair of the IPCC working group II, said that “What began with a single unfortunate error over Himalayan glaciers has become a clamour without substance” and the IPCC had investigated the other alleged mistakes, which were “generally unfounded and also marginal to the assessment”. Question: What substantial error put the IPCC research in doubt? Question: Who was Count Ludwig von Nassau-Saarbucken’s father? | • Answerable: 0<br>• Ground truth: ’error over Himalayan glaciers’<br>• QATnet: Himalayan glaciers<br>• QAnet: ask why it happened |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: Around 1800 Richard Trevithick and, separately, Oliver Evans in 1801 introduced engines using high-pressure steam; Trevithick obtained his high-pressure engine patent in 1802. These were much more powerful for a given cylinder size than previous engines and could be made small enough for transport applications. Thereafter, technological developments and improvements in manufacturing techniques (partly brought about by the adoption of the steam engine as a power source) resulted in the design of more efficient engines that could be smaller, faster, or more powerful, depending on the intended application. Question: In what year did Oliver Evans patent his device?<br>Question: Who was Count Ludwig von Nassau-Saarbucken's father? | • Answerable: 0<br>• Ground truth: '1802'<br>• QATnet: 1802<br>• QAnet: 1801 |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: None of the original treaties establishing the European Union mention protection for fundamental rights. It was not envisaged for European Union measures, that is legislative and administrative actions by European Union institutions, to be subject to human rights. At the time the only concern was that member states should be prevented from violating human rights, hence the establishment of the European Convention on Human Rights in 1950 and the establishment of the European Court of Human Rights. The European Court of Justice recognised fundamental rights as general principle of European Union law as the need to ensure that European Union measures are compatible with the human rights enshrined in member states' constitution became ever more apparent. In 1999 the European Council set up a body tasked with drafting a European Charter of Human Rights, which could form the constitutional basis for the European Union and as such tailored specifically to apply to the European Union and its institutions. The Charter of Fundamental Rights of the European Union draws a list of fundamental rights from the European Convention on Human Rights and Fundamental Freedoms, the Declaration on Fundamental Rights produced by the European Parliament in 1989 and European Union Treaties.<br>Question: What other entity was established at the same time as the European Convention on Human Rights? | • Answerable: 1<br>• Ground truth: 'European Court of Human Rights.', 'the European Court of Human Rights', 'European Court of Human Rights'<br>• QATnet: the European Court of Human Rights<br>• QAnet: European Court of Human Rights |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: Orientalism, as theorized by Edward Said, refers to how the West developed an imaginative geography of the East. This imaginative geography relies on an essentializing discourse that represents neither the diversity nor the social reality of the East. Rather, by essentializing the East, this discourse uses the idea of place-based identities to create difference and distance between "we" the West and "them" the East, or "here" in the West and "there" in the East. This difference was particularly apparent in textual and visual works of early European studies of the Orient that positioned the East as irrational and backward in opposition to the rational and progressive West. Defining the East as a negative vision of itself, as its inferior, not only increased the West's sense of self, but also was a way of ordering the East and making it known to the West so that it could be dominated and controlled. The discourse of Orientalism therefore served as an ideological justification of early Western imperialism, as it formed a body of knowledge and ideas that rationalized social, cultural, political, and economic control of other territories.<br>Question: Early Western texts referencing the East describe the people as being what? | • Answerable: 1<br>• Ground truth: 'irrational and backward', 'them', 'as irrational and backward', 'irrational and backward', 'irrational and backward'<br>• QATnet: irrational and backward in opposition to the rational and progressive West<br>• QAnet: negative vision of itself |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: Notable alumni in the field of government and politics include the founder of modern community organizing Saul Alinsky, Obama campaign advisor and top political advisor to President Bill Clinton David Axelrod, Attorney General and federal judge Robert Bork, Attorney General Ramsey Clark, Prohibition agent Eliot Ness, Supreme Court Justice John Paul Stevens, Prime Minister of Canada William Lyon Mackenzie King, 11th Prime Minister of Poland Marek Belka, Governor of the Bank of Japan Masaaki Shirakawa, the first female African-American Senator Carol Moseley Braun, United States Senator from Vermont and 2016 Democratic Presidential Candidate Bernie Sanders, and former World Bank President Paul Wolfowitz.<br>Question: Who serves as Attorney General as well as top political advisor to the President? | • Answerable: 0<br>• Ground truth: 'Bill Clinton David Axelrod'<br>• QATnet: Bill Clinton David Axelrod<br>• QAnet: amsey Clark |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: After Malaysia's independence in 1957, the government instructed all schools to surrender their properties and be assimilated into the National School system. This caused an uproar among the Chinese and a compromise was achieved in that the schools would instead become "National Type" schools. Under such a system, the government is only in charge of the school curriculum and teaching personnel while the lands still belonged to the schools. While Chinese primary schools were allowed to retain Chinese as the medium of instruction, Chinese secondary schools are required to change into English-medium schools. Over 60 schools converted to become National Type schools.<br>Question: What language is used in Chinese primary schools in Malaysia? | • Answerable: 1<br>• Ground truth: 'Chinese', 'Chinese', 'Chinese'<br>• QATnet: Chinese<br>• QAnet: English |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: The definition of imperialism has not been finalized for centuries and was confusedly seen to represent the policies of major powers, or simply, general-purpose aggressiveness. Further on, some writers[who?] used the term imperialism, in slightly more discriminating fashion, to mean all kinds of domination or control by a group of people over another. To clear out this confusion about the definition of imperialism one could speak of "formal" and "informal" imperialism, the first meaning physical control or "full-fledged colonial rule" while the second implied less direct rule though still containing perceivable kinds of dominance. Informal rule is generally less costly than taking over territories formally. This is because, with informal rule, the control is spread more subtly through technological superiority, enforcing land officials into large debts that cannot be repaid, ownership of private industries thus expanding the controlled area, or having countries agree to uneven trade agreements forcefully.<br>Question: colonial rule, or physical occupation of a territory is an example of what kind of imperialism? | • Answerable: 1<br>• Ground truth: '"formal"', 'formal', 'formal', 'formal', 'formal'<br>• QATnet: formal<br>• QAnet: informal" imperialism |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: In particular, this norm gets smaller when a number is multiplied by p, in sharp contrast to the usual absolute value (also referred to as the infinite prime). While completing Q (roughly, filling the gaps) with respect to the absolute value yields the field of real numbers, completing with respect to the p-adic norm $\|-\|p$ yields the field of p-adic numbers. These are essentially all possible ways to complete Q, by Ostrowski's theorem. Certain arithmetic questions related to Q or more general global fields may be transferred back and forth to the completed (or local) fields. This local-global principle again underlines the importance of primes to number theory.<br>Question: Completing Q with respect to what will produce the field of real numbers? | • Answerable: 1<br>• Ground truth: 'the absolute value', 'the absolute value', 'absolute value', 'the absolute value'<br>• QATnet: absolute value yields<br>• QAnet: p-adic norm $\|-\|p$ |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: The centre-left Australian Labor Party (ALP), the centre-right Liberal Party of Australia, the rural-based National Party of Australia, and the environmentalist Australian Greens are Victoria's main political parties. Traditionally, Labor is strongest in Melbourne's working class western and northern suburbs, and the regional cities of Ballarat, Bendigo and Geelong. The Liberals' main support lies in Melbourne's more affluent eastern and outer suburbs, and some rural and regional centres. The Nationals are strongest in Victoria's North Western and Eastern rural regional areas. The Greens, who won their first lower house seats in 2014, are strongest in inner Melbourne.<br>Question: Which party is strongest in Victoria's northwestern and eastern regions? | • Answerable: 1<br>• Ground truth: 'National Party', 'National Party of Australia', 'Nationals'<br>• QATnet: The Nationals<br>• QAnet: Labor is strongest in Melbourne's working class western and northern suburbs, and the regional cities of Ballarat, Bendigo and Geelong. The Liberals' main support lies in Melbourne's more affluent eastern and outer suburbs, and some rural and regional centres. The Nationals |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: In contrast, during wake periods differentiated effector cells, such as cytotoxic natural killer cells and CTLs (cytotoxic T lymphocytes), peak in order to elicit an effective response against any intruding pathogens. As well during awake active times, anti-inflammatory molecules, such as cortisol and catecholamines, peak. There are two theories as to why the pro-inflammatory state is reserved for sleep time. First, inflammation would cause serious cognitive and physical impairments if it were to occur during wake times. Second, inflammation may occur during sleep times due to the presence of melatonin. Inflammation causes a great deal of oxidative stress and the presence of melatonin during sleep times could actively counteract free radical production during this time.<br>Question: Melatonin during sleep can actively counteract the production of what? | • Answerable: 1<br>• Ground truth: 'free radical production', 'free radical', 'free radical production'<br>• QATnet: free radical production<br>• QAnet: Inflammation |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: All Recognized Student Organizations, from the University of Chicago Scavenger Hunt to Model UN, in addition to academic teams, sports club, arts groups, and more are funded by The University of Chicago Student Government. Student Government is made up of graduate and undergraduate students elected to represent members from their respective academic unit. It is led by an Executive Committee, chaired by a President with the assistance of two Vice Presidents, one for Administration and the other for Student Life, elected together as a slate by the student body each spring. Its annual budget is greater than $2 million.<br>Question: Who leads the Student Government? | • Answerable: 1<br>• Ground truth: 'an Executive Committee', 'Executive Committee', 'an Executive Committee'<br>• QATnet: fan Executive Committee<br>• QAnet: Executive Committee |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: A deterministic Turing machine is the most basic Turing machine, which uses a fixed set of rules to determine its future actions. A probabilistic Turing machine is a deterministic Turing machine with an extra supply of random bits. The ability to make probabilistic decisions often helps algorithms solve problems more efficiently. Algorithms that use random bits are called randomized algorithms. A non-deterministic Turing machine is a deterministic Turing machine with an added feature of non-determinism, which allows a Turing machine to have multiple possible future actions from a given state. One way to view non-determinism is that the Turing machine branches into many possible computational paths at each step, and if it solves the problem in any of these branches, it is said to have solved the problem. Clearly, this model is not meant to be a physically realizable model, it is just a theoretically interesting abstract machine that gives rise to particularly interesting complexity classes. For examples, see non-deterministic algorithm.<br>Question: What fixed set of factors determine the actions of a deterministic Turing machine? | • Answerable: 1<br>• Ground truth: 'rules', 'rules', 'a fixed set of rules to determine its future actions'<br>• QATnet: aa fixed set of rules<br>• QAnet: Turing machine |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: With modern insights into quantum mechanics and technology that can accelerate particles close to the speed of light, particle physics has devised a Standard Model to describe forces between particles smaller than atoms. The Standard Model predicts that exchanged particles called gauge bosons are the fundamental means by which forces are emitted and absorbed. Only four main interactions are known: in order of decreasing strength, they are: strong, electromagnetic, weak, and gravitational.:2–10:79 High-energy particle physics observations made during the 1970s and 1980s confirmed that the weak and electromagnetic forces are expressions of a more fundamental electroweak interaction.<br>Question: What fixed set of factors determine the actions of a deterministic Turing machine? | • Answerable: 1<br>• Ground truth: 'Standard Model', 'Standard Model', 'Standard Model', 'Standard Model', 'a Standard Model', 'a Standard Model'<br>• QATnet: a Standard Model<br>• QAnet: particles smaller than atoms |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: Reserved matters are subjects that are outside the legislative competence of the Scotland Parliament. The Scottish Parliament is unable to legislate on such issues that are reserved to, and dealt with at, Westminster (and where Ministerial functions usually lie with UK Government ministers). These include abortion, broadcasting policy, civil service, common markets for UK goods and services, constitution, electricity, coal, oil, gas, nuclear energy, defence and national security, drug policy, employment, foreign policy and relations with Europe, most aspects of transport safety and regulation, National Lottery, protection of borders, social security and stability of UK's fiscal, economic and monetary system.<br>Question: Where are issues like abortion and drug policy legislated on? | • Answerable: 1<br>• Ground truth: 'Westminster', 'Westminster', 'Westminster'<br>• QATnet: Westminster<br>• QAnet: The Scottish Parliament |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: Since its invention in 1269, the 'Phags-pa script, a unified script for spelling Mongolian, Tibetan, and Chinese languages, was preserved in the court until the end of the dynasty. Most of the Emperors could not master written Chinese, but they could generally converse well in the language. The Mongol custom of long standing quda/marriage alliance with Mongol clans, the Onggirat, and the Ikeres, kept the imperial blood purely Mongol until the reign of Tugh Temur, whose mother was a Tangut concubine. The Mongol Emperors had built large palaces and pavilions, but some still continued to live as nomads at times. Nevertheless, a few other Yuan emperors actively sponsored cultural activities; an example is Tugh Temur (Emperor Wenzong), who wrote poetry, painted, read Chinese classical texts, and ordered the compilation of books.<br>Question: How poorly did the Mongol Emperors know spoken Chinese? | • Answerable: 1<br>• Ground truth: 'converse well in the language'<br>• QATnet: converse well in the language<br>• QAnet: large palaces and pavilions |

Continued on next page

**Table A.1 – continued from previous page**

| Context and Question | Answer |
|---|---|
| Context: Some theories of civil disobedience hold that civil disobedience is only justified against governmental entities. Brownlee argues that disobedience in opposition to the decisions of non-governmental agencies such as trade unions, banks, and private universities can be justified if it reflects "a larger challenge to the legal system that permits those decisions to be taken." The same principle, she argues, applies to breaches of law in protest against international organizations and foreign governments.<br>Question: Browlee also applies that civil disobedience is okay regarding? | • Answerable: 1<br>• Ground truth: 'international organizations and foreign governments', 'a larger challenge to the legal system that permits those decisions to be taken', 'international organizations and foreign governments', 'breaches of law in protest against international organizations and foreign governments', 'opposition to the decisions of non-governmental agencies such as trade unions, banks, and private universities'<br>• QATnet: breaches of law in protest against international organizations and foreign governments<br>• QAnet: governmental entities |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: In the meantime, on August 1, 1774, an experiment conducted by the British clergyman Joseph Priestley focused sunlight on mercuric oxide (HgO) inside a glass tube, which liberated a gas he named "dephlogisticated air". He noted that candles burned brighter in the gas and that a mouse was more active and lived longer while breathing it. After breathing the gas himself, he wrote: "The feeling of it to my lungs was not sensibly different from that of common air, but I fancied that my breast felt peculiarly light and easy for some time afterwards." Priestley published his findings in 1775 in a paper titled "An Account of Further Discoveries in Air" which was included in the second volume of his book titled Experiments and Observations on Different Kinds of Air. Because he published his findings first, Priestley is usually given priority in the discovery.<br>Question: What did Priestley name the air he created? | • Answerable: 0<br>• Ground truth: dephlogisticated air<br>• QATnet: dephlogisticated air<br>• QAnet: mercuric oxide (HgO) inside a glass tube |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
| --- | --- |
| Context: The historical measure of a steam engine's energy efficiency was its “duty.” The concept of duty was first introduced by Watt in order to illustrate how much more efficient his engines were over the earlier Newcomen designs. Duty is the number of foot-pounds of work delivered by burning one bushel (94 pounds) of coal. The best examples of Newcomen designs had a duty of about 7 million, but most were closer to 5 million. Watt's original low-pressure designs were able to deliver duty as high as 25 million, but averaged about 17. This was a three-fold improvement over the average Newcomen design. Early Watt engines equipped with high-pressure steam improved this to 65 million.<br>Question: What is the weight of a bushel of coal in pounds? | • Answerable: 0<br>• Ground truth: ’94’, ’94 pounds’, ’94 pounds’<br>• QATnet: 94 pounds<br>• QAnet: 94 |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: The Scotland Act 1998, which was passed by the Parliament of the United Kingdom and given royal assent by Queen Elizabeth II on 19 November 1998, governs the functions and role of the Scottish Parliament and delimits its legislative competence. The Scotland Act 2012 extends the devolved competencies. For the purposes of parliamentary sovereignty, the Parliament of the United Kingdom at Westminster continues to constitute the supreme legislature of Scotland. However, under the terms of the Scotland Act, Westminster agreed to devolve some of its responsibilities over Scottish domestic policy to the Scottish Parliament. Such "devolved matters" include education, health, agriculture and justice. The Scotland Act enabled the Scottish Parliament to pass primary legislation on these issues. A degree of domestic authority, and all foreign policy, remain with the UK Parliament in Westminster. The Scottish Parliament has the power to pass laws and has limited tax-varying capability. Another of the roles of the Parliament is to hold the Scottish Government to account.<br>Question: Who has the role of holding the Scottish Government to account? | • Answerable: 1<br>• Ground truth: 'Scottish Parliament', 'Parliament', 'the Parliament'<br>• QATnet: The Scottish Parliament<br>• QAnet: Scottish Parliament |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: Following the Peterloo massacre of 1819, poet Percy Shelley wrote the political poem The Mask of Anarchy later that year, that begins with the images of what he thought to be the unjust forms of authority of his time—and then imagines the stirrings of a new form of social action. It is perhaps the first modern[vague] statement of the principle of nonviolent protest. A version was taken up by the author Henry David Thoreau in his essay Civil Disobedience, and later by Gandhi in his doctrine of Satyagraha. Gandhi's Satyagraha was partially influenced and inspired by Shelley's nonviolence in protest and political action. In particular, it is known that Gandhi would often quote Shelley's Masque of Anarchy to vast audiences during the campaign for a free India.<br>Question: Inspired by Shelley what was the name of Gandhi's doctrine? | • Answerable: 1<br>• Ground truth: 'Satyagraha', 'Satyagraha', 'Satyagraha', 'Satyagraha', 'Satyagraha'<br>• QATnet: Satyagraha<br>• QAnet: Masque of Anarchy |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: Tymnet was an international data communications network headquartered in San Jose, CA that utilized virtual call packet switched technology and used X.25, SNA/SDLC, BSC and ASCII interfaces to connect host computers (servers)at thousands of large companies, educational institutions, and government agencies. Users typically connected via dial-up connections or dedicated async connections. The business consisted of a large public network that supported dial-up users and a private network business that allowed government agencies and large companies (mostly banks and airlines) to build their own dedicated networks. The private networks were often connected via gateways to the public network to reach locations not on the private network. Tymnet was also connected to dozens of other public networks in the U.S. and internationally via X.25/X.75 gateways. (Interesting note: Tymnet was not named after Mr. Tyme. Another employee suggested the name.)<br>Question: What was Tymnet? | • Answerable: 1<br>• Ground truth: 'an international data communications network headquartered in San Jose, CA', 'an international data communications network', 'international data communications network'<br>• QATnet: an international data communications network<br>• QAnet: international data communications network |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: The Rhine emerges from Lake Constance, flows generally westward, as the Hochrhein, passes the Rhine Falls, and is joined by its major tributary, the river Aare. The Aare more than doubles the Rhine's water discharge, to an average of nearly 1,000 m3/s (35,000 cu ft/s), and provides more than a fifth of the discharge at the Dutch border. The Aare also contains the waters from the 4,274 m (14,022 ft) summit of Finsteraarhorn, the highest point of the Rhine basin. The Rhine roughly forms the German-Swiss border from Lake Constance with the exceptions of the canton of Schaffhausen and parts of the cantons of Zürich and Basel-Stadt, until it turns north at the so-called Rhine knee at Basel, leaving Switzerland.<br>Question: What is the major tributary of the Rhine? | • Answerable: 1<br>• Ground truth: 'river Aare', 'Aare', 'river Aare'<br>• QATnet: the river Aare<br>• QAnet: Aare |
| | Continued on next page |

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: The fundamental theorem of arithmetic continues to hold in unique factorization domains. An example of such a domain is the Gaussian integers Z[i], that is, the set of complex numbers of the form $a+bi$ where i denotes the imaginary unit and a and b are arbitrary integers. Its prime elements are known as Gaussian primes. Not every prime (in Z) is a Gaussian prime: in the bigger ring Z[i], 2 factors into the product of the two Gaussian primes $(1+i)$ and $(1-i)$. Rational primes (i.e. prime elements in Z) of the form $4k+3$ are Gaussian primes, whereas rational primes of the form $4k+1$ are not.<br>Question: What theorem remains valid in unique Gaussian primes? | • Answerable: 0<br>• Ground truth: 'The fundamental theorem of arithmetic'<br>• QATnet: fundamental theorem of arithmetic<br>• QAnet: arithmetic |
| | Continued on next page |

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: The University of Chicago Library system encompasses six libraries that contain a total of 9.8 million volumes, the 11th most among library systems in the United States. The university's main library is the Regenstein Library, which contains one of the largest collections of print volumes in the United States. The Joe and Rika Mansueto Library, built in 2011, houses a large study space and an automatic book storage and retrieval system. The John Crerar Library contains more than 1.3 million volumes in the biological, medical and physical sciences and collections in general science and the philosophy and history of science, medicine, and technology. The university also operates a number of special libraries, including the D'Angelo Law Library, the Social Service Administration Library, and the Eckhart Library for mathematics and computer science, which closed temporarily for renovation on July 8, 2013. Harper Memorial Library no longer contains any volumes; however it is, in addition to the Regenstein Library, a 24-hour study space on campus.<br>Question: Which University's library system has over 10 millionvolumes? | • Answerable: 0<br>• Ground truth: 'University of Chicago'<br>• QATnet: The University of Chicago Library<br>• QAnet: University of Chicago Library |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: During the mid-Eocene, it is believed that the drainage basin of the Amazon was split along the middle of the continent by the Purus Arch. Water on the eastern side flowed toward the Atlantic, while to the west water flowed toward the Pacific across the Amazonas Basin. As the Andes Mountains rose, however, a large basin was created that enclosed a lake; now known as the Solimões Basin. Within the last 5–10 million years, this accumulating water broke through the Purus Arch, joining the easterly flow toward the Atlantic.<br>Question: Where did water to the east of the Amazon drainage basin flow towards? | • Answerable: 1<br>• Ground truth: ’the Atlantic’, ’the Atlantic’, ’Atlantic’<br>• QATnet: the Atlantic<br>• QAnet: the Pacific across the Amazonas Basin |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: Telenet was the first FCC-licensed public data network in the United States. It was founded by former ARPA IPTO director Larry Roberts as a means of making ARPANET technology public. He had tried to interest AT&T in buying the technology, but the monopoly's reaction was that this was incompatible with their future. Bolt, Beranack and Newman (BBN) provided the financing. It initially used ARPANET technology but changed the host interface to X.25 and the terminal interface to X.29. Telenet designed these protocols and helped standardize them in the CCITT. Telenet was incorporated in 1973 and started operations in 1975. It went public in 1979 and was then sold to GTE.<br>Question: Telnet was sold to? | • Answerable: 0<br>• Ground truth: 'Telenet was incorporated in 1973 and started operations in 1975. It went public in 1979 and was then sold to GTE', 'GTE', 'GTE'<br>• QATnet: GTE<br>• QAnet: X.29 |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: Forces act in a particular direction and have sizes dependent upon how strong the push or pull is. Because of these characteristics, forces are classified as “vector quantities”. This means that forces follow a different set of mathematical rules than physical quantities that do not have direction (denoted scalar quantities). For example, when determining what happens when two forces act on the same object, it is necessary to know both the magnitude and the direction of both forces to calculate the result. If both of these pieces of information are not known for each force, the situation is ambiguous. For example, if you know that two people are pulling on the same rope with known magnitudes of force but you do not know which direction either person is pulling, it is impossible to determine what the acceleration of the rope will be. The two people could be pulling against each other as in tug of war or the two people could be pulling in the same direction. In this simple one-dimensional example, without knowing the direction of the forces it is impossible to decide whether the net force is the result of adding the two force magnitudes or subtracting one from the other. Associating forces with vectors avoids such problems. Question: How do you avoid problems when determining forces involved on an object from two or more sources? | • Answerable: 1<br>• Ground truth: ’Associating forces with vectors’, ’Associating forces with vectors’, ’Associating forces with vectors’, ’Associating forces with vectors’, ’know both the magnitude and the direction of both forces to calculate the result’<br>• QATnet: Associating forces with vectors<br>• QAnet: it is necessary to know both the magnitude and the direction of both forces to calculate the result |
| | Continued on next page |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: A regulation of the Rhine was called for, with an upper canal near Diepoldsau and a lower canal at Fußach, in order to counteract the constant flooding and strong sedimentation in the western Rhine Delta. The Dornbirner Ach had to be diverted, too, and it now flows parallel to the canalized Rhine into the lake. Its water has a darker color than the Rhine; the latter's lighter suspended load comes from higher up the mountains. It is expected that the continuous input of sediment into the lake will silt up the lake. This has already happened to the former Lake Tuggenersee.<br>Question: What is expected with the continuous input of sediment into the Dornbirner Ach? | • Answerable: 1<br>• Ground truth: 'silt', 'silt up the lake', 'the continuous input of sediment into the lake will silt up the lake', 'silt up the lake'<br>• QATnet: silt up the lake<br>• QAnet: lighter suspended load |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
| --- | --- |
| Context: Private schooling in the United States has been debated by educators, lawmakers and parents, since the beginnings of compulsory education in Massachusetts in 1852. The Supreme Court precedent appears to favor educational choice, so long as states may set standards for educational accomplishment. Some of the most relevant Supreme Court case law on this is as follows: Runyon v. McCrary, 427 U.S. 160 (1976); Wisconsin v. Yoder, 406 U.S. 205 (1972); Pierce v. Society of Sisters, 268 U.S. 510 (1925); Meyer v. Nebraska, 262 U.S. 390 (1923).<br>Question: Who was the opposing party in the Runyon case? | • Answerable: 1<br>• Ground truth: 'McCrary', 'McCrary', 'McCrary'<br>• QATnet: Runyon v. McCrary<br>• QAnet: Pierce v. Society of Sisters |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: Trevithick continued his own experiments using a trio of locomotives, concluding with the Catch Me Who Can in 1808. Only four years later, the successful twin-cylinder locomotive Salamanca by Matthew Murray was used by the edge railed rack and pinion Middleton Railway. In 1825 George Stephenson built the Locomotion for the Stockton and Darlington Railway. This was the first public steam railway in the world and then in 1829, he built The Rocket which was entered in and won the Rainhill Trials. The Liverpool and Manchester Railway opened in 1830 making exclusive use of steam power for both passenger and freight trains.<br>Question: What was the name of the locomotive that debuted in 1808? | • Answerable: 1<br>• Ground truth: 'Catch Me Who Can', 'Catch Me Who Can', 'Catch Me Who Can'<br>• QATnet: Catch Me Who Can<br>• QAnet: Salamanca |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: In 1979, the Soviet Union deployed its 40th Army into Afghanistan, attempting to suppress an Islamic rebellion against an allied Marxist regime in the Afghan Civil War. The conflict, pitting indigenous impoverished Muslims (mujahideen) against an anti-religious superpower, galvanized thousands of Muslims around the world to send aid and sometimes to go themselves to fight for their faith. Leading this pan-Islamic effort was Palestinian sheikh Abdullah Yusuf Azzam. While the military effectiveness of these “Afghan Arabs” was marginal, an estimated 16,000 to 35,000 Muslim volunteers came from around the world came to fight in Afghanistan.<br>Question: How effective was the military use of the ”Afghan Arabs”? | • Answerable: 1<br>• Ground truth: ’marginal’, ’marginal’, ’marginal’<br>• QATnet: marginal<br>• QAnet: 16,000 to 35,000 |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: Stadtholder William III of Orange, who later became King of England, emerged as the strongest opponent of king Louis XIV after the French attacked the Dutch Republic in 1672. William formed the League of Augsburg as a coalition to oppose Louis and the French state. Consequently, many Huguenots considered the wealthy and Calvinist Dutch Republic, which led the opposition to Louis XIV, as the most attractive country for exile after the revocation of the Edict of Nantes. They also found many French-speaking Calvinist churches there.<br>Question: William would eventually gain what throne? | • Answerable: 1<br>• Ground truth: 'King of England', 'King of England', 'King of England'<br>• QATnet: King of England<br>• QAnet: Louis XIV |
| | Continued on next page |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
| --- | --- |
| Context: One of the first known experiments on the relationship between combustion and air was conducted by the 2nd century BCE Greek writer on mechanics, Philo of Byzantium. In his work Pneumatica, Philo observed that inverting a vessel over a burning candle and surrounding the vessel's neck with water resulted in some water rising into the neck. Philo incorrectly surmised that parts of the air in the vessel were converted into the classical element fire and thus were able to escape through pores in the glass. Many centuries later Leonardo da Vinci built on Philo's work by observing that a portion of air is consumed during combustion and respiration.<br>Question: What was the title of Philo's work? | • Answerable: 1<br>• Ground truth: 'Pneumatica', 'Pneumatica', 'Pneumatica', 'Pneumatica', 'Pneumatica'<br>• QATnet: Pneumatica<br>• QAnet: Leonardo da Vinci |

Continued on next page

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: The English name "Normans" comes from the French words Normans/Normanz, plural of Normant, modern French normand, which is itself borrowed from Old Low Franconian Nortmann "Northman" or directly from Old Norse Noromaor, Latinized variously as Nortmannus, Normannus, or Nordmannus (recorded in Medieval Latin, 9th century) to mean "Norseman, Viking". Question: What is the original meaning of the word Norman? | • Answerable: 1 <br> • Ground truth: 'Viking', 'Norseman, Viking', 'Norseman, Viking' <br> • QATnet: "Norseman, Viking <br> • QAnet: Old Low Franconian Nortmann "Northman |
| | Continued on next page |

**Table A.1 – continued from previous page**

| **Context and Question** | **Answer** |
|---|---|
| Context: In July 2013, the English High Court of Justice found that Microsoft's use of the term "SkyDrive" infringed on Sky's right to the "Sky" trademark. On 31 July 2013, BSkyB and Microsoft announced their settlement, in which Microsoft will not appeal the ruling, and will rename its SkyDrive cloud storage service after an unspecified "reasonable period of time to allow for an orderly transition to a new brand," plus "financial and other terms, the details of which are confidential." On 27 January 2014, Microsoft announced "that SkyDrive will soon become OneDrive" and "SkyDrive Pro" becomes "OneDrive for Business." Question: What did Microsoft announce that it would rename OneDrive for Business to? | • Answerable: 0<br>• Ground truth: 'SkyDrive Pro'<br>• QATnet: SkyDrive Pro<br>• QAnet: their settlement |