Attention Mechanisms in Natural Language Processing

JOB VAN DIETEN, University of Twente, The Netherlands

This paper examines the impact of attention mechanisms starting from their initial use cases up to their present-day roles in the field of Natural Language Processing (NLP). Traditional machine learning models struggle to capture contextual dependencies, which have been largely resolved by incorporating attention mechanisms into NLP models. Through an exploration of attention mechanisms, this research offers a comprehensive overview, looking into their evolutionary trajectory, performance enhancements, inherent limitations, and visualization techniques. Key findings highlight the remarkable performance improvements brought by attention mechanisms, particularly evident in tasks like Machine Translation and Sentiment Analysis. Challenges, including computational complexity and interpretability, are discussed, providing insights into the more nuanced landscape of attention in NLP.

Additional Key Words and Phrases: Attention, Deep Learning, Natural Language Processing, Review

1 INTRODUCTION

When processing human language, the individual components of the source text contribute differently, depending on the task that has to be done. Certain words may be important in one instance, but may not matter in another. This becomes problematic for machines, as they have trouble discerning context and prioritizing certain information. For us humans, this is no problem, as we prioritize information in a dynamic and nuanced way.

Traditionally, NLP models used a combination of feature extraction and a classifier, to classify text [25]. The main limitation in these models' accuracy lies in the feature extraction process. The problem with these processes, particularly Bag-Of-Words, is that they treat each input element as a single thing, resulting in a loss of its context, and thus treating every word with the same level of significance.

As an example, consider the sentence *"The bank was crowded, so I had to stand in line"*. In this sentence, the word *"bank"* holds a dual meaning; namely one as a financial institution and the other as a physical space. The early NLP models mentioned earlier, struggle with this. They used to process each word in isolation, ignoring the word's context within a sentence [25].

While the feature extraction processes worked for their time, they were still limited across many NLP tasks. Word2vec [37] and GloVe [42] were introduced as advanced embedding techniques to address some of the limitations of the traditional feature extraction processes, such as semantic meaning, by providing continuous, distributed representations of words. They capture semantic relationships and consider contextual information, allowing for more effective modeling of language semantics and improving the performance of all kinds of NLP tasks.

TScIT 40, February 2, 2024, Enschede, The Netherlands

Ultimately, these early approaches to understanding text fail to emulate the way we humans process language. People understand text by reading individual words, whose meaning and context are incrementally integrated with the preceding context, to finally build up the sentence's meaning [16]. This difference between how the traditional machine learning models process text and how we humans understand text becomes especially evident in Natural Language Processing (NLP), where precise comprehension and interpretation of text are crucial.

To address this inherent difference, Deep Neural Networks (DNNs) emerged as a potential solution. One of the first successful attempts were Recurrent Neural Networks (RNNs) [44]. They excel in capturing sequential dependencies in language since they process sequences by maintaining hidden states that carry information from previous steps to the current one. While they showed promise in capturing temporal dependencies, they are not capable of handling long-range dependencies between words. These networks also suffer from the vanishing gradient problem [19], further limiting their effectiveness in practice.

Despite the advancements, these early DNN structures struggled to capture the nuanced and hierarchical relationships that exist in natural language. The breakthrough came with the introduction of attention mechanisms.

Attention was first introduced to aid in the machine translation of text by Bahdanau et al. [5]. They found that the translation from a model utilizing a form of attention performed significantly better than just the conventional encoder-decoder model at the time. The results showed that the use of attention improved their model's performance by about 50%. This paved the way for the widespread adoption of attention mechanisms across the field of NLP.

This paper aims to elucidate the history of how attention mechanisms have enhanced a model's understanding of text. Section 6 provides insights into the historical development of attention mechanisms. In Section 7, the performance improvements that have happened over the last decade are explored, since the introduction of attention. Furthermore, in Section 8, inherent challenges and drawbacks associated with attention mechanisms are examined, as well as potential ways to mitigate them. Lastly, Section 9 looks at some common visualization techniques for attention mechanisms.

2 TECHNICAL BACKGROUND

Traditionally NLP models grappled with the challenge of correctly processing human language, where the significance of words varies depending on the context. These models consisted of a feature extraction and a classifier [25], where the feature extraction process was the main limitation in the model's accuracy.

Early Feature Extraction approaches:

- **Bag-Of-Words (BOW):** One of the initial techniques was BOW [66]. This method transforms sentences into a mapping array, linking unique words to their total occurrences

in a dataset. This way, a computer can easily understand the extracted features, but a word's context is lost in the process.
- **N-Grams:** N-Gram models, such as Trigrams, improve on BOW by considering combinations of words and their probabilities of occurrence in the dataset [9]. Essentially, it is a sliding window of size N going over the input. For large datasets, this provided an efficient and quick way to check for spelling errors for its time.
- **Term Frequency-Inverse Document Frequency (TF-IDF):** TF-IDF introduces a numerical representation of term importance by combining local information (term frequency, TF), with global information (inverse document frequency, IDF) [4]. It combines this information to identify terms that are both frequent in a document, as well as across the entire dataset.

In the pursuit of better semantic representation, Word2Vec [37] and GloVe [42] were introduced. Word2Vec generates continuous vector representations of words using either the Continuous Bag-Of-Words or SkipGram model, resulting in dense word embeddings that capture some semantic relationships and linguistic regularities. GloVe, on the other hand, operates globally, constructing a word-to-word co-occurrence matrix to optimize the embeddings, resulting in vectors with linear structures that contain meaningful algebraic operations. Both models meaningfully contributed to advancing NLP models to process human language.

As the NLP models evolved, RNNs emerged to capture sequential dependencies in language by maintaining hidden states that carry information from previous steps to the current one [44]. However, RNNs faced problems with vanishing gradients during learning [19], which limited their practical ability. Long-Short Term Memory (LSTM) networks and Gated Recurrent Unit (GRU) networks were introduced as variations on RNNs to overcome the vanishing gradient problem [19]. These networks use mechanisms to selectively update and forget information, which improves their handling of long-range dependencies.

3 PROBLEM STATEMENT

Within Natural Language Processing, traditional machine-learning models struggle to capture nuanced relationships and the contextual dependencies that exist in human language. This is due to the idea behind these models, as they treat each word as a separate input, consequently losing the important context [25]. For instance, in the example from the Introduction, the word "bank" can refer to either a financial institution or a physical space. The failure to accurately disambiguate such terms leads to misinterpretations of text and thus reduces the model's accuracy in all kinds of NLP tasks, ranging from Machine Translation and Sentiment Analysis to Question Answering.

3.1 Research Question

The problem statement leads to the following research question:

How do attention mechanisms enhance the contextual understanding and performance of natural language processing tasks?

This research question will be answered by the following sub-questions:

1. How have attention mechanisms evolved since their initial use cases?
2. What performance improvements do attention mechanisms offer over traditional machine-learning models?
3. What are the inherent challenges and limitations of attention mechanisms, and how can they be mitigated or addressed?
4. How can the visualization of attention mechanisms help with the interpretability of NLP models?

4 RELATED WORK

In the last decade, numerous studies have looked into the challenges that are associated with traditional machine-learning models. Notably, the work of *Bahdanau et al.* [5] laid the foundation for incorporating attention mechanisms in NLP models, especially focusing on the translation of human language. While not called attention at the time of their paper, the underlying principle is the same. Their model demonstrated a remarkable improvement in translation accuracy compared to the state-of-the-art recurrence-based encoder-decoder models of the time [5].

Building on Bahdanau's work, subsequent research introduced the concept of global and local attention mechanisms, further refining the adaptability of attention [33]. The global attention mechanism allows a model to consider the entire input sequence when assigning weight to different elements, while the local mechanism narrows its focus to a specific region. This increases efficiency and performance in all kinds of sequence-to-sequence tasks.

Furthermore, in 2017 the Transformer model was presented, an advancement that uses newly introduced multi-head attention to attend to differently attend to different parts of the input, along with a self-attention mechanism to process input sequences in parallel so that the limitations of sequential processing could be overcome [58]. Ever since, the Transformer architecture has become a vital component of the modern NLP models, demonstrating a state-of-the-art performance in a multitude of NLP tasks. The self-attention mechanism allows the model to weigh the importance of each word in the context of the entire sequence, close to how we humans understand language.

Attention mechanisms have not only proven to be effective in sequence-to-sequence tasks but also in tasks that require contextual understanding, such as sentiment analysis. In 2016, an attention-based neural network was proposed for sentiment analysis and demonstrated high performance in capturing nuances in sentiment across different inputs [64].

In the context of information extraction, research into more refined transformer architectures stands out, as they employed attention mechanisms to improve the extraction of relevant information from unstructured text [32]. This model showed superior precision in identifying key entities and relationships, highlighting the potential of attention mechanisms in tasks demanding very fine-grained analysis of text.

While models with attention mechanisms have shown great promise, some studies have explored their limitations [50]. A comprehensive analysis of attention mechanisms has been done and

revealed problems with computational complexity, as well as the challenge of interpretability. These findings show that it is needed to think carefully before utilizing attention mechanisms for every model.

5 METHODOLOGY

Due to the inherent limitations of traditional machine learning models in NLP and how attention mechanisms solve many of the problems, the proposed methodology aims to explore the evolution, performance improvements, challenges, and visualization aspects associated with attention mechanisms in a comprehensive way, using literature.

5.1 Literature Search

To ensure a comprehensive review, the literature search was conducted on the following reputable scientific databases: Google Scholar and IEEE Explore.

Papers were retrieved using a systematic search strategy that uses relevant keywords on the scientific databases. Additionally, a snowballing technique was employed, to find relevant papers through citation analysis and references in retrieved papers.

5.1.1 Keywords Selection. To improve the precision and relevance of the review, a set of keywords was used to find relevant papers. The choice of keywords was iterative, refined through trial searches, and changed depending on the terms commonly found in the titles.

The following is a list of used keywords: *"Attention Mechanisms, NLP, Transformer, Challenges in Attention Mechanisms, Self-Attention, Multi-Head Attention, Machine Translation, Sentiment Analysis, Question Answering, Text Summarization"*.

Finding papers by year for the Performance Improvements section slightly breaks away from this, due to the sheer number of retrieved papers. As such, finding the relevant papers is done in a slightly different way. For each of the tasks, the dataset along with the evaluation metric is entered in either of the scientific databases, after which the publication date can be altered to the desired year. This way, a limited number of more relevant papers is retrieved and we can be near-certain that papers presenting new performance improvements were not missed.

5.1.2 Inclusion Criteria. The inclusion criteria for selecting papers are as follows:

- Papers focusing on the evolution, performance improvements, challenges, and visual aspects of attention mechanisms
- Papers published in reputable conferences or journals [1]
- Papers written in English

5.1.3 Exclusion Criteria. Papers will be excluded if they:

- Are not related to attention mechanisms in NLP
- Are not published in reputable conferences or journals
- Are not written in English

5.1.4 Data Collection. The data extraction process will involve extracting relevant information from each selected paper, such as the publication year, key findings, methodologies, datasets used, and performance on the analyzed task.

5.1.5 Quality Assessment. To ensure the credibility of the selected papers, a quality assessment was performed, and factors such as the credibility of the conference or journal were considered.

5.1.6 Paper Selection Process. The selected papers for review were chosen based on their alignment with the research objectives. Specifically, these papers provide valuable insights into the evolution, performance gains, challenges, and interpretability of attention in NLP. The inclusion criteria ensure that the chosen papers are from reputable sources, are directly related to the research focus, and are written in English, to make sure that the reliability and relevance of the information synthesized in this review is of good quality.

6 EVOLUTION OF ATTENTION MECHANISMS

Attention mechanisms have emerged as a transformative solution in NLP, addressing many of the inherent challenges of the earlier machine learning models. The conventional approach of treating each word in isolation is inherently flawed in capturing a word's nuance and context in a sentence. This limitation became especially evident [5] in tasks like machine translation, where the translation of a word heavily depends on the context of the entire sentence.

Before the introduction of attention mechanisms, the conventional machine learning models faced challenges and limitations that reduced their effectiveness in capturing contextual dependencies. One prevalent model for sequence-to-sequence tasks, like machine translation, was the recurrence-based encoder-decoder architecture [55]. In this architecture, an encoder processes the input text into a fixed-size context vector, which is then used by a decoder to generate the output sequence. It showed good performance in machine translation tasks and can handle medium-length sequences and maintain syntactic structure. However, this rigid one-size-fits-all approach fails in handling longer sequences and capturing very small nuances between the words in varying contexts.

In 2014, the field of NLP experienced a huge advancement, as the first attention mechanism was introduced and implemented [5]. They recognized that in translation, the importance of a word may vary significantly based on its context within a sentence. Their insight led to the introduction of a mechanism that allowed a model to dynamically focus on different parts of the source sequence when generating each word of the translation. It was found that this performed significantly better than a state-of-the-art RNN encoder-decoder model of the time. While these results were very promising, the newly proposed model still struggled with the translation of rare words.

After the introduction of the first attention mechanism in machine translation, subsequent research built upon the previous work by introducing Global and Local attention mechanisms [33]. Global attention allowed a model to consider the entire source sequence when assigning weights to the different elements. This made the model have a broader perspective, which facilitated a more comprehensive understanding of the input context to overcome the limitations of the one-size-fits-all approaches seen up to that point.

[1] Conferences are checked for their credibility by their Publication Forum rating, done via https://www.tsv.fi/julkaisufoorumi/haku.php?lang=en

Concurrently, their introduction of a Local attention mechanism allowed a model to narrow its focus to specific regions within the source sequence [33]. This refinement increased the model's efficiency by concentrating on relevant portions of the input, which turned out to be particularly advantageous when processing lengthy sequences. With these new attention mechanisms, they performed significantly better than the other state-of-the-art machine translation models of the time. Furthermore, it was shown that for machine translation, attention-based models are superior to non-attentional ones in many cases, such as translating names and handling long sequences [33].

The next advancement came in the form of self-attention, also known as intra-attention, introduced by Cheng et al. [10]. They recognized the limitations of the existing attention mechanisms in capturing dependencies within a sequence. Traditional attention mechanisms, while effective at attending to different parts of the input sequence, were inherently sequential in their processing. This leads to constraints in capturing more complex relationships and hinders a model's ability to consider all possible positions of the input sequence.

Self-attention addresses this limitation by enabling a word to attend to all other words in the sequence simultaneously [10]. Each word in the input sequence could dynamically adjust its attention weights based on its relationship with every other word. This allows the model to process the input in parallel, which not only significantly speeds up computation, but also allows the model to capture long-range dependencies in the input [10].

This laid the groundwork for the biggest advancement yet: the introduction of the Transformer model [58]. This model marked a paradigm shift in NLP architectures. The primary innovation of the Transformer model lies in the incorporation of self-attention, as well as the introduction of Multi-Head attention. Each attention head focuses on a different part of the input, allowing the model to attend to multiple representations of the input sequence simultaneously.

The Transformer architecture proved to be highly scalable and efficient, being able to handle long sequences with ease [58]. It quickly became a cornerstone in NLP, achieving state-of-the-art performance in many NLP tasks.

Expanding on the multi-head attention introduced with the Transformer model, subsequent research has explored its applications in capturing diverse representations in input sequences. It has shown promising results in many NLP tasks, such as machine translation [12], semantic role labeling [54], and subject-verb agreement task [56], among others. The reason for these performance gains is that multi-head attention allows a model to attend to information from different representation subspaces at different positions, from the same input [58]. This flexibility becomes especially beneficial in the aforementioned tasks, where a holistic understanding of language is important for getting good performance.

Inspired by Self-attention, relative attention was introduced to improve the modeling of relationships within the input. Relative attention mechanisms use the positional information of words, to address challenges related to word order and positional encoding [51]. This led to small performance gains in the machine translation task, compared to absolute position representations.

7 RESULTS

The evolution of attention mechanisms, as explored in the preceding section, shows their transformative impact on NLP. From their initial use cases to the more sophisticated adaptations seen in current state-of-the-art models, attention mechanisms are seen as indispensable tools for capturing contextual nuances within input sequences. This section delves into the outcomes of this evolution, by looking at the performance improvements that attention mechanisms have brought to a specific subset of NLP tasks.

As these attention mechanisms evolved, they not only addressed inherent limitations but also showed remarkable enhancements in handling diverse NLP challenges. This section will illuminate some of the advancements achieved by attention mechanisms in a selected set of tasks, showcasing their performance compared to earlier models in tasks such as machine translation, sentiment analysis, text classification, question answering, and text summarization. By looking into these tasks, this section aims to show the benefits that attention mechanisms bring to NLP.

7.1 Machine Translation

Machine translation is perhaps the most popular NLP task. It involves the translation of one language to another, done by a machine translation system. Evaluating the performance of machine translation systems commonly employs metrics like BLEU (Bilingual Evaluation Understudy), which quantifies the similarity between the generated translation and one or more human reference translations [41]. A higher BLEU score generally indicates a better translation quality. For this section, the WMT2014 English-French [7] dataset will be used due to the large number of models that have been tested on it over the years, providing a decade's worth of results from NLP models.

Before the integration of attention mechanisms, machine translation models relied on RNNs with encoder-decoder architectures. In these models, the encoder processed the entire input sequence into a fixed-size context vector, which the decoder used to generate the output sequence [55]. With this approach, a BLEU score of 26.71 was achieved [5].

With the incorporation of attention for the same model architecture, the BLEU score was improved to 36.15 [5]. While not the best performance for that year, this laid the groundwork for future machine translation models to incorporate attention mechanisms, as the performance was greatly improved by just incorporating attention.

A significant improvement was made in 2021, with the introduction of the Re-Transformer [28]. This transformer-based model focused on reducing the number of parameters and modifying the ratio of self-attention to feed-forward layers in the encoder layer. This adjustment was aimed at further increasing the model's understanding of natural language and to improve the translation efficiency. Using this new model architecture, a large improvement was made in the achieved BLEU score, where the score jumped from 46.4 to 55.6. Another advantage of this model comes in its total training time, which is much less than comparable models. Compared to the regular Transformer, the training time was 44% shorter, while achieving a score that is 17.5 BLEU points higher [28].

Table 1. Performance on the WMT2014 English-French Dataset

Year	Model name	BLEU	Reference
2014	LSTM Ensemble + PosUnk	37.5	[34]
2015	"	"	"
2016	Deep-Att. Ensemble + PosUnk	40.4	[67]
2017	ConvS2S (10 models)	41.6	[14]
2018	Noisy Back-translation	45.6	[13]
2019	"	"	"
2020	Transformer + Back-translation	46.4	[31]
2021	Re-Transformer-2	55.6	[28]
2022	X-Transformer	**55.63**	[29]
2023	"	"	"

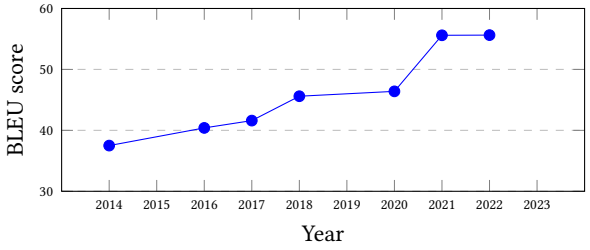

Fig. 1. **Performance over the years of the best-performing model per year.** This figure illustrates the evolution of the best-performing models of the past decade, showing the transformative impact of attention mechanisms and model optimizations, and showcasing the continual progress in this NLP task. If a certain year has no new best-performing model, the year is skipped in the graph.

The current state-of-the-art in machine translation is the X-Transformer model, which is a refinement of the Re-Transformer [29]. The main architecture of this model is similar to that of the Re-Transformer, with minor changes made. With this, a state-of-the-art score of 55.63 was achieved (Figure 1, Table 1).

7.2 Sentiment Analysis

Sentiment analysis is a task in NLP that involves determining the sentiment expressed in a piece of text, typically classified as positive, negative, or neutral. Assessing the performance of sentiment analysis models is usually done via accuracy metrics on standardized datasets. The accuracy metric quantifies the ratio of correctly predicted sentiments to the total number of instances in the dataset. This provides a clear measure of a model's ability to discern and classify sentiments correctly. There are many standardized datasets for sentiment analysis, each with its downsides and benefits. For this section, the *SST-2 Binary Classification* [52] dataset will be analyzed due to the number of models that have been benchmarked with it, and due to there being accuracy metrics of models before the use of attention.

Before attention, the best performance on sentiment analysis was seen with convolutional neural networks [23]. They used pre-trained word embeddings from Word2Vec [37] to improve the accuracy of the final model. In the end, the best-performing model of this era achieved an accuracy of 88.1%.

Table 2. Performance on the SST-2 Binary Classification Dataset

Year	Model name	Accuracy	Reference
2014	CNN-multichannel	88.1	[23]
2015	DMN	88.6	[24]
2016	Neural Semantic Encoder	89.7	[38]
2017	Block-sparse LSTM	93.2	[15]
2018	BERT$_{\text{LARGE}}$	94.9	[11]
2019	T5-11B	**97.5**	[45]

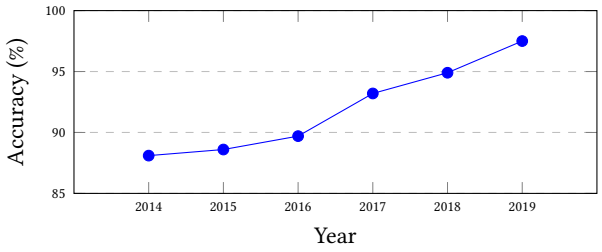

Fig. 2. **Best-performing models over the years.** The figure shows the performance of models on the dataset since its introduction in 2014. Since no model has beaten the performance of T5-11B, the graph only shows data up to 2019.

The first use of attention in sentiment analysis was seen in 2015 when a Dynamic Memory Network employed an attention mechanism [24]. The model was used on a variety of NLP tasks, like Question Answering, Sequence Tagging, and Sentiment Analysis. The accuracy of this model on the *SST-2 Binary classification* dataset came to 88.6%, a 0.5 percent-point increase above the model that did not use attention. While not a large increase in itself, it showed the potential that attention could have in this NLP task.

In 2017, the Block-Sparse LSTM brought a substantial improvement in sentiment analysis, introducing highly optimized GPU kernels for gradient-based learning with block-sparse weights [15]. This innovation allowed the model to scale up to much wider states than typically used in LSTMs, achieving state-of-the-art results for the time.

The current state-of-the-art in sentiment analysis involves transformer-based architectures. They make heavy use of attention mechanisms to perform very well on a variety of NLP tasks. The current highest-ranking model on the *SST-2 Binary Classification* dataset is T5-11B, with an accuracy of 97.5% [45], a substantial improvement over the Dynamic Memory Network that first utilized attention (Figure 2, Table 2). Since this performance in 2019, no other model has beaten the accuracy score of 97.5 on the SST-2 dataset. Models like *RoBERTa Large + MUPPET* have gotten close with an accuracy of 97.4 [1], but none have surpassed T5-11B in the sentiment analysis task.

7.3 Question answering

Question Answering is an NLP task that involves designing models that are capable of providing accurate responses to questions posed in natural language.

Evaluating the performance of QA systems often relies on standardized datasets, with the Stanford Question Answering Dataset

Table 3. Performance on the SQuAD1.1 Dataset

Year	Model name	EM-score	Reference
2016	BIDAF	73.3	[49]
2017	R.M-Reader	82.3	[18]
2018	$\text{BERT}_{\text{LARGE}}$	87.4	[11]
2019	XLNet	89.9	[63]
2020	LUKE	90.2	[61]
2021	$\text{ANNA}_{\text{LARGE}}$	**90.6**	[21]

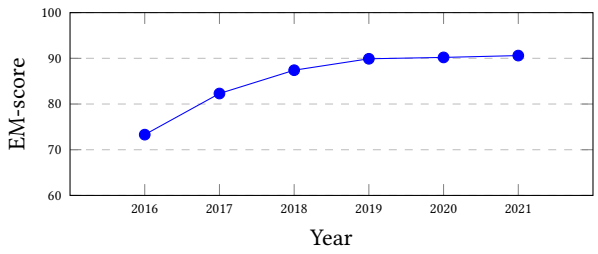

Fig. 3. **Performance of the best QA models over the past years.** The figure illustrates the performance over the last few years. If a certain year has no new best-performing model, the year is skipped in the graph.

(SQuAD1.1) [46] being a widely adopted benchmarking dataset. The dataset consists of questions posed on a set of Wikipedia articles, with corresponding answers. The Exact Match (EM) score measures the percentage of the model's given answers that match the human-annotated answer. This dataset was chosen for comparison due to its diversity, complexity, and real-world relevance.

Given the recency of the SQuAD1.1 dataset, the first models tackling this challenge already incorporated attention mechanisms. In 2016, the year the dataset was released, the best-performing model was BIDAF, which achieved an EM score of 73.3 [49]. This model used a bi-directional attention flow to attend to information from the passage and the question simultaneously. In contrast to other attention mechanisms, this allowed it to capture not only local dependencies within the passage but also intricate semantic relationships between both the question and the passage.

Attention mechanisms as a whole play a crucial role in improving the performance of QA systems. Since the introduction of the SQuAD dataset, every best-performing model by year has utilized a form of attention.

As of the latest advancements in QA, the Approach of Noun-phrase based language representation with Neighboraware Attention (ANNA) model stands as the current state-of-the-art model, with an EM score of 90.6 [21] (Figure 3, Table 3). With this model, a new attention mechanism was proposed; a neighbor-aware self-attention mechanism. This attention mechanism is aimed at mitigating a recognized limitation in traditional transformer encoders, where a single self-attention layer may prove to be insufficient to understand certain nuanced relationships between words [6]. The goal of the neighbor-aware self-attention mechanism is to overcome this limitation by disregarding the diagonality in the attention matrix, meaning that the computed attention weights focus more on other tokens than the token itself.

Table 4. Performance on the GigaWord dataset.

Year	Model name	ROUGE-1	Reference
2015	Abs+	31.0	[48]
2016	MRT	36.5	[3]
2017	FTSum_{g}	37.3	[8]
2018	"	"	"
2019	ControlCopying + BP Norm	39.4	[53]
2020	BART + R3F	40.5	[2]
2021	Pegasus + DotProd	40.6	[22]
2022	GENIE	**45.7**	[27]

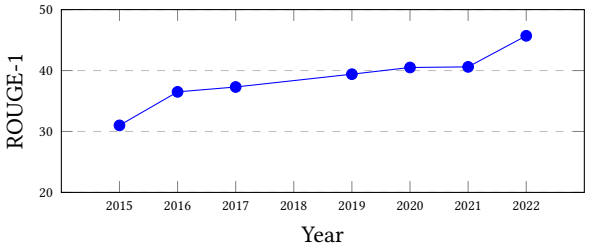

Fig. 4. **Performance of the text summarization models over the last years.** It illustrates the increase in performance of the models operating on the GigaWords dataset. If a certain year has no new best-performing model, the year is skipped in the graph.

7.4 Text Summarization

Text Summarization is an NLP task that involves distilling the essential information from a given text while retaining its core meaning. It is relevant in various domains, aiding in information retrieval, and content condensation.

The evaluation of text summarization models is done by using the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric [26]. This metric assesses the quality of summaries by comparing them to reference summaries. Often, performance is measured in ROUGE-1, ROUGE-2 or ROUGE-L. ROUGE-1 and ROUGE-2 measure the overlap of unigrams and bigrams respectively, while ROUGE-L measures the longest overlapping sequence. This section looks at just the ROUGE-1 metric.

For rigorous evaluation, the annotated GigaWord dataset [40] is commonly used due to its extensive collection of news articles, that encompass a wide variety of topics and writing styles.

The use of attention in text summarization models was rather quick since the first models tested on GigaWord already used attention mechanisms back in 2015. The best-performing model of this time was the tuned Attention-Based Summarization system (Abs+), with a ROUGE-1 score of 31.0 [48]. This model used a local attention mechanism, which allowed the model to generate each word of the output summary conditioned on the input sequence. It proved to be structurally simple, while also scaling well to large amounts of data.

As of the latest advancements in text summarization, the current state-of-the-art is represented by the Diffusion Language model, named GENIE, which achieves an exceptional ROUGE-1 score of 45.7 [27] (Figure 4, Table 4). GENIE introduces a groundbreaking

diffusion language model pre-training framework for text generation, comprised of an encoder and a diffusion-based decoder. This enables it to generate coherent text sequences by gradually transforming sequences of random noise. Another feature of GENIE is the use of cross-attention. This form of attention allows for enhanced interaction between the transformer's encoder and decoder.

8 DISCUSSION

The evolution of attention mechanisms has significantly shaped how NLP models understand human language. Starting from traditional methods like Bag-Of-Words and N-Gram, the field transitioned to more sophisticated methods like Word2Vec and GloVe, to address some of the limitations of traditional models.

The advent of RNNs aimed to capture sequential dependencies, but ultimately lacked practical use cases due to challenges with long-range dependencies and the vanishing gradient problem. Attention mechanisms emerged as a breakthrough, initially applied to machine translation tasks. Soon enough, attention would be adopted in many NLP applications.

With its Self-attention mechanism, the Transformer model revolutionized NLP further, demonstrating scalability and efficiency in handling long input sequences. Subsequent research, inspired by self-attention, introduced variations like sparse attention and co-attention.

Performance improvements across tasks like machine translation and sentiment analysis showcased the improvement brought by the incorporation of attention mechanisms into NLP models. For Question Answering and Text Summarization, the chosen dataset does not provide enough details to compare the attention models to the models before the introduction of attention.

For Question Answering, looking at an older dataset like WikiQA [62] may provide insights into the performance improvements brought by the introduction of attention. The first model utilizing attention on this dataset was an LSTM, with a score of 0.664 [36]. In the same paper, they implemented an LSTM without any attention mechanism, and achieved a score of 0.655, showcasing not much of an improvement. This is backed by Hao et al. [17], who showed that using an attention mechanism for their Question Answering models increased the score by at most 5%, suggesting that attention mechanisms may not be as important for the Question Answering task as initially thought.

Performing the same analysis on the Text Summarization task, research done in 2016 [39] provides useful information. In it, a model was trained with a temporal attention model with their newly introduced CNN/Daily Mail dataset and compared it to a baseline model that did not utilize attention. They found that the model with temporal attention achieved a full-length ROUGE-F1 score that is almost 10% higher than the baseline. It is important to note that the full-length ROUGE-F1 metric was used to evaluate their models, to not unfairly favor long summaries, while not imposing a length restriction.

While the BLEU and ROUGE performance metrics have been widely used in NLP as evaluation metrics, it is important to note that they are not perfect tools for evaluating NLP models. These metrics are based on n-gram overlap, which may not capture nuanced improvements made by models and the incorporation of attention. The main problem with these metrics is that they do not account for word synonyms and their order in the sentence, producing a worse score than otherwise [43].

An important consideration in this paper is the selection of the NLP tasks and the analyzed datasets. The chosen datasets often have inherent biases, and the selected tasks may not fully represent the diversity of real-world applications. Extending the exploration to a more extensive range of datasets and tasks would yield a better understanding of the effectiveness of attention across NLP.

8.1 Challenges and Limitations

Attention mechanisms have significantly advanced the capabilities of NLP models by allowing them to dynamically focus on relevant parts of the input sequence. However, the integration of attention mechanisms introduces challenges and limitations that need to be considered.

8.1.1 Computational Complexity. One of the challenges associated with attention mechanisms in NLP models is their computational complexity. The nature of attention requires the calculation of attention scores for each element in the sequence concerning all other elements. As such, the complexity scales to $O(n^2)$. In large-scale Transformer models, this process becomes resource-intensive, leading to longer training times [60].

One way to mitigate this is to re-implement a version of attention that is of a lower time complexity. A recent study showed an implementation of an attention mechanism that is of linear time complexity and demonstrated that the proposed "EcoFormer" has an on-chip energy footprint reduction of 73%, while only dropping performance by 0.33% [30].

Sparse attention offers another potential solution to the problem. Instead of having each element attend to every other element in the sequence, Sparse attention has a subset of the sequence for each token to attend to [57]. This way, the time complexity ultimately scales to $O(n)$, greatly reducing the number of calculations. However, this form of attention comes with its own problem, as it is difficult to determine what tokens are relevant for the subset that each token may attend to.

8.1.2 Interpretability. While attention mechanisms improve model interpretability compared to the traditional approaches, their black-box nature remains a limitation. Understanding the inner workings of attention, particularly how the model assigns importance to specific elements in a sequence, is a complex topic.

The challenge lies in interpreting attention scores effectively. While it has been shown that the highest-weighted words do have more impact on the model's decisions, the assumption that attention directly corresponds to importance is much more nuanced [50]. Serrano and Smith's comprehensive examination of attention's role in interpretation suggests that attention does not always align perfectly with the significance of elements in a sequence.

8.1.3 Limited Sequence Length. Deep Transformer models utilizing attention may also encounter problems with the vanishing gradient problem when processing long input sequences [65]. As the input

sequence grows in length, gradients can diminish during backpropagation, which hinders the effective training of the Transformer. The vanishing gradient problem is especially pronounced in deep architectures, where the influence of gradients diminishes exponentially with every layer.

One way to deal with this is to use depth-scaled initialization, which scales down parameters by a certain factor depending on the layer [65]. This method addresses the vanishing gradient problem by adapting the initialization of the weights based on the depth of the model. Depth-scaled initialization has the advantage of not having to change the model's architecture, and as such is much easier to implement.

8.2 Visualization and Interpretation

Understanding attention mechanisms usually involves visualizing attention weights to gain some insights into a model's decision-making process. Visualization techniques vary depending on the type of attention that is used. For NLP models using self-attention, attention heatmaps show how much focus the different parts of an input sequence receive during processing [35]. These heatmaps display the calculated attention weights as gradients or colors and provide a view into what the model focuses on.

However, it is crucial to note that while visualization offers insights into a model, it does not provide a comprehensive understanding of a model's decision-making process [20, 50]. Attention weights do not fully explain the reasons behind a model's decision due to the non-linear nature of neural networks. Despite this, it still provides useful in helping to interpret an NLP model by showing what parts of an input sequence the model attends to [5].

8.2.1 Attention Heatmap. An attention heatmap is a common visualization technique used to understand how attention mechanisms distribute their focus across an input sequence [5, 47, 48]. These heatmaps display the attention weights as a gradient of color, giving a visual representation of what information a model attends to. An example can be found in Appendix A.

8.2.2 BertViz. BertViz is a powerful tool for visualizing the multi-head attention weights in a Transformer model [59]. It offers several visualizations of attention, the first one being the *Multi-Head View*, which can be used to interpret how different attention heads contribute to the overall understanding of the input. Besides this, BertViz also offers a *Model View*, as well as a *Neuron View*. The *Model View* shows all attention heads simultaneously, providing a birds-eye view of the underlying attention weights. It shows users how attention patterns evolve through the layers of the model. Finally, the *Neuron View* shows how individual neurons of the model interact with each other to calculate the final attention weights. While the *Multi-Head View* and the *Model View* show what patterns the Transformer learns, the *Neuron View* shows how they learn the patterns. Appendix B shows the visualizations offered by BertViz.

9 CONCLUSION

In conclusion, the trajectory of attention mechanisms in NLP has reshaped how computer models understand language. Beginning with the limitations of traditional feature-extraction approaches like Bag-Of-Words and N-Gram, the field transitioned to more sophisticated methods such as Word2Vec and GloVe, addressing some contextual challenges but still falling short. RNNs initially showed promise in capturing sequential dependencies but struggled with long-range dependencies and the vanishing gradient problem. The pivotal moment came with the introduction of attention mechanisms, particularly highlighted by the impact of the Transformer architecture. Subsequent variations like sparse attention and co-attention, contributed to scalability and efficiency.

However, the story of attention is incomplete without acknowledging the associated challenges and limitations, predominantly interpretability concerns, as other challenges can be mitigated without too much effort. Despite this hurdle, attention mechanisms consistently emerge as valuable tools for NLP models. Performance improvements in tasks like Machine Translation and Sentiment Analysis highlight the positive impact of attention mechanisms. It is crucial to recognize that this paper focussed on specific NLP tasks and datasets, and future research may explore a broader spectrum of applications to improve our overall understanding of how attention mechanisms have helped NLP models.

REFERENCES

[1] Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive Multi-task Representations with Pre-Finetuning. https://doi.org/10.48550/arXiv.2101.11038

[2] Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2020. Better Fine-Tuning by Reducing Representational Collapse. https://doi.org/10.48550/arXiv.2008.03156

[3] Ayana, Shiqi Shen, Yu Zhao, Zhiyuan Liu, and Maosong Sun. 2016. Neural Headline Generation with Sentence-wise Optimization. https://doi.org/10.48550/arXiv.1604.01904

[4] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern information retrieval.* Vol. 463. ACM press New York.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural Machine Translation by Jointly Learning to Align and Translate. https://doi.org/10.48550/arXiv.1409.0473

[6] Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. 2020. On the Ability and Limitations of Transformers to Recognize Formal Languages. https://doi.org/10.48550/arXiv.2009.11264

[7] Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation.* Association for Computational Linguistics, Baltimore, Maryland, USA. https://doi.org/10.3115/v1/W14-3302

[8] Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2017. Faithful to the Original: Fact Aware Neural Abstractive Summarization. https://doi.org/10.48550/arXiv.1711.04434

[9] William Cavnar and John Trenkle. 2001. N-Gram-Based Text Categorization. *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval* (2001).

[10] Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long Short-Term Memory-Networks for Machine Reading. https://doi.org/10.48550/arXiv.1601.06733

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/arXiv.1810.04805

[12] Tobias Domhan. 2018. How Much Attention Do You Need? A Granular Analysis of Neural Machine Translation Architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Association for Computational Linguistics, Melbourne, Australia. https://doi.org/10.18653/v1/P18-1167

[13] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding Back-Translation at Scale. https://doi.org/10.48550/arXiv.1808.09381

[14] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. https://doi.org/10.48550/arXiv.1705.03122

[15] Scott Gray, Alec Radford, and Diederik P. Kingma. 2017. GPU Kernels for Block-Sparse Weights. 3 (2017). https://openai-assets.s3.amazonaws.com/blocksparse/

blocksparsepaper.pdf
[16] Peter Hagoort. 2005. On Broca, brain, and binding: a new framework. *Trends in Cognitive Sciences* 9 (2005). https://doi.org/10.1016/j.tics.2005.07.004
[17] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An End-to-End Model for Question Answering over Knowledge Base with Cross-Attention Combining Global Knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada. https://doi.org/10.18653/v1/P17-1021
[18] Minghao Hu, Yuxing Peng, Zhen Huang, Xipeng Qiu, Furu Wei, and Ming Zhou. 2018. Reinforced Mnemonic Reader for Machine Reading Comprehension. https://doi.org/10.48550/arXiv.1705.02798
[19] Yuhuang Hu, Adrian Huber, Jithendar Anumula, and Shih-Chii Liu. 2019. Overcoming the vanishing gradient problem in plain recurrent networks. https://doi.org/10.48550/arXiv.1801.06105
[20] Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. https://doi.org/10.48550/arXiv.1902.10186
[21] Changwook Jun, Hansol Jang, Myoseop Sim, Hyun Kim, Jooyoung Choi, Kyungkoo Min, and Kyunghoon Bae. 2022. ANNA: Enhanced Language Representation for Question Answering. https://doi.org/10.48550/arXiv.2203.14507
[22] Akhil Kedia, Sai Chetan Chinthakindi, and Wonho Ryu. 2021. Beyond Reptile: Meta-Learned Dot-Product Maximization between Gradients for Improved Single-Task Regularization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic. https://doi.org/10.18653/v1/2021.findings-emnlp.37
[23] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar. https://doi.org/10.3115/v1/D14-1181
[24] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. https://doi.org/10.48550/arXiv.1506.07285
[25] Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2022. A Survey on Text Classification: From Traditional to Deep Learning. *ACM Transactions on Intelligent Systems and Technology* 13 (2022). https://doi.org/10.1145/3495162
[26] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain. https://aclanthology.org/W04-1013
[27] Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Nan Duan, and Weizhu Chen. 2023. Text Generation with Diffusion Language Models: A Pre-training Approach with Continuous Paragraph Denoise. https://doi.org/10.48550/arXiv.2212.11685
[28] Huey-Ing Liu and Wei-Lin Chen. 2021. Re-Transformer: A Self-Attention Based Model for Machine Translation. *Procedia Computer Science* 189 (2021). https://doi.org/10.1016/j.procs.2021.05.065
[29] Huey-Ing Liu and Wei-Lin Chen. 2022. X-Transformer: A Machine Translation Model Enhanced by the Self-Attention Mechanism. *Applied Sciences* 12 (2022). https://doi.org/10.3390/app12094502
[30] Jing Liu, Zizheng Pan, Haoyu He, Jianfei Cai, and Bohan Zhuang. 2022. EcoFormer: Energy-Saving Attention with Linear Complexity. *Advances in Neural Information Processing Systems* 35 (2022). https://proceedings.neurips.cc/paper_files/paper/2022/hash/4310ae054ce265e56d8ea897971149b5-Abstract-Conference.html
[31] Xiaodong Liu, Kevin Duh, Liyuan Liu, and Jianfeng Gao. 2020. Very Deep Transformers for Neural Machine Translation. https://doi.org/10.48550/arXiv.2008.07772
[32] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. https://doi.org/10.48550/arXiv.1907.11692
[33] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. https://doi.org/10.48550/arXiv.1508.04025
[34] Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the Rare Word Problem in Neural Machine Translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China. https://doi.org/10.3115/v1/P15-1002
[35] Xingjing Mao, Li Chen, Shenggen Ju, Xia Wan, and Yuezhong Liu. 2022. *Toward Fact-aware Abstractive Summarization Method Using Joint Learnin*. Technical Report. https://doi.org/10.21203/rs.3.rs-2206382/v1
[36] Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural Variational Inference for Text Processing. https://doi.org/10.48550/arXiv.1511.06038
[37] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. https://doi.org/10.48550/arXiv.1301.3781
[38] Tsendsuren Munkhdalai and Hong Yu. 2017. Neural Semantic Encoders. https://doi.org/10.48550/arXiv.1607.04315
[39] Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. https://doi.org/10.48550/arXiv.1602.06023
[40] Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*. Association for Computational Linguistics, Montréal, Canada. https://aclanthology.org/W12-3018
[41] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02)*. Association for Computational Linguistics, USA. https://doi.org/10.3115/1073083.1073135
[42] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar. https://doi.org/10.3115/v1/D14-1162
[43] Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. https://doi.org/10.48550/arXiv.1804.08771
[44] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and S. S. Iyengar. 2018. A Survey on Deep Learning: Algorithms, Techniques, and Applications. *Comput. Surveys* 51 (2018). https://doi.org/10.1145/3234150
[45] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. https://doi.org/10.48550/arXiv.1910.10683
[46] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. https://doi.org/10.48550/arXiv.1606.05250
[47] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. 2016. Reasoning about Entailment with Neural Attention. https://doi.org/10.48550/arXiv.1509.06664
[48] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. https://doi.org/10.48550/arXiv.1509.00685
[49] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2018. Bidirectional Attention Flow for Machine Comprehension. https://doi.org/10.48550/arXiv.1611.01603
[50] Sofia Serrano and Noah A. Smith. 2019. Is Attention Interpretable? https://doi.org/10.48550/arXiv.1906.03731
[51] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. https://doi.org/10.48550/arXiv.1803.02155
[52] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA. https://aclanthology.org/D13-1170
[53] Kaiqiang Song, Bingqing Wang, Zhe Feng, Liu Ren, and Fei Liu. 2019. Controlling the Amount of Verbatim Copying in Abstractive Summarization. https://doi.org/10.48550/arXiv.1911.10390
[54] Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-Informed Self-Attention for Semantic Role Labeling. https://doi.org/10.48550/arXiv.1804.08199
[55] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. https://doi.org/10.48550/arXiv.1409.3215
[56] Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. Why Self-Attention? A Targeted Evaluation of Neural Machine Translation Architectures. https://doi.org/10.48550/arXiv.1808.08946
[57] Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. 2020. Sparse Sinkhorn Attention. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR. https://proceedings.mlr.press/v119/tay20a.html
[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. https://doi.org/10.48550/arXiv.1706.03762
[59] Jesse Vig. 2019. BertViz: A Tool for Visualizing Multi-Head Self-Attention in the BERT Model. (2019).
[60] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-Attention with Linear Complexity. https://doi.org/10.48550/arXiv.2006.04768
[61] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing (EMNLP). Association for Computational Linguistics, Online. https://doi.org/10.18653/v1/2020.emnlp-main.523

[62] Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal. https://doi.org/10.18653/v1/D15-1237

[63] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. XLNet: Generalized Autoregressive Pretraining for Language Understanding. https://doi.org/10.48550/arXiv.1906.08237

[64] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California. https://doi.org/10.18653/v1/N16-1174

[65] Biao Zhang, Ivan Titov, and Rico Sennrich. 2019. Improving Deep Transformer with Depth-Scaled Initialization and Merged Attention. https://doi.org/10.48550/arXiv.1908.11365

[66] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. 2010. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics* 1 (2010). https://doi.org/10.1007/s13042-010-0001-0

[67] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation. https://doi.org/10.48550/arXiv.1606.04199

A ATTENTION HEATMAP

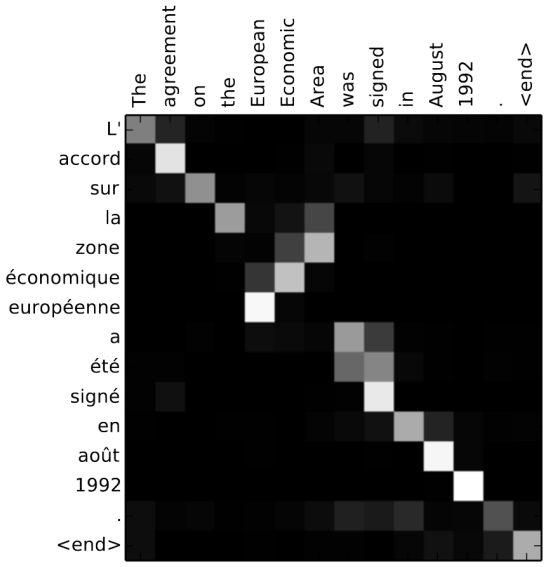

Fig. 5. A heatmap visualization of the attention weights, showing what input tokens were relevant for the given output token [5]. Here, the English sentence is translated into a French sentence.

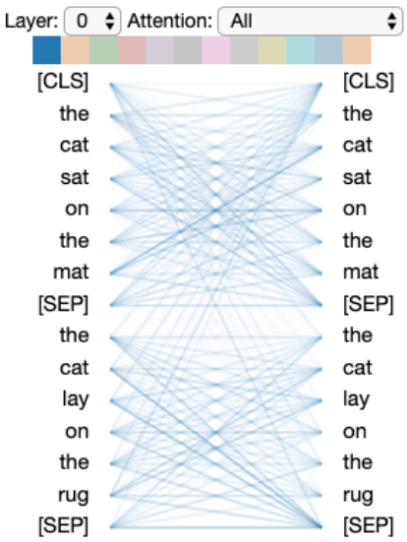

Fig. 6. The Attention Head view offered by BertViz [59]. It shows the attention patterns produced by one or more attention heads. The colors identify the corresponding attention heads, and the lines show the weight of the attention

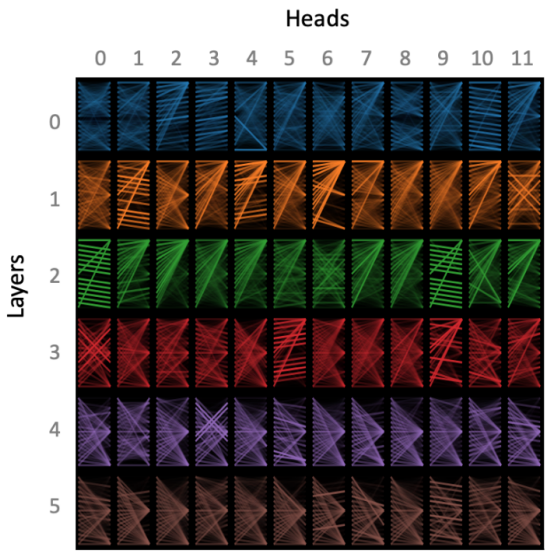

Fig. 7. The Model view offered by BertViz [59]. It shows all the heads and layers simultaneously, providing an overview of the learned patterns.

B BERTVIZ

The three visualizations offered by BertViz are the *MultiHead View* (Figure 6), the *Model View* (Figure 7), and the *Neuron View* (Figure 8).

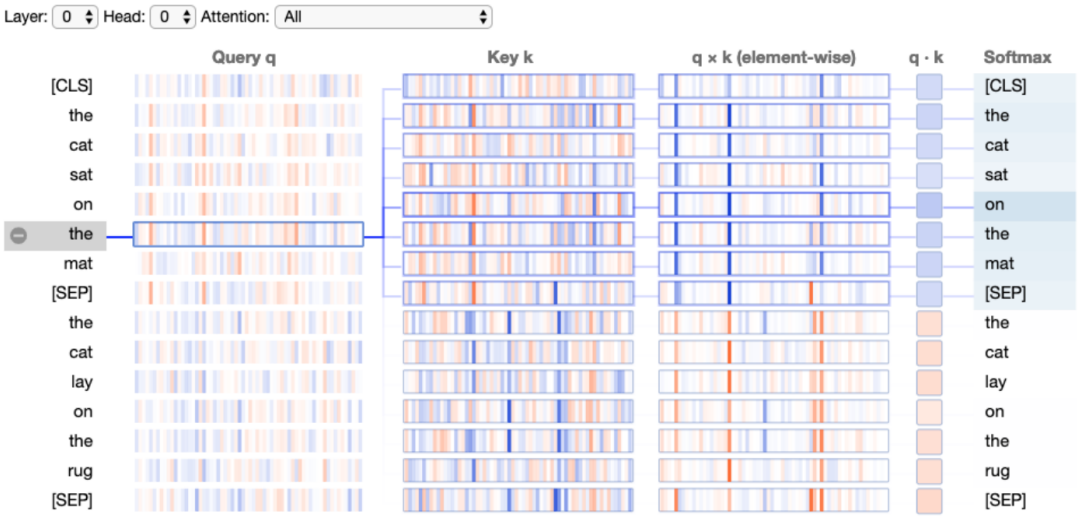

Fig. 8. The Neuron view offered by BertViz [59]. It provides a comprehensive visualization of attention computation in the Transformer model. Focussing on individual neurons in the query and key vectors, this view illustrates their interaction to calculate the attention weights. For a selected token, the computation goes from left to right across several columns, showcasing essential components such as the 64-element query vector (q), the 64-element key vector (k) for each token receiving attention, the element-wise product (q × k), the dot product (q · k) and the softmax of the scaled dot product. Attention weights are visualized through color coding, with blue representing positive values and orange representing negative values. The saturation reflects the magnitude.