# Attention is all you need

## Technical summary

The paper introduced a new model called the Transformer model. For a long time, Recurrent neural network architectures have dominated the field of NLP. However, they are difficult to parallelize and they struggle with long-range dependencies within the input and output sequences. The Transformer models those dependences by using attention mechanism coupled with many techniques. Instead of using traditional one sweep of attention, the Transformer uses multiple attention distributions and multiple outputs for a single input (they called them "heads"). The Transformer also uses layer normalization and residual connections to speed up the optimization process. Realizing that attention cannot take advantage of the position of the input, the Transformer adds explicit position encodings to the input embeddings.

## Description of contribution

Compared to the prevalent recurrent layers used in encoder-decoder architecture, the proposed model is a novel architecture in language embedding. The Transformer is the first sequence transduction model that built entirely on attention. It has better training time while accomplishing higher score in language translate tasks.

The model achieved the new state of the art in WMT 2014 English-to-German and WMT 2014 English-to-French translation tasks. Although the Transformer architecture was only evaluated in language translate tasks, it has potential to be applied to other tasks in natural language processing.

## Major critiques

The authors have researched carefully state-of-the-art models using recurrent or convolutional neural networks. Throughout the paper, they demonstrated knowledge of other sequence to sequence models and pointed out their problems. The Transformer is the first architecture that relies only on attention mechanism to perform the task of encoder-decoder.

It is easy to follow the paper since the authors first introduced the general architecture and then elaborate on each component. The multi-head attention idea is novel and effective. It helps capturing various aspects of the input by using different linear transformations to the values, keys and queries for each "head" of attention. The explanation for the scale factor of $1/d_k$ for the scale dot-product attention is reasonable. The use of positional encoding is necessary because unlike recurrent networks, the multi-head attention network cannot make use of position of the words in the input sequence. The two equations for positional encoding mentioned in the paper allowed relative position between different embedding to be easily inferred because PE[pos + k] can be represented as a linear function of PE[pos]

The statistics of the training process were transparent with hyperparameters stated clearly in the paper. The publicly available code used to train and evaluate the model is also a plus point. The evaluation on variations of different components of the model helped the reader better understanding their importance and contribution to the result. Comparing the complexity per layer, sequential operations and maximum path length of the Transformer to Recurrent and Convolutional model help demonstrating the superior of the model.

One question in 6.2 Model Variations: Will the result be better if we make increase N and h and reduce the dimensions of the vector. The best model had N = 6, h = 16, $d_{model}$ = 1024. What about N = 8, h = 20 and $d_{model}$ = 512?

# BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

**Technical summary**

Unlike traditional language models which use previous n tokens to predict the next one, BERT (which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformer) trains a language model that takes both the previous and the next tokens into account when predicting. BERT is trained on two tasks. In the first task, masked language model training, 15% of the tokens in each example are replaced with a special [MASK] token and the model is required to predict the masked tokens. In addition to that task, BERT also uses the next sentence prediction task to pretrain the model since it is useful for tasks that require an understanding of the relationship between two sentences (e.g. question answering or natural language inference). BERT uses the Transformer architecture for encoding sentences. Even with small datasets, BERT's performance increases when it is given more parameters.

**Description of contributions**

When published, BERT outperformed the state-of-the-art across 7 tasks, including new models like ELMo or GPT. BERT has become the gold-standard for language representation ever since its appearance. Similar to ELMo, BERT is an effective transfer learning method.

**Major critiques**

The method of masking words in the sentence and predicting them is effective because it forces the model to incorporate information from the entire sentence to learn to deducing the missing words. Left-to-right architecture is not able to capture useful information from the right context of the current token. Bidirectional LSTM based language models (such as ELMo, ULMFiT) do not take both the previous and subsequent tokens into account at the same time since there is a single LSTM for the forward and backward language model each.

The next sentence prediction task used in the training process is helpful for tasks which requires relationship between two sentences such as Question Answering or Natural Language Inference. The [CLS] embedding is useful for classification tasks such as sentiment analysis.

Fine-tuning BERT is easy because attention mechanism in the Transformer, which is a component of BERT, give BERT the flexibility to model may downstream tasks by swapping proper inputs and outputs. Additionally, the model can be easily incorporated into downstream supervised learning tasks.

The authors spent a lot of effort to experiment the model on a variety of tasks. The results showed that BERT can even outperformed task-specific methods to become a new state of the art in GLUE, SQuAD v1.1, SQuAD v2.0 and SWAG. These results imply that BERT can be applicable in a wide range of tasks in natural language processing.

The authors conducted a thorough ablation studies over a number of facets of BERT. They showed the significance of both masked language model and next sentence prediction task in pre-training the model. After experimenting with various numbers of layer, hidden activation sizes, and number of attention heads, the authors claimed that across all tasks, the performance of larger models are better, even for very small datasets like MRPC. Although fine-tuning the model gives the best performance, using fixed embeddings extracted from BERT does not produce remarkably worse results on the CoLL-NER dataset. This suggests that BERT is both effective for fine-tuning and feature-based approaches.

**Minor**

Error: typo "befor" (page 6), 2$^{nd}$ last paragraph

# Deep contextualized word representations

## Technical summary

Since the meaning of a word depends on its context, its embedding should also take context into consideration. Embedding from language models (ELMo) use language models to obtain embeddings for each word while taking the entire sentence or paragraph into account. ELMo was trained on a multilayer RNN and learned word embeddings from context. Specifically, ELMo uses pre-trained, multi-layer, bi-directional, LSTM-based language models and extract the hidden state of each layer for the input sequence of words. The more layers you have, the more context you can learn from the input. Then, they compute weighted sum of these hidden states to obtain an embedding for each word. The weight of each hidden state is task-dependent and is learned.

## Description of contribution

ELMo improves the performance of models in a wide range of tasks, spanning from question answering and named entity recognition to sentiment analysis. Despite its simplicity, ELMo can elevate a baseline model to outperform sophisticated models and become a new state of the art.

ELMo is also an important progress in transfer learning since the pre-trained embedding can be used in other domains and still accomplishes good results.

## Major critiques

The idea of computing representations for tokens and passing them through a bidirectional LSTM network to form bidirectional language models (biLM) is not new. The proposed method (ELMo) combines intermediate layers in the biLM in a task specific manner. For each task, the vector representations will be adjusted to better fit the task. ELMo can be easily incorporated into a supervised model without complex modification.

The author evaluated ELMo across a variety of NLP tasks: question answering, textual entailment, semantic role labeling, conference resolution, named entity extraction and sentiment analysis, in which adding ELMo achieves new state of the art results. The evaluation was done thoroughly, with a baseline model implemented for each task. The results show that using ELMo improve the performance of the baseline methods. Many of the baseline models become new states of the art in their tasks after adding ELMo.

The analysis section provides more insights into different aspects of the proposed method. The authors show the importance of combining multiple biLM layers because each layer can capture different types of information of language. For example, top layer is more essential for word sense disambiguation, while first layer better represents basic syntax for POS tagging. ELMo is claimed to be good in term of sample efficiency and models using ELMo also require smaller training sets. Other constituents of ELMo such as the significance of sub-word information or the necessity of pre-trained vectors are also examined sufficiently and objectively.

Overall, the proposed method is very promising since it is applicable to a wide range of tasks in natural language processing. Unlike other methods (Glove, RNN), ELMo can become a potention competitor in transfer learning as it can be retrained to adapt to a new context.

# Distributed representations of words and phrases and their compositionality

**Technical summary**

In this paper, the authors proposed a method (word2vec) to represent words and phrases in a vector space. To do this, the paper's idea is to capture the context of a word by predicting its surrounding words (skip-gram model). A feed-forward neural network is trained to to the following task: given a word, maximize the probability of nearby words that appear along with that word in the corpora. The layers of the neural network is used as the vector representation for words after the training process. They use gradient descent to solve this optimization problem. Additionally, to improve the computational efficiency, the paper uses two innovative techniques: subsampling of frequent words and negative sampling.

**Description of contribution**

The paper shows how to train distributed representations of words and phrases with Skip-gram model using a huge amount of data. The proposed techniques, which are negative sampling and subsampling of frequent words not only results in faster training time but also provide better representations. The vector representations learned from the model demonstrate some interesting findings and can be used to make analogical reasoning.

Although this work is not the first one which tries to build distributed representations of words and phrases, the word2vec exhibits significant improvement in training time and accuracy. This model has great potential to be applied to other tasks of natural language processing.

**Major critiques**

At the time when the paper was published, the idea of representing words or phrases in the vector space is new and little progress in this area was made. The solution provided in the paper is novel and it successfully overcomes challenges of the other previous work.

The way the authors presented their final model was logical and easy to follow. They first came up with the formula for the skip-gram model by trying to maximize the probability of context words given a word. Realizing the difficulty in computational time of this model, they introduced methods that can be used to solve the problem. The negative sampling technique, which is a simplified version of the Noise Contrastive Estimation, was used to limit the calculation of each pair of words to only a small portion of data drawn from the noise distribution instead of looping through the whole vocabulary. The subsampling of frequent words is also necessary since words like "the" or "a" are not very informative. These techniques made the training with enormous amount of data possible, which other related work failed to achieve. The expansion to represent phrases using Skip-gram is reasonable and it showed that phrases can also be captured in the vector space.

The paper also carried out a thorough evaluation for their work using the analogical reasoning task. The word2vec model was superior than the Hierarchical Softmax and Noise Contrastive Estimation, which strengthened the author's choice for the Negative Sampling method. The parameters (subsampling rate, noise distribution) were chosen based on practical performance.

There are still several parts in the paper that need some more consideration. In section 6, comparison to published word representations, the paper chose some infrequent words and showed their nearest neighbors to claim that their model outperformed other models. However, having better representations for some words or phrases might not mean that the model's distributed representations are better overall. The empirical method for choosing the noise distribution of negative sampling and the threshold of $10^{-5}$ for the subsampling rate may not work well in some specific domain of language.

# Glove: Global vectors for word representation

**Technical summary**

The proposed model (Glove) aims to achieve two goals: to create word vectors that capture meaning in vector space and taking advantage of global count statistics instead of local information from other models like what word2vec did. Glove learns word representation from co-occurrence matrix and use co-occurrence ratios as its training goal. Matrix factorization methods like LSA, while using global count information, do not possess the desired behaviors of vectors learned from word2vec. Glove tries to take the best of both worlds: take global information into account while learning dimensions of meaning.

**Description of contributions**

Glove outperformed other models, including recent one like word2vec, in several tasks: word analogies, word similarity and named entity recognition. The training time of Glove is also faster compared to word2vec. The model provided a novel perspective on the word embedding problem. It created vector representations of word with meaningful features of window-based method while still taking advantage of the global co-occurrence statistics from the corpus.

The authors showed the resemblance between Glove and word2vec in their objective function for optimization. Despite their starting points, the two model surprisingly turned out to be extremely similar.

**Major critiques**

The paper carefully examined related work in generating low-dimensional word representations and pointed out weaknesses of the two approaches. Matrix factorization methods such as LSA do not capture dimensions of meaning. Shallow window-based methods like word2vec, on the other hand, do not take advantage of the co-occurrence statistics of the corpus. The proposed model (Glove) operate on global count from the data while still learning useful vector representations for words.

It is beneficial for readers when the author explained step by step how to came to the final formulation. Since it is mathematically heavy, showing the evolving of the equations and the reasons to do so is remarkably conducive to understanding the model. The initial example of “ice” and “cream” was very intuitive and helped introducing the underlying principle behind Glove, which can be stated as follows: the co-occurrence ratios between two words in a context are strongly connected to meaning. After that, the authors gradually modified the equations in a systematical and logical way. Therefore, the final equation, even though significantly different and more complex than the first one, still sounded convincing.

Another great point was that the paper related Glove to other models such as word2vec to show that although they are from different starting point, the final formulas were very similar. The transition in this relationship was persuasive and not too difficult to follow. The authors also estimated the complexity of the model in order to demonstrate that it could be even more efficient than the window-based methods.

Evaluation of the Glove model was done carefully in three tasks: word analogies, word similarity and named entity recognition. The corpora and decisions on hyperparameters were explained in detail. Factors that affects the model such as vector length, context size or corpus size were discussed. To better compare Glove to word2vec, in addition to the experimented tasks, the authors should use them on tasks that use pre-trained word representation such as sentiment analysis to evaluate their empirical performance when applied to real world problems.