Research Article

RBN: enhancement in language attribute prediction using global representation of natural language transfer learning technology like Google BERT

Chiranjib Sur[1]

Received: 18 May 2019 / Accepted: 22 November 2019 / Published online: 4 December 2019

Abstract
Transfer learning can replace the long and costly data collection, labeling and training session by effective and the most efficient representations. BERT, trained by Google, is a language representation generator and is far more global to be effectively determine the representations of natural languages and create the numerical version of grammatical structures and inter-dependencies of language attributes. In this work, we introduced recurrent BERT network and singular BERT network and have demonstrated the effective way of utilization of BERT for applications like parts-of-speech tagging and phrase tagging, which are integral part of understanding languages structure and interpretation of message. We have achieved extraordinarily high accuracy for prediction with these models and have done a comparative study using different datasets and is aimed at applications related to sentence generation. We created an accuracy of 96.65% for parts-of-speech detection and 95.24% for phrase prediction for Penn Tree Bank sentences and 99.64% and 97.94% for MSCOCO dataset sentences. Different origin sentences will ensure that while human generated sentences are complex for high accuracy and prone to errors, effective machine generated sentence attribute detection provision must be kept open for progress in language understanding.

Keywords POS tagging · Phrase tagging · BERT · Deep representation learning · Natural language global embedding

1 Introduction

Language modeling [33, 40] has gained much momentum in industry and academics due to high demand of making machine learn to generate and understand languages. As large volumes of data are produced each day, automatic parsing of the language data and language attribute prediction become inevitable problems. Language parsing will help in language understanding and also in parsing of contexts and many researchers have provided many deterministic and prediction based algorithms. In this work, we provided some important and fundamental analysis of BERT model related to language attribute prediction in real time scenario, where the machine generates and learns at the same time. Enhancing these procedures is challenging and the best effectiveness in accuracy of detection is still around 97.20% [1] for these language attributes (like Parts-of-Speech, Phrases, Constituent Trees etc.). In this work, we have devised analysis of the novel transfer learning approach BERT for language attribute detection from sentences. Transfer learning is already widely used in many computer vision applications like ResNet [2], Inception [3], AlexNet, VGG [4] etc. With BERT (Bidirectional Encoder Representations from Transformers) [5], the prospect of a global representation for natural language is also now feasible. Previous language models took the form of the data, without worrying the underlying implications that

✉ Chiranjib Sur, chiranjibsur@gmail.com | [1]Computer and Information Science and Engineering Department, University of Florida, Gainesville, USA.

the language attributes should carry, but BERT makes that possible and we have analyzed some of the underlying language attribute properties that BERT bring into the table. Researchers claim that BERT used so much of training data and resources that each training epoch cost is very high and correspondingly no other embedding can be more generalized than this, though it may not be specific and need fine-tuning. In fact, Deep Learning's Uncertainty Principle states "Networks with greater generalization are less interpret-able, while networks that are interpret-able don't generalize well". In this work, we have demonstrated that we can deliver a better parser and a better POS tagger, when we utilize prediction model consisting of the BERT representation and infrastructure. We achieved considerable improvement in prediction considering the fact that we predict the language attribute of words of sentences, without knowing the next series of context. English is a non-structural language, where the language attribute of the word depends on the words before and after. In that sense, we achieved comparable accuracy of prediction with our model, whereas previous works utilized both past and present topological contexts as dependencies. Mathematically, we can define the tagger prediction problems as the followings,

$$P_t(\text{prediction} \mid \text{context}) = P_t(\text{prediction} \mid \text{past words, present word, future words}) \quad (1)$$

where $P_t(.)$ is the traditional way of tag prediction based on the presence of the whole sentence. However, here, we try to predict real time based on the sentence sequence generated and not on what will be generated as we consider the scenario of machine generating and understanding what it is generating. Mathematically, the problem is denoted as,

$$P_r(\text{prediction} \mid \text{context}) = P_r(\text{prediction} \mid \text{past words, present word}) \quad (2)$$

and $P_r(.)$ is the real time model. For proper language understanding higher accuracy will be highly desirable. All the previous models utilized the whole sentences for prediction of the individual components of the sentences and that is suitable for many applications, related to understanding the existing text data. But a real time prediction is not feasible as the current prediction will be also dependent on the future words. Our goal is different and we argue that our real-time model as good as theirs in performance and thus is much better. If we had utilized the future knowledge, we would had surely outperformed them in every respect (Fig. 1).

The rest of the document is arranged with revisit of the existing works in literature in Sect. 2, the details and scope of the BERT architecture in Sect. 3, the description and statistics of the training data in Sect. 4, the intricacies of our methodology in Sect. 5, experiments, results and analysis in Sect. 6, concluding remarks with future prospects in Sect. 7,

The main contributions are in many folded and it is important to point these out.

- For the first time we used a global representation for language attribute prediction, without worrying about the dependency strategy.
- We formulated the language attribute prediction problem into diverse ways (single word based prediction, multiple (only previous) word based prediction, etc) and used diverse datasets (Penn, MSCOCO) and different set of prediction classes (refer Tables 1, 2).
- We showed that we can predict the POS and phrases of words on the go without concerning about generating the whole sentence.

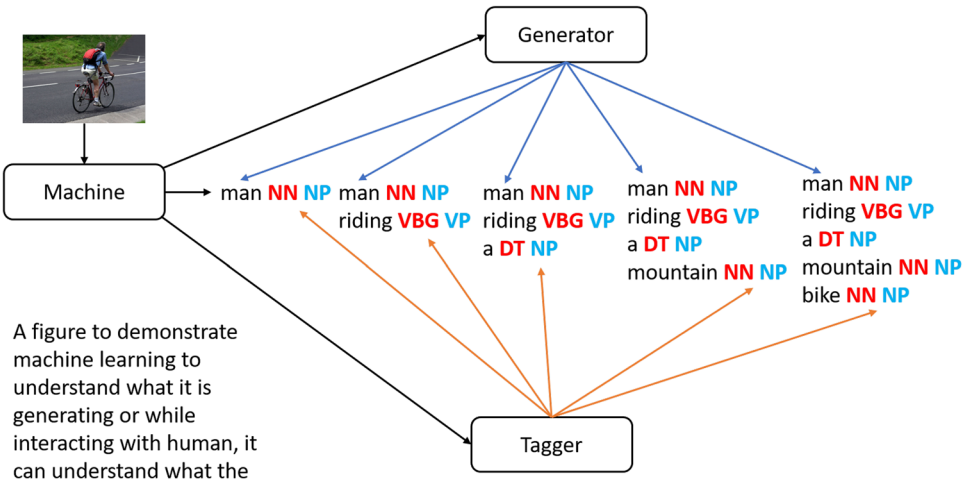


Fig. 1 Example of machine learning based generator model to understand generation of language

Table 1 Performance comparison of prediction accuracy for POS (parts-of-speech) tags of languages with fine-tuning

Datasets ⇒	Penn Tree Bank	Penn Tree Bank (reduced)	MSCOCO data	Generated caption
# Number of classes ⇒	80 classes	34 classes	34 classes	34 classes
	Evaluation accuracy	Evaluation accuracy	Evaluation accuracy	Evaluation accuracy
Stanford Parser [1] (human annotated)	–	0.9720 (prev + post words)	–	–
Dependency network [7]	–	0.9692 (multiple window)	–	–
Google [6]	–	0.9280 (end-to-end)	–	–
Caption generator ensemble† [40]	–	–	0.8769 (prev + post words)	0.9565 (prev + post words)
Word representation BERT-1	0.9377 (single word)	0.9301 (single word)	0.9755 (single word)	0.9868 (single word)
Sentence representation BERT-1	**0.9647** (previous words)	0.9640 (Previous Words)	0.9845 (previous words)	**0.9964** (previous words)
Sentence representation BERT-3	0.9634 (previous words)	0.9619 (previous words)	**0.9960** (previous words)	0.9865 (previous words)
Sentence representation BERT-4	0.9626 (previous words)	**0.9665** (previous words)	0.9845 (previous words)	0.9963 (previous words)

Bold indicates the best results for performance

†Caption generator ensemble: likelihood of word representation

**BERT-1: uncased L-12 H-768 A-12, BERT-2: uncased L-24 H-1024 A-16

**BERT-3: cased L-12 H-768 A-12, BERT-4: multilingual L-12 H-768 A-12

*Penn Tree Bank and Penn Tree Bank (reduced) test dataset consists of WSJ 22 and WSJ 23 stories

***BERT-2: uncased L-24 H-1024 A-16 cannot run on Tesla K80 GPU

Previous words—RBN, single word—SBN

–: Results not available

Table 2 Performance comparison of prediction accuracy for phrase tags of languages with fine-tuning

Test datasets ⇒	Penn Tree Bank	Penn Tree Bank (reduced)	MSCOCO data	Generated caption
# Number of classes ⇒	256 classes	48 classes	21 classes	21 classes
	Evaluation accuracy	Evaluation accuracy	Evaluation accuracy	Evaluation accuracy
[41]	–	0.904 (multiple window)	–	–
Caption generator ensemble [40]	–	–	0.9425 (prev + post words)	0.9759 (prev + post words)
Word representation BERT-1	0.7280 (single word)	0.8734 (single word)	0.936 (single word)	0.9578 (single word)
Sentence representation BERT-1	**0.8879** (previous words)	**0.9531** (previous words)	0.9610 (previous words)	0.9786 (previous words)
Sentence representation BERT-3	0.8863 (previous words)	0.9524 (previous words)	0.9625 (previous words)	**0.9795** (previous words)
Sentence representation BERT-4	0.8856 (previous words)	0.9514 (previous words)	**0.9640** (previous words)	0.9794 (previous words)

Bold indicates the best results for performance

†Caption generator ensemble: likelihood of word representation

**BERT-1: uncased L-12 H-768 A-12, BERT-2: uncased L-24 H-1024 A-16

**BERT-3: cased L-12 H-768 A-12, BERT-4: multilingual L-12 H-768 A-12

*Penn Tree Bank and Penn Tree Bank (reduced) test dataset consisted of WSJ 22 and WSJ 23 stories

***BERT-2: uncased L-24 H-1024 A-16 cannot run on Tesla K80 GPU

Previous words—RBN, single word—SBN

–: Results not available

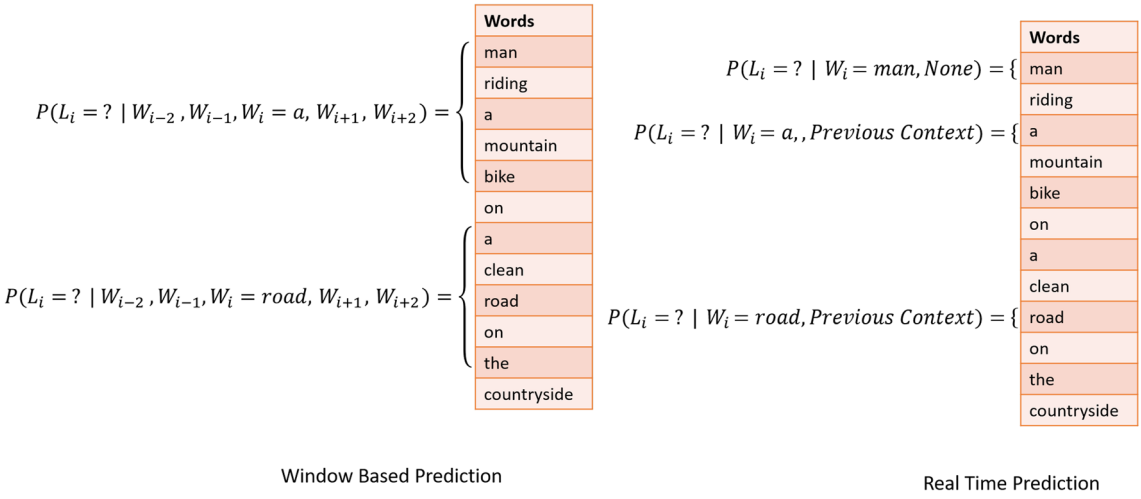


Fig. 2 Real time prediction model versus window model (series of sentences and likelihood)

- We also established facts whether it is beneficial to go for very high end BERT models while prediction is concerned.

2 Existing works

There are lots of work being done in this narrow field and is considered as the fundamental problem of natural language processing (NLP) researchers. We revisit some of their starte-of-the-art works in this section. Manning [1] utilizes manually annotated Penn Tree Bank for sentence tagging, while [6] utilized end-to-end and attention strategy for prediction, where during encoding the model has seen the whole sentence as some form of embedding and during attention, it also utilizes an attention for prediction. In [7], the task is defined as window based prediction, where likelihood is defined as a probability model based on the words representations like the followings,

$$\mathcal{P}(\mathbf{x}_t = c \mid \mathbf{x}_{t-n}, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+m}) \tag{3}$$

where we have $(n + m)$ as the window size for each position t and much of the effectiveness is dependent on both previous and post positional information. However, this procedure cannot be utilized for real time prediction of sentence tagging as future words are yet to be predicted. On the contrary, we can claimed that if effectiveness of likelihood of

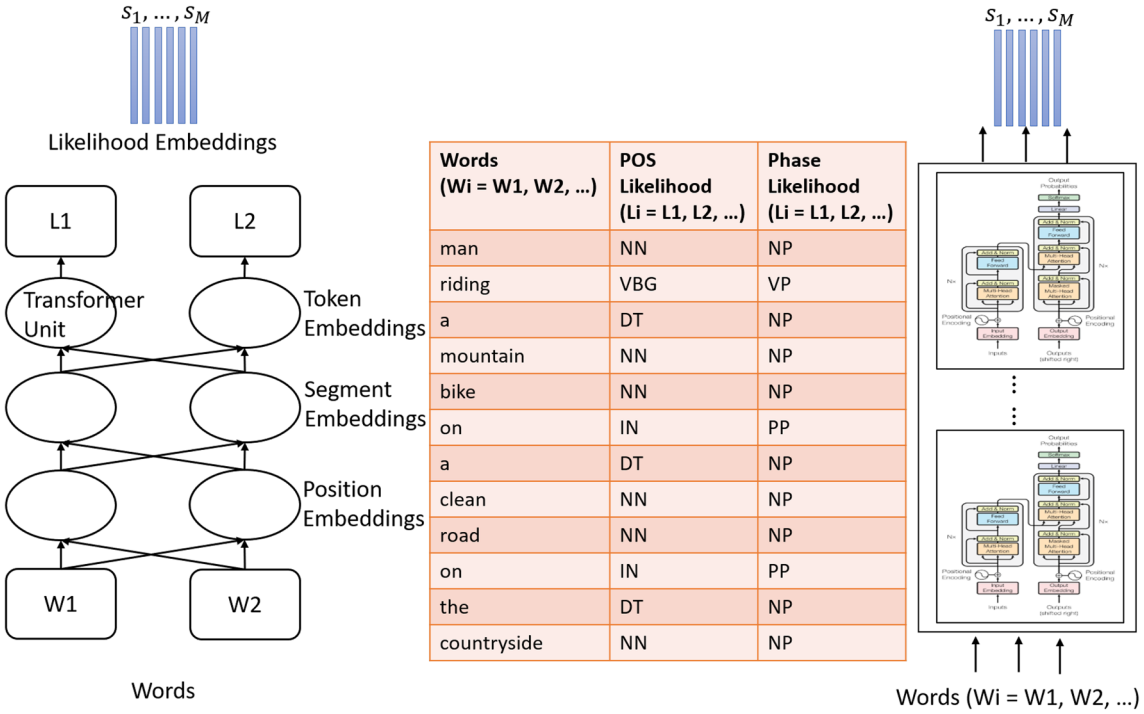
$$\mathcal{P}_r(\mathbf{x}_t = c \mid \mathbf{x}_{t-m}, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t) \tag{4}$$

is enhanced, it can benefit likelihood of

$$\mathcal{P}_t(\mathbf{x}_t = c \mid \mathbf{x}_{t-m}, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+m}) \tag{5}$$

as well and can help in real time prediction, without depending on future appearing sequences. Real time prediction is a challenge for many applications related to language understanding like dialog systems, query understanding for Knowledge Bases, understanding document content, differentiating sentences that have similar bag of words like in "equivalence finding" tasks, question answering, etc. Hence, this work emphasized more on likelihood of models that can be represented as, $\mathcal{P}_r(\mathbf{x}_t = c \mid \mathbf{x}_{t-m}, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t)$ and similar kinds of situation and we have shown improvements and reported state-of-the-art accuracy and results with Penn Tree Bank dataset [1], which is the state-of-the-art ground-truth reference and is reported to have accuracy around 97%.

Some of the popular part-of-speech tagging models were like boosting based model [8], tagger based on statistical methods [9], equation based tagger [10], stochastic program [11], statistical estimator based [12], conditional structure and conditional estimation [13], hidden Markov model assuming independence of variables [14], probabilistic method [15], maximum of entropy model [16], second order hidden Markov model[17], maximum of entropy model with enriched sources for estimation of POS [18]. Phrase tagging works included modeling like through transition-based system for joint part-of-speech tagging and labeled dependency parsing with non-projective trees in [19], perceptron algorithm with dynamic programming methods to recover full constituent-based parse trees in [20], maximum entropy based parser in [21], dependency parsing with subtrees in [22], dependency language mode



Words (Wi = W1, W2, ...)	POS Likelihood (Li = L1, L2, ...)	Phase Likelihood (Li = L1, L2, ...)
man	NN	NP
riding	VBG	VP
a	DT	NP
mountain	NN	NP
bike	NN	NP
on	IN	PP
a	DT	NP
clean	NN	NP
road	NN	NP
on	IN	PP
the	DT	NP
countryside	NN	NP

Fig. 3 BERT architectures for prediction for languages

in [23], perceptron algorithm classifier in [24], self-training PCFG grammars with latent annotations in [25], conditional random fields in [26], a data-driven parser in [27], a classifier-based parser in [28], new string-to-dependency machine translation in [29], paradigmatic and syntagmatic lexical relations in [30], support vector machines based classifier in [31], online large-margin training for multiclass classifier in [32].

A pictorial difference between the real time tag predictor and the traditional model is provided in Fig. 2. In Fig. 2, "Previous Context" refers to the recurrent neural network parameters. Language attribute prediction accuracy is quite high when we process the whole generated sentence together and each word attribute prediction is defined as a function of probability of maximum likelihood $\mathcal{P}(\mathbf{x}_t = c \mid \mathbf{x}_{t-m}, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+m})$ for sentence with N words $\{\mathbf{x}_0, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_N\}$. But, while researching on image captioning generation [33], it is also very useful to consider the partially generated sentence and derive the language attributes on the fly for real time understanding and inference and we can achieve this effectiveness with BERT which was not possible before. We can do this with BERT embedding with very high accuracy and this can be regarded as a game changer for big data applications, where a second time traversing (of huge number of sentences) would cost lots of dollars. Even if we consider sentences, the highly connected BERT layer can produce very good results, but since BERT is itself a bi-directional model and is dependent on the whole sentences, we utilized the model with single words and more details are provided in Tables 1 and 2 in Sect. 6. The main idea is to test BERT for diverse applications and experiment settings and also provide some neutral representation, beyond some likelihood and distance based similarity. While, most of the network models tend to converge diverse representations to a similar space, BERT kinds of architecture encourages more generalized framework learning [34]. Large part of the image captioning framework is also attributed to generalized and easily distinguishable representations and researchers are trying to establish this. But, BERT kind of architecture is very costly and it is estimated that if someone tries to train the lowest BERT model with Amazon AWS, it will cost 30,000 USD for each epoch of training, a task way beyond the reach of many researchers. However, in comparison to that, BERT can be fine-tuned in much lesser time and with low infrastructure and is highly efficient. BERT has been introduced recently and has only been used for prediction and classification tasks through enhancing the capability of the machine to learn representation. However, at the same time, it, also, establishes the local positional and contextual influences of neighboring words.

Results around 97% effectiveness is not absolute for a word, but is based on its positioning in the sentences and is also based on influences of neighboring word(s), both backward and forward. In this work, we have shown

that we can beat the existing performance, when we are equipped with a much global representation and that even without any influences of neighbor(s). As the computational power increases and the number of available training data is very high, such systems and global representations can be trained. Google released such a pre-trained system and this system has the capability to handle different representations and combinations of representations and this can used for proper determination of contexts and situations and meanings of sentence. In this work, we have tried to answer the question, whether we can also do proper determination of the different language attribute tagging through this model. We find very good result when we experimented. Moreover, the representation of BERT is far beyond the normal expectation, where word embedding techniques like GloVe [35], Word2Vec can achieve only 76% when processed individually, BERT can predict those situations with 98% accuracy. Definitely, when combined with forward and backward influences, this 98% accuracy can sought to 100% with no doubt.

3 BERT description

BERT (Bidirectional Encoder Representations from Transformers) [5] is a multi-layered network, with several cooperated attention feedback to gather informational fusion and representation generation. These units are known as Transformer and mathematically the attention representation of a Transformer can be represented as the followings,

$$\Phi_{MH}(Q, K, V) = [\Theta_1;\Theta_2; \ldots ;\Theta_n]W_T \tag{6}$$

where $\Phi_{MH}(.)$ is the multiple head function and we define the individual head function $\Theta_i(.)$ as,

$$\Theta_i = \text{Attention}(QW_{Q_i}, KW_{K_i}, VW_{V_i}) \tag{7}$$

where we define Attention(.) function as,

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{8}$$

where $\sqrt{d}$ is a favorable scaling factor. Overall, these equation defines every language data into three quantities Q (queries), K (keys) and V (values) and train an encoder-decoder model using the multi-head attention components and the depth and scale of the model depends on the data and the complexity of the problem. There are two kind of attention: additive attention, where compatibility function is calculated and dot-product (multiplicative) attention, where the product is calculated and then softmaxed and if required scaled favorably.

Since, BERT pre-trained models are in public domain, it is expected that the inside version of BERT in Google is much better and more effective. While, most of the academic research solutions are particularly useful, industrial products are much more complex and need to go through β-testing, where the developers are mostly interested in the untrained operational sectors and how they behave. Transfer learning has been able to gather much momentum and provide lots of improvement in many applications in the intersection of media and natural language processing (NLP) and survived β-testing. BERT will now contribute from the NLP side through powerful representations that will enhance the understanding of the models. Figure 3 provided an architectural overview of BERT and how can this model be utilized for determination of the tags and can also be defined as a language parser. However, with a new prediction parameter, several other layers add-in to the traditional training framework and the maximum likelihood weights gets trained with the fine-tuning of the BERT network. We observed that more than 2 epochs of fine-tuning of the BERT network can create over-fitting and can decrease the testing accuracy by 0.5–1%. The Annotated Transformer is a coherent multi-dimensional model with more parameters (weights) than the LSTM and also more skip-through connection and more numbers of feedback. Thus, the ability to learn is much higher for these Annotated Transformers and when kept in layers, it can generalize very well. However, due to high number of parameters, the CUDA requirement is also higher and transfer learning is a way out for many researchers.

With new and more effective solutions for many applications are generated everyday, these kinds of pre-trained networks will help in quick development and estimation of the final product capability much earlier than before. It will speed up the process of training and also lower down cost and data collection. Also, many researchers and university settings do not have high-end GPUs (like TPUs) for training of these kinds of networks. Another useful factor that BERT brings in is the notion of uniformity of representations that unite all infrastructures, creating the capability to communicate. Such inter-operatable communication platform will enable machine (or application) communication with natural languages and the capability of understanding and without the bound of limited command combination. This work is not only for identification of language attributes, but a demonstration of the applicability of the transfer learning infrastructure for next generation applications for machine interpretation for inference.

There are many different dimension varieties of BERT model and these are publicly available at their github sites [5]. Out of them there is BERT-1: uncased L-12 H-768

A-12 consists of 12-layer, 768-hidden, 12-heads, 110M parameters. BERT-2: uncased L-24 H-1024 A-16 consists of 24-layer, 1024-hidden, 16-heads, 340M parameters. BERT-3: cased L-12 H-768 A-12, which consists of 12-layer, 768-hidden, 12-heads , 110M parameters. BERT-4: multilingual L-12 H-768 A-12 consists of 12-layer, 768-hidden, 12-heads, 110M parameters. BERT-1 is the baseline model and is very effective in many small applications, while the others are specialized and are more effective when the modeling involves more data and more number of classes. Uncased version were mostly used for name-entity recognition (NER) application, where the entities were paid more attention so that they are recognized without difficulty, while the cased version was more generalized for different applications. In our task of POS tagger or parser, we hardly find difference in performance. However, K-80 GPU failed to run BERT-2 (uncased L-24 H-1024 A-16) and hence the results were not reported. BERT architecture is characterized by series of fully connected layers that are provided with attention from previous layers, while a likelihood is estimated with influences from both previous and next word levels. Mathematically, each Transformer Units of BERT can be denoted pictorially by the following Fig. 4. The equations of this Transformer Units are not clearly provided except a conceptual embedding, but it is evident that the skip-through and multiple headed attentions work well for generalization and learning of the embedding. In Transformer Unit, there are positional encoding vector, and follows a query based model, where the input (query) predicts the probability of the output from the output (context). While, BERT is extensively trained with input-output pairs where both query and context are same and the task is to predict itself, so that a better representation is learned amid noises in the form of Masked tokens instead of original ones.

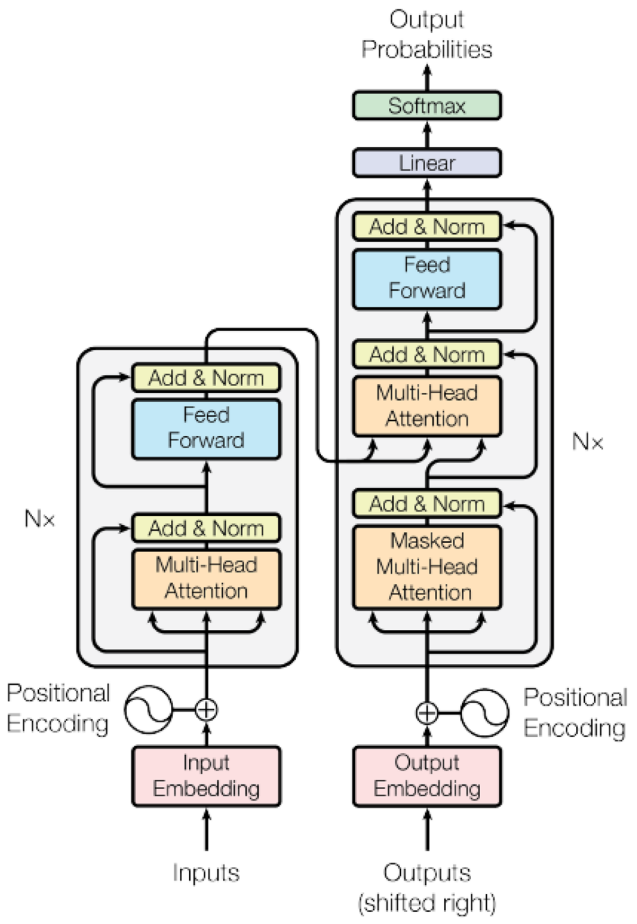


Fig. 4 Transformer units of BERT. Image source is [36]. Several such layers form the BERT architecture

Series of these layers differ for different BERT architectures (denoted as $A = 12, 24$) and are very fine tuned with huge amount of data that are made to trained to predict itself with noise addition in the form of masks. However, the success of likelihood of prediction is also dependent on the ability of recurrent neural network (here transformer) to predict the sequentiality of data with high accuracy like above 80% BLEU_4 and when trained with huge amount of data and with multiple layers of attention, these network can produce very fine representation and prediction of itself. Physically, the multiple class prediction of BERT is gathered by adding an extra layer after the last layer consisting of 768 dimension and when a new prediction model is trained, another likelihood layer is attached. We, instead, separated this 768 dimension embedding and used as a constant model for prediction. This helped in quicker experimentation and avoid the hefty BERT model again and again. The most unique characteristics of BERT is that for a given token (a word in sentence either in query or context), the input representation is constructed by summing the corresponding token, segment and position embedding. The token embedding is defined as the physical entities like words. The segment embedding relates to either query or contexts and have different interpretation in different applications. The position embedding is related to the position of the words in the sentence and its context based on the previous words and next words.

4 Language data description

Language description problem arises due to the constructive capability of the languages. In ancient times, people tried to give shape to each of expressions (ideas), but the feasibility and scope of such large number of definitions gradually faded away with time and people came up with languages so that expressions (ideas) can be constructed with subset of meaning representations and the positional information was far more important.

This work is an effort to make the machine understand the subset of meaning representations through POS and Phrase analysis. While, natural language processing was dealt as bag of words, there were ambiguity in representation of sentences and the non-linear approximation of network algorithms also contributed to it. Hence, there is always requirement to define representations that can separate it and help in solving problems. While researchers worked on local applications and provided solutions for specific solutions, BERT provides a global prospect and the promise to scale and re-usability. Also, sentences are diverse in construction and hence the ambiguity also varies, mainly with many words being characterized into different parts-of-speech and even phrases. This work does not claim to tackle that problem with intensive care as it will require a different set of ground truth data, but we use different sets of data that actually characterize these words into multiple-disciplines. We have provided more details of the data in the next few subsection. We, not only, used human generated sentences, but also evaluated our model for machine generated captions. In the future, most of the data will be machine generated and hence, this kind of initiative of machine generated data analysis is unique and we have done it for the first time.

Accurate phrase prediction helps in Phrase Chunking, a process that helps in separating and segmenting a sentence into its sub-constituents, such as noun, verb, and prepositional phrases and thus helps in understanding the content of sentences. Phrases are combinations of words to describe certain combination of meanings or some sense, which cannot be described by one word. Also, each Phrase comprises of certain combination of POS tags, while, each Phrase also comprises of certain combination of Phrase tags. If we can predict these POS tags and/or Phrase tags, the Phrase tree can be generated, which in return will help in Phrase Chunking, that is establishment of deeper meaning of sentence segments, that can be utilized for indexing and machine understanding of what is the content and their interrelationships. Also when queried, these conglomeration of words can be used for identification of the best possibility. There are many applications that are directly related to these tagging. Like, when we query "big Apple", it is important to denote that the document is not related to the apple, which is big, but it is a definite argument where the "big Apple" entity is related to specific definition and the structure of the sentence will help in understanding these. Here, Named entity recognition is important as 'Apple' is related to organization, events, apart from being fruit. Annotation of document using tags to identify extracted structure is costly and this kind automatic extraction is important to handle billions of data. While, the data amount is very large, it is important to make sure that the effectiveness and accuracy of the algorithms is high. In this work, we devised an approach that uses BERT embedding and can detect POS tags and Phrases effectively and with high precision.

4.1 Language attribute utility

Language Attribute detection has lots of applications in language understanding through components identification. This ability to detect and predict the language attributes will also enhance other applications that require language processing and understanding. Presently, many applications are dependent on understanding languages and create specific meaning for inference instead of mere topic modeling. We have provided a demonstration of language attribute detection and inference procedure in Fig. 5. Like, in a sentence, the parts-of-speech tags can be used to detect the utility of the words in the sentences based on its position and neighbors, while the Phrases can establish higher level coupling of the sequential relationship and can help in establishing the different attributes in sentences. Similarly, a POS description based understanding is also included in Fig. 6. POS attribute detection is very important for many applications for the machines to understand individual words to help in processing queries and determination of proximity of likelihoods for those queries and one such example application is provided in Fig. 6. Overall, language representations are gradually replacing the specific identity of language attributes and principles. Most of the present day approaches of machines understanding of these languages are through

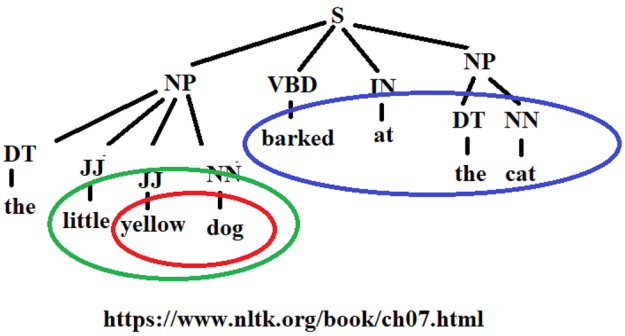


Fig. 5 Language attribute example from NLTK for phrase detection based language understanding. The coloured rounds denote the descriptions and inter-relationships of semantics. Here, if we consider each segment starting with 'dog', 'yellow dog', 'little yellow dog' and so on, we can achieve different level of information, which are hierarchical. Also, if we consider the NP part as "the little yellow dog" and VBD part as "bark", we can associate them as an actor and action with another NP as "the cat". *Source*: https://www.nltk.org/book/ch07.html

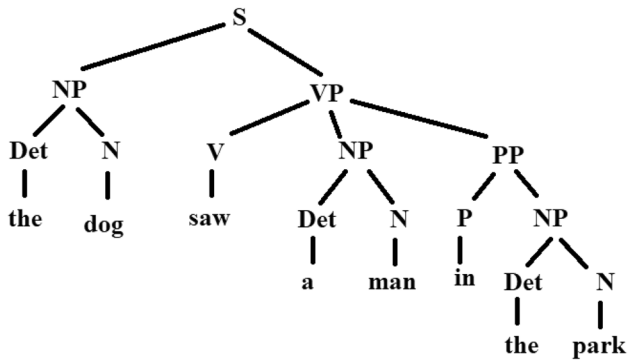


Fig. 6 Language attribute example for POS tagger to understand individual word contribution. The individual descriptions of the semantics are denoted. Here, NP, VP and PP divide the sentence into different meaningful segments, which are themselves associate and interact with the rest of the sentence through some activity like 'saw'. *Source*: https://www.nltk.org/book/ch08.html

these representations. However, the impact of language attribute prediction will not get diminished as these are inevitable part of languages and are required for identification of what has been learned.

4.2 PennTree bank data

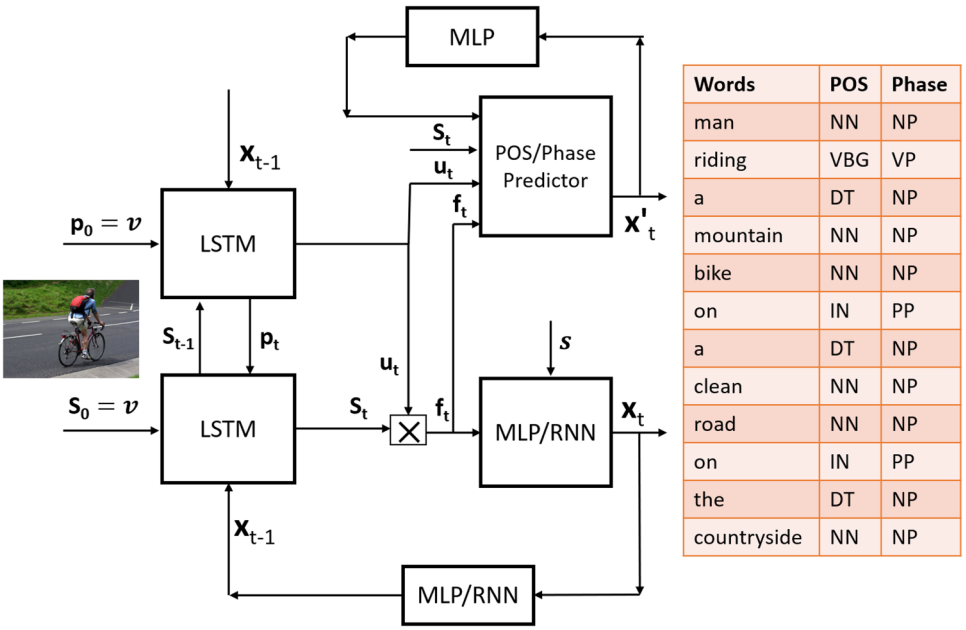
Penn TreeBank dataset [37] is a highly annotated set of stories, using the sentences of Wall Street Journal (WSJ), which can be used for POS tagging task and Phrase tagging task. There are around 25 stories where researchers use story WSJ 22 and WSJ 23 as test dataset, while WSJ 0 to WSJ 24 (excluding WSJ 22 and WSJ 23) as training data. We evaluated each experiment in this traditional way and the results are reported in Tables 1 and 2. The training is performed through a end-to-end sequential way, where the sentences are feed into BERT and the extracted representational is feed into a classifier to predict the different tagging of the sentences. Testing with WSJ 22 and WSJ 23 will help in comparing the results with Stanford parser (Manning, 2017, [38]), which provides the best possible results in POS tagging for sentences and has been widely used for language attribute annotation of other sentences. The success of BERT will depend on its ability to provide very distinguishable representation and thus helping in perfect prediction. The results clearly reflects that we can easily use BERT representation for capturing of grammatical and language attribute representations and can provide the structural intricacies for different tagging. Penn TreeBank dataset consists of (39.8K Training + 1.7K Development + 2.4K Test) sentences with varied length of sentences in the overall 25 stories. However, when it comes to tagging, there are around 80 total categories for POS and 256 for Phrases. The 80 POS categories consists of both POS tags, multiple ambiguous tags (like one word can categorically belong to different POS classes) and also identification of punctuation and other special characters as these stories are in narration format. We experimented with both 80 category and a reduced category version with 34 POS category version. Similarly, for phrases, we experimented with a multiple-factor based prediction with 256 classes and a reduced class data with only 48 phrase categories.

4.3 MSCOCO image captions and generated image captions

MSCOCO Image Captions [39] consisted of huge amount of sentences for image caption and we used these data for evaluation of the tag prediction models based on the BERT infrastructure representations. We used Karpathy's validation set sentences of MSCOCO dataset as our data. This comprises of 25,010 sentences, out of which 23,010 sentences were used as training set and the last 2000 sentences were used as test set. We used Stanford POS tagger and Stanford Parser for identification of the POS tags and the Phrase tags for these sentences. While, these MSCOCO validation set sentences are human generated sentences, we also used a test set comprising of generated image caption from a caption generator model with effectiveness of 30.7% for BLEU_4. Figure 7 provided a pictorial overview of the model. These 5000 sentences were also tested on the model trained with 23,010 sentences of MSCOCO data. We denoted the evaluation of 2000 sentences of validation set in "MSCOCO Data" column and the evaluation of 5000 sentences (generated from trained model) as "Generated Caption" column. Also, we performed different experiments with the data and with varied number of classes of tags. We have "MSCOCO Data" *5K* data in one sets with 21 phrase tags and 34 POS tags. While the "Generated Caption" *2K* data had two sets: one with 256 Phrase tags and 80 POS tags, while another with 48 Phrase tags and 34 POS tags. 256 Phrase tags and 80 POS tags version contains special characters and other non-essential tags that were part of the PennTree dataset, but is not actually part of the language understanding infrastructure. Hence, when we removed those tags, the effectiveness improved and is clearly reflected in results in Tables 1 and 2 for POS tag and Phrase tag respectively.

5 Our methodology

In this section, we will discuss the different networks that demonstrated the individual capability of the BERT representations and also as a topologically connected sequence. We emphasized the real time analysis of the



Words	POS	Phase
man	NN	NP
riding	VBG	VP
a	DT	NP
mountain	NN	NP
bike	NN	NP
on	IN	PP
a	DT	NP
clean	NN	NP
road	NN	NP
on	IN	PP
the	DT	NP
countryside	NN	NP

Fig. 7 Language attribute prediction from image captioning likelihood model. This model is used to generate sentences, denoted as "generated caption" in Tables 1 and 2 [40]

sentences, which do not depend on the future contextual words and is much realistic and has not been considered before. The aim of this work, is to demonstrate the capability of generator of simultaneous generation of contextual sentences and also provide the semantic analysis, which can help in producing better analysis for machines and would help in real time frameworks. A similar framework for image captioning (with attribute tagger) is provided in Fig. 7. We have provided a comparative study of different model performance based on the works done before. In this context, we will discuss the modifications of the softmax layer for maximum likelihood layer of BERT and the physical interpretations, considering the fact that the true training softmax maximum likelihood layer of BERT is completely different.

5.1 Sentence and word representation BERT

BERT is mostly aimed at defining unique representations for sentences and segments through the combination of token, segment and positional information. Sentence Representation BERT is most popular in many applications including Question-Answering, Sentence Classification etc, but the joint probability (Sentence Representation BERT) will not be helpful in tag prediction. BERT representation can be regarded as recurrent representation where the later word representations had influences from the previous and future ones, which are distant in position and not in time and is unique. However, we assume that we devise a model which is dependent on only previous ones and not from the next ones as they are yet to be generated and the partial sentence is informative. If we consider a sentence $\{\mathbf{x}_0, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_N\}$ with N words and each word is denoted as $\mathbf{x}_i$, then we can define the prediction models for language parser and POS tagging in many different ways. We have revisited some of such definitions.

Word Representation BERT is a subset of the Sentence Representation from BERT. When we experimented with Word Representation BERT, we provided ample evidence that it can be utilized with much more efficiency like word embedding and has a great prospect for many applications, where word embedding is required. From the effectiveness of performance of this data category based prediction, researchers will think of considering the BERT representation as the next best word embedding for many applications. The main prospect of doing a comparison study of different network models is to compare with the existing works, where the state-of-the-art architecture was to consider the window based features and in this regard they also take into account the future (next) word contexts. We devised a RBN (Recurrent BERT Network) architecture to denote a real time detector for language attributes and has outperformed or at least at par with the state-of-the-art model and performance.

5.1.1 Multiple window

Toutanova et al. [7] used Multiple Window for their experiments. Mathematically, we can denote the Multiple Window as the following set of equations.

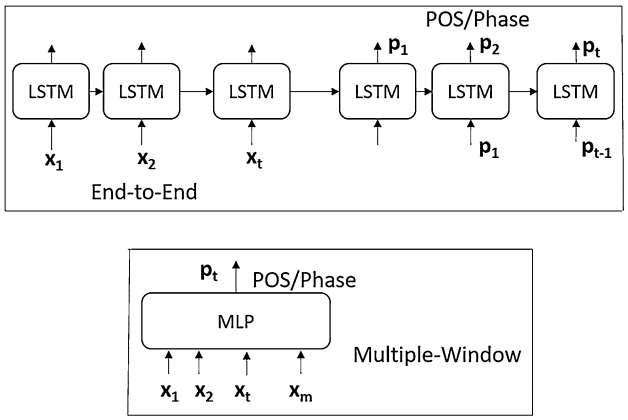


Fig. 8 Different architectures for POS/phrase tagging prediction for languages

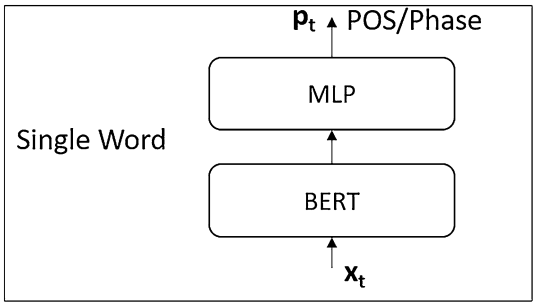


Fig. 9 Singular BERT network (SBN) architectures for POS/phrase tagging prediction for languages. **B** denotes a BERT block

$$\mathcal{P}(\mathbf{x}_t = c \mid \text{MLP}(\mathbf{x}_{t-m}, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+m})) \tag{9}$$

where the window size is 2*m* and we have provided a pictorial overview in Fig. 8.

5.1.2 End-to-end

An end-to-end is used by [6] for Phrase detection of sentences. Mathematical probability likelihood of this model can denoted as the following,

$$\mathcal{P}(\mathbf{x}_t = c \mid \text{LSTM}_d(\text{LSTM}_e(\mathbf{x}_t, \mathbf{h}_{e,t-1}), \mathbf{h}_{d,t-1})) \tag{10}$$

where we have defined the LSTM function set as $\text{LSTM}_e(.)$ for encoder and $\text{LSTM}_d(.)$ for decoder. Figure 8 provided a pictorial description of the end-to-end architecture.

5.1.3 Singular BERT network (SBN)

We introduce a new architecture called Singular BERT Network (SBN) where the representation is for single words. In result tables, this architecture is denoted as "Single Word". Here, we used Word Representation BERT and the mathematical model can be denoted as the following, considering that we do not provide topological relevance among different components and use a multi-layered perceptron function for likelihood estimation.

$$\mathcal{P}(\mathbf{x}_t = c \mid \text{MLP}(\mathbf{B}(\mathbf{x}_t))) \tag{11}$$

where we have defined the multi-layered perceptron function as MLP(.). $\mathbf{B}(.) \rightarrow \mathbb{R}^{768}$ is the BERT function. Figure 9 provided the diagram for SBN architecture and is very basic model.

5.1.4 Recurrent BERT network (RBN)

Recurrent BERT Network (RBN) is devised to improve over the SBM model and we achieved many fold improvements. Here, we establish the topological sequentiality among the representations. In result tables, this architecture is denoted as "Previous Words". Mathematically, RBN with Word Representation BERT can be denoted as the followings,

$$\mathcal{P}(\mathbf{x}_t = c \mid \text{LSTM}(\mathbf{B}(\mathbf{x}_t), \mathbf{h}_{t-1})) \tag{12}$$

where we have defined the LSTM function set as LSTM(.).

$$\mathbf{h}_t = \text{LSTM}(\mathbf{B}(\mathbf{x}_t), \mathbf{h}_{t-1}) \tag{13}$$

$$\mathcal{P}(\mathbf{x}_t = c \mid \text{LSTM}(\mathbf{B}(\mathbf{x}_t), \mathbf{h}_{t-1})) = \mathbf{h}_t \mathbf{W}_h \tag{14}$$

This equation does not include Sentence Representation BERT, which better in terms of performance (at least 1–3% improvement in accuracy) and the mathematical equations can be denoted as the followings,

$$\mathcal{P}(\mathbf{x}_t = c \mid \text{LSTM}(\mathbf{B}(\mathbf{x}_1, \ldots, \mathbf{x}_t), \mathbf{h}_{t-1})) \tag{15}$$

$$\mathbf{h}_t = \text{LSTM}(\mathbf{B}(\mathbf{x}_1, \ldots, \mathbf{x}_t), \mathbf{h}_{t-1}) \tag{16}$$

$$\mathcal{P}(\mathbf{x}_t = c \mid \text{LSTM}(\mathbf{B}(\mathbf{x}_1, \ldots, \mathbf{x}_t), \mathbf{h}_{t-1})) = \mathbf{h}_t \mathbf{W}_h \tag{17}$$

where we have $\mathbf{B}(.)$ as the BERT function, replacing the BERT architecture. Figure 10 provided the diagram of the architecture for Recurrent BERT Network (RBN). In the next section, we will provide some description of the experiments and analysis of the results. The choice of LSTM is because of its popularity and the ability to initialize the model at two different gates, although GRU can be equally good if initialization is not required, however, if Bi-LSTM is considered with future prospects, the performance will be much better and can be a future prospect of this work. The mathematical model for Bi-LSTM (function denoted as bi-LSTM(.)) will be the following and the model will need the whole sentence for detection of the tags.

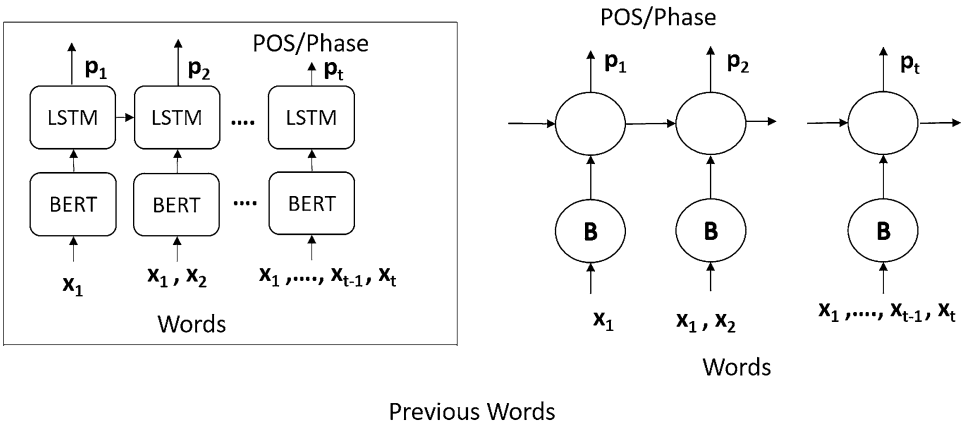


Fig. 10 Recurrent BERT network (RBN) architectures for POS/phrase tagging prediction for languages. Real time model for identification of tags with topological connection with previous predictions. **B** denotes a BERT block

$$\mathcal{P}(\mathbf{x}_t = c \mid \text{bi-LSTM}(\mathbf{B}(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{x}_N), \mathbf{h}_{t-1})) \tag{18}$$

$$\mathbf{h}_t = \text{bi-LSTM}(\mathbf{B}(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{x}_N), \mathbf{h}_{t-1}) \tag{19}$$

$$\mathcal{P}(\mathbf{x}_t = c \mid \text{bi-LSTM}(\mathbf{B}(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{x}_N), \mathbf{h}_{t-1})) = \mathbf{h}_t \mathbf{W}_h \tag{20}$$

where we consider a sentence $\{\mathbf{x}_0, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_N\}$ with N words and each word is denoted as $\mathbf{x}_i$.

6 Results and analysis

This section will discuss the experiments, their results and compare them with previous state-of-the-art works in language attribute prediction. For some comparative study and analysis with our introduced architectures, we used accuracy as metric, though it does not provide any topological information or connections among different semantic structures. However, high accuracy will help in language understanding and the architecture can develop the capability to generalize well utilizing the BERT architecture. We performed our experiments on different forms of data from different sources and with diverse vocabulary. While Word BERT Representation is utilized in some, we utilized Sentence BERT Representation to achieve the state-of-the-art results. Sentence Representation uses different parts of the sentences for prediction of the tags for the end word. Figure 10 provided the diagrammatic overview of the Sentence Representation scheme and the corresponding results are provided in Tables 1 and 2 for POS tags and Phrase respectively.

From Table 1, the results on POS tag prediction for different aspects of data and for different dimensional BERT models denote that the variation in the trained BERT models do not create much variations in the accuracy of detection of the architectures. However, RBN performs much better than SBN and is evident from the design of topological significance of recurrent neural networks and also from the Sentence BERT Representation prospective, where the Sentence BERT Representation is better than the Word BERT Representation. Similarly, Table 2

Table 3 Performance comparison of prediction accuracy for POS (parts-of-speech) tags of languages without fine-tuning of word BERT embedding

Test datasets ⇒	MSCOCO data	Generated caption
# Number of classes ⇒	34 classes	34 classes
	Evaluation accuracy	Evaluation accuracy
Bi-directional LSTM with GloVe [35] embedding	0.9865 (single word)	**0.9982** (single word)
Bi-directional LSTM with BERT [5] embedding	0.9892 (single word)	**0.9988** (single word)

Bold indicates the best results for performance

†BERT-1: uncased L-12 H-768 A-12 embedding is used

Previous words—RBN, single word—SBN

Table 4 Performance comparison of prediction accuracy for phrase tags of languages without fine-tuning of word BERT embedding

Test datasets ⇒	MSCOCO data	Generated caption
# Number of classes ⇒	21 classes	21 classes
	Evaluation accuracy	Evaluation accuracy
Bi-directional LSTM with GloVe [35] embedding	0.9723 (single word)	**0.9845** (single word)
Bi-directional LSTM with BERT embedding [5]	0.9750 (single word)	**0.9885** (single word)

Bold indicates the best results for performance

†BERT-1: uncased L-12 H-768 A-12 embedding is used

Previous words—RBN, single word—SBN

demonstrates the effectiveness of phrase tag prediction based on different dataset and for different BERT models. Next, we compared the Word BERT Representation with word embedding and Table 3 provided results comparing the POS tags and Table 4 for Phrase tags. The comparison in performance between BERT embedding and GloVe word embedding clearly denotes the space of the features, it acquires for specificity and uniqueness. These kinds of embedding can be used for many classification and generative application while detection of the Phrase tag of the words in a sentence with high accuracy can help machine identify the contents. We have tested the case on MSCOCO sentences, as the MSCOCO dataset is meant for language generation and is highly dependent on language embedding and representation. Here, MSCOCO do not contain any special characters and has 21 classes. From the numerical results, we can say that for Phrase tags, our results has outperformed the state-of-the-art results in [41] for PennTree data and [40] for MSCOCO data. While, for POS tag, our performance is at par with [7] and outperformed [6] for PennTree data. Stanford Parser is actually human annotated version of PennTree data and is actually the ground-truth and utilized for annotation of unlabeled datasets through matching from PennTree dataset. Stanford Parser is still the best and is not a prediction or detection model.

We used the uncased version as it is generalized for name-entity recognition (NER) application, where the entities were paid more attention. This uncased version is required as it will help in detection of the special characters and also help in recognition of the ambiguous name-entities. It must be denoted that time comparison could not be made as we use a server with K-80 GPU and other demand based allocated resources like RAM and CPUs (through SLURM). While, the speed depends on how many people have requested jobs completion, the running time of submitted jobs will vary from time to time. In comparison of the embedding performance, BERT embedding for individual words out performed the word embedding in all respects, though the gain is between 0.06 and 0.40% when we consider the MSCOCO data and we use Bi-Directional LSTM, where the whole sentence can be utilized for prediction of POS tags and Phrase for any specific word of the sentences. These results are evident in Tables 3 and 4. Overall, we can conclude that the BERT is a very strong model with much generalization and can be utilized in many NLP applications.

6.1 Statistical analysis of results

In Table 1, we demonstrated through simulation that our model, when used domain adaptation with BERT (cased L-12 H-768 A-12) trained counterpart, has surpassed the performance of state-of-the-art results in [1, 6, 7], and [40] for Part-of-Speech language attributes. When we used the t-distribution to find the confidence interval for our analysis and we found that Mean $\mu = 0.9795$, Standard Deviation $\sigma = 0.328$ and Confidence Interval (CI) to be $U = 0.9819$ and $L = 0.9772$, where we have U and L as the upper limit of the confidence interval and lower limit of the confidence interval. Similarly, in Table 2, we have shown that compared with the works like [40, 41] the Mean $\mu = 0.9964$, Standard Deviation $\sigma = 0.07$ and Confidence Interval (CI) to be $U = 0.9969$ and $L = 0.9959$ For Table 3, we compared the work with [5, 35] for Mean $\mu = 0.9988$, Standard Deviation $\sigma = 0.15$ and Confidence Interval (CI) to be $U = 0.9998$ and $L = 0.9977$ For Table 4, comparing works like [5, 35], Mean $\mu = 0.9885$, Standard Deviation $\sigma = 0.67$ and Confidence Interval (CI) to be $U = 0.9933$ and $L = 0.9837$.

7 Conclusion and future works

In this work, we have provided some qualitative and also qualitative analysis of the BERT model, mainly focusing on applications related to parts-of-speech tagging and phrase tagging, which helps in understanding the parsing tree structure of sentence. This work provided very comparable results and sometimes outperformed the present state-of-the-art works. We can say that for Phrase tags, our results has outperformed the state-of-the-art results in [41] for PennTree data and [40] for MSCOCO data. While, for POS tag, our performance is at par with [7]and outperformed [6] for PennTree data. In conclusion, this must be admitted that BERT is a really powerful representation generator, that can handle large amount of problems single-handed even bypassing some of the data like in media and images, which are more structured, invariant and organized. However, BERT still requires descent amount of resources, memory, time of training and costs for operation and need to be compressed for IOT devices. Future works can be extended with analysis of the other layers of the BERT model and utilization of the ensemble models for the representations, which stand as different prospective of the same thing and also complement of each other for applications related to language understanding.

Compliance with ethical standards

Conflict of interest The author states that there is no conflict of interest.

References

1. Manning C (2017) Stanford parser. nlp.stanford.edu/software/lex-parser.shtml
2. He K et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition
3. Szegedy C et al (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition
4. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
5. Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
6. Vinyals O, Kaiser Ł, Koo T, Petrov S, Sutskever I, Hinton G (2015) Grammar as a foreign language. In: Advances in neural information processing systems, pp 2773–2781
7. Toutanova K, Klein D, Manning CD, Singer Y (2003) Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol 1. Association for Computational Linguistics, pp 173–180
8. Abney S, Schapire RE, Singer Y (1999) Boosting applied to tagging and PP attachment. In: 1999 joint SIGDAT conference on empirical methods in natural language processing and very large corpora
9. Brants T (2000) TnT: a statistical part-of-speech tagger. In: Proceedings of the sixth conference on applied natural language processing. Association for Computational Linguistics
10. Charniak E, Hendrickson C, Jacobson N, Perkowitz M (1993) Equations for part-of-speech tagging. In: AAAI, vol 11, pp 784–789
11. Church KW (1989) A stochastic parts program and noun phrase parser for unrestricted text. In: International conference on acoustics, speech, and signal processing. IEEE
12. Johnson Mark, Geman Stuart, Canon Stephen, Chi Zhiyi, Riezler Stefan (1999) Estimators for stochastic "unificationbased grammars". ACL 37:535–541
13. Klein D, Manning CD (2002) Conditional structure versus conditional estimation in NLP models. In: Proceedings of the ACL-02 conference on empirical methods in natural language processing, vol 10. Association for Computational Linguistics
14. Lee S-Z, Tsujii J, Rim H-C (2000) Part-of-speech tagging based on hidden Markov model assuming joint independence. In: Proceedings of the 38th annual meeting on association for computational linguistics. Association for Computational Linguistics
15. Marshall I (1987) Tag selection using probabilistic methods. The computational analysis of English: a corpus-based approach, pp 42–65
16. Ratnaparkhi A (1996) A maximum entropy model for part-of-speech tagging. In: Conference on empirical methods in natural language processing
17. Thede SM, Harper MP (1999) A second-order hidden Markov model for part-of-speech tagging. In: Proceedings of the 37th annual meeting of the association for computational linguistics
18. Toutanova K, Manning CD (2000) Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th annual meeting of the association for computational linguistics, vol 13. Association for Computational Linguistics
19. Bohnet B, Nivre J (2012) A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In: Proceedings of the 2012 Joint conference on empirical methods in natural language processing and computational natural language learning. Association for Computational Linguistics
20. Carreras X, Collins M, Koo T (2008) TAG, dynamic programming, and the perceptron for efficient, feature-rich parsing. In: Proceedings of the twelfth conference on computational natural language learning. Association for Computational Linguistics
21. Charniak E (2000) A maximum-entropy-inspired parser proceedings of NAACL. Philadelphia, US
22. Chen W et al (2009) Improving dependency parsing with subtrees from auto-parsed data. In: Proceedings of the 2009 conference on empirical methods in natural language processing, vol 2. Association for Computational Linguistics
23. Chen W, Zhang M, Li H (2012) Utilizing dependency language models for graph-based dependency parsing models. In: Proceedings of the 50th annual meeting of the association for computational linguistics: long papers, vol 1. Association for Computational Linguistics
24. Collins M, Roark B (2004) Incremental parsing with the perceptron algorithm. In: Proceedings of the 42nd annual meeting on association for computational linguistics. Association for Computational Linguistics
25. Huang Z, Harper M (2009) Self-training PCFG grammars with latent annotations across languages. In: Proceedings of the 2009 conference on empirical methods in natural language processing, vol 2. Association for Computational Linguistics
26. Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of ICML. Massachusetts, USA, pp 282–289
27. Nivre J, Hall J, Nilsson J (2006) Maltparser: a data-driven parser-generator for dependency parsing. In: Proceedings of LREC, pp 2216–2219
28. Sagae K, Lavie A (2005) A classifier-based parser with linear runtime complexity. In: Proceedings of IWPT. Vancouver, Canada, pp 125–132
29. Shen L, Xu J, Weischedel R (2008) A new string-to-dependency machine translation algorithm with a target dependency language model. In: Proceedings of ACL. Ohio, USA, pp 577–585
30. Sun W, Uszkoreit H (2012) Capturing paradigmatic and syntagmatic lexical relations: towards accurate Chinese part-of-speech tagging. In: Proceedings of ACL, Jeju, Republic of Korea
31. Yamada H, Matsumoto Y (2003) Statistical dependency analysis with support vector machines. In: Proceedings of IWPT. Nancy, France, pp 195–206
32. McDonald R, Crammer K, Pereira F (2005) Online large-margin training of dependency parsers. In: Proceedings of ACL. Ann Arbor, Michigan, pp 91–98
33. Sur C (2019) Survey of deep learning and architectures for visual captioning—transitioning between media and natural languages. Multimed Tools Appl 78(22):32187–32237
34. Sur C (2019) UCRLF: unified constrained reinforcement learning framework for phase-aware architectures for autonomous vehicle signaling and trajectory optimization. Evol Intell 12(4):689–712
35. Pennington J, Socher R, Manning C (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
36. Vaswani A et al (2017) Attention is all you need. Adv Neural Inf Process Syst
37. Marcus MP, Santorini B, Marcinkiewicz MA, Taylor A (1999) Treebank-3. Linguistic Data Consortium, Philadelphia, p 14

38. Dozat T, Qi P, Manning CD (2017) Stanford's graph-based neural dependency parser at the conll 2017 shared task. In: Proceedings of the CoNLL 2017 shared task: multilingual parsing from raw text to universal dependencies, pp 20–30
39. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision. Springer, Cham, pp 740–755
40. Sur C (2018) Representation for language understanding. University of Florida, Gainesville, pp 1–90. https://drive.google.com/file/d/15Fhmt5aM_b0J5jtE9mdWInQPfDS3TqVw. Accessed 16 Oct 2018
41. Zhu M, Zhang Y, Chen W, Zhang M, Zhu J (2013) Fast and accurate shift-reduce constituent parsing. In: Proceedings of the 51st annual meeting of the association for computational linguistics, vol 1: Long Papers, vol 1, pp 434–443