International Journal on Artificial Intelligence Tools
© World Scientific Publishing Company

Unexpectedness as a Measure of Semantic Learning when Training Transformer Models

Ricardo A. Calix[*]
Purdue University Northwest
Hammond, IN, USA
rcalix@pnw.edu

Leili Javadpour
University of the Pacific
Stockton, CA, USA
ljavadpour@pacific.edu

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Many problems in NLP such as language translation and sentiment analysis have shown a lot of improvement in recent years. As simpler language problems are solved or better understood, the focus shifts to more complex problems such as semantic analysis and understanding. Unfortunately, a lot of studies in the literature suffer from a too much specificity problem. The algorithms and datasets are too domain specific. In this study, we analyze and elaborate on this notion of generality. Instead of selecting a highly specialized data set for semantic analysis, we take a generic and possibly dry data set, and we study how a plain vanilla Transformer performs in learning higher level semantic patterns beyond what was obvious or expected. We tune our Transformer model on a classic language task to ensure correct performance. Once tuned, the goal is to select sentences with specific key words and study whether higher level semantic patterns may have been learned by our model. We believe that we obtained promising results. The average BLEU score for sentences less than 25 words is equal to 39.79. Our initial qualitative analysis of possible semantic content of interest shows a 17 percent rate in finding interesting semantic patterns. We provide discussion of data driven results of unexpectedness as a measure of semantic learning.

Keywords: Deep learning; Transformers; Semantic Analysis; Attention.

1. Introduction

The field of artificial intelligence (AI) has made a lot of progress in recent years. Many problems such as language translation and sentiment analysis have shown a lot of improvement in recent years [38]. Deep learning has certainly done a lot to help the field of AI and is expected to continue to produce rapid advancement

[*] Corresponding author

in the future. As simpler language problems are solved or better understood, the focus shifts to more complex problems such as semantic analysis and understanding. Computational semantic analysis deals with automated higher understanding of concepts from language with the goal of developing more robust AI. Recently, the Transformer and the Attention mechanism have helped to advance several areas of natural language processing such as question answering, text summarization, language translation, and many more. Machine learning does suffer from some problems, however. A lot of studies in the literature suffer from a "too much specificity" problem. The basic approach these days seems to be to develop elaborate methodologies using data sets from highly specific domains to solve highly specific problems. True AI systems development should be opposite to this approach. In fact, a robust AI model needs to generalize well to be considered to be effective [1]. In this study, we analyze and elaborate on this notion of generality. Instead of selecting a highly specialized data set for semantic or sentiment analysis, we take a generic and possibly dry data set and we study how a plain vanilla Transformer, with a plain vanilla Attention mechanism, performs in learning higher level semantic patterns beyond what was obvious or expected. We tune our Transformer model on a classic language task to ensure correct performance. Once tuned, we select sentences with specific key words (e.g. affect or common sense signals) and study whether semantic patterns were learned by our model. The goal is to understand what patterns it seems to be learning, what errors it makes, and to determine how to improve the Transformer model's results. Along the way, we try to explain the intricacies of the Transformer and Attention mechanisms and how they may tie to semantic or cognitive computing. Another term similar in meaning to semantic analysis is sentic computing. Sentic computing is an approach to NLP that is multidisciplinary. It relates loosely to semantic computing or common sense computing or also to sentiment computing. The term Sentic is derived from Latin. The term comes from 2 Latin words. One related to common sense (Latin: sensus) and the other related to the senses (Latin: sentire). The word "sentire" in spanish is probably "sentir" (as in: "siento hambre" or "siento alegria"). Here, the first just means "I feel hunger" in English, whereas the second one means "I feel happy". Clearly, one is directly related to emotions and the other one is more related to "wants". In [2], using sentic computing as a way to better process and understand human text has been suggested. Sentic computing uses a Common Sense Computing [3, 4] approach to better derive emotion from text. In [3] it was suggested that a sentic computing approach can extract more useful information from short lengths of text (i.e. paragraphs and sentences) compared to the traditional methods involving larger amounts of data (e.g. papers and books). In this study, we propose that Transformer models can be used to address many problems in semantic computing and cognitive computing with some expected degree of success. We use the classic example of machine translation (MT) to discuss the point. In particular, our focus is on the Attention mechanism and its importance to semantic and cognitive computing. Attention is all about ignoring things. This is similar to the notion of common sense in that common sense implies knowing better and only using that which is important. Additionally, traditional neural networks are seen as black box algorithms. The mechanism of attention helps to better explain how neural networks learn and, as such, is consistent with the transparency goal of semantic computing and robust AI. To demonstrate the power of Attention mechanisms, we implemented a Transformer model on the classic language translation problem. Language translation is a well-established problem. Specifically, in this work, we study a Transformer-based methodology to perform English to Spanish translation and its implications for semantic or sentiment analysis. So, if we give language translation data to a Transformer model, does that not mean that we are building a machine translation (MT) model? Usually, one would think yes. However, the point we wish to elucidate is that we do not tell a Transformer to be a language translator. Instead, we tell it to learn patterns from the data. The point we wish to explore is about what additional patterns the Transformer picks up that could be considered of interest for semantic understanding (i.e. inductive bias). As part of our analysis, we perform quantitative and qualitative language analysis as well as semantic analysis, to identify interesting observations about how the Transformer model learns. The goal of this work is to better understand what patterns the model seems to be learning, and what errors it makes. Many papers in the literature such as [5], for instance, focus on just the model characteristics, and quantitative aspects of the Transformer-based model results such as the BLEU score. Fewer works also include a qualitative analysis. In our work we perform both a qualitative and a quantitative analysis to try to gain better insights about how Transformers work, and how they learn. We use the Europarl corpus of English-to-Spanish sentences [6]. To tune the transformer, we use the standard metric of the BLEU score [7] to evaluate the translation models. Our average BLEU score for sentences less than or equal to 25 words is equal to 39.79 which is consistent with the state of the art as of this writing. Results of our qualitative semantic analysis are encouraging. Our initial analysis of possible semantic content of interest shows a 17 percent rate in finding interesting semantic patterns on a subset of interest (candidates obtained by simple word lookups). The contributions of our work include the following: 1) we applied a Transformer model to the English to Spanish translation task and found that it performs very well, 2) we performed a qualitative analysis of the sentences the model generated and found that the translations had interesting semantic meaning, and 3) we propose that semantic meaning can, in some way, be measured by the unexpectedness (probability) of the words used by the Transformer to generate the translations. Finally, the results of both the quantitative and qualitative analysis are presented and discussed.

2. Literature Review

An important measurement of AI effectiveness was introduced in [1]. In essence, their conceptualization indicates that we measure the effectiveness of AI by its ability to generalize. Specifically, they define intelligence as a measure of an agent's ability to effectively achieve goals in a wide range of environments (Eq. 1).

$$\gamma(\pi) \sim \sum_{\mu \in E} 2^{-K(\mu)} V_{\mu}^{\pi} \tag{1}$$

where $\gamma(\pi)$, as a measure of intelligence, is a function of a policy (π). Where a policy determines what action to take in a given state. The measure of intelligence γ is expressed as the sum over environments $\mu \in E$. So it is the sum over all computable environments. The value V_{μ}^{π} determines the value of success. That is, the value (success) that policy (π) obtains in environment (μ). The complexity penalty $2^{-K(\mu)}$ is a weighting term. The value $K(\mu)$ is the Kolmogorov complexity of the environment μ. Basically, if $K(\mu)$ is low, then $2^{-K(\mu)}$ is high. And if $K(\mu)$ is high, then $2^{-K(\mu)}$ is lower. As such, this definition gives more weight to simpler environments, and gives less weight to more complex environments. In general, there are more complex environments than simpler ones, so it also serves as a normalization. In the previous section we defined that semantic or sentic computing deals with automated higher understanding from many mediums such as text. We propose that Transformer models can be used to address almost every problem in sentic, semantic or cognitive computing with some expected degree of success. We use a classic example of machine translation to discuss the point. There are some studies that have used Transformers or Attention for semantic or Sentic computing. In [8] an attention based model to identify Self-Deprecating Sarcasm (SDS) in text such as tweets on Twitter is developed. Their model, CAT-BiGRU, uses a bi-directional GRU with a convolution and several attention layers. When compared to current state-of-the-art models, the authors found that their experiments performed considerably better in the detection of SDS. In [9] a multi-head attention based model, referred to as FP2GN, is created for aspect-based opinion summarization. Similar to [8] , they have implemented a bi-directional RNN based system with the addition of multi-head attention layers. When compared to baseline data, the authors found that their model outperformed the baselines suggesting that, with sentic computing methods, accurate and readable summaries could be generated. In [10] the “Natural Language Decathlon” (decaNLP) was proposed which is a benchmark that casts a question-answering format for a suite of ten NLP tasks: question answering, machine translation, summarization, natural language inference, sentiment analysis, semantic role labelling, relation extraction, goal-oriented dialogue, semantic parsing, and commonsense pronoun resolution. The Natural Language Decathlon also uses a multitask question answering network to learn all the tasks jointly and simultaneously tackle all of them at once. In [11] the zero-shot learning capabilities of language models was evaluated by feeding some input to the model as a prefix and then autoregressively sampling an output. The authors demonstrated that using generative pre-training of a language model on a unlabelled diverse corpus and then fine-tuning for specific tasks will show promising results. Recently, many language tasks used Transformer models such as in [5] to solve many problems. This work has established state of the art results. Transformers are very interesting and seem to be very powerful. Many researchers suggest that they are better than RNNs for NLP because they parallelize better and because of the Attention mechanism. So far, Transformers have been used to develop very impressive implementations such as BERT [12], GPT-2 [11] , and GPT-3 as of this writing, which seem to be very good at language understanding. Transformers have been applied to language translation [5] , question answering [13] , document summarization [14], automatic text generation [15], creation of commonsense knowledge bases, image classification [16], and long sequences processing [17], to name a few. In the future, Transformers will no doubt be applied more extensively to semantic computing related applications.

2.1. *Overview of Transformers*

This section assumes an understanding of Transformers and Attention. The emphasis, therefore, is only in pointing out specific parts which may be important for semantic analysis. For more detail see [5] or [18]. Transformers, as introduced in [5], are sequence to sequence models. They are much more complicated, with deeper and more resource intensive networks. The Encoder Decoder with Multi Head Attention Transformer (used in this work) is a very deep network (Figure 1).

The architecture has an encoder followed by a decoder. The encoder and decoder have 6 sublayers each called encoder or decoder layers. Each encoder layer has a Multi-Head Attention layer followed by a standard fully connected feed forward layer (Figure 2). The input to the encoder goes through all these layers in the encoder and is converted into an encoder output. The “encoder output” is an example in semantic or Sentic computing of top-down analysis because some meaning of the input (e.g. English sentence) is encoded in the “encoder output”. The input to the encoder and the output of the encoder have the same dimensions.

The decoder layer has 2 inputs. One input is the encoder output. The second input to the decoder varies based on whether you are training or predicting. If you are training, the input to the decoder is the other sequence (e.g. sentence in the other language). In the decoder, when training the Transformer, a mask is needed to prevent the model from seeing all the words it is trying to predict. This is called a look ahead mask. If you are testing, the input to the decoder is the encoder output and just the previously predicted words before the word you are trying to predict. You start with a start of sentence token (e.g. <sos>) and predict iteratively. The predicted word is then added to the previous input tokens and the process is repeated. The attention mechanism in Transformers is the heart of the whole algorithm. Both the encoder and decoder layers have attention mechanisms (Figure 2 and Figure 3). The attention mechanism is nothing more than a dot product matrix multiplication between all the words in a sentence (e.g. the input English sentence). The idea is that, given the input and output, the model learns to correlate the words in the sentence to determine their importance.

There are many types of attention mechanisms. Some of the most important

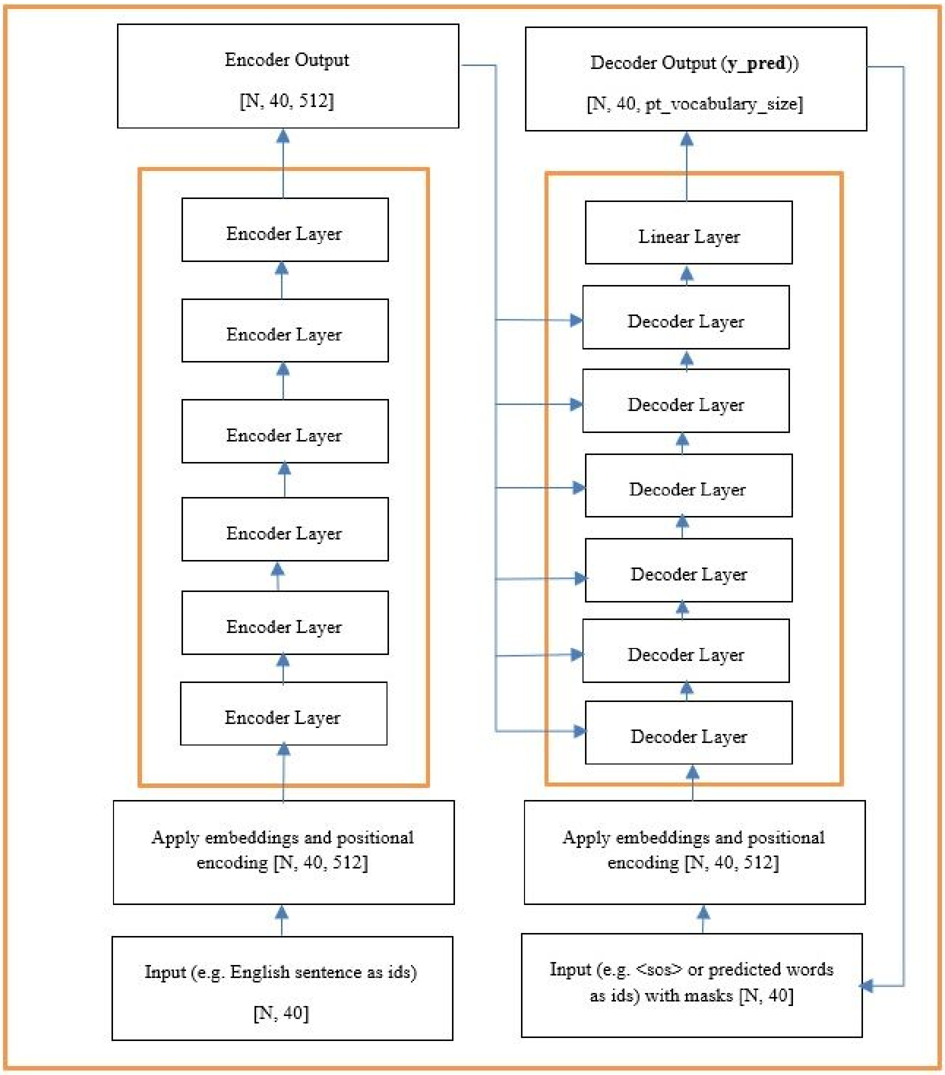
Encoder Output
[N, 40, 512]
Decoder Output (**y_pred**))
[N, 40, pt_vocabulary_size]
Encoder Layer
Encoder Layer
Encoder Layer
Encoder Layer
Encoder Layer
Encoder Layer
Linear Layer
Decoder Layer
Decoder Layer
Decoder Layer
Decoder Layer
Decoder Layer
Decoder Layer
Apply embeddings and positional encoding [N, 40, 512]
Apply embeddings and positional encoding [N, 40, 512]
Input (e.g. English sentence as ids)
[N, 40]
Input (e.g. <sos> or predicted words as ids) with masks [N, 40]

Fig. 1. Transformer Model [18]

types include: location based attention, associative attention, and introspective attention. Location based attention is one of the most basic types. As its name implies, it calculates scores based on position in sequences. Usually, the values are on the matrix diagonal. The most popular type of attention, as of this writing, is the associative or content based attention [19]. This is the standard attention used in Transformers and the one that is used in this paper. Introspective Attention can best be understood in the context of Neural Turing Machines (NTM) proposed in [20]. The NTM architecture mimics the ability of a Turing machine [21] to read

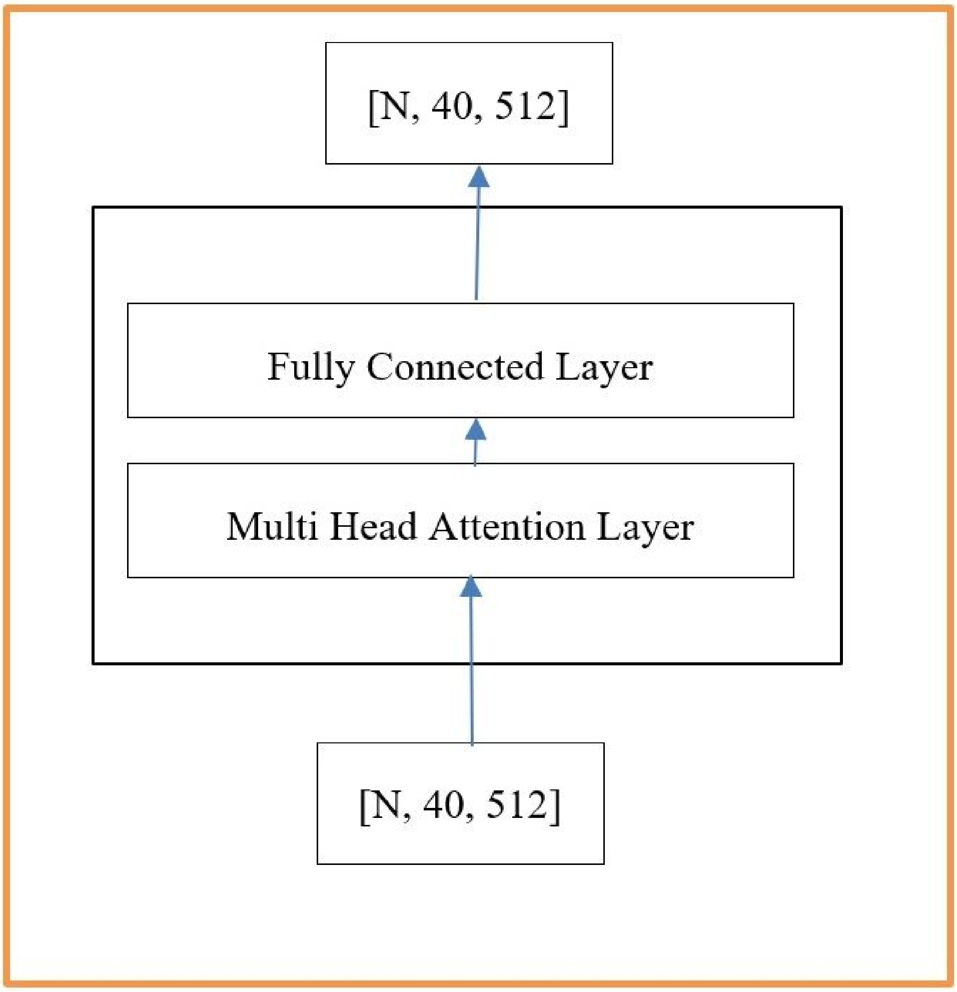

Fig. 2. Encoder Layer [18]

and write to a memory tape. Via a neural network, the NTM selects portions of the memory to pay attention to. If you define “memory” as attention through time, then the NTM can use an Attention mechanism to look at specific parts of its memory as needed. Importantly, in the NTM, memory (the tape), and computation (neural networks) are separate elements. And the attention mechanism helps the neural network to select from memory. In [20] it was shown that the NTM can learn basic algorithms (e.g. code) such as a simple copy algorithm. Therefore, NTM could possibly be used for implementing First Order Logic [22], etc. An extension of the NTM is the Differentiable Neural Computer (DNC) by [23]. Instead of being used to represent code or first order logic like in the NTM, the DNC can be used to

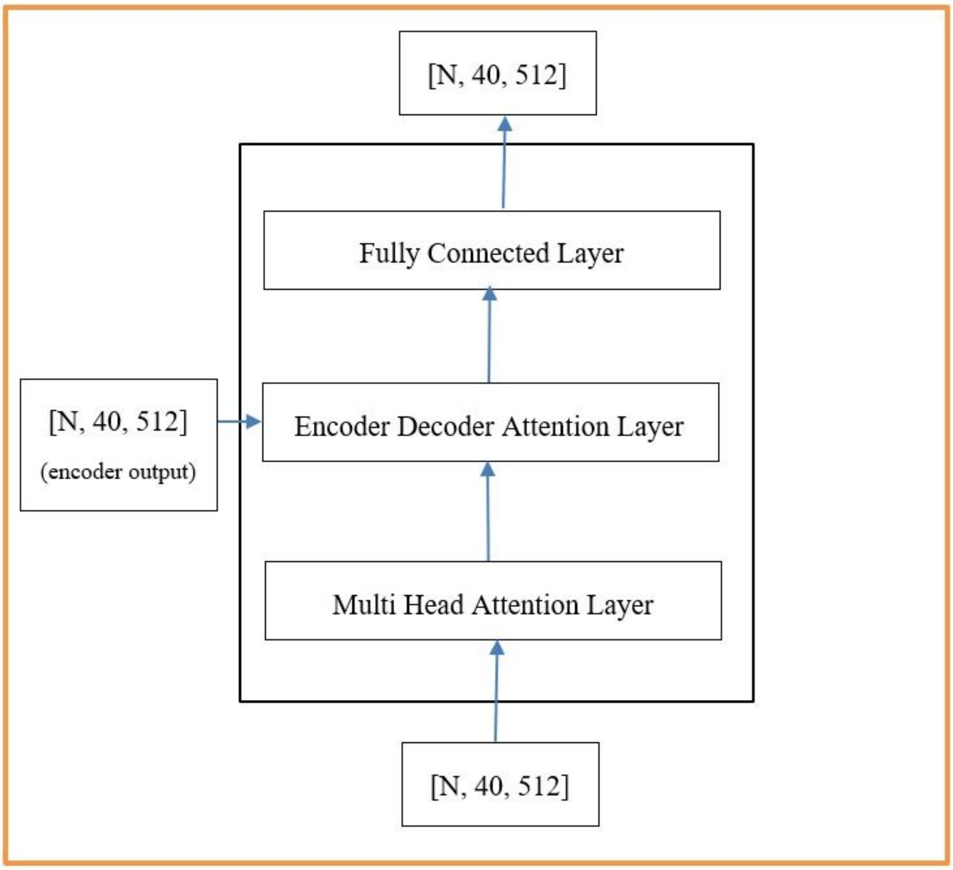

Fig. 3. Decoder Layer [18]

represent and process graphs. Many problems in cognition and semantic computing are represented as graphs (i.e. links and nodes). Universal Transformers [24] have introduced recurrence. Cognitively speaking, this could be compared to how a human spends an amount of time thinking about something (e.g. the number of times the algorithm attends to something). In our work we use the associative attention mechanism [19] . However, it is clear that all these variations focus on higher level semantics and could therefore provide more powerful capabilities. As a side note, in neural networks, a Jacobian matrix can also be used as a whole neural network wide attention mechanism [25]. In the associative or content based attention [19], the goal is to attend to the content that you want to look at. In this type of attention, the attention parameter emitted by the network is a key vector. That key vector is then compared to all the elements in the input data using some similarity (e.g. a dot product). This is then normalized with a softmax function and that gives you the attention weights. Intuitively, you input some key (a query) and you look through everything in the data to see which parts best match that key. The fact that this is a key (query) means that we can match to anything like a concept, common sense knowledge, an emotion, etc. As a result, you get back an attention vector that tells you which parts of the data correspond more closely to the key. This allows you to construct a type of natural lookup between many keys (queries) and the data. An extension of this approach is to split the data into key-value pairs (k, v) and use the keys to define the attention weights and the values to define the readout. In the context of a lookup, you use the key (k) to search and (v) is what you get out (the readout). Mechanically, first we calculate the keys (K), values (V), and queries (Q). This is what is needed to calculate and use the Attention mechanism (Eq. 2).

$$AttentionScores = softmax(\frac{Q * K^T}{\sqrt{d_k}}) * V \tag{2}$$

The K, V, and Q symbols are nothing more than tensors that are the result of a matrix multiplication between the input ([N, 40, 512]) matrix multiplied by a respective weight matrix. The dimension of the resulting Q, K, and V is now 64 and not 512. You reduce it and perform the attention 8 times. Then you concatenate the 8 results and get back the original dimensionality of 512 (8*64). You calculate a score of word i's importance to all other words in the input sentence by effectively doing a dot product. That is the magic of the attention mechanism and consistent with the definition of bottom-up analysis in Sentic computing (e.g. kernel learning, dot product, etc). The intuition is that at the sentence level, every word in the sentence looks at every other word, for the given problem (e.g. translation, semantic understanding, etc.), and during the learning process it assigns weights to words that are important given other words. That is why it is called Attention (as in paying attention). The queries (Q) tensor is matrix multiplied by the transpose of the keys (K) tensor. The resulting attention scores matrix is of size [N, len, len]. Where “len” is length of sentence. We don't really want the network to pay attention to the padding so we are going to mask it with the mask of size [N, len]. We multiply the padding by -1e9 which is basically negative infinity. What this does is that when we run the result tensor through the softmax function, the softmax function will make those infinities close to zero which will allow us to ignore the padding. The last step is to matrix multiply the values (V) tensor with the scores matrix after running it through a softmax and a dropout layer. The resulting tensor is of size [N, 40, 64]. Remember that this is done 8 times. That is why the attention mechanism is called the 8 headed attention mechanism. The attention layer consists of 8 parallel attention sub layers that are later concatenated. The intuition is that each of these 8 layers can learn something new and different. So this gives more capacity to the network to learn semantics. For instance, here, the input would be the English sentence. Six identical encoder layers are created in the encoder where the outputs of one become the inputs of the next layer. The dimensions of all inputs and output at this stage are the same (e.g. [N, 40, 512]). The encoder layer consists of the multi-head attention segments followed by the fully connected layer segment (Figure 2). The input sentence and the padding mask go into the encoder Multi-head Attention 8 times, in parallel, and the results are concatenated. The researchers in [5] found that they could reduce the dimensionality of these 8 attention heads to improve performance. The dot product attention calculation is a bottleneck in Transformers. The residual is the output of the 8 headed Multi-Head Attention. This residual becomes the input to the fully connected layer. The result of the fully connected layer is once again added to the original input and normalized. The output of the encoder is a tensor of size [N, 40, 512]. This will be the input to the decoder and is also called the encoder output. The intuition here is that you took English sentences and you encoded them with this scheme. The "encoder output" is an example in semantic computing of top-down analysis because some meaning of the input (e.g. English sentence) is encoded in the encoder output. The last part of the encoder is the fully connected layer. The Feed Forward layer provides for non-linearity using RELU and changes the representation of data to a higher dimension of 2048 before going back down to 512. The encoder output is passed to the decoder and a similar process happens. Here, though, the decoder layer uses 2 consecutive attentions (Fig 3) and the output of the decoder is predicted words (one at a time in sequence). To predict, you start with a start of sentence token (e.g. <sos>) and predict. The predicted word is then added to the previous tokens and the process is repeated. The decoder consists of 6 decoder layers, followed by a linear layer (Figure 1). Each decoder layer has a decoder multi-head attention layer, followed by a decoder-encoder attention layer, and a fully connected layer (Figure 3). The attention layers consist of 8 parallel attention sub layers that are later concatenated (like in the encoder). There are now in the decoder layer 2 attention mechanisms instead of just one. Basically, one Attention mechanism is for the decoder input which is the spanish sentence and corresponding padding and look ahead masks. The second attention mechanism is for the encoder output and the output from the first attention mechanism (plus corresponding masks). Just like with the encoder there are 6 decoder layers where the outputs of one layer become the inputs of the next layer. Unlike the encoder, the decoder has one last layer. This final layer (Figure 1) maps a tensor of size [N, 40, 512] to a tensor of size [N, 40, vocab size] where vocab size is the size of the vocabulary. This is what allows us to select the predicted word.

2.2. *Embeddings*

Embedding converts the sequence of ids (e.g. words) of the inputs into a sequence of embeddings much like word2vec [26, 27]. The dimensions are learned by the model and can represent many semantic and cognitive concepts. Given that the words are represented by vectors (much like in word2vec), the dimensions in the vectors can represent concepts such as common sense, meaning, sentiment, etc. Therefore, the attention mechanism can serve as a type of semantic construct enhanced by the embeddings. In fact, in the context of neural networks, attention is a method that can mimic a type of cognitive attention. After converting the sequence of ids (words) into a sequence of embeddings, you will go from a 2d tensor to a 3d tensor of size: (N, Seqlenmax, EmbeddingDimension), where N is the batch size, Seqlengthmax is the length of a sentence (e.g. maximum number of words allowed in a sentence), and embedding dimension is n for R^n. The embedding vectors of size "n" are learned by the model so initially they are random data.

3. Methodology

To demonstrate the power of attention mechanisms, we implemented a Transformer model on a generic language translation problem and performed semantic analysis. In this section, the methodology for our research is discussed. We used the Europarl corpus [28] of English and Spanish sentences. We used a Transformer model based on the attention mechanism to train and test the model. For tuning purposes we measured the BLEU score which is the standard metric for language translation tasks. Finally, we search for semantic pattern learning examples as part of the qualitative analysis and measure results. The code and data set are available online [29] . The implementation was done on the Tensorflow API on a GPU box with an Nvidia RTX 2080 Ti (11 VRAM), AMD processor with 12 cores, and with 128 GB of RAM. The Transformer uses batches of size 64, an embedding layer of size 512, and maximum sentence length of 40.

3.1. *Loss Function*

The loss function of the Transformer is similar to other cross entropy based loss functions except that we want to use a mask and that the y_pred and y_real tensors have different dimensions. We mask the loss incurred by the padded tokens to 0 so that they do not contribute to the mean loss. Therefore, we want to ignore padding when calculating the loss function. For this loss, y_real is of size [N, 40] and y_pred is one hot encoded of size [N, 40, vocab_size], which is big. Conveniently, Tensorflow has the loss function: Sparse softmax cross entropy with logits. This function by definition takes tensors with different dimensions as inputs. Once the loss is defined, we can specify the training function which uses the Adam optimizer. In [5] the authors recommended to use a certain set of parameters for Adam as follows: beta1=0.9, beta2=0.98, epsilon=1e-9.

3.1.1. *Input and Outputs*

In deep learning, you usually use to 2 tensors for the input (features) and output (classes). But with the Transformer we have 3 (a, b, c). The best way to understand this is to think of the translation problem. In translation, we have, for example, an English sentence and a Spanish sentence. The English sentence is the input to the encoder (a). The spanish sentence is what you want to predict (c). Therefore, (c) should be the data we feed to the loss function to be compared with what we predict (y_pred). So, why do we need a third tensor ("b")? The third tensor (b) is the decoder input and it is a bit more complex. The simplest explanation is that the decoder input (b) is also the spanish sentence. So "b" and "c" are both the spanish sentence but on one sentence the words are all shifted to the left. The spanish sentence or the batch (of 64 spanish sentences), call it batch_sp, is divided into batch_sp_inp and batch_sp_real for the decoder. The tensor batch_sp_inp is passed as an input to the decoder. The batch_sp_real is that same input shifted by 1 and only used by the loss function. It is compared to y_pred. At each location "i" in each sentence in batch_sp_inp, the tensor batch_sp_real contains the next token that should be predicted.

3.2. *Data*

The dataset used is Europarl which is a set of translated sentences from the European parliament (Europarl). The details about the dataset can be obtained from [6]. The Europarl English-to-Spanish dataset consists of about 2,000,000 sentence pairs. The train set has about 1,965,734 English to Spanish pairs. After removing sentences longer than 40 tokens, and further processing, the data set used contained a train set with around 1,300,000 samples and a test set with around 32,000 samples. We used all 1.3 million sentences for training. The BLEU scores were calculated on the 32,000 sentence pairs test set. The qualitative semantic analysis was also done on the test set.

3.3. *Metrics*

The standard metric to evaluate language translation tasks is the BLEU score. According to [7] , the BLEU score can be calculated by multiplying a brevity penalty factor by the modified precision values of a corpus. In [30] several smoothing techniques have been discussed that can be applied to BLEU scores to help with some of the original inconsistencies that existed with using the original BLEU score. We have applied smoothing function 4 to the BLEU score.

3.4. *Final Details*

In this work we developed a Transformer using TensorFlow 2.0 based on [5]. The parameters of the model included a batch size of 256, a dff of 1024, a d_model value of 256, and number of heads equal to 8. The Transformer model trained for about 12 hours. Each batch of 256 pairs took about 22 seconds to run. The Transformer was run on a computer with 1 RTX 2080 ti with 11 vram model GPU, on Python 3.7, and Linux.

4. Analysis and Discussion

In this section, the results and analysis are presented and discussed.

4.1. *Quantitative Language Translation Analysis (Tuning)*

Examples of predicted Spanish sentences can be viewed at [29]. The BLEU scores average for all translations (31,238 pairs) was equal to 33.80. The BLEU scores average for all translations where real Spanish sentences were less than 40 tokens and English sentences were not equal to nothing (26,066) was 36.7244. The Table below shows length statistics of predicted sentences (Table 1).

Table 1. Length statistics for predicted sentences.

Metrics	Values
max	41
min	2
average	25.41
median	25

The Transformer seems to learn an optimal predicted sentence size and it seems predisposed to predict sentences up to a certain length. Using 25 as the sentence cutoff given that, in general, it usually stops before reaching 40, the average BLEU score for sentences less than 25 (17,654 pairs) was equal to 39.7982 (Table 2).

Table 2. BLEU scores.

	quantity	BLEU
All translations	31,238	33.80
sent <= 40 and eng sent != none	26,066	36.72
Sentence length <= 25	17,654	39.79

Table 3 shows BLEU scores that have been achieved by others using various methods. In [31] the authors measured if translating into a language was harder than out of the language. Their study did not include a qualitative analysis and mostly focused on metrics. They used 190,733 sentences for training and 2,000 for testing. In [32] the authors trained with another dataset of 21 million sentence pairs but tested on Europarl. Their study included a speech component and was mainly quantitative. In [33] multilingual machine translation is focused on zero shot prediction. They trained on 0.6 million samples and tested on just 2,000 sentences. In [34], the authors trained using syntactic data about a sentence as well as the sentence itself. They trained with 170 thousand samples and tested with about 10 thousand samples. This study is particularly interesting. They used POS (Part of Speech) tagging to enrich the sentences. However, one can easily see how an emotion tagger, for instance, could have been used instead. In [35], the authors took sentences and converted them into abstract meaning representations before feeding them to the transformer for translation. All of these studies either used the standard Transformer model we used or slightly modified versions of it. Of note, are the studies that enriched the original sentence. In our study, we did not use byte pair encoding(BPE) [36] which many of the other studies did use. We would expect that BPE would improve our results. Our study's test size was also much bigger than most of these other studies. We, therefore, consider our results to be consistent with the results of these other studies and the state of the art.

Table 3. Other BLEU Scores

english to spanish	BLEU
Currey et al. (2019)	43.10
Liao et al. (2021)	34.98
Iranzo-Sánchez et al. (2020)	48.20
Fan et al. (2020)	25.10
Bugliarello et al. (2020)	50.20
Our study	39.79

4.2. *Qualitative Language Translation Analysis*

Out of a total of 26,639 sentence pairs under study, we randomly selected and analyzed 453 sentence pairs for our qualitative language translation analysis. We made a few initial observations. The translation of the sentences from English to Spanish, in general, was very good. One important observation we noticed is that our Transformer does not do so well with sentences that are very long (e.g. that have 70, 90, or even 100 terms). In these cases, the Transformer, it seems, learns a basic length for average sentences and has a probability to end predicting terms for sentences after a certain length is reached. For example, sentence (12743), which has more that 70 terms, should be, in English as follows.

> The next item on the agenda is the recommendation for a second reading (A4-0151/98) on behalf of the Committee on Economic and Monetary Affairs and Industrial Policy, on the common position adopted by the Council with a view to the adoption of a directive of the European Parliament and the Council amending Directive 83/189/EEC, relating to the provision of information in the field of technical standards and regulations (C4-0035/98-96/0220(COD)) for the third time (Rapporteur: Mr Hendrick).

The human translation was:

> De conformidad con el orden del día, se procede al debate de la recomendación para la segunda lectura (A40151/98), en nombre de la Comisión de Asuntos Económicos y Monetarios y de Política Industrial, sobre la posición común aprobada por el Consejo con vistas a la adopción de la Directiva del Parlamento Europeo y del Consejo que modifica por tercera vez la Directiva 83/189/CEE (C4-0035/98-96/0220(COD)), por la que se establece un procedimiento de información en materia de normas y reglamentaciones técnicas (Ponente: Sr. Hendrick).

But the Transformer provides only the following translation

> <sos>El punto siguiente del orden del día es la recomendación para una segunda lectura ( ) , en nombre de la Comisión de Asuntos Económicos y Monetarios y Política Industrial , sobre la posición común aprobada por .

As can be seen, the Transformer's predicted sentence is shorter. We found this to be a common pattern. In fact, Table 1 shows the Transformer's predicted sentence average length statistics. A summary of other observations is provided in Table 4 as follows (see [29] online for examples of translation by reference number).

Table 4. Comments and Observations

Additional observations
Generally speaking, short sentences, are translated correctly. Example sentence: 3737
Translations sometimes are missing names, numbers, or acronyms (Example sentences: 8, 20).
Sentences that are considered as very good translations are those where the Transformer omits a word or adds a new word, or where it says things in a different way without changing the meaning of the original sentence (Example: 25499). We will elaborate on this further from a semantic analysis perspective.
In general, the translation of sentences from English to Spanish was excellent or very good.
Examples of excellent translations include sentences: 29, 30, 1845, 1846, 3636, 6069, 7274, 9094, 11526, 12720, 12745, 14647, 16641, 16642, 18884, 20132, 23013, 23920, 25455, 25490.
Examples of a very good translations include: 1840, 25499.
There are some very long sentences with 70, 90, or more than 100 words where the Transformer does not finish translating the whole sentence. It translates the first part very well but it then just stops. We believe this is an effect of the sequence size being set to 40. Example 12743.

One other observation, is that it sometimes seems to be missing words from its vocabulary. To assess this issue, we tried the analysis two times with a vocabulary size of 12,000 words and with a vocabulary size of 36,000 words. The results were similar. Unique and infrequent words may be harder to predict. We observe that

Transformer models can capture syntax, grammar, and polarity really well. For example, in: “The iphone 12 is nice but expensive” versus “The iphone 12 is expensive but nice” (Example from Sentic.net). We can see that both sentences can be interpreted as having different polarity or sentiment. On one sentence, the intent to buy the iphone is stronger than in the other. The Attention mechanism has shown to be very good at correctly catching word order. So, in a sense, this is not a worry of this study.

4.3. *Discussion and Semantic Analysis*

Intuitively, we can define a generalization of Equation 1, to measure the level of intelligence for a Transformer model applied to multiple domain problems as follows

$$\gamma(A) \sim \sum_{\tau \in E} 2^{-K(\tau)} V_{\tau}^{A} \tag{3}$$

where, for convenience, we have changed some parameters to reflect the specifics of Transformer models. In particular, we replaced the policy π, in Equation 1 with the attention mechanism (A) defined in Equation 2. We can think of a Q-table, for instance, as analogous to the attention mechanism. The environments set (E) now represents the different problems τ that Transformers can be applied to (e.g. translation, question answering, inference, text summarization, etc.). And to calculate the score (V), in a supervised learning problem, given an attention and Transformer architecture, applied to a specific problem τ, we can use the metric for the specific problem if it exist (e.g. BLEU, ROGUE, precision, recall, F-measure, etc.). We define a supervised learning problem in this context as one where we have a data set with real input sequences, real output sequences, and predicted output sequences (For example, the problem discussed in this paper of English-to-Spanish Translation from Europarl). In this context, so far, we can address supervised learning and problems $\tau \in E$ for which we have metrics like BLEU and ROGUE. Our semantic analysis focuses on looking for evidence of higher semantic understanding by the Transformer. So, what is higher semantic understanding? This is obviously a subjective task and we started by defining what that could be. We concluded that we would look for examples that seemed interesting from the point of view of semantics and that appeared to be beyond simple translation patterns where we could find more richness of meaning. Still, how does a human interpret this description? In Table 6 we provide some examples of cases that we considered as having more interesting semantic understanding or richness of meaning. Part of the focus ended up being mostly on emotion. However, it is important to point out that this is just an example case to let the data drive our analysis. To obtain the semantic interest candidates, we looked at our test set of translated sentences and we focused on the original English sentence and the predicted Spanish translation (given that this is what the Transformer learned). We picked a set of words related to emotions or common-sense words (Table 5) and performed look ups of our test samples. After gathering the example sentence pairs via key word search, we expected the number of cases of semantic interest to be very low or non-existent. We were actually surprised to find more than expected.

Table 5. Look up words

Type of words	Word examples
Emotion	Feel, sense, sad, happy, mad, angry, surprised, neutral, scared, fear
Common sense	Sensible, judgement, logic, prudent, sagacity, wit, acumen, ingenious, reasonable, practical, prudent

In this way, we easily obtained a small number of candidates to analyze qualitatively. We found 17 percent of the candidates to be of interest and that we thought had semantic richness based on our definition (see Table 6). This resulted in 17 percent of the candidates being of interest. This result was actually quite unexpected and we anticipated finding zero or close to zero candidates of interest. Given that we only performed the word lookup on less than 10 percent of our test set, we could anticipate that we can actually find many more cases of semantic interest even in a supposedly very dry corpus of EU parliamentary discussions. Our future work and efforts will no doubt focus on finding more samples and possibly creating a sub corpus of semantic or emotional samples from Europarl (e.g. Senti-Europarl). The list of words for our lookup methodology included emotion terms but also other terms we deemed common sense. It is important to note that it was easier to find emotion related semantic candidates than common sense related semantic candidates. Therefore, our 17 percent rate for just emotions could be higher. In our example case, we then looked at the pairs of interest to see why we had selected them (e.g. why a human intuitively picks them). Interestingly, 2 patterns emerged. It seems we selected these pairs for only 2 reasons and one of the reasons (or patterns) was much more common than the other. The 2 patterns the humans used to select candidate pairs were: (a) the Transformer inferred in the translation unexpected Spanish words from the English sentence (or, more generally, produces unexpected results), and (b) the Transformer translated words that were completely out of order but were the translation was still semantically correct. Examples of case (a) include: 9, 117, 212, 1059, 1535, 271, 272, 1295, 1270, 24, 146, 30782. Examples of case (b) include: 977, 24. See Table 6 for ids.

We found it interesting, that what we loosely defined as “find pairs of semantic interest” almost completely boiled down to “the Transformer infers unexpected words” or sometimes “infers sentences very much out of order but correct”. In both cases, the predicted keywords are unexpected or surprising. This insight is important. So, how can we quantify our insight from our example case that “semantic interest to a human” ended up meaning “predicted translations that are unexpected or surprising”? It is well known that languages such as English and Spanish have

Table 6. Examples of Semantic Interest

ID	Original English Sentence	Predicted Spanish Translation
9	Mr President, less than three weeks ago an important peace agreement was reached in Northern Ireland, which we should all be very happy about. We should congratulate ourselves and, especially, the people of Northern Ireland as well as all the political actors who have had the courage to reach this agreement, including the former British Government under Mr Major, the current one under Mr Blair, and the Governments of the Republic of Ireland.	<sos>Señor Presidente , hace menos de tres semanas se llegó a un importante acuerdo de paz en Irlanda del Norte , que todos deberíamos celebrar con gran alegría en el marco de los Gobiernos de Irlanda del Norte .
117	It is fair to say, as expressed by Mo Mowlam, that the immense welcome the agreement has received from our friends across the world and in particular from within the European Union underlines its truly historic and momentous nature.	<sos>Es justo decir , tal como se ha expresado en este debate , que la inmensa alegría recibida por nuestros amigos de todo el mundo y , en particular , desde la Unión Europea subraya su carácter histórico y .
212	Mr President, I have to say that I am very pleased that the leadership of my own group has asked me to speak for the group today because I am quite sure that the vast majority of the people of Northern Ireland would want me to speak for them today to express their deep gratitude to all of the European institutions for the powerful strength and support that they have given to the peace movement on our streets.	<sos>Señor Presidente , tengo que decir que me alegra mucho que el liderazgo de mi propio Grupo me haya pedido que hable hoy por el Grupo porque estoy totalmente seguro de que la inmensa mayoría de los ciudadanos .
1059	For once we can be delighted with this, and I am pleased with the extremely favourable response that the Commiss'on's recommendation for the 11 Member States has received.	<sos>Por una vez podemos felicitarnos por ello y me alegro de la respuesta extraordinariamente favorable que ha recibido la recomendación de la Comisión para los 11 Estados miembros .
1535	Mr President, ladies and gentlemen, I am happy to be able to explain to you the Commiss'on's position on the overall policy it is pursuing, or attempting to pursue, with regard to the textiles and clothing industry.	<sos>Señor Presidente , Señorías , me complace poder explicarles la posición de la Comisión sobre la política global que sigue o intenta seguir la ropa de vestir .
271	So it is easy for them not to fear peace or democracy today.	<sos>Por lo tanto , es fácil que no se teme hoy la paz ni la democracia .
272	As the President of the Commission, Mr Santer, said here: there is no need to be frightened of peace.	<sos>Como ha dicho aquí el Presidente de la Comisión , Sr. Santer , no hay que tener miedo de la paz .
1295	At the same time, the new European society also has to adapt with almost breathtaking speed to the new technologies which surprise us daily and which call for a radical change in the traditional concept of work and business.	<sos>Al mismo tiempo , la nueva sociedad europea también tiene que adaptarse con una velocidad casi increíble a las nuevas tecnologías que nos sorprenden todos los días y que exigen un cambio radical en el concepto tradicional .
1270	We know that we can only follow this approach through to ensure a high level of employment, as is rightly required and emphasized by Article 2 of the Treaty on European Uni–n - as well as through the employment chapter in the Treaty of Amsterdam, the resolutions of the Employment Summit –nd the strategies during the British Presidency of the Council and the forthcoming Austrian Presidency which we hope will point the w–y - if, together with structural policy and measures in the labour market, education and training policies, a further basket of measures makes economic growth of 3 to 3.5 possible.	<sos>Sabemos que sólo podremos seguir este enfoque si la Presidencia del Consejo de Asuntos Económicos y Monetarios se propone , como es lógico , en las decisiones de empleo y en el capítulo de la política de .
977	I therefore believe that these peoples, who no-one consulted on this monetary union, will soon make the weight of their own opinion felt.	<sos>Por lo tanto , creo que estos pueblos , que nadie ha consultado sobre esta Unión Monetaria , pronto harán sentir el peso de su propia opinión .

specific Probability Distributions [37]. So, assuming that we know these probability distributions, we can use them to calculate likelihood probabilities of producing a Spanish word in a sentence given an English word in a sentence. In Table 7 we can see an example of this.

Table 7. Example Conditional Language Translation Probabilities

	gran	alegria	celebrar
very	0.8	0.1	0
happy	0	0.9	0.2
congratulate	0	0.1	0.3

From our insight we know we need a metric that can measure unexpectedness or surprise. Again, in [37], we know that Shannon Information can be thought of as a measure of information content or surprise (Equation 4). Intuitively, the amount of information you get increases as the probability of the event gets smaller. We use this to motivate a metric. Additionally, we know that for language problems we can get probabilities from the corpus. In this case, we are interested in the probabilities to identify the unlikely words, unlikely word orderings, etc.

$$H(X) = \sum_{x \in X} p(x) log_2 \frac{1}{p(x)} \tag{4}$$

In Equation 4, H is information content or unexpectedness. And from here we can generalize a metric for various unexpected cases such as unexpected words, unexpected word order, etc. (Equation 5). We can call the metric Semantic Unexpectedness Score (SUS).

$$SUS = \sum_{i \in U} w_i U_i \tag{5}$$

where U is the set of all possible unexpected situations, w_i is an associated weight for each U_i. U can be defined by (Eq. 6) where P is the unexpected probability per word in sentence “s”, and w represents each word in sentence (s).

$$U = \frac{1}{\sum_{w \in S} P_w} \tag{6}$$

This semantic unexpectedness measure can be included in Equation 3 to train generalizable Transformers that can learn semantics.

5. Conclusions

In this paper we have discussed Transformer and Attention models in the context of language translation and semantic analysis. The Transformer seems to learn an optimal predicted sentence size and it seems predisposed to predict sentences to a certain length. The average BLEU score for sentences less than 25 words is equal to 39.79. Our initial analysis of possible semantic content of interest shows a 17 percent rate in finding interesting semantic patterns. We observed that unexpectedness has some relation to semantic learning. Our future work will focus on testing efficient ways of incorporating the findings from our qualitative analysis into the overall Transformer model training process.

References

1. S. Legg, and M. Hutter, *Universal intelligence: A definition of machine intelligence* (Minds and machines, 17(4), pp.391-444. https://doi.org/10.1007/s11023-007-9079-x, 2007).
2. E. Cambria, A. Hussain, C. Havasi, and C. Eckl, C., *Sentic computing: Exploitation of common sense for the development of emotion-sensitive systems* (In Development of Multimodal Interfaces: Active Listening and Synchrony (pp. 148-156). Springer, Berlin, Heidelberg. 2010).
3. E. Cambria, A. Hussain, C. Havasi, and C. Eckl, *Common sense computing: From the society of mind to digital intuition and beyond* (In European Workshop on Biometrics and Identity Management (pp. 252-259). Springer, Berlin, Heidelberg, 2009).
4. Y. Ma, H. Peng, and E. Cambria, *Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM*, (In Thirty-second AAAI conference on artificial intelligence, 2018).
5. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin, *Attention is all you need*, (In Advances in neural information processing systems (pp. 5998-6008), 2017)
6. P. Koehn, Europarl: A parallel corpus for statistical machine translation, In MT summit (Vol. 5, pp. 79-86), 2005.
7. K. Papineni, S. Roukos, T. Ward, and W. Zhu, Bleu: a method for automatic evaluation of machine translation, in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311-318, 2002.
8. A. Kamal, and M. Abulaish, Cat-bigru: Convolution and attention with bi-directional gated recurrent unit for self-deprecating sarcasm detection, in *Cognitive Computation, pp.1-19, 2021.*
9. A. Kumar, S. Seth, S. Gupta, and S. Maini, Sentic computing for aspect-based opinion summarization using multi-head attention with feature pooled pointer generator network, *Cognitive Computation*, , pp.1-19, 2021.
10. B. McCann, N. Keskar, C. Xiong, and R. Socher, The natural language decathlon: Multitask learning as question answering, *arXiv preprint arXiv:1806.08730, 2018*
11. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, Language models are unsupervised multitask learners, *OpenAI blog* 1(8), p.9, 2019.
12. J. Devlin, M. Chang, K. Lee, and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805.
13. T. Shao, Y. Guo, H. Chen, and Z. Hao, Transformer-based neural network for answer selection in question answering, *IEEE Access*, 7, pp.26146-26156, 2019.
14. K. Kreimeyer, M. Foster, A. Pandey, N. Arya, G. Halford, S. Jones, R. Forshee, M. Walderhaug, and T. Botsis, Natural language processing systems for capturing and standardizing unstructured clinical information: a systematic review, *Journal of biomedical informatics*, 73, pp.14-29, 2017.

15. A. Amin-Nejad, J. Ive, and S. Velupillai, Exploring transformer text generation for medical dataset augmentation, *In Proceedings of the 12th Language Resources and Evaluation Conference* , (pp. 4699-4708), 2020.
16. N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, Image transformer, *In International Conference on Machine Learning*, pp. 4055-4064, PMLR, 2018.
17. R. Child, S. Gray, A. Radford, and I. Sutskever, Generating long sequences with sparse transformers, arXiv preprint arXiv:1904.10509, 2019.
18. R. Calix, Deep Learning Algorithms: Transformers, gans, encoders, cnns, rnns, and more, *Amazon KDP Publishing*, 2020.
19. D. Bahdanau, K. Cho, and Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473., 2014.
20. A. Graves, G. Wayne, and I. Danihelka, Neural turing machines, arXiv preprint arXiv:1410.5401, 2014.
21. A. Turing, Computing machinery and intelligence, *In Parsing the turing test*, pp. 23-65, Springer, Dordrecht.
22. D. Jurafsky, J. Martin, Speech and Language Processing, 2nd Edition. New Jersey: Prentice Hall, 2008.
23. A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, and A. Badia, Hybrid computing using a neural network with dynamic external memory, *Nature*, 538(7626), pp.471-476, 2016.
24. M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser, Universal transformers, arXiv preprint arXiv:1807.03819, 2018.
25. Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, Dueling network architectures for deep reinforcement learning, *In International conference on machine learning*, pp. 1995-2003, PMLR, 2016.
26. Q. Le, and T. Mikolov, In International conference on machine learning, pp. 1188-1196, PMLR, 2014.
27. T. Mikolov, K. Chen, G. Corrado, and J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781, 2013.
28. Europarl dataset (retrieved August 2020) which can be retrieved from: https://www.statmt.org/europarl/
29. Rcalix.com 2021. The results of the analysis are available at http://www.rcalix.com/research/transformers/spanish/
30. B. Chen, and C. Cherry, A systematic comparison of smoothing techniques for sentence-level bleu, *In Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 362-367, 2014.
31. E. Bugliarello, S. Mielke, A. Anastasopoulos, R. Cotterell, and N. Okazaki, It's Easier to Translate out of English than into it: Measuring Neural Translation Difficulty by Cross-Mutual Information, arXiv preprint arXiv:2005.02354, 2020.
32. J. Iranzo-Sánchez, J. Silvestre-Cerdà, J. Jorge, N. Roselló, A. Giménez, A. Sanchis, J. Civera, and A. Juan, Europarl-st: A multilingual corpus for speech translation of parliamentary debates, *In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8229-8233, IEEE, 2020.
33. J. Liao, Y. Shi, M. Gong, L. Shou, H. Qu, and M. Eng, Improving Zero-shot Neural Machine Translation on Language-specific Encoders-Decoders, arXiv preprint arXiv:2102.06578, 2021.
34. A. Currey, and K. Heafield, Incorporating source syntax into transformer-based neural machine translation, *In Proceedings of the Fourth Conference on Machine Translation*,

Volume 1: Research Papers, pp. 24-33, 2019.

35. A. Fan, and C. Gardent, Multilingual AMR-to-text generation, arXiv preprint arXiv:2011.05443, 2020.
36. R. Sennrich, B. Haddow, and A. Birch, Neural machine translation of rare words with subword units, arXiv preprint arXiv:1508.07909, 2015.
37. C. Shannon, A mathematical theory of communication, *ACM SIGMOBILE mobile computing and communications review*, 5(1), pp.3-55, 2001.
38. Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, *http://arxiv.org/abs/1907.11692* 2019.