Assignment Guidance and Front Sheet

This front sheet for assignments is designed to contain the brief, the submission instructions, and the actual student submission for any WMG assignment. As a result the sheet is completed by several people over time, and is therefore split up into sections explaining who completes what information and when. Yellow highlighted text indicates examples or further explanation of what is requested, and the highlight and instructions should be removed as you populate ‘your’ section.

This sheet is only to be used for components of assessment worth more than 3 CATS (e.g. for a 15 credit module, weighted more than 20%; or for a 10 credit module, weighted more than 30%).

**To be completed by the student(s) prior to final submission:**

Your actual submission should be written at the end of this cover sheet file, or attached with the cover sheet at the front if drafted in a separate file, program or application.

|  |  |
| --- | --- |
| **Student ID or IDs for group work** | **2222673** |

**To be completed (highlighted parts only) by the programme administration after approval and prior to issuing of the assessment; to be consulted by the student(s) so that you know how and when to submit:**

|  |  |
| --- | --- |
| **Date set** | 28/04/2023 |
| **Submission date (excluding extensions)** | 30/05/2023 by 12:00pm (UK Time) |
| **Submission guidance** | To be submitted electronically via Tabula |
| **Late submission policy** | If work is submitted late, penalties will be applied at the rate of **5 marks per University working day** after the due date, up to a **maximum of 10 working days** late. After this period the mark for the work will be reduced to 0 (which is the maximum penalty). “Late” means **after the submission deadline time as well as the date** – work submitted after the given time even on the same day is counted as 1 day late.  For **Postgraduate** students only, who started their **current course before 1 August 2019**, the daily penalty is **3 marks** rather than 5. |
| **Resubmission policy** | If you fail this assignment or module, please be aware that the University allows students to remedy such failure (within certain limits). Decisions to authorise such resubmissions are made by Exam Boards. Normally these will be issued at specific times of the year, depending on your programme of study. More information can be found from your programme office if you are concerned.  **If this is already a resubmission attempt, this means you will not be eligible for an additional attempt. The University allows as standard a maximum of two attempts on any assessment (i.e. only one resubmission). Students can only have a third attempt under exceptional circumstances via a Mitigating Circumstances Panel decision.** |

**To be completed by the module owner/tutor prior to approval and issuing of the assessment; to be consulted by the student(s) so that you understand the assignment brief, its context within the module, and any specific criteria and advice from the tutor:**

|  |  |
| --- | --- |
| **Module title & code** | WM9B7-15 Artificial Intelligence & Deep Learning |
| **Module owner** | Michael Mortenson |
| **Module tutor** | Michael Mortenson |
| **Assessment type** | PMA |
| **Weighting of mark** | 80% |

|  |
| --- |
| **Assessment brief** |
| You have been appointed to write a chapter for a book on deep learning methods. You can choose which deep learning algorithm you wish to focus on within the field of **natural language processing**. Whichever approach you choose, your chapter should include:   * An explanation of how the algorithm works (intuitively and/or mathematically) * A discussion of the practical/applied scenarios where your algorithm is most applicable * The relative merits of your algorithm in comparison to alternative methods * An implementation/code tutorial of your algorithm in TensorFlow or similar library.   References from both **academic and commercial sources** should be included. |

|  |  |
| --- | --- |
| **Word count** | Suggested word count = 3,200 |
| **Module learning outcomes (numbered)** | Interpret and evaluate various use-cases and the applicability of artificial intelligence and deep learning.  Adopt best practices for data processing and engineering for artificial intelligence and deep learning models.  Implement, interpret and critique current, professional standard learning models.  Automate deployment-ready deep learning pipelines and algorithms.  Evaluate and interpret the results of deep learning models and tune them to optimise performance. |
| **Learning outcomes assessed in this assessment (numbered)** | All 5 |
| **Marking guidelines** | *Above expectation*  A thorough and detailed explanation of their chosen algorithm that demonstrates a complete (or near-complete) understanding of the approach (LO1 and LO3)  Excellent understanding and analysis of where the chosen algotihm could be applied and its relative merits in comparison to related approaches (LO1)  A well explained and professional-standard implementation of the model. Appropriate use of automation methods and tuning/optimisation methods (LO2, LO4 and LO5)  *Expectation*  A sensible explanation of their chosen algorithm that may show some areas of misunderstanding but is largely correct (LO1 and LO3)  Some thought given to the use-cases where the algorithm could be applied. Comparisons are made with related methods and show a broader understanding of the space (LO1)  Implementation is largely correct and some relevant discussion and documentation of the methods to show understanding. Some evidence of optimisation and consideration of best practice (LO2, LO4 and LO5) |
| **Academic guidance resources** | *Contact the module tutor for guidance* |

**The following is pre-populated for PGT assignments only:**

|  |
| --- |
| **Writing your Post-Module Assignment (PMA): specific additional advice for WMG’s Postgraduate Taught Students** |
| As a postgraduate level student in WMG you may have some concerns about your ability to write at the high standard required. This short guide is intended to provide general guidance and advice. It is important that if you have any questions you discuss them with your module tutor. Remember, in writing your PMA you need to meet the expectations of the reader and university. |
| **A good PMA generally requires you to answer the question and to include**…   1. A title, with your student number, module, lecturer’s name and any other documentation required by the university. 2. A table of contents. 3. An introduction which acts as a ‘map’ to the rest of the document, describing the aim or purpose of the work and explaining how this aim is achieved. 4. Evidence of an appropriate level of background reading of relevant texts. 5. Evidence of systematic and clear thinking, indicative of good planning and organisation. 6. Writing which makes sense, is clearly and carefully presented (proof-read and grammar checked). 7. A critical style of writing which compares and contrasts the main theories, concepts and arguments with conclusions that are based in evidence presented. 8. A logical and well-defined structure with headings and subheadings. 9. Clearly labelled and well-presented diagrams and other graphics that are discussed in the text. 10. Adherence to usual academic standards including length and a timely submission. 11. A reference section in which every source that is cited in the text is listed and please ensure that you underpin the discussion throughout with relevant academic material to support the content, using the Harvard approach. |
| **Where to get help:**   1. **Talk to your module tutor if you don’t understand the question or are unsure as to exactly what is required.** 2. Study, Professional and Analytical Skills (SPA) Moodle site – we have a lot of resources on this website with workbooks, links and other helpful tools. <https://moodle.warwick.ac.uk/> 3. Numerous online courses provided by the University library to help in academic referencing, writing, avoiding plagiarism and a number of other useful resources. <https://warwick.ac.uk/services/library/students/your-library-online/> 4. Wellbeing support services <https://warwick.ac.uk/services/wss> |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PMA marking criteria** | | | | |
| **Intended LO or Evaluative Criteria** | **70+ Distinction** | **60-69**  **Merit** | **50-59 Pass** | **0-49**  **Fail** |
| Overarching descriptor for the PMA | An outstanding or excellent piece of work, showing mastery or complete mastery of the subject, with a highly developed and mature ability to analyse, synthesise and apply concepts, models, and techniques. All requirements of the set work are covered, and work is free from all but very minor or no errors. There is good critical reflection and the ability to tackle questions and issues not previously encountered. Ideas are explained very clearly and in a highly structured manner. | A strong piece of work, showing a sound grasp of the subject and a skilful attempt at analysis, synthesis and application of concepts, models, and techniques. Most requirements of the set work are covered, but there may be a few gaps leading to some errors. There is some critical reflection, and a reasonable attempt is made to tackle questions and issues not previously encountered. Ideas are explained clearly and in a well organised manner, with some minor exceptions. | A satisfactory piece of work, showing a grasp of major areas of the subject, but probably with areas of ignorance. Analysis, synthesis and application of concepts, models and techniques is mechanical, with a heavy reliance on course materials. The requirements of the set work are covered but with significant gaps. Little or no critical reflection and limited ability to tackle questions or issues not previously encountered. Ideas are explained adequately but with some confusion and lack of organisation. | An unsatisfactory piece of work. There is a weak attempt at analysis, synthesis and application of concepts, models, and techniques. Only some of the requirements of the set work are covered. Inability to reflect critically and difficulty in beginning to address questions and issues not previously encountered. Ideas are poorly explained and organised. |
| LO1 Compare and contrast different supply chain management approaches for the effective supply of different products and services in a range of global and domestic industries.  study . | Shown by selection, analysis, synthesis and application of appropriate concepts, models, or techniques to compare and contrast the chosen company’s approach drawing on academic materials from the module’s reading list, lecture materials and industrial examples given during the module lectures and seminars and research on the chosen company. Maybe no or a few small gaps. | Shown by selection, analysis, synthesis and application of concepts, models, or techniques to compare and contrast the chosen company’s approach drawing on some of the academic materials from the module’s reading list, lecture materials and industrial examples given during the module lectures and seminars and research on the chosen company. Some gaps. | Shown by selection, analysis, synthesis and application of concepts, models, or techniques to compare and contrast the chosen company’s approach drawing on some of the academic materials from the module’s reading list, lecture materials and industrial examples given during the module lectures and seminars and research on the chosen company. Some significant gaps. | Poor or selection of relevant concepts, models, or techniques. Weak or no application or theory and no or weak comparison or contrasting of the chosen company’s approach to other approaches. Very little evidence drawn from the academic materials from the module’s reading list, lecture materials and industrial examples given during the module lectures and seminars or evidence of research on the chosen company. |
| LO2 Comprehensively understand the role of operations management in achieving supply chain integration. | Shown by matching the appropriate selection of aspect(s) of operations management to evaluate how the selected technology can be implemented to improve operations management to achieve supply chain integration. Comprehensive understanding demonstrated by drawing on relevant academic materials from the module’s reading list, lecture materials and industrial examples given during the module lectures and seminars and research on the chosen company. Maybe no or a few small gaps. | Shown by matching the appropriate selection of aspect(s) of operations management to evaluate how the selected technology can be implemented to improve operations management to achieve supply chain integration. Sound understanding demonstrated by drawing on relevant academic materials from the module’s reading list, lecture materials and industrial examples given during the module lectures and seminars and research on the chosen company. Some gaps and/or lack of depth of analysis e.g. by tackling too many aspects of operations management. | Shown by matching the appropriate selection of aspect(s) of operations management to evaluate how the selected technology can be implemented to improve operations management to achieve supply chain integration. Some understanding demonstrated by drawing on some relevant academic materials from the module’s reading list, lecture materials and industrial examples given during the module lectures and seminars and research on the chosen company. Some significant gaps and/or lack of depth of analysis e.g. by tackling too many aspects of operations management. | Poor matching of the aspect(s) of operations management to evaluate how the selected technology can be implemented to improve operations management to achieve supply chain integration. Very limited comprehension demonstrated with very evidence of knowledge of any relevant academic materials from the module’s reading list, lecture materials and industrial examples given during the module lectures and seminars presented or and research on the chosen company. |
| LO3 - Critically identify and assess supply chain integration requirements for the effective relationships with, and the coordination of, customers and suppliers. | Shown by the selection of appropriate aspect(s) of supply chain integration to evaluate with analysis and examples to show how the selected technology can be implemented to improve effective relationships with customers and suppliers for the chosen company. Comprehensive evidence drawn from relevant academic materials from the module’s reading list, lecture materials and industrial examples given during the module lectures and seminar exercises and research on the chosen company. A few or no, gaps. | Shown by the selection of appropriate aspect(s) of supply chain integration to evaluate with analysis and examples to show how the selected technology can be implemented to improve effective relationships with customers and suppliers for the chosen company. Evidence drawn from relevant academic materials from the module’s reading list, lecture materials and industrial examples given during the module lectures and seminar exercises and research on the chosen company. Some gaps. | Shown by the selection of appropriate aspect(s) of supply chain integration to evaluate with analysis and examples to show how the selected technology can be implemented to improve effective relationships with customers and suppliers for the chosen company. Some evidence drawn from relevant academic materials from the module’s reading list, lecture materials and industrial examples given during the module lectures and seminar exercises and research on the chosen company. Some significant gaps. | Poor selection of aspect(s) of supply chain integration with limited linkage or evaluation of the selected technology could improve effective relationships with customers and suppliers for the chosen company. Very little evidence drawn from the academic materials from the module’s reading list, lecture materials and industrial examples given during the module lectures and seminars or evidence of research on the chosen company. |
| LO4 - Critically evaluate how appropriate digital technologies can be applied to improve supply chain management and integration. | Selection of the technology & examples of their industrial application in supply chain management & integration are clearly relevant to this company and show mastery of the subject. Evaluation of the technology supported by qualitative and quantitative evidence drawing on the in-module work and good further research on current industrial practices. Appraised potential risks, disadvantages and challenges during the implementation and use phase of the technology. A few small gaps. | Selection of the technology &  examples of their industrial application in supply chain management & integration are relevant to this company and show a sound grasp mastery of the subject. Evaluation of the technology supported by some qualitative and quantitative evidence drawing on the in-module work and some further research on current industrial practices. Some appraisals of potential risks, disadvantages and challenges during the implementation and use phase of the technology. Some gaps. | Selection of the technology & examples of their industrial application in supply chain management & integration may have gaps or their relevance to this company may be unclear/weak showing a grasp of the subject but some areas of ignorance. Evaluation of the technology supported by some qualitative and quantitative evidence drawing largely on the in-module work, with little evidence of further research.  Acceptable appraisals of potential risks, disadvantages and challenges during the implementation and use phase of the technology. Some significant gaps. | Selection of the technology & examples of their industrial application in supply chain management & integration may have gaps or their relevance to this company may be very unclear/weak showing a limited grasp of the subject but some areas of ignorance. Very limited evaluation of the technology supported by little qualitative or quantitative evidence. May have tackled more than 1 technologies, leading to insufficient depth of analysis. Limited evidence of use of the in-module work or further research. Little or no appraisal of potential risks, disadvantages and challenges during the implementation and use phase of the technology. Significant gaps. |
| Academic practice & writing | Showing exceptional written communication skills. Well-structured with excellent use of headings and sub-sections that show the development of a logical argument.  Figures and tables where used are appropriately titled and referenced in the text. All pages are correctly and clearly numbered.  Excellent number and appropriate mix of academic, grey, and recently published references. Majority of statements appropriately supported with evidence via in-text references to short quotes, facts, or numerical evidence. All reference details presented in an appropriately sequenced list. | Showing effective written communication skills. Reasonable overall structure and use of headings with some evidence of development of logical argument.  Figures and tables where used are appropriately titled and referenced in the text, all pages are correctly and clearly numbered.  Good number and mix of academic, grey, and recently published references but a few gaps. Most statements appropriately supported with evidence via in-text references to short quotes, facts, or numerical evidence. Most reference details presented in an appropriately sequenced list. | Showing competent written communication skills. Fair overall structure and use of headings with some evidence of development of logical argument.  Figures and tables where used are appropriately titled and referenced in the text, all pages are correctly and clearly numbered, and all references are properly cited and listed however there may be some errors. Too few references or a poor mix of academic, grey, and recently published references but gaps. Some statements appropriately supported with evidence via in-text references to short quotes, facts, or numerical evidence. Some gaps in the reference details presented and may be errors in the sequence of the reference list. | Showing less than optimal written communication skills. Weak overall structure and use of headings with little evidence of development of logical argument.  Figures and tables where used may be poorly titled and referenced in the text, pages may not be correctly and clearly numbered, and some references are not cited or listed. There may be some major errors.  Significant gaps in the number and choice of references. Many statements not supported by evidence via in-text references or supported by non-credible reference sources. Many gaps in the reference details. There may be a poorly sequenced reference list. |

**Bidirectional Encoder Representations from Transformers (BERT)**

**Table of Contents**

[**1.** **Introduction** 12](#_Toc136916247)

[**2.** **Relevance of BERT** 12](#_Toc136916248)

[**3.** **BERT Architecture and Working** 12](#_Toc136916249)

[**3.1.** **Transformers** 13](#_Toc136916250)

[**3.1.1.** **Encoder** 15](#_Toc136916251)

[**3.1.1.1.** **Input Embedding** 15](#_Toc136916252)

[**3.1.1.2.** **Positional Encoding(PE)** 15](#_Toc136916253)

[**3.1.1.3.** **Multi-Head Attention(MHA)** 18](#_Toc136916254)

[**3.1.1.4.** **Feed-Forward Network(FFN)** 21](#_Toc136916255)

[**3.1.2.** **Decoder** 21](#_Toc136916256)

[**3.1.2.1.** **Masked Multi-Head Attention(MMHA)** 22](#_Toc136916257)

[**3.1.3.** **Linear and Softmax Layer** 22](#_Toc136916258)

[**3.2.** **BERT Architecture** 23](#_Toc136916259)

[**3.2.1.** **Differences between BERT and Transformers** 23](#_Toc136916260)

[**3.2.2.** **BERT Framework Steps** 24](#_Toc136916261)

[**3.2.2.1.** **BERT Pretraining** 24](#_Toc136916262)

[**3.2.2.1.1.** **Inputs and Outputs** 24](#_Toc136916263)

[**3.2.2.1.2.** **Masked Language Modelling(MLM)** 26](#_Toc136916264)

[**3.2.2.1.3.** **Next Sentence Prediction(NSP)** 26](#_Toc136916265)

[**3.2.2.2.** **BERT Fine-tuning** 27](#_Toc136916266)

[**3.2.2.2.1.** **BERT Fine-tuning for Question Answering Task** 28](#_Toc136916267)

[**3.2.2.2.2.** **Recommended Hyper Parameter Values** 29](#_Toc136916268)

[**4.** **BERT Real-World Applications** 30](#_Toc136916269)

[**5.** **Benefits and Limitations of BERT** 30](#_Toc136916270)

[**6.** **BERT Model for Spam Message Classification** 31](#_Toc136916271)

[**6.1.** **Step 1: Import Libraries and Dataset** 31](#_Toc136916272)

[**6.2.** **Step 2: Divide the data for testing and training** 32](#_Toc136916273)

[**6.3.** **Step 3: Import BERT and Load Tokenizer** 33](#_Toc136916274)

[**6.4.** **Step 4: Tokenize and Encode the Input Sequence** 33](#_Toc136916275)

[**6.5.** **Step 5: Convert Lists to Tensors** 34](#_Toc136916276)

[**6.6.** **Step 6: Hyper Parameter Optimisation** 34](#_Toc136916277)

[**6.7.** **Step 7: Initialise Data Loader** 35](#_Toc136916278)

[**6.8.** **Step 8: BERT Model for Pretraining** 35](#_Toc136916279)

[**6.9.** **Step 9: BERT Fine-tuning** 38](#_Toc136916280)

[**6.10.** **Step 10: Model Prediction** 43](#_Toc136916281)

[**6.11.** **Step 11: Model Evaluation** 43](#_Toc136916282)

[**7.** **Conclusion** 44](#_Toc136916283)

[**8.** **References** 45](#_Toc136916284)

**Table of Figures**

[Figure 1.Limitations of RNN and LSTM Architectures(Giacaglia,2019) 13](#_Toc136916212)

[Figure 2.High-level Architecture of Transformers(Vaswani et al.,2017) 14](#_Toc136916213)

[Figure 3.Encoder-Decoder Architecture of Transformers(Alammar,2018b) 14](#_Toc136916214)

[Figure 4.Tokenisation and Vectorisation(Jain,2022) 15](#_Toc136916215)

[Figure 5.Positional Encoding Matrix for 'I am an Engineer'(Saeed,2023) 16](#_Toc136916216)

[Figure 6.Positional Encoding Equations(Jain,2022;Saeed,2023) 16](#_Toc136916217)

[Figure 7.Positional Encoding Matrix for the sequence 'I am an Engineer'(Jain,2022;Saeed,2023) 17](#_Toc136916218)

[Figure 8.Word Embedding Space(Padmanabhan,2022) 17](#_Toc136916219)

[Figure 9.Example of Self-Attention Mechanism-Long Dependencies in a Sequence(Kortschak,2022) 18](#_Toc136916220)

[Figure 10.Example of Self-Attention Mechanism for the word ‘him’(Kortschak,2022) 18](#_Toc136916221)

[Figure 11.Self-Attention Representational Vectors(Kortschak,2022) 18](#_Toc136916222)

[Figure 12.Steps for Self-Attention(Jain,2022;Karim,2023; Saeed,2022) 19](#_Toc136916223)

[Figure 13.Example of Multi-Head Attention Mechanism(Karim,2023) 19](#_Toc136916224)

[Figure 14.Weights for Key,Value and Query vectors(Karim,2023) 20](#_Toc136916225)

[Figure 15.(a)Scaled Dot-Product Attention(left) (b)Multi-Head Attention with Multiple Attention Layers(right)(Maxime,2019) 20](#_Toc136916226)

[Figure 16.Example of Masked Multi-Head Attention(Kortschak,2022) 22](#_Toc136916227)

[Figure 17.Example of probability scores for the next word prediction(Kortschak,2022) 22](#_Toc136916228)

[Figure 18.BERT- Base and Large Versions(Alammar,2018a;Kana,2019) 23](#_Toc136916229)

[Figure 19.Key Differences between BERT and Transformers(Alammar,2018a ;DeLucia,2023;Hui,2019;StackExchange,2020) 23](#_Toc136916230)

[Figure 20.Component Size Differences of Transformers and BERT(Alammar,2018a) 24](#_Toc136916231)

[Figure 21.BERT Framework Phases(Devlin et al.,2019) 24](#_Toc136916232)

[Figure 22.Inputs and Outputs of BERT(Hui,2019) 25](#_Toc136916233)

[Figure 23.BERT Input Representation(Devlin et al.,2019;Hui,2019;Khalid,2020) 25](#_Toc136916234)

[Figure 24.Example of BERT Bidirectional Processing(Khalid,2020) 26](#_Toc136916235)

[Figure 25.Mitigation Steps for MLM Mask Limitation(Devlin et al.,2019) 26](#_Toc136916236)

[Figure 26.NSP Training Examples(Devlin et al.,2019) 27](#_Toc136916237)

[Figure 27.NSP Training Example(Hui,2019) 27](#_Toc136916238)

[Figure 28.Fine-tuning of BERT for different NLP tasks(Devlin et al.,2019) 27](#_Toc136916239)

[Figure 29.Example for Question Answering(Hui,2019) 28](#_Toc136916240)

[Figure 30.Equations for finding start and end of answer in Question Answering task(Devlin et al.,2019;Hui,2019) 28](#_Toc136916241)

[Figure 31.Candidate Span Score Calculation Equation(Devlin et al.,2019) 29](#_Toc136916242)

[Figure 32.Recommended Optimal Hyper Parameter Values(Devlin et al.,2019) 29](#_Toc136916243)

[Figure 33.Applications of BERT(Alammar,2018a;Chakraborty,2020;Jain,2022;Persson,2021) 30](#_Toc136916244)

[Figure 34.Benefits of BERT(DeLucia,2023;Ravichandran,2022) 30](#_Toc136916245)

[Figure 35.Limitations of BERT(Devlin et al.,2019; McCormick,2019;Ravichandran,2022) 31](#_Toc136916246)

**Abbreviations**

|  |  |
| --- | --- |
| BERT | Bidirectional Encoder Representations from Transformers |
| CUDA | Compute Unified Device Architecture |
| DL | Deep Learning |
| FFN | Feed-Forward Network |
| LSTM | Long Short Term Memory |
| MHA | Multi-Head Attention |
| MLM | Masked Language Model |
| MMHA | Masked Multi-Head Attention |
| NLI | Natural Language Inference |
| NLP | Natural Language Processing |
| NN | Neural Network |
| NSP | Next Sentence Prediction |
| PE | Positional Encoding |
| QA | Question Answering |
| RNN | Recurrent Neural Network |
| SQuAD | Stanford Question Answering Dataset |

# **Introduction**

Bidirectional Encoder Representations from Transformers(BERT) is an open-sourced Natural Language Processing(NLP) technique developed by researchers at Google in 2018(Khalid,2020). Several reasons made BERT a breakthrough in the Deep Learning(DL) space, including better state-of-the-art performance and pre-trained on a bidirectional language representation model enabling fine-tuning of specific NLP tasks more efficiently than its predecessors(Devlin et al.,2019;Khalid,2020). BERT can be used as an embedding to extract features from data and for finetune models for NLP tasks such as text summarisation, question answering, language translation and sentimental analysis (Khalid,2020; ProjectPro,2023).

# **Relevance of BERT**

DL-based NLP models require huge volumes of data on specific task training to perform well, however, there is no sufficient human-annotated data available for training these kinds of models(Khalid,2020). This is one of the major challenges faced in this field and this problem can be resolved using pre-training and fine-tuning models(ibid). Researchers have developed various pre-training techniques that can leverage unlabelled data on the internet for training general-purpose language models and then fine-tuning those models for specific tasks allowing them to perform well for those tasks without training them from scratch(ibid).

Furthermore, the standard language models before BERT were trained unidirectional, restricting them from understanding the context of the text from both directions which is crucial for fine-tuning based models for token-level tasks such as question answering (Devlin et al.,2019). BERT leverages a Masked Language Model(MLM) that enables it to understand the context from both directions(ibid). MLM masks the tokens randomly from the input text and attempts to predict the masked vocabulary id correctly by understanding the context of the text(ibid). This feature of BERT also makes it suitable for fine-tuning the models more efficiently than predecessor models(ibid).

# **BERT Architecture and Working**

Having understood the relevance of BERT, this section covers the architecture and its working in more detail. BERT employs a multi-layered bidirectional encoder based on the architecture of Transformers for processing text (Devlin et al.,2019; Ravichandran,2022). Therefore, it is good to have an understanding of Transformers before diving into BERT architecture.

# **Transformers**

Transformers are widely used Neural Network(NN) architectures because they overcame the limitations of its predecessors including Recurrent Neural Networks(RNNs) and Long Short Term Memory(LSTM) (Giacaglia,2019). Fig.1 depicts the limitations of RNNs and LSTM architectures.

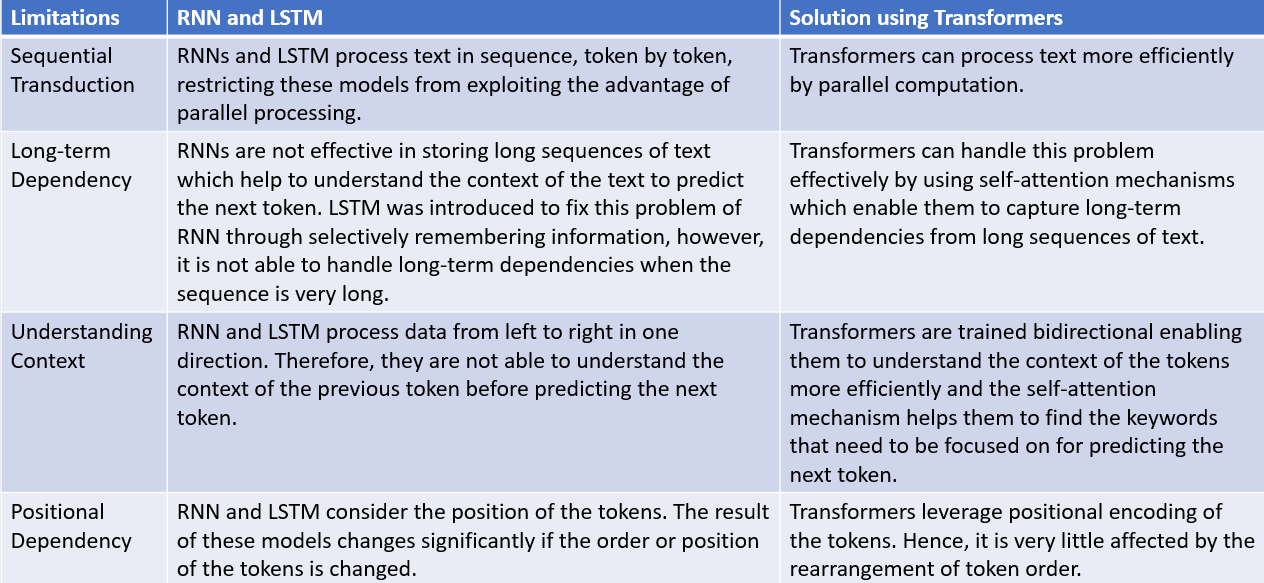


Figure 1.Limitations of RNN and LSTM Architectures(Giacaglia,2019)

It’s good to look at the architecture to get clear understanding on underlying architecture of BERT and how transformers overcome the limitations of its predecessors. As illustrated in Fig.2, the architecture of transformers consists of two components: Encoder and Decoder(Jain,2022). The Encoder extracts features from the input sequence and Decoder uses the features to generate an output(KiKaBeN,2021). The transformer model comprises a stack of Encoders and Decoders as demonstrated in Fig.3. Each component is covered in the next sections.

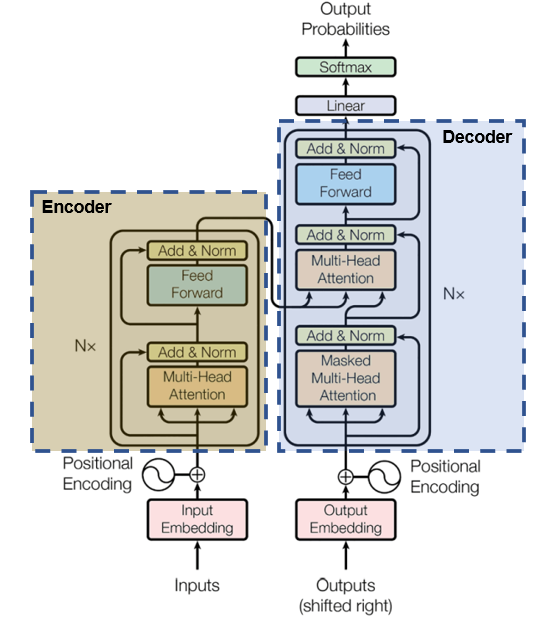


Figure 2.High-level Architecture of Transformers(Vaswani et al.,2017)

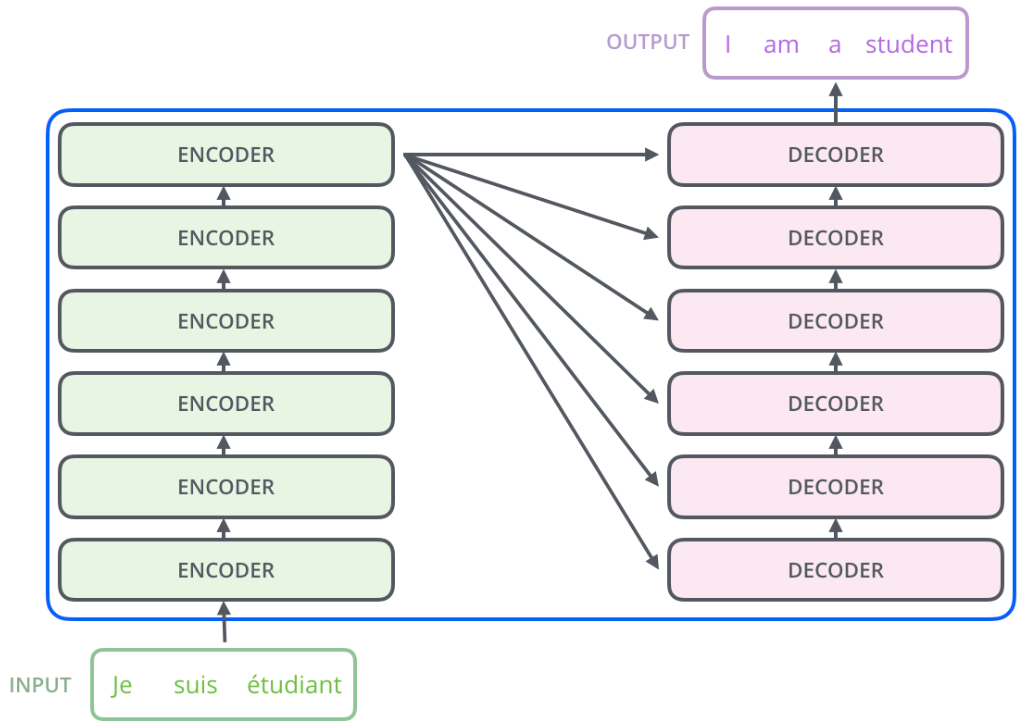


Figure 3.Encoder-Decoder Architecture of Transformers(Alammar,2018b)

# **Encoder**

Transformer encoder comprises a stack of identical layers where each layer has two main sub-layers: Multi-Head Attention(MHA) and Feed-Forward Network(FFN) (Cristina,2023). The input embedding layer will pre-process the input sequence before it is processed for self-attention(Jain,2022).

# **Input Embedding**

The input embedding layer maps each word or token in the input text sequence to a learned vector representation through tokenisation and vectorisation(Jain,2022). NNs learn through numerical values, so each word is represented using a continuous value in the vector(ibid). This process is also known as Word Embedding(ibid). As illustrated in Fig.4, each word in the input sentence is converted into tokens during the tokenisation process and then each token is converted into a vector which is a numerical representation of the word using Word2Vec or GloVe encoding during the vectorisation process(ibid).

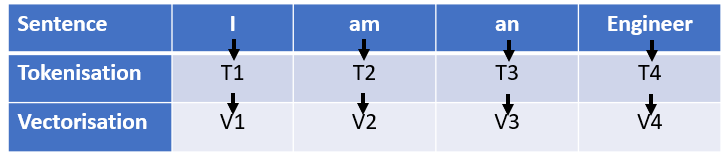


Figure 4.Tokenisation and Vectorisation(Jain,2022)

# **Positional Encoding(PE)**

The order and position of the words in a sentence are important in human language to interpret its meaning and context(Saeed,2023). The NLP using the RNN model leverages techniques to keep track of the order of words in the input sentence(ibid). However, the Transformers model considers each word as an independent token and adds PE to retain information about the position of the token within a text sequence(ibid).

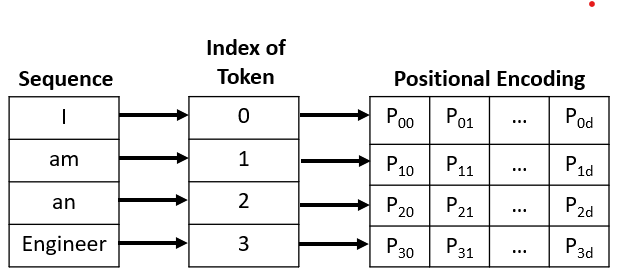


Figure 5.Positional Encoding Matrix for 'I am an Engineer'(Saeed,2023)

Instead of using numerical values such as index values, PE assigns a unique numerical representation that identifies the position of each token in a given sequence(Saeed,2023). One reason for not using index value is that index value can be a large number for long text sequences and normalising these indices between 1 and 0 might not provide expected results for variable length sequences(ibid). Hence, PE leverages a positional matrix in which each row is the summation of the encoded object with its positional information as illustrated in Fig.5(ibid). This transformation is achieved through sine and cosine functions as depicted in Fig.6 and an example is illustrated in Fig.7.

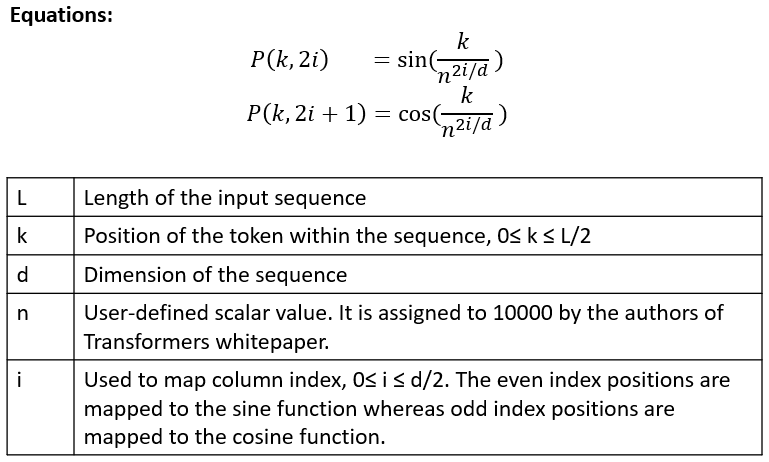


Figure 6.Positional Encoding Equations(Jain,2022;Saeed,2023)

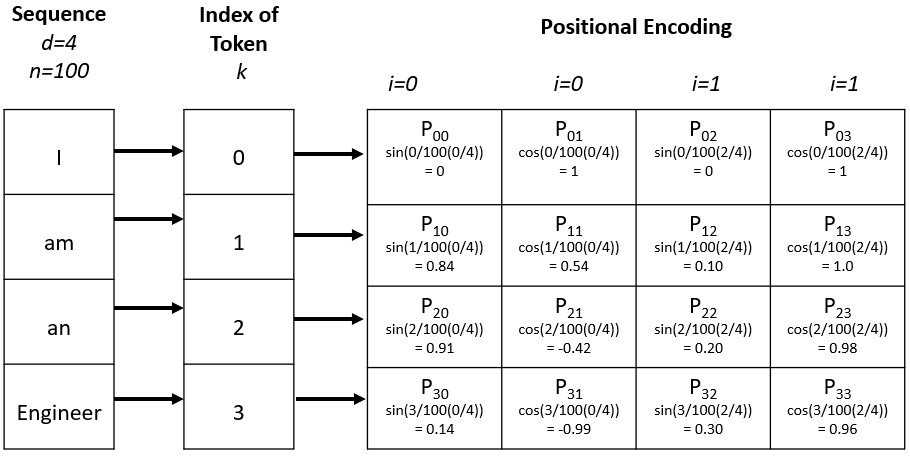


Figure 7.Positional Encoding Matrix for the sequence 'I am an Engineer'(Jain,2022;Saeed,2023)

The Input Embedding layer and PE will create a vector space where the words with similar semantics occur closer to each other as demonstrated in the Fig.8(Jain,2022). This vector space helps the model to easily learn to pay attention to the words represented in it(ibid).

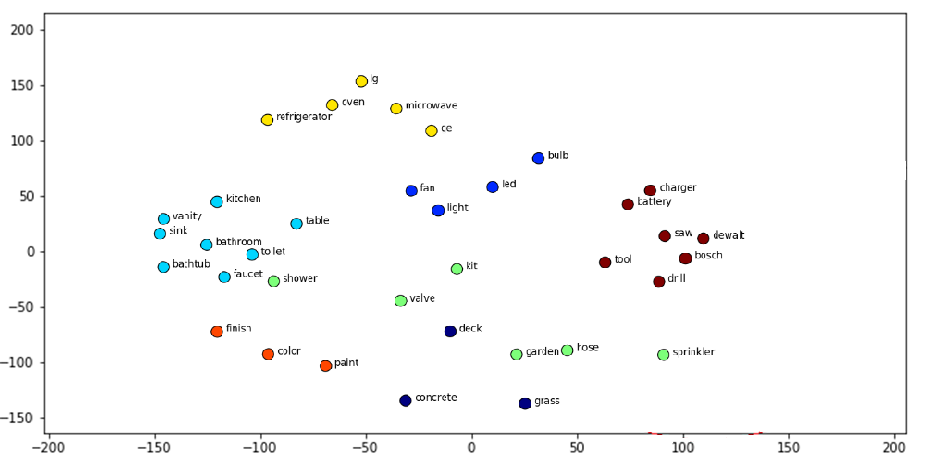


Figure 8.Word Embedding Space(Padmanabhan,2022)

# **Multi-Head Attention(MHA)**

MHA layer capture the context of the input sequence using self-attention mechanism which assist model to pay attention to different parts of the input sequence in parallel(Jain,2022;Karim,2023). An example of self-attention and its working are illustrated in Fig.9 and Fig.10.

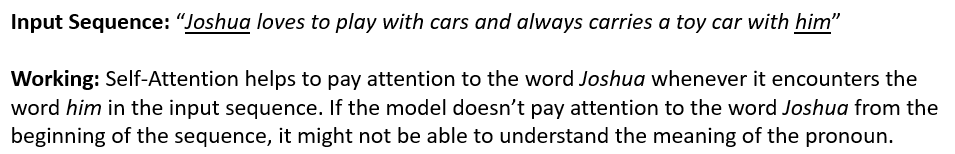


Figure 9.Example of Self-Attention Mechanism-Long Dependencies in a Sequence(Kortschak,2022)

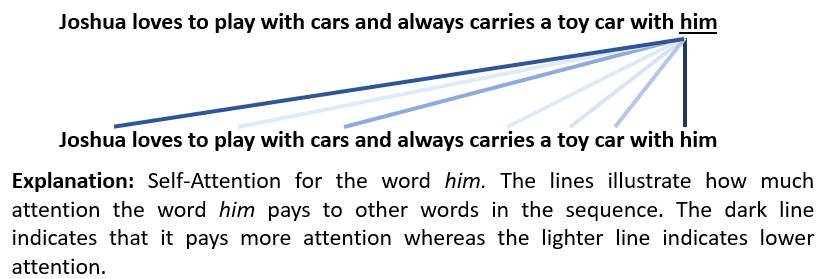


Figure 10.Example of Self-Attention Mechanism for the word ‘him’(Kortschak,2022)

MHA comprises multiple self-attention modules focusing on various attention types(Jain,2022).Self-attention receives an input vector and converts it into three representational input vectors – Query, Key and Value, explained in Fig.11(Kortschak,2022). The three vectors help in calculating scores for determining how much attention should be given to other words in the sequence as shown in Fig.10(ibid).

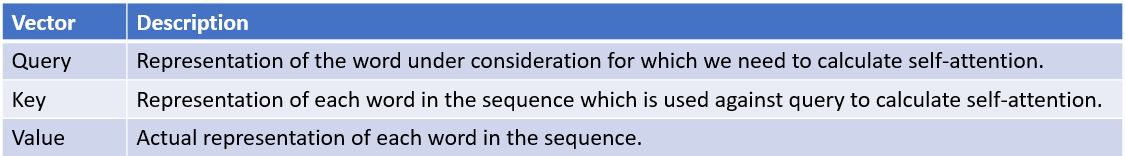


Figure 11.Self-Attention Representational Vectors(Kortschak,2022)

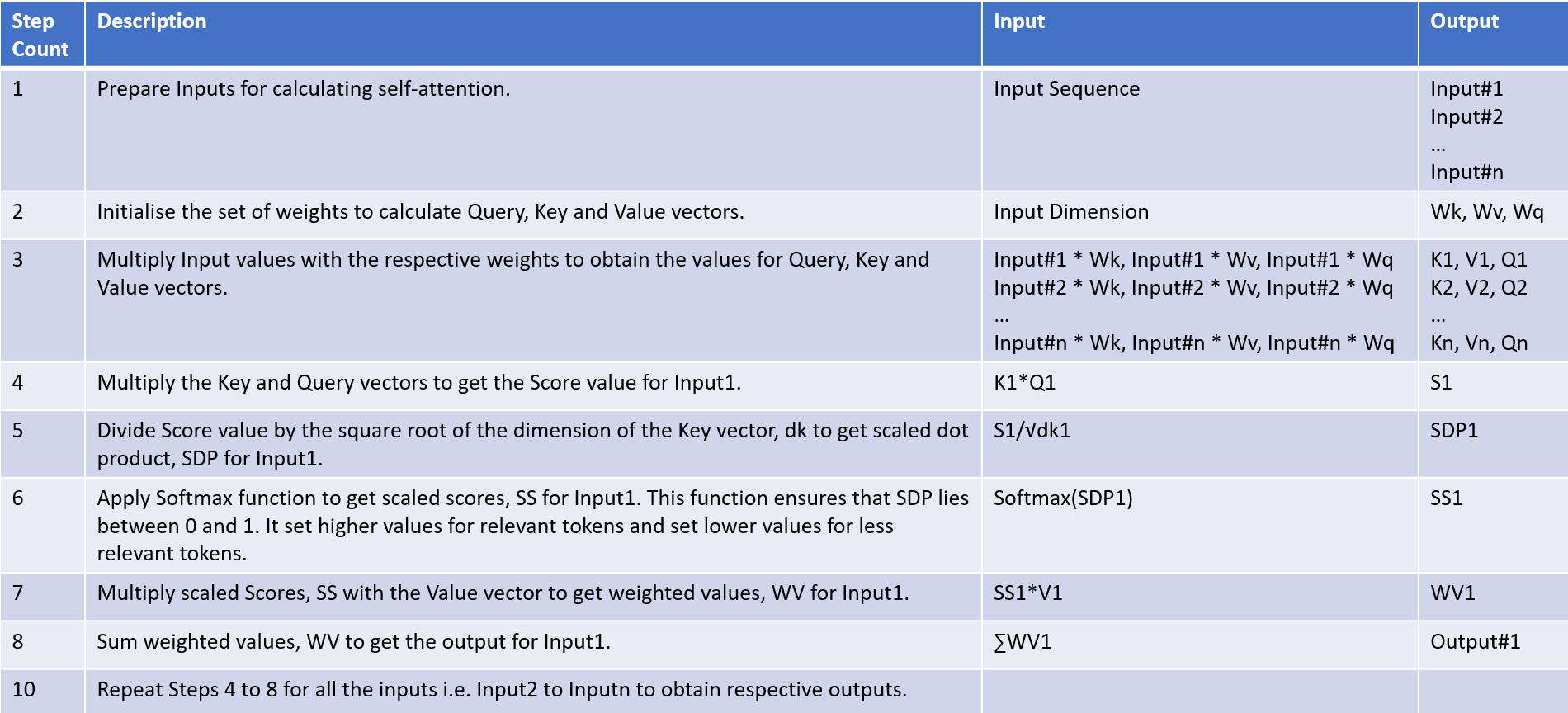


Figure 12.Steps for Self-Attention(Jain,2022;Karim,2023; Saeed,2022)

A set of steps are followed during the working of self-attention as illustrated in Fig.12 and all mathematical calculations followed in this process are vectorised(Karim,2023). Fig.13 depicts an example for the self-attention mechanism. The mechanism is explained with the help of the example demonstrated in Fig.13.

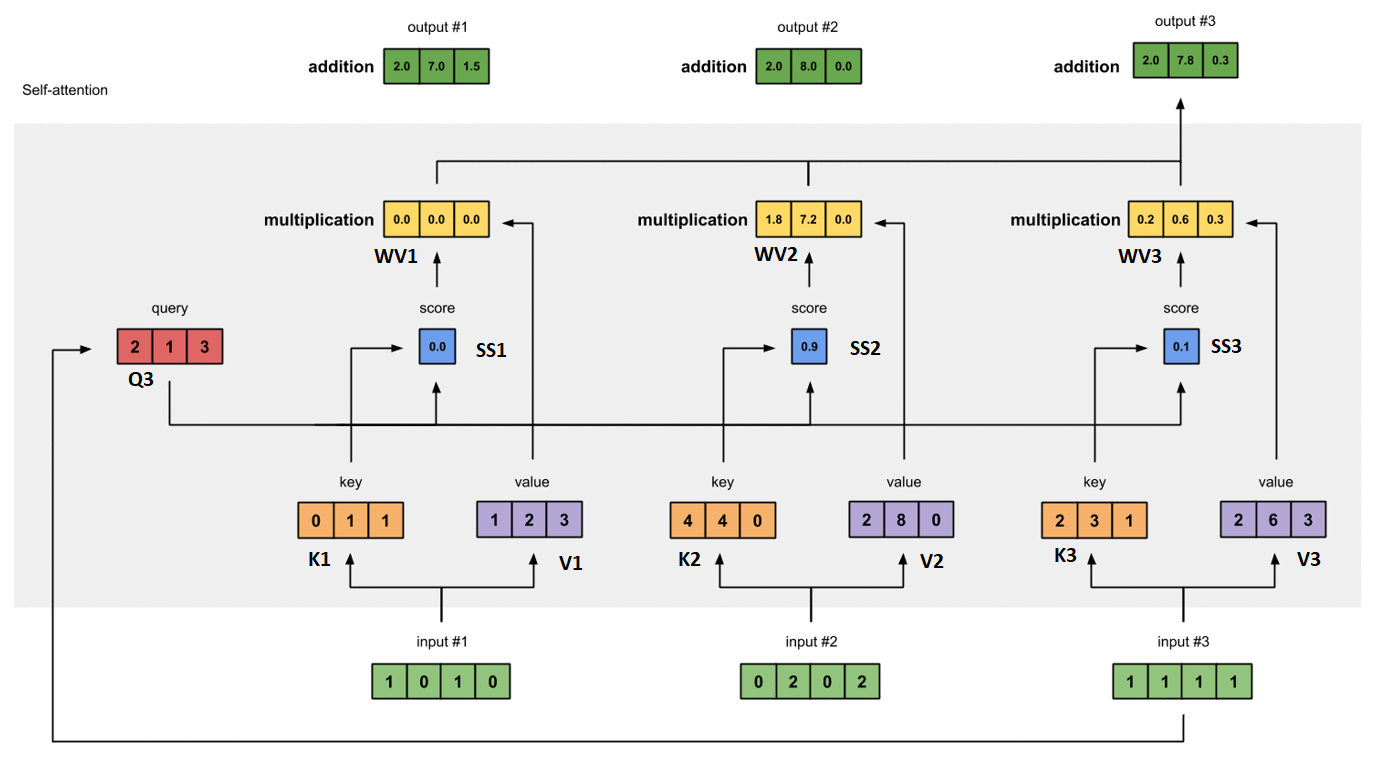


Figure 13.Example of Multi-Head Attention Mechanism(Karim,2023)

Firstly, the inputs for calculating self-attention are prepared(Karim,2023). In the example shown in Fig.13, inputs are highlighted as green boxes and the dimension of each input is 4(ibid). Secondly, the weights for calculating Query, Key and Value vectors are initialised(ibid). Each set of weights is set to 4x3 since every input dimension is 4 and it is illustrated in Fig.14(ibid). Thirdly, the three representational vectors for Query, Key and Value are obtained by multiplying it with each input and the results are highlighted in Fig.13 using yellow, purple and red boxes(ibid). The dimensions of the Query and Key vectors must be the same for performing multiplication operation in the upcoming step whereas the dimension of the Value vector can be different from the other two vectors(ibid).

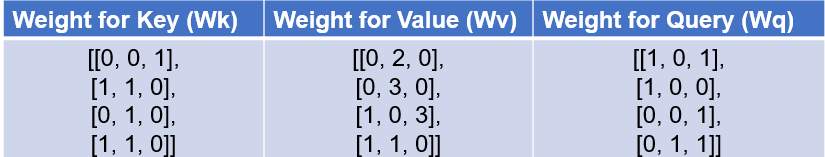


Figure 14.Weights for Key,Value and Query vectors(Karim,2023)

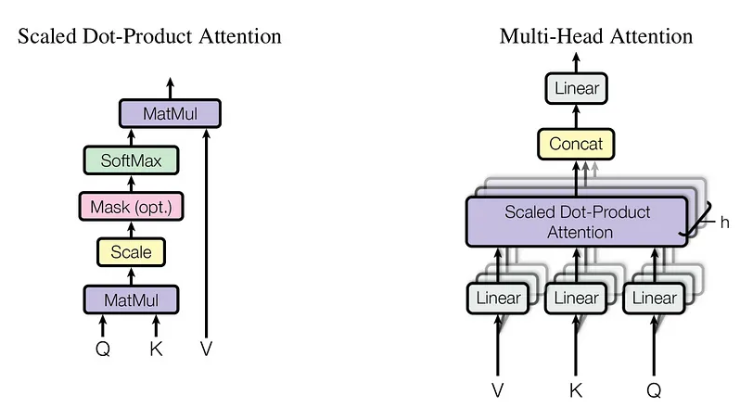


Figure 15.(a)Scaled Dot-Product Attention(left) (b)Multi-Head Attention with Multiple Attention Layers(right)(Maxime,2019)

Fourthly, the attention score is calculated by applying the score function on the Query and Key vectors and the dot product function is used to obtain score values in the above example(Jain,2022;Karim,2023). Fifthly, the attention score value is divided by the square root of the dimension of the Key vector, dk to get scaled dot product as illustrated in Fig.15(a)(Jain,2022). Sixthly, the softmax function is applied to the updated scores to normalise the value and it lies between 0 and 1(Saeed,2022). In the example, the scores were [2,4,4] and after applying softmax function, it became [0,0.5,0.5](Karim,2023). It exaggerates relevant value and lowers the non-relevant value(Jain,2022;Saeed,2022).

Seventhly, the normalised attention scores are multiplied with the Value vector to get the weight values(Karim,2023). Eighthly, calculate the sum of all the weighted values in the yellow boxes to get the output of the first input. Similarly, steps 4 to 8 are repeated for all other inputs(ibid). Thus MHA performs multiple self-attentions in parallel as illustrated in Fig.15(b), each focusing on different parts of the input sequence that helps the model to capture the context of the input(Jain,2022). As discussed earlier, transformers overcome the limitation of its predecessors in handling long-term dependencies and capturing context of input using MHA.

# **Feed-Forward Network(FFN)**

The output from the self-attention layer is forwarded to a fully connected neural network called the FFN layer which is applied independently to each position of the sequence(Alammar,2018b). It comprises two linear layers-GeLu and Dropout layers and a non-linear function, ReLu in between (Kortschak,2022;Kosar,2021;Suresh,2022). The goal of this layer is to transform the output vectors from the self-attention layer to an acceptable input form for the next encoder or decoder layer(Ankit,2020). Since each self-attention layer can work independently in parallel to generate output, FFN is the most scaled-up component because it can take only one input at a time(Ankit,2020;Suresh,2022). As we discussed in the previous section, this is how transformers overcome the limitation of its predecessors which follows sequential processing(Ankit,2020).

# **Decoder**

The decoder has a similar architecture of encoder. However, unlike encoder, the decoder leverages Masked MHA(MMHA)(Kortschak,2022). Since BERT leverages only the encoder of transformer for its implementation, the decoder section is not covered in detail. However, the next section will give a brief overview on MMHA.

# **Masked Multi-Head Attention(MMHA)**

In MMHA, some parts of the decoder input is masked before it is forwarded to self-attention layer which helps to train the model for better predictions(Cristina,2022; Kortschak,2022). The MMHA is designed to predict the word at a given position based on the known outputs of the words that are at preceding positions in the input sequence without knowing the words at succeeding positions as shown in Fig.16(ibid).

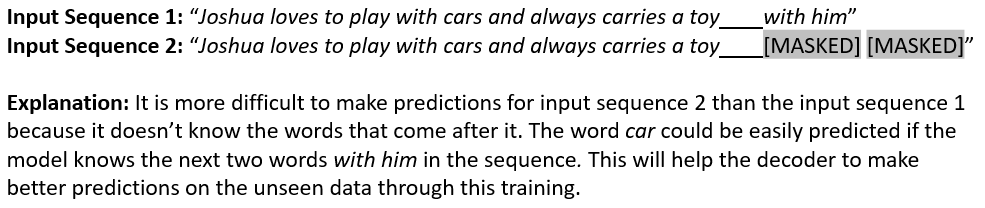


Figure 16.Example of Masked Multi-Head Attention(Kortschak,2022)

# **Linear and Softmax Layer**

To obtain the output predictions, the last decoder output is transformed into words using a linear function layer, generating a logit vector of vocabulary size(Kortschak,2022). Then softmax function is applied to obtain the probability scores of each word as illustrated in Fig.17, and the word with the highest probability is chosen as prediction as it will be the word closer in the word embedding space(ibid).

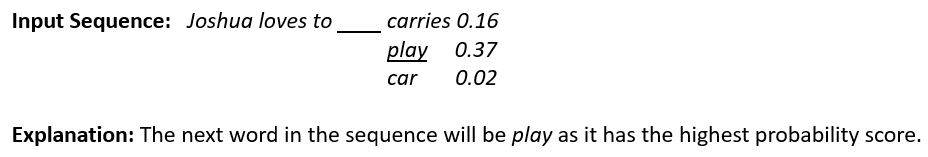


Figure 17.Example of probability scores for the next word prediction(Kortschak,2022)

Having understood the transformer architecture in detail, the next section will cover the difference between BERT and Transformers.

# **BERT Architecture**

BERT is a pretrained model built on transformer encoder stack(Kana,2019). While original transformer model used a 6 encoders stack, BERT’s base version is 12 encoders stack whereas its large version is 24 encoders stack as illustrated in Fig.18(ibid).

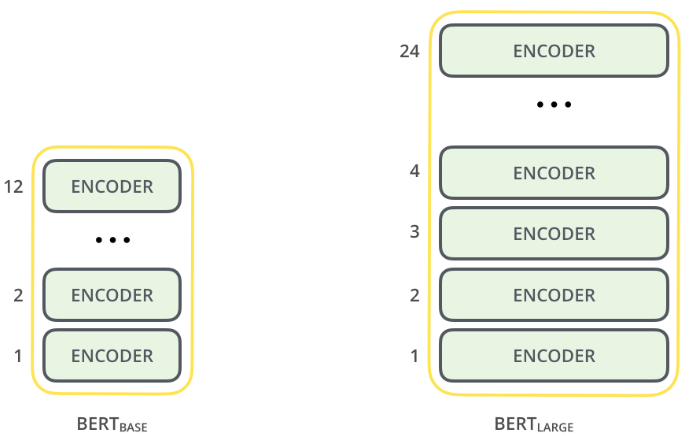


Figure 18.BERT- Base and Large Versions(Alammar,2018a;Kana,2019)

# **Differences between BERT and Transformers**

The key differences and component size differences of BERT and Transformer are depicted in Fig.19 and Fig.20 respectively.

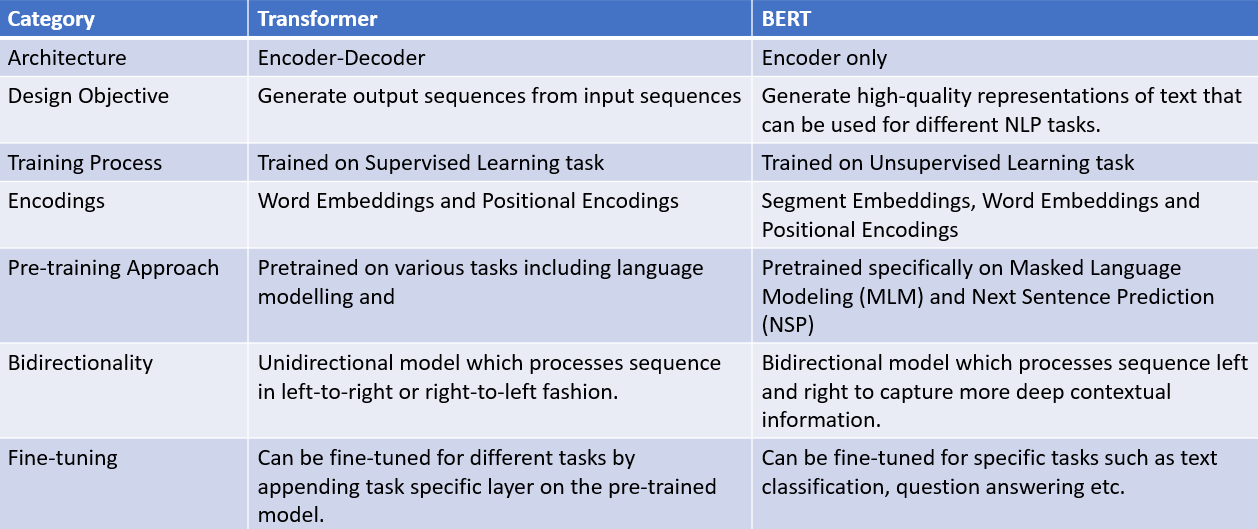


Figure 19.Key Differences between BERT and Transformers(Alammar,2018a ;DeLucia,2023;Hui,2019;StackExchange,2020)

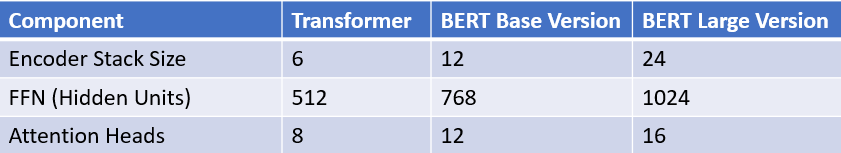


Figure 20.Component Size Differences of Transformers and BERT(Alammar,2018a)

# **BERT Framework Steps**

BERT undergoes through two phases-Pretraining and Fine-tuning(Devlin *et al.*,2019) as depicted in Fig.21.

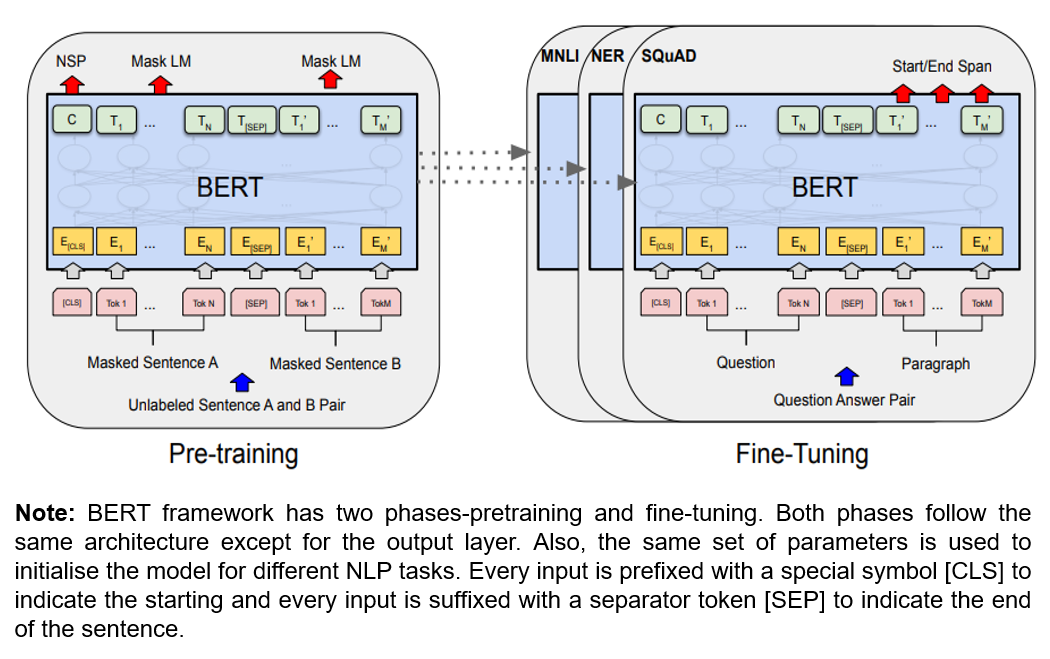


Figure 21.BERT Framework Phases(Devlin et al.,2019)

# **BERT Pretraining**

BERT is the first pretrained model on unlabelled data and two NLP tasks are performed during this phase-MLM and NSP which will be covered in the upcoming sections(Hui,2019). Before diving into sub-tasks, it’s good to understand the inputs and expected outputs of the pretraining.

# **Inputs and Outputs**

BERT requires one or two text sequences as input and as mentioned in Fig.21 input is prefixed with [CLS] and suffixed with [SEP](Hui,2019). The model will generate N output tokens for N input tokens including appended tokens as shown in Fig.22(ibid).

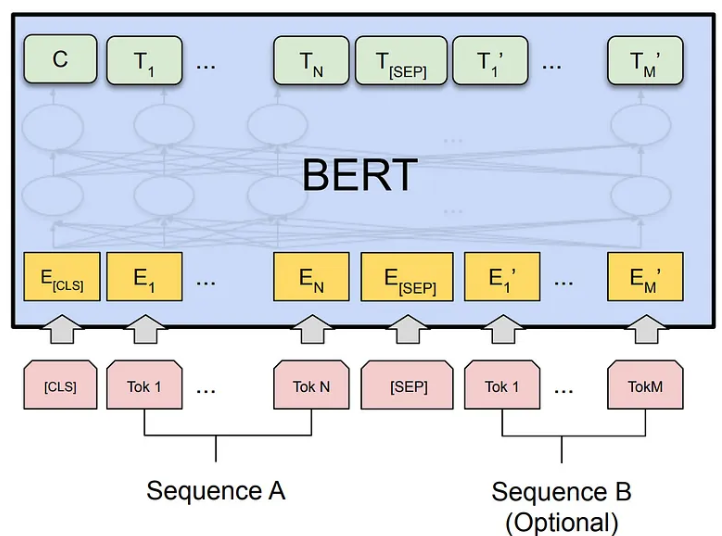


Figure 22.Inputs and Outputs of BERT(Hui,2019)

The appended input token [CLS] will generate an output token C which is used only for classification problems and ignored for non-classification problems(Hui,2019). Fig.23 illustrates the input representations used in BERT.

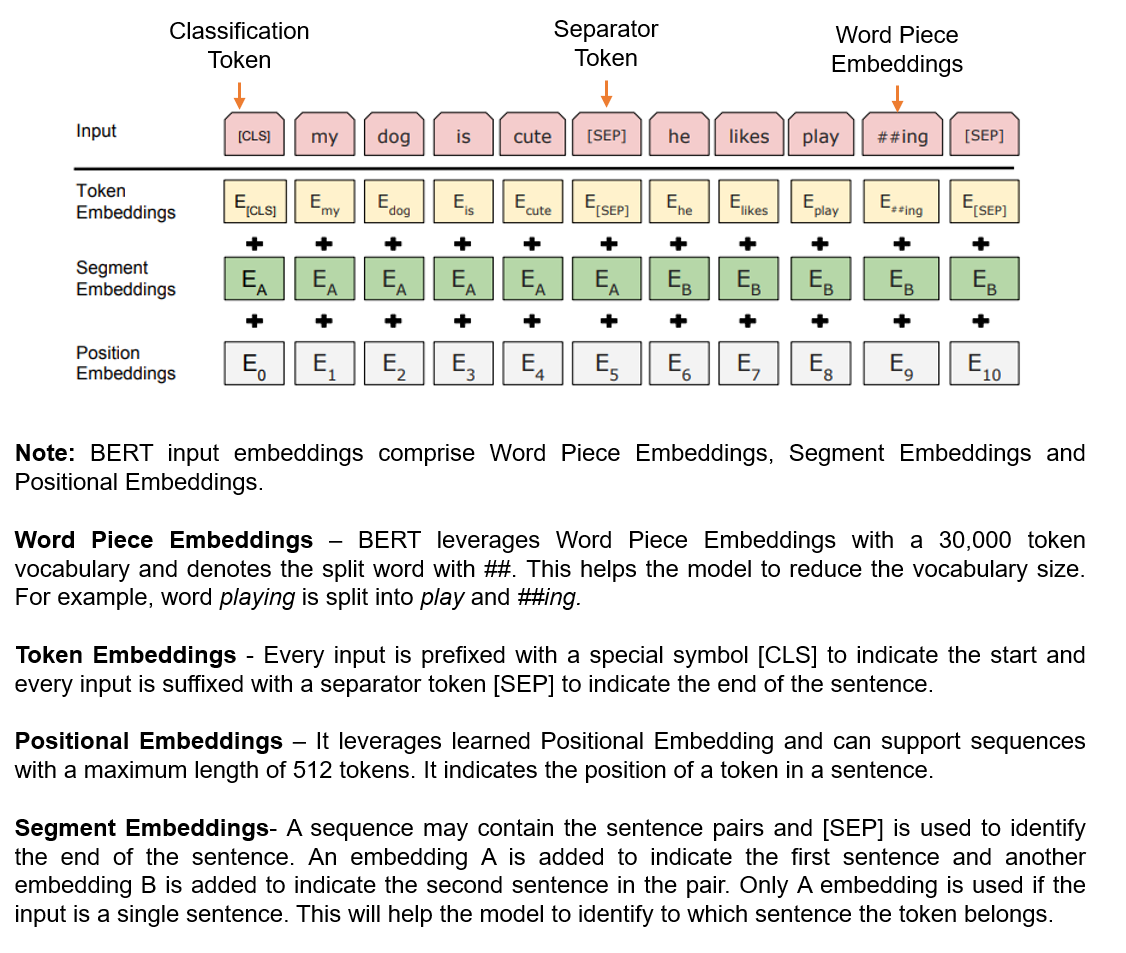


Figure 23.BERT Input Representation(Devlin et al.,2019;Hui,2019;Khalid,2020)

# **Masked Language Modelling(MLM)**

In order to train deep bidirectional representation, MLM randomly masks 15% of words in the input sequence by replacing words with [MASK] token and then predict those masked words based on the contextual information obtained from unmasked words in the sequence as illustrated in Fig.24(Devlin *et al.*,2019;Khalid,2020). The drawback of this approach is that the [MASK] token will not be present in fine-tuning phase, resulting in a mismatch with pre-training phase and it can be overcome using different techniques listed in Fig.25.

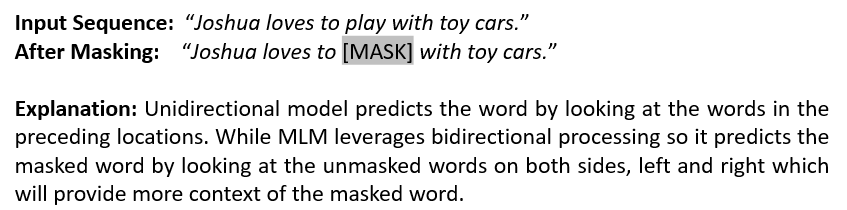


Figure 24.Example of BERT Bidirectional Processing(Khalid,2020)

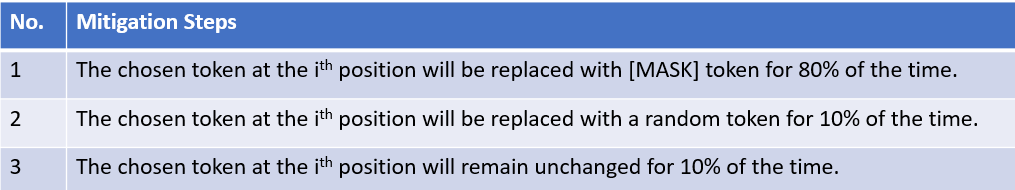


Figure 25.Mitigation Steps for MLM Mask Limitation(Devlin et al.,2019)

BERT uses a cross entropy loss function which considers only the prediction of masked tokens and ignores the unmasked token predictions(Devlin *et al.*,2019;Khalid,2020).

# **Next Sentence Prediction(NSP)**

The model needs to understand the relationship between sentences in an input in order to perform different NLP tasks such as Question Answering(QA), Natural Language Inference(NLI) etc.(Devlin *et al.*,2019). NSP helps the model to understand the relationship between two sentences in an input sequence and also helps it to predict next sentence(Khalid,2020). The model is trained on inputs with sentence pairs and learns to predict next sentence in the original input(ibid). In this training, the output token C is considered for classification as mentioned in the previous section(Hui,2019). Fig.26. shows how training samples are selected for NSP and Fig.27 illustrates an example of it.

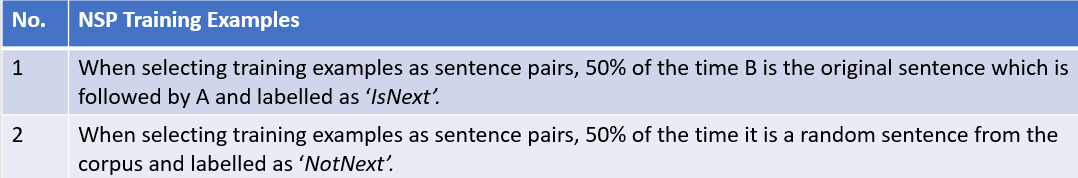


Figure 26.NSP Training Examples(Devlin et al.,2019)



Figure 27.NSP Training Example(Hui,2019)

The combination of MLM and NSP approaches during pretraining helps the model to minimise the loss function(Khalid,2020).

# **BERT Fine-tuning**

Once the model completes the pretraining, it can be finetuned for specific NLP tasks as illustrated in Fig.28(Khalid,2020). In this phase, the model parameters are finetuned end-to-end by providing task related data and corresponding labels(Hui,2019). Fine-tuning phase is relatively cheaper than pre-training phase(Devlin *et al.*,2019).

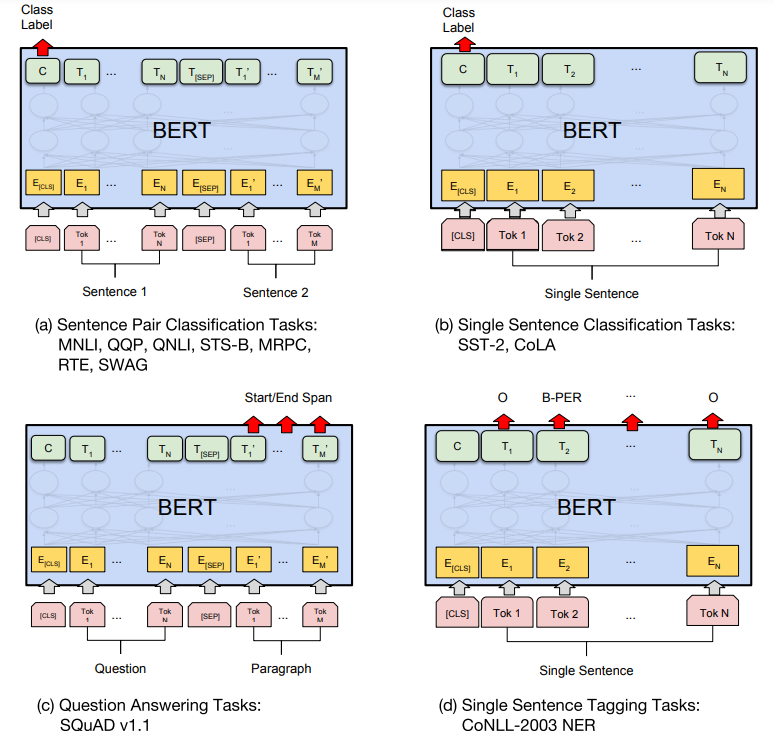


Figure 28.Fine-tuning of BERT for different NLP tasks(Devlin et al.,2019)

# **BERT Fine-tuning for Question Answering Task**

BERT can be finetuned different downstream tasks as illustrated in Fig.28 including sequence-level tasks such as sentence pair classification and single sentence classification, and token-level tasks such as question answering and single sentence tagging tasks(Devlin *et al.*,2019). In this section, BERT finetuning is explained using question answering task.

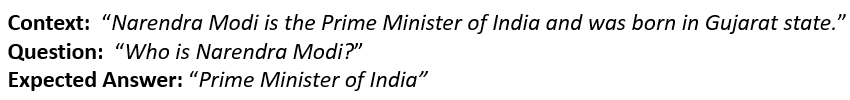


Figure 29.Example for Question Answering(Hui,2019)

During finetuning, only two vectors are introduced which is Start(S) vector and End(E) vector(Devlin *et al.*,2019;Hui,2019). *Ti*  is the output token at ith position of the sequence(ibid). The objective is to find whether the token at ith position is the start of answer, for example above, it should determine the position of the word *‘Prime’* which is the first word of the answer(ibid). For achieving this goal, the model performs a dot product of S and *Ti* and the output value is softmaxed over all the tokens in the paragraph. Fig.30-Equation-1 illustrates the mathematical equation for this.

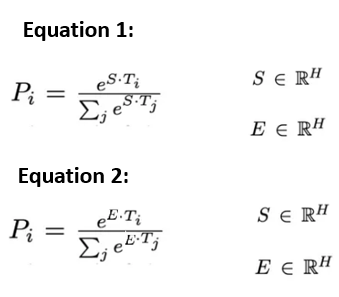


Figure 30.Equations for finding start and end of answer in Question Answering task(Devlin et al.,2019;Hui,2019)

Similarly, to find the end of the answer, it performs the same equation with End vector as depicted in Fig.30-Equation-2. Then score of each candidate span (phrase or word) is calculated by the equation in Fig.31(Devlin *et al.*,2019). The candidate span with highest score is considered as the answer to the question(ibid). In the above example, the candidate span with highest score will be *“Prime Minister of India”.*

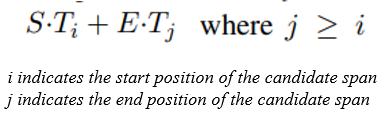


Figure 31.Candidate Span Score Calculation Equation(Devlin et al.,2019)

The Stanford Question Answering Dataset(SQuAD) is a collection of crowd-sourced question and answer pairs dataset used for finetuning BERT model for the question answering task(Devlin *et al.*,2019).

# **Recommended Hyper Parameter Values**

Although most model parameters are same as that of pretraining, there are some exceptions such as batch size, learning rate, epoch count and dropout probability(Devlin *et al.*,2019). The authors of BERT recommend a set of hyperparameter values that works well across all tasks based on their experiments are listed in Fig.32.

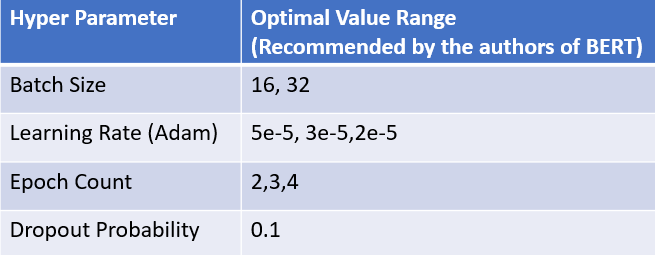


Figure 32.Recommended Optimal Hyper Parameter Values(Devlin et al.,2019)

# **BERT Real-World Applications**

As discussed in the previous sections, BERT has several applications and a few are listed in Fig.33.

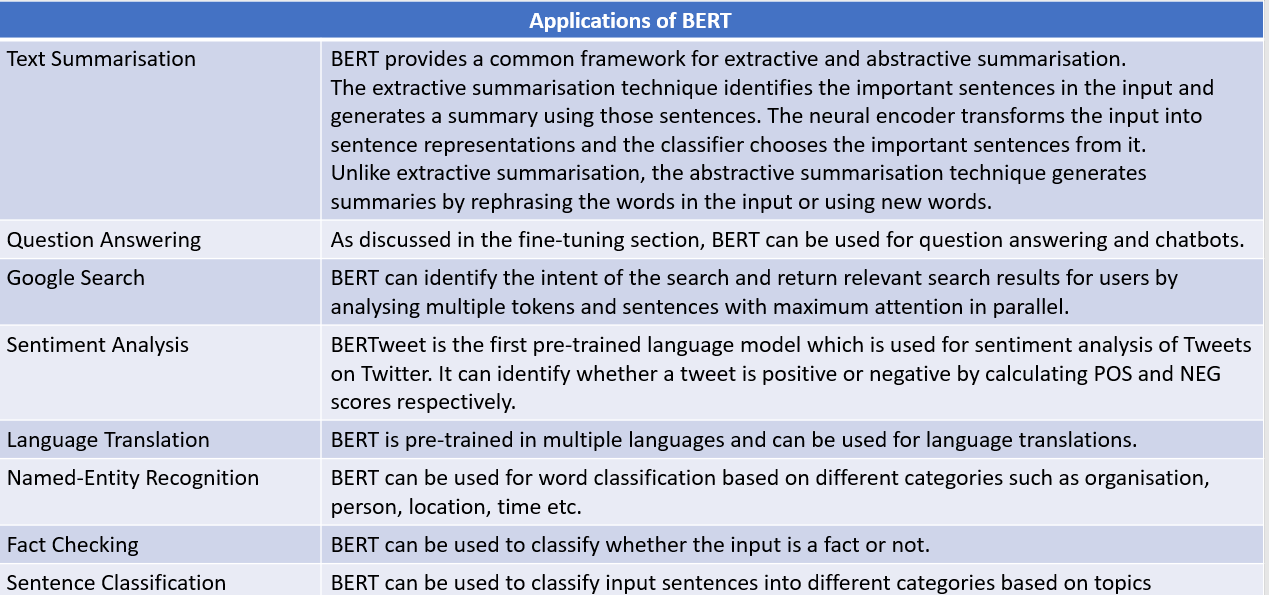


Figure 33.Applications of BERT(Alammar,2018a;Chakraborty,2020;Jain,2022;Persson,2021)

# **Benefits and Limitations of BERT**

As we discussed in the previous sections, there are several benefits of using BERT as shown in Fig.34. Also, it has some drawbacks and a few are listed in Fig.35.

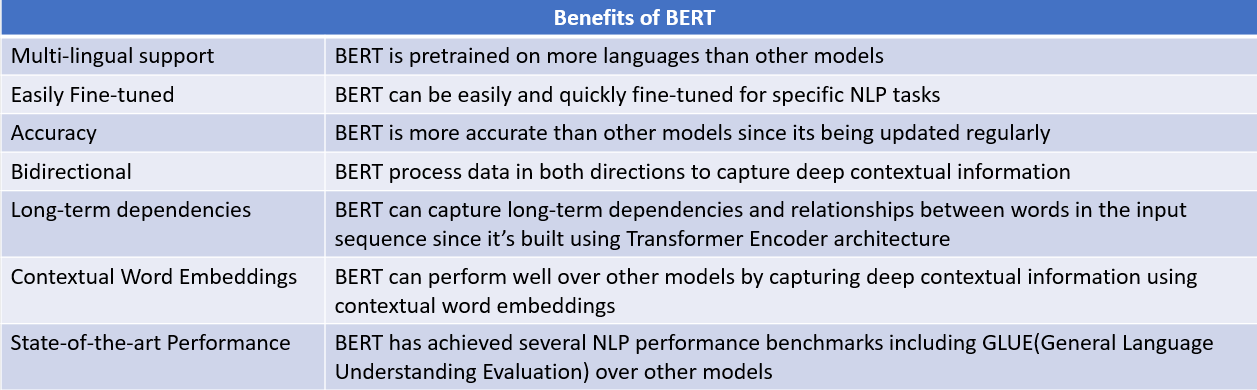


Figure 34.Benefits of BERT(DeLucia,2023;Ravichandran,2022)

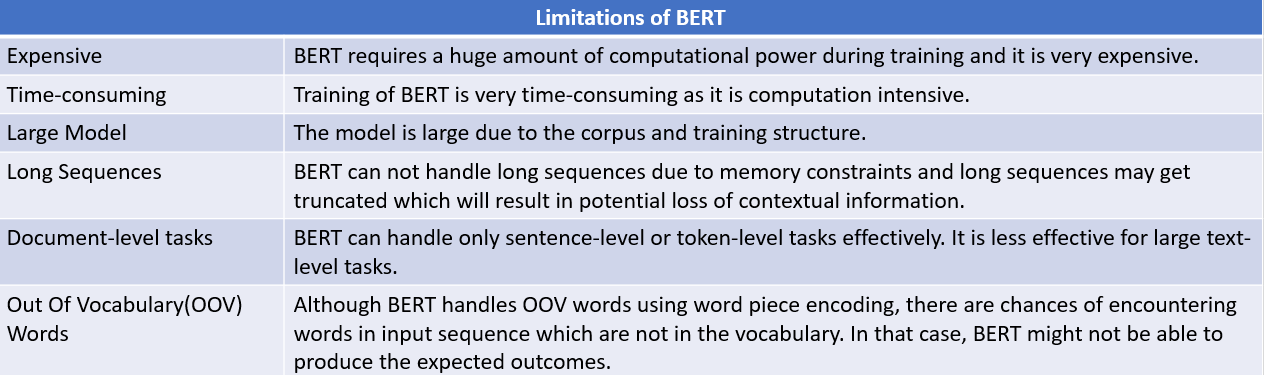


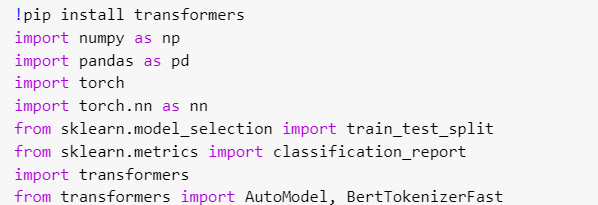
Figure 35.Limitations of BERT(Devlin et al.,2019; McCormick,2019;Ravichandran,2022)

# **BERT Model for Spam Message Classification**

This section will cover step by step tutorial to build a model using BERT for classifying the message whether it is spam or not.

# **Step 1: Import Libraries and Dataset**

Firstly, to setup the environment, install required libraries and import as needed.

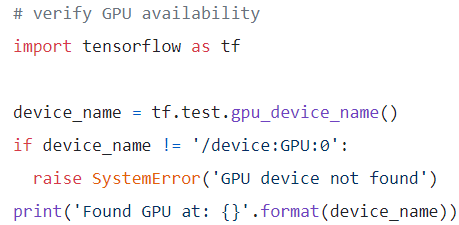




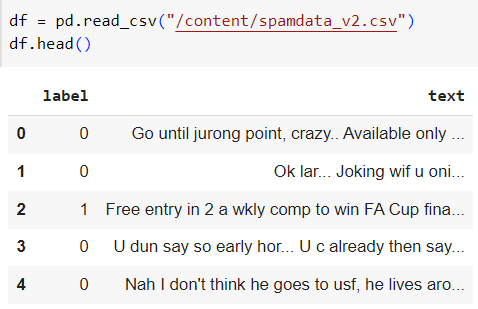
Once required libraries are imported, specify GPU as CUDA(Jain,2021). CUDA(Compute Unified Device Architecture) is developed by NVIDIA to provide a platform to accelerate the GPU capabilities(Turing,2022).



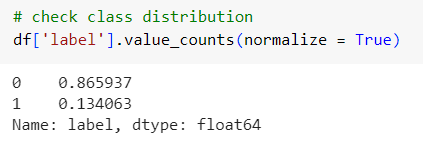
To check the availability of the GPU, the following code is executed and if it is not found it will throw an error(Kana,2019).



Then, read the data from the local file and load data from the dataset into a dataframe(Jain,2021). The dataset contains two columns label and text, where label 1 indicates spam and 0 indicates genuine(ibid)

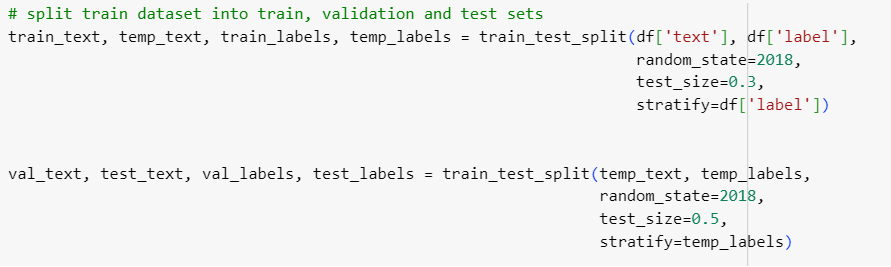


Then, check the label class distribution as shown below and it will show the percentage of spam and genuine messages in the dataset(Jain,2021).



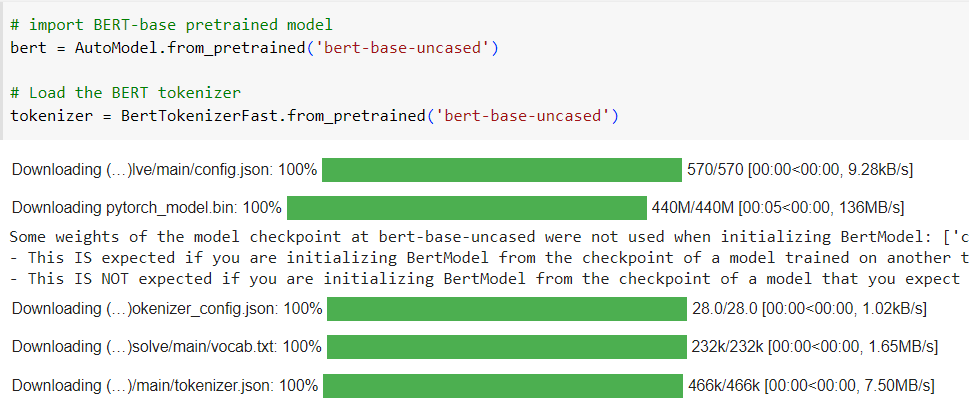
# **Step 2: Divide the data for testing and training**

In this step, data is divided for testing and training purposes(Jain,2021).

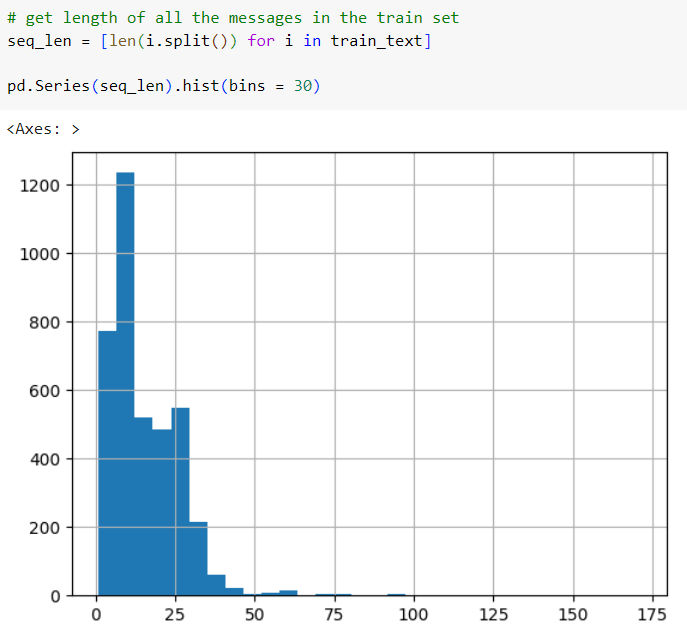


# **Step 3: Import BERT and Load Tokenizer**

In this step, hugging face BERT model, uncased version is imported and load the BERT tokenizer for preparing input representations(Jain,2021).

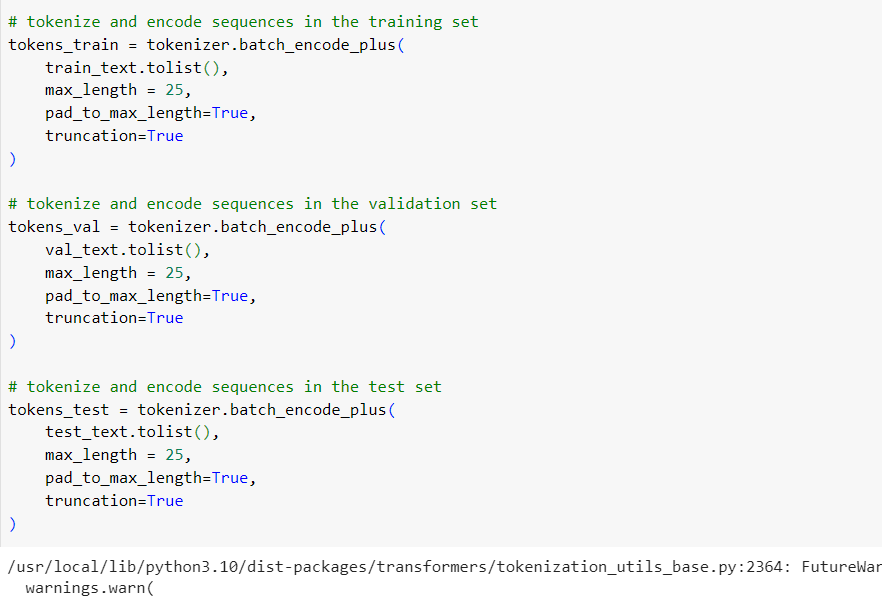


Check the length of messages in training data set(Jain,2021).



# **Step 4: Tokenize and Encode the Input Sequence**

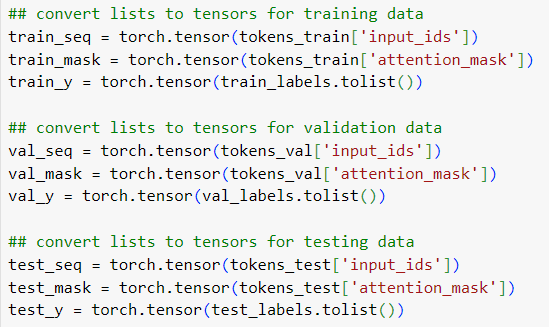
As the fourth step, tokenize the input for testing, training and validation datasets(Jain,2021)



As we discussed in previous sections, BERT uses Word Piece tokenization and maximum sequence length of the sequence can be upto 512(Jain,2021). In this example, the max\_length is set to 25 so that input token length can be upto 25. Also, the pad\_to\_max\_length is set to true for padding the tokens that are less than the max\_length. The results of the tokenization is stored into respective lists.

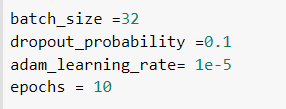
# **Step 5: Convert Lists to Tensors**

The model expects the input datatype as torch tensors(Kana,2019). Therefore, all the data which is stored in the lists during the previous step is converted into tensors(Jain,2021).



# **Step 6: Hyper Parameter Optimisation**

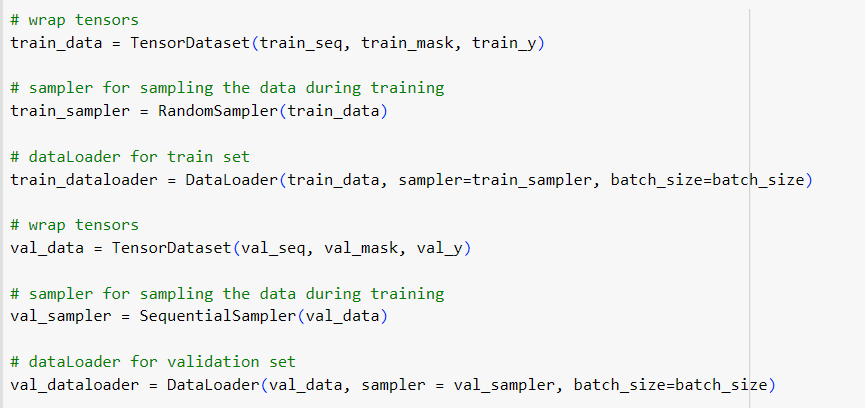
In this step, optimal values are assigned to hyper parameters. As discussed in the previous section, recommended optimal values can be used. However, in this example, learning rate and epoch is selected differently.



Also, automated hyper parameter optimisation can be done using libraries such as GridSearchCV for better accuracy(Brownlee,2022).

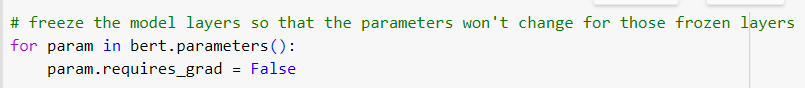
# **Step 7: Initialise Data Loader**

As seventh step, the data loaders are initialised for training and validation datasets(Jain,2021). This will create iterators required for fine-tuning the model using the data during BERT fine-tuning phase(Kana,2019).

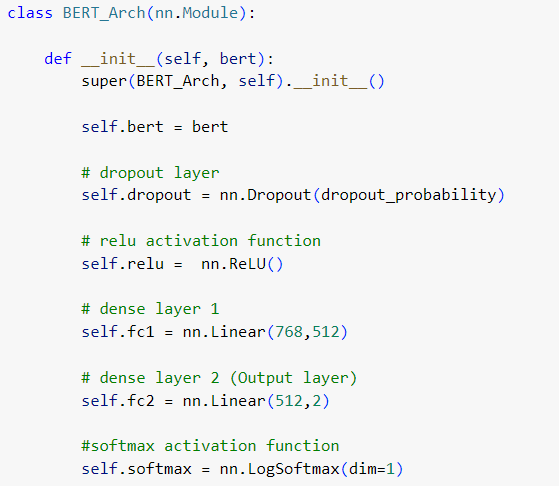


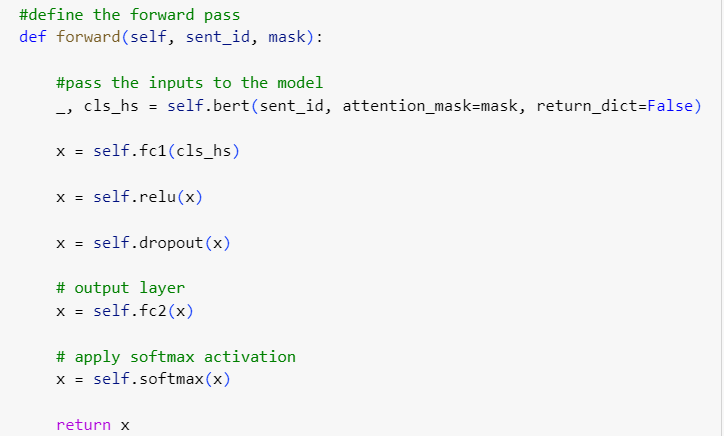
# **Step 8: BERT Model for Pretraining**

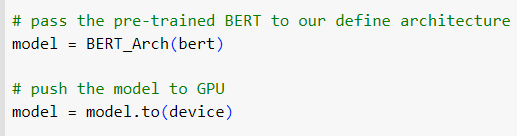
In this step, all the layers involved in the pretraining phase are defined including attention layers, FFN layer, linear and softmax layers. This code will freeze the layers of the model and the parameter values will not change for the frozen layers(Jain,2021).



In this example, BERT\_Arch class defined with two functions – init() and forward(). This function will initialise and set layers required for pretraining.

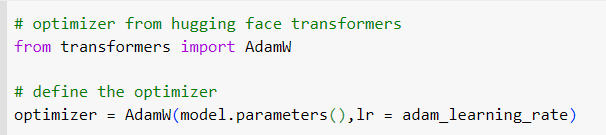


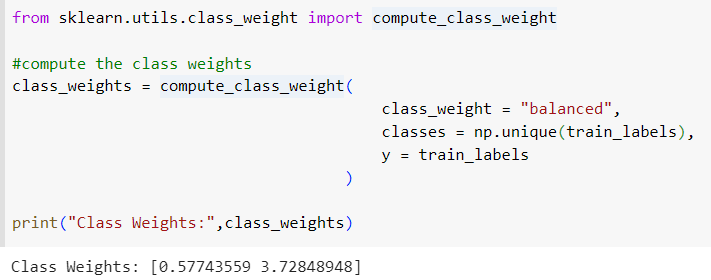




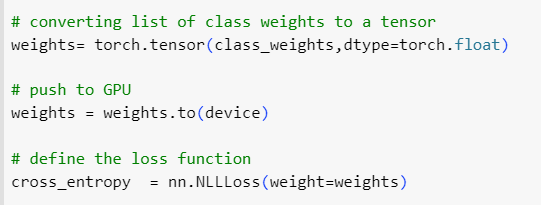
# **Step 9: BERT Fine-tuning**

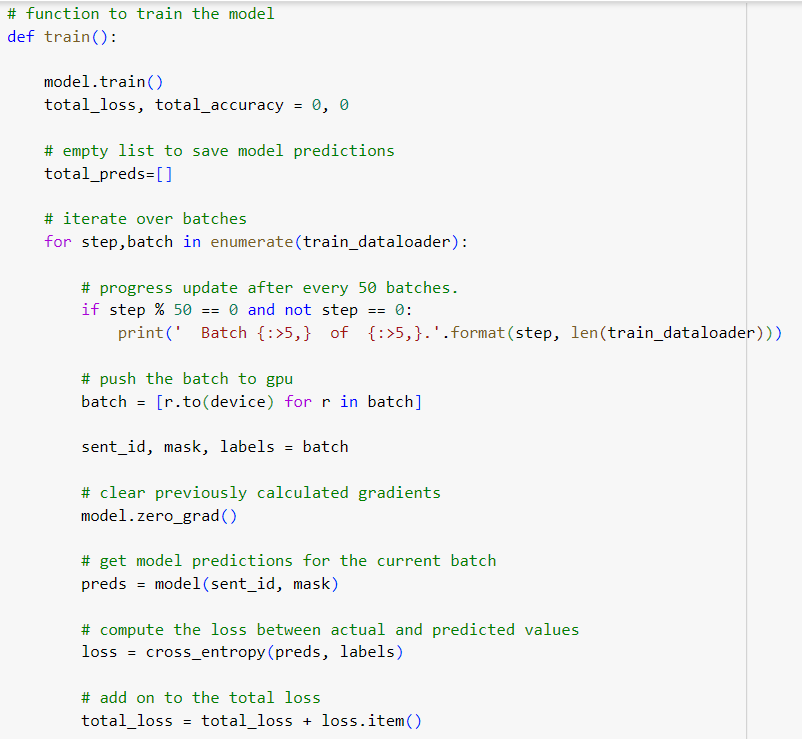
Once pretraining is completed, the next step is fine-tuning the values(Jain,2021). Adam optimiser is used tune the learning rate to update the individual network weights(Brownlee,2021).

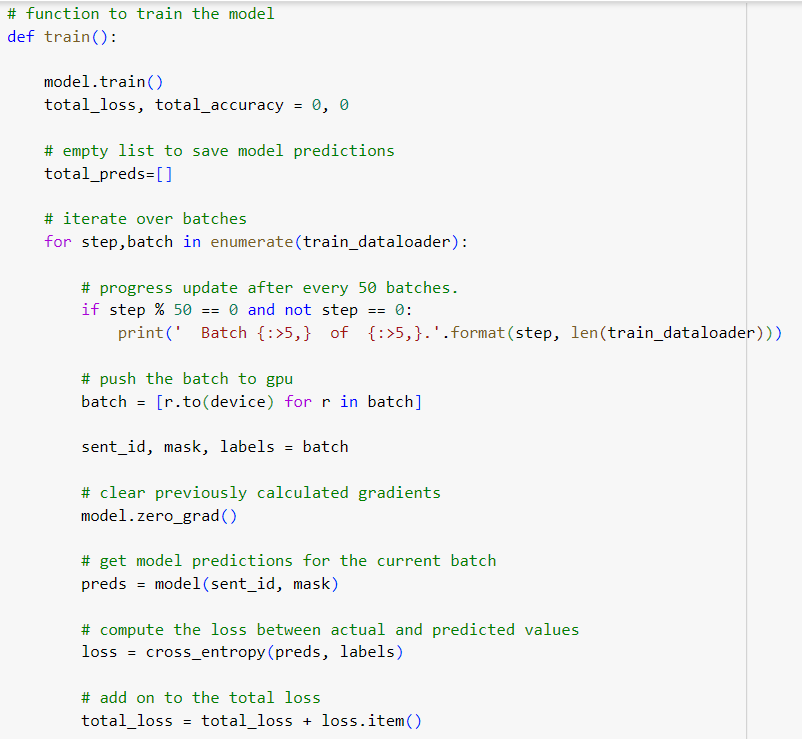


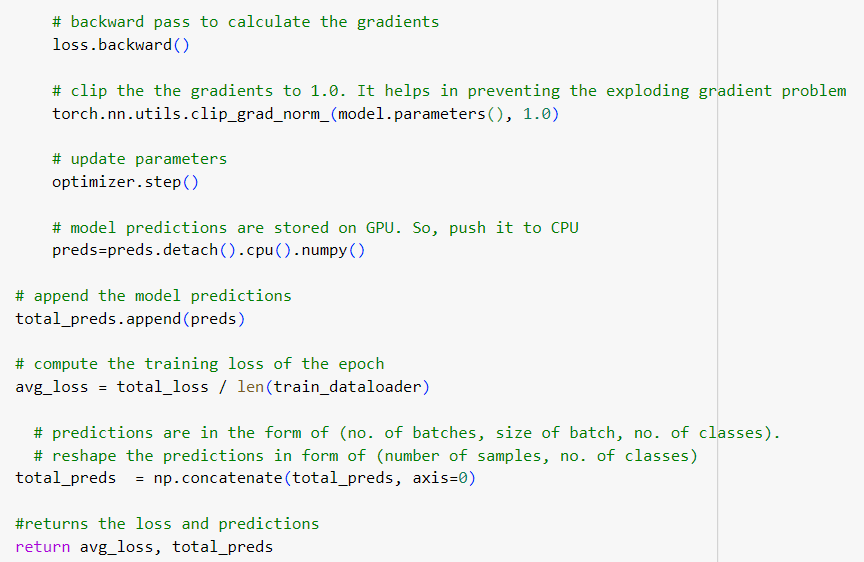


It also define the cross entropy loss function with updated weights(Jain,2021).

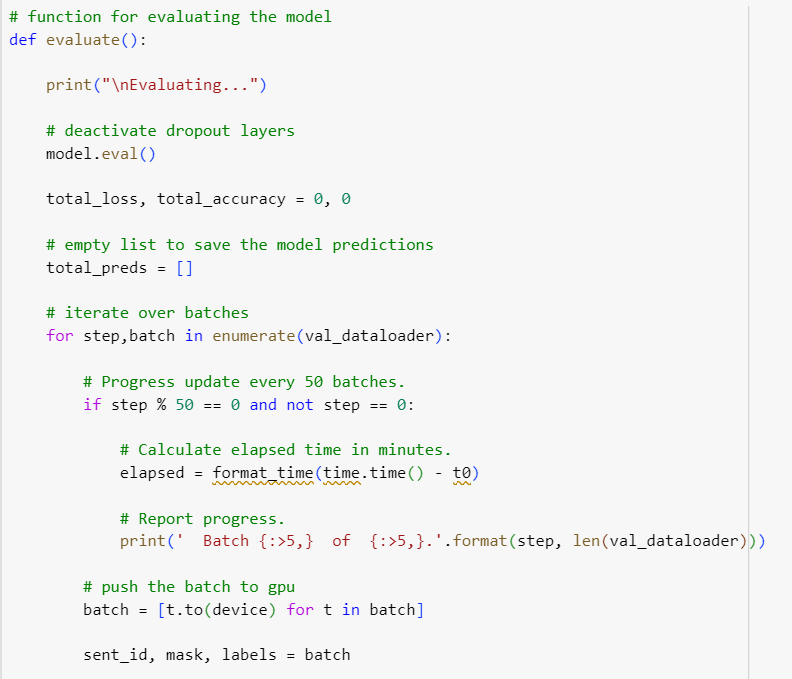


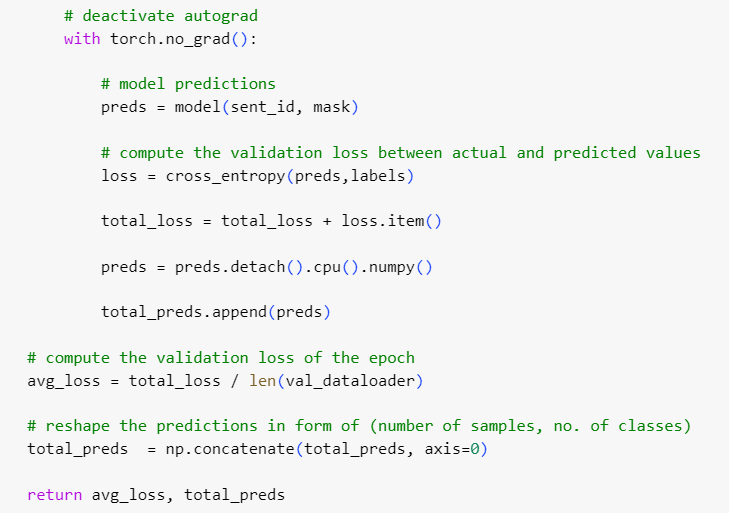




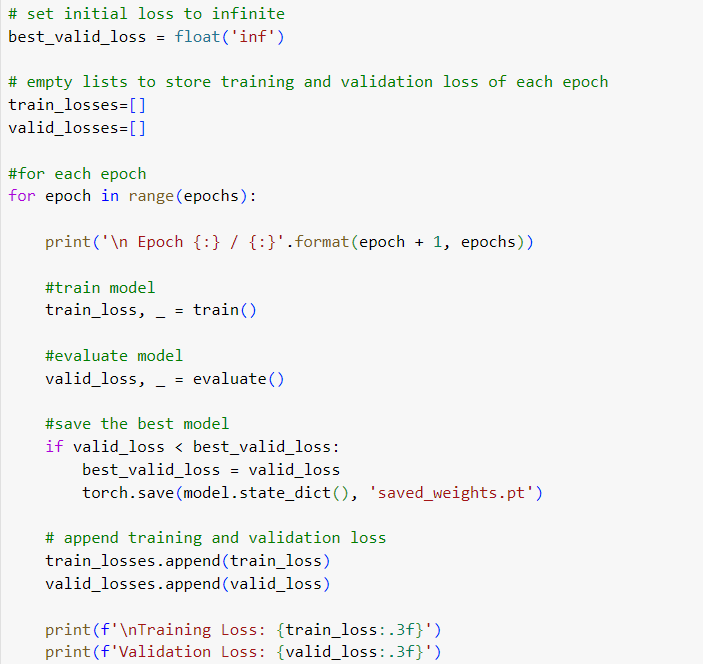


The model is trained with training dataset and after each iteration(epoch), the model accuracy is evaluated(Jain,2021). The progress report is displayed after every 50 batches for monitoring the model(ibid).

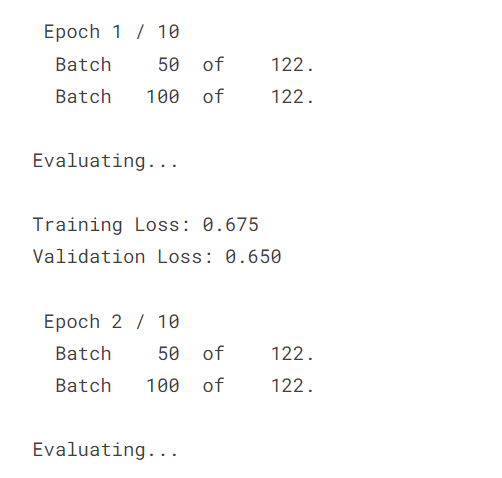




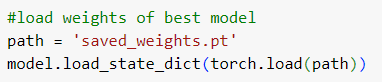
Also, it calculates the training and validation loss after each iteration(Jain,2021).



Sample output is shown below.



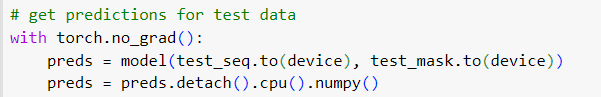
Once the fine-tuning process is completed, the model is loaded with the best weights for performing predictions(Jain,2021).





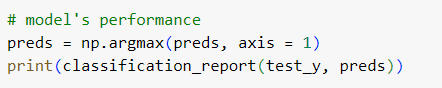
# **Step 10: Model Prediction**

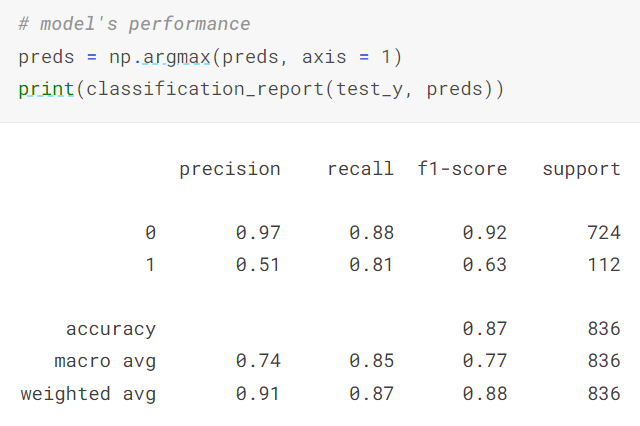
In this phase, model predicts whether a message is spam or not on unseen data(Jain,2021).



# **Step 11: Model Evaluation**

In this step, the accuracy of the model is evaluated(Jain,2021)





The recall score shows 0.88 indicates the percentage of correctly predicting spam messages by the model. The accuracy score indicates the overall correctness of model predictions. Overall, the model performance is good.

# **Conclusion**

This tutorial provided an overview on BERT model. It covered the relevance of the model. It discussed the architecture of Transformers which is used to built BERT. Then, it explained the BERT architecture, the two phases- Pretraining and Fine-tuning. Furthermore, this tutorial briefly explained the fine-tuning of BERT in question answering task. Then provided a brief overview of the real-world applications, benefits and limitations of BERT. Finally, it covered step by step tutorial for building a model using BERT for spam message classification.

# **References**

Alammar, J. (2018a) *The illustrated Bert, Elmo, and Co. (how NLP cracked transfer learning)*, *The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning) – Jay Alammar – Visualizing machine learning one concept at a time.* Available at: http://jalammar.github.io/illustrated-bert/ (Accessed: 05 June 2023).

Alammar, J. (2018b) *The illustrated transformer*, *The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time.* Available at: http://jalammar.github.io/illustrated-transformer/ (Accessed: 04 June 2023).

Ankit, U. (2020) *Transformer neural network: Step-by-step breakdown of the beast.*, *Medium*. Available at: https://towardsdatascience.com/transformer-neural-network-step-by-step-breakdown-of-the-beast-b3e096dc857f (Accessed: 04 June 2023).

Brownlee, J. (2021) *Gentle introduction to the adam optimization algorithm for deep learning*, *MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/ (Accessed: 06 June 2023).

Brownlee, J. (2022) *How to grid search hyperparameters for deep learning models in python with keras*, *MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/ (Accessed: 06 June 2023).

Chakraborty, S. (2020) *Practical uses of bert*, *Medium*. Available at: https://sayanchak.medium.com/practical-uses-of-bert-c384ae3a5c2a (Accessed: 05 June 2023).

Cristina, S. (2022) *The Transformer model*, *MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/the-transformer-model/ (Accessed: 05 June 2023).

Cristina, S. (2023) *Implementing the transformer encoder from scratch in tensorflow and keras*, *MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/implementing-the-transformer-encoder-from-scratch-in-tensorflow-and-keras/#:~:text=The%20Transformer%20encoder%20consists%20of,%2Dconnected%20feed%2Dforward%20network. (Accessed: 31 May 2023).

DeLucia, S.-A. (2023) *Unleashing the power of bert: How the Transformer model revolutionized NLP*, *Arize AI*. Available at: https://arize.com/blog-course/unleashing-bert-transformer-model-nlp/#:~:text=One%20of%20the%20main%20differences,wide%20range%20of%20NLP%20tasks. (Accessed: 05 June 2023).

Devlin, J. *et al.* (2019) *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* [Preprint]. doi:https://doi.org/10.48550/arXiv.1810.04805.

Giacaglia, G. (2019) *Transformers*, *Medium*. Available at: https://towardsdatascience.com/transformers-141e32e69591 (Accessed: 17 May 2023).

Hui, J. (2019) *NLP - Bert & Transformer*, *Medium*. Available at: https://jonathan-hui.medium.com/nlp-bert-transformer-7f0ac397f524 (Accessed: 05 June 2023).

Jain, H. (2021) *Bert for ‘everyone’ (tutorial + implementation)*, *Kaggle*. Available at: https://www.kaggle.com/code/harshjain123/bert-for-everyone-tutorial-implementation/notebook (Accessed: 06 June 2023).

Jain, S.M. (2022) ‘Introduction to Transformers’, in *Introduction to transformers for NLP: With the hugging face library and models to solve problems*. New York, Berkeley: Apress, pp. 19–36.

Kana, M. (2019) *Bert for dummies -  step by step tutorial*, *Medium*. Available at: https://towardsdatascience.com/bert-for-dummies-step-by-step-tutorial-fb90890ffe03 (Accessed: 05 June 2023).

Karim, R. (2023) *Illustrated: Self-attention*, *Medium*. Available at: https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a (Accessed: 03 June 2023).

Khalid, S. (2020) *Bert explained: A Complete Guide with Theory and tutorial*, *Medium*. Available at: https://medium.com/@samia.khalid/bert-explained-a-complete-guide-with-theory-and-tutorial-3ac9ebc8fa7c (Accessed: 13 May 2023).

KiKaBeN (2021) *Transformer’s encoder-decoder: Let’s understand the model architecture*, *KiKaBeN*. Available at: https://kikaben.com/transformers-encoder-decoder/ (Accessed: 04 June 2023).

Kortschak, H. (2022) *Attention and Transformer models*, *Medium*. Available at: https://towardsdatascience.com/attention-and-transformer-models-fe667f958378 (Accessed: 03 June 2023).

Kosar, V. (2021) *Feed-forward, self-attention & key-value*, *Vaclav Kosar Blog*. Available at: https://vaclavkosar.com/ml/Feed-Forward-Self-Attendion-Key-Value-Memory (Accessed: 04 June 2023).

Maxime (2019) *What is a Transformer?*, *Medium*. Available at: https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04 (Accessed: 04 June 2023).

McCormick, C. (2019) *Bert word embeddings tutorial*, *BERT Word Embeddings Tutorial · Chris McCormick*. Available at: https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/#21-special-tokens (Accessed: 06 June 2023).

Padmanabhan, A. (2022) *Word embedding*, *Devopedia*. Available at: https://devopedia.org/word-embedding (Accessed: 02 June 2023).

Persson, S. (2021) *Paper summary - bert: Pre-training of deep bidirectional Transformers for language understanding*, *Medium*. Available at: https://medium.com/analytics-vidhya/paper-summary-bert-pre-training-of-deep-bidirectional-transformers-for-language-understanding-861456fed1f9 (Accessed: 05 June 2023).

ProjectPro (2023) *Bert NLP model explained for complete beginners*, *ProjectPro*. Available at: https://www.projectpro.io/article/bert-nlp-model-explained/558 (Accessed: 15 May 2023).

Ravichandran, A. (2022) *How bert NLP optimization model works*, *How BERT NLP Optimization Model Works*. Available at: https://www.turing.com/kb/how-bert-nlp-optimization-model-works#er:-encoder-representations (Accessed: 17 May 2023).

Saeed, M. (2022) *Transformers: What they are and why they matter - blog*, *Scale Virtual Events*. Available at: https://exchange.scale.com/public/blogs/transformers-what-they-are-and-why-they-matter (Accessed: 04 June 2023).

Saeed, M. (2023) *A gentle introduction to positional encoding in transformer models, part 1*, *MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/a-gentle-introduction-to-positional-encoding-in-transformer-models-part-1/ (Accessed: 01 June 2023).

StackExchange (2020) *How is Bert different from the original transformer architecture?*, *Artificial Intelligence Stack Exchange*. Available at: https://ai.stackexchange.com/questions/23221/how-is-bert-different-from-the-original-transformer-architecture (Accessed: 05 June 2023).

Suresh, S.K. (2022) *Understanding transformers - encoder*, *Medium*. Available at: https://medium.com/mlearning-ai/understanding-transformers-encoder-1f269b1cc943 (Accessed: 04 June 2023).

Turing (2022) *Understanding nvidia cuda: The basics of GPU parallel computing*, *Turing*. Available at: https://www.turing.com/kb/understanding-nvidia-cuda (Accessed: 06 June 2023).

Vaswani, A. *et al.* (2017) ‘Attention is All you Need’, in *Advances in neural information processing systems: 31st annual conference on neural information process ..* Long Beach, CA: NEURAL INFO PROCESS SYS F, pp. 6000–60010.