Semantic role labelling using transfer learning model

Archana M Nair¹,² and Bindu K R¹,³

¹Dept of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India
²cb.en.p2aid19004@cb.students.amrita.edu
³j_bindu@cb.amrita.edu

Abstract. Semantic role labelling plays a major role in determining the main predicate and related arguments and its relationship to the predicate. These arguments identify the semantic roles in a sentence such as agent, goal, result, etc. Many approaches have been made to perform the task of semantic role labelling. Many recent works incorporated neural networks making use of syntactic features of the given text such as part of speech tags, POS tagged dependency tree, etc. Other works included the use of BERT along with LSTM and MLP to create semantic role labeling models. In this work, BERT model has been fine tuned to be used as a classifier to detect the main predicate, arguments of the predicate and also the relationship arguments maintain with the predicate in the sentence. Using BERT to build a simple classifier creates an efficient semantic role labelling model. Model performance was assessed using metric accuracy, precision, recall and F1 score. This is achieved by comparing the output of each word with the predicted word and labels as well as the arguments in the prediction of the output.

1. Introduction
The motivation behind semantic role labelling is that it can be applied to infer the meaning of a sentence, which is advantageous to an expansive range of natural language processing applications. The main task of semantic role labelling is to extract the relation between the argument and predicate or the predicate-argument structure from a sentence in order to deduce what was done to whom, the cause and when that event has occurred, etc, from the given sentence. Semantic role labeling is used for downstream applications where logical reasoning is important.

Predicate can be defined as that part of a clause or sentence that contains a verb and stating about the subject. Predicates are mostly verbs, nouns or other forms of verb. An argument refers to the noun phrases in the sentence that are directly related to the verb such as subject or object in the sentence. Understanding arguments defines semantic roles in a sentence and is an important function in the understanding of natural language that is found in many areas such as information extraction, emotional analysis, answering questions, data search, machine translation, etc.

Consider a sentence ‘She shouldn't have expected anything in return from those she had helped.’. The arguments and the predicates in given sentence can be exemplified as ['He': acceptor] ['should': modal] ['n’t': negation] ['have': verb] ['expected': verb] ['anything in return': thing accepted] 'from' ['those she had helped': accepted from]. This classification to arguments and predicates along with the relationship between them gives a better understanding of the given sentence.
Consider the following examples:

- Alice [AGENT] cleaned the kitchen [THEME].
- Alice [AGENT] cleaned the kitchen [THEME] with a mop [INSTRUMENT].
- Mop [INSTRUMENT] cleaned the kitchen [THEME].
- Kitchen [THEME] is cleaned.
- The kitchen [THEME] was cleaned by Alice [AGENT].

In all of the above examples, the syntactic representations differ in each sentence whereas the predicate-argument relationship remains the same in the mentioned sentences. It can be conjectured that the syntactic features alone cannot be used to derive the meaning of a sentence, rather the meaning of a sentence changes depending on the context, structure of the sentence, etc. A layer of abstraction can be imparted using semantic roles that is beyond the syntactic dependency relations like subject and object. The semantic labels bestowed are insensitive to syntactic reformations.

Most of the semantic role labelling models makes use of syntactic and lexical information of the input text such as POS tags, etc. Syntactic information or syntactic features are essential in giving a better insight about the text but syntactic parsers are not available for most languages. Also, they may not be much efficient in cases where there are out-of-domain texts which affects the robustness of the system thereby effecting the performance.

2. Related works

Understanding the intent or the latent meaning of the input text is very important to use it for any applications related to understanding of natural languages [6]. In semantic role labeling, every token in the sentence should be classified into argument relationship with the predicate in the sentence. A variety of algorithms can be used for the classification of arguments’ relationship with the predicate into the set of possible labels such as agent, cause, etc [4].

Semantic role labeling can be performed by making use of syntactic features such as POS tag, lemma embedding and BiLSTM. Syntactic information can be used on advanced labeling models that is compatible with the conflicting k-order of algorithm pruning. This algorithm has been used in [5] for the datasets CoNLL-2008, 2009 benches in both English and Chinese.

Semantic role labeling can be performed separately for dependency and span-based annotations as in [9]. This task is also performed as unified approach [7] which attempts to select appropriate spans for each label and all labelled spans are scored directly based on span representations that is induced from the neural networks. The model greedily selects top scoring labelled spans at the decoding time. This can also be performed for sentences where predicates are unknown. Two separate models can be used for tasks where predicate is known and unknown as in [3]. This paper also implements a complete pipeline for Russian texts. Argument classification is performed using Russian semantically annotated corpus (Frame Bank) trained on a supervised neural network model.

Deep neural networks can be used to decipher the text to classify into the given target classes [2]. Neural network models can also be used in applications such as dependency parsing, part-of-speech tagging, predicate detection and semantic role labelling by incorporating multi-head self-attention with multi-task learning [8]. LISA accepts raw tokens as input, encodes the sequence only once to simultaneously perform parsing, predicate detection and role labelling for all predicates to incorporate syntax.

Semantic role labeling can be performed by using BERT [1] model along with a multilayer perceptron model to surpass the performance of previous model that used a BiLSTM. [1] claims to be the first to successfully perform semantic role label using BERT. A the simple MLP model used in [1] provides better accuracy using strong embedding of content compared to research using language features and other in-depth learning strategies.
3. Methodology
Semantic role labelling can be shaped as four subtasks, including predicate detection, predicate sense disambiguation, argument identification, and argument classification. There are two different annotations for arguments: span-based and dependency-based. In dependency-based annotation for semantic role labelling, identifying the syntactic heads of arguments is the main goal rather than the entire span. Whereas in span-based annotation, arguments are represented as syntactic constituents (or spans). For most of the benchmark datasets for semantic role labelling (e.g. CoNLL 2005, 2009, and 2012) takes in predicate as input along with the sentence for the training and testing process. Predicate is not given as an input to the model for training and testing and only one predicate and its corresponding arguments are predicted at a time. Thus, all four tasks predicate detection, predicate sense disambiguation, argument identification, and argument classification are performed. The dataset COLX 563 has been used for the experiments. This dataset consists of 83 files with different number of sentences consisting of the words in the sentence as each row with the attributes like lemmatized form of the verb, argument that each word formed for each predicate in the sentence.

Predicate is also given as an input to the model in training of the state-of-the-art model for semantic role labelling using BERT \[^{[1]}\]. Using BERT alone for semantic role labelling gives a good performance as the embedding in BERT tried to apprehend the contextual connotation of the words in the sentence rather than just creating vectors based on the tokens unlike other embedding methods available. Here, the ability of BERT to capture the contextual connotation of words are being exploited.

BERT (Bidirectional Encoder Representations from Transformers) is a multi-layer bidirectional Transformer encoder. BERT has been pretrained on a large corpus like Wikipedia and BooksCorpus consisting of 2,500 million and 800 million words respectively. The available models for BERT are: BERT Base and BERT Large. 12 layers of transformer blocks, 12 attention heads and 110 million parameters constitute BERT Base. 24 transformer block layers, 16 attention heads and 340 million parameters constitute the BERT Large model.

This model processes words in relation to all other words in a sentence and not in the given order. Thus, it considers the whole context of the word in that sentence by taking into consideration the word that comes before and after the word in the sentence. The main motivation for the development of this model was to infer the intend behind the search queries in Google search engine.

Other embedding methods like Word2vec and Glove word embeddings produces just one vector (embedding) for each word in the sentence by combining all the different senses of the word into one vector thus they can be called as context-independent models. Models like BERT and ELMo produce different word embedding of the same word in different sentences depending on the context in which the word is used in the sentence. ELMo in contrast to BERT is a character-based model and can handle out of vocabulary words using character convolutions. BERT uses word piece tokenizer to transform the input to sub-words and learns embeddings for these sub-words. BERT has a vocabulary that contains embeddings with 768 features for 30,000 words which is much smaller in comparison with Glove, Word2vec, or ELMo model trained on the same corpus.

![Vocab Token Lengths](image)

**Figure 1.** The diagram shows the distribution of the token length of each token in the BERT vocabulary.
Figure 1 makes it very clear that the BERT vocabulary already contains tokens of varying lengths of tokens. The embedding of BERT can be considered as a dictionary that hashes each word string with the word ID. This hash table contains the word ID with the corresponding vector. The maximum length of tokens possible in a sentence is 512 tokens. Language models using BERT can be developed quickly by fine tuning of the model by adding required layers after the final hidden layer. Some of the applications which BERT has be applied are named entity recognition, POS tagging, question answering, etc.

3.1. Predicate detection and predicate sense disambiguation

The first task for semantic role labelling is predicate detection. This is the process of detecting the predicates in a sentence. A sentence may contain more than one predicate in a sentence, the main task in this step is to determine all the predicates in the sentence. All the sentences are IOB(Inside-Outside-Beginning) tagged before they are given to word piece tokenizer. IOB tagging is a common tagging technique that is incorporated for tagging tokens in a chunk. In this tagging method, the argument of the first word of a chunk is prefixed with the token ‘B-’, all the arguments that follow the first word in the chunk is prefixed with the token ‘I-’ and the tag for tokens in the sentence that do not come under any of the chunks are given as ‘O’. This gives a better representation of tags for arguments in the sentence that consists of more than one token in an argument.

The words in the sentence must be split into tokens, and these tokens must be mapped to their index in the tokenizer vocabulary. The sentences are tokenized using Word-Piece tokenizer that splits some words into sub-tokens. After the words are tokenized and the tokens are converted to the corresponding embeddings, the vector formed should be padded and truncated in order to change all the sentences to the same length. Special tokens should be added in the start and end of the resultant embedding corresponding to the input sentence and attention masks should be added.

BERT uses special tokens like [CLS], [UNK], [PAD], [SEP], [MASK], etc. The special token [SEP] is appended at the end of every sentence. [CLS] is a special token that is added at the beginning of all the sentences to signify classification tasks. The maximum sentence length in BERT is 512 tokens. The input given to BERT for embedding should be of the same length. In order to achieve this, the sentences are padded and truncated to the same length. Attention masks are also added to differentiate between the tokens that are tokens for padding or not. These are given to prevent giving attention on the padded token indices. This is done by giving 0s and 1s. 1s denote tokens that are not masked and 0s denote tokens that are masked.

![Figure 2](image_url)

**Figure 2.** Figure illustrates the distribution of the number of tokens in sentences in the COLX 563 dataset.
Figure 2 illustrates the number of sentences with each length of tokens. This distribution gives an idea about the length of different sentences in the corpus. As mentioned before, the sentences are tokenized using Word-Piece tokenizer, padded and truncated based on the desired length of the sentence. Attention masks are also used since there is very large variation in the length of the sentences in the corpus as shown in the figure above. These masks are given to ignore the attention to the special tokens in the vector for all sentences. The input sentences are shuffled during the training time to give diversity of type of sentences if the order in training set contains similar sentences adjacent to each other. For testing, the sentences are passed sequentially to test the performance of the model trained.

3.2. Argument identification and argument classification
This task is to identify the arguments from the sentence and classify the arguments to what is the relationship of the argument with the predictor. The different arguments contained in the dataset are ARG-0, ARG-1, ARG-2, ARG-3, ARG-MOD, ARG-NEG, V. These arguments unveil the relationship of the argument with the predicate. The ARG-0 denotes the acceptor, ARG-1 to denotes the thing that is accepted, ARG-2 denotes the source from which the thing is accepted, ARG-3 denotes the attribute, ARG-MOD denotes modal, ARG-NEG denotes the negation and V denotes the verb.

Here, token classification class of the BERT model was used for classification of different arguments. The model BERT for token classification fine tunes BERT by wrapping a classification layer on top of it. The classification layer consists of a linear layer that incorporates input from the last hidden layer of the sequence. This model can be used either by only training the final linear layer with all the possible labels or this entire model can be trained. Training the entire model makes it computationally challenging. Training only the linear layer also gives good results. A scheduler was also added in the model to reduce the learning rate through the epochs.

Figure 3. Figure shows the architecture for semantic role labeling task. Here, BERT is used for sequence classification of the labels i.e., the argument relations to the predicate

The above image illustrates architecture of semantic role labelling model. The input sentence is tokenized, padded truncated, added attention masks and these tokens are converted to the embeddings using the BERT model as mentioned before. The output from the last hidden layer of the BERT is given to the linear layer to perform the final classification to the arguments. F1-score and accuracy metrics are used for evaluating the performance of this model. Accuracy gives a token level evaluation
of the model. The model is trained such that the number of training samples in each epoch is determined based on the number of epochs.

\[
F1\ score = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]

Applications like information retrieval commonly use F1-score as a metric for evaluating performance. This measures the accuracy using the statistic recall and precision. Ratio of true positives to all predicted positives constitutes precision. The ratio of true positives to all actual positives constitutes recall. F1-score metric weights recall and precision. The equation for calculating F1-score is as shown in the equation (1).

4. Results and analysis

The dataset used is the COLX 563 dataset. The model was mainly evaluated using the metric mean F1-Score. The dataset consisted of information such as words, predicate indicator to show whether it is a verb or not, the lemmatized formed of the verbs present in the sentence and the different arguments corresponding to the predicates of the sentence.

The original dataset consisted of 83 files with different number of sentences along with the sentence number in that particular file, the word number in that sentence, word, part of speech, tree representation of part of speech, lemmatized form of all the predicates in the sentence and followed by the predicate and the corresponding arguments. The columns are given such that each column contains a predicate and the corresponding arguments. If a sentence contains 5 predicates, then the words corresponding to the sentence will contain 5 columns for each predicate denoting the different arguments present. The IOB tags are added to every word in the sentence such that the tag corresponding to the beginning of the argument is prefixed with ‘B’, the tags of tokens that follow the argument are prefixed with ‘I’ and the tokens that do not form any argument are given tag as ‘O’. This tagging is performed for all the words in the sentence. The dataset after performing tagging contains the fields: Sentence number, words and argument. The dataset contained 1601 unique sentences after preprocessing.

The sequence length is set to 75 as that was the sentence with the maximum length after preprocessing and batch size is set to 32. BERT can support a maximum length of 512 tokens. The BERT has a pretrained tokenizer and a vocabulary that consists of approximately 30000 tokens. The BERT model is based on Word Piece tokenizer, it will split the words into sub-word tokens. For example, for the word separately, the tokenizer splits it to “separate” and “##ly”. Once the tokenization is complete, the tokens and labels are padded and truncated to the desired length. Attention masks are created to ignore the padded elements in the sequence. The validation data is chosen as 10% of the entire data. The sentences during training is shuffled using random shuffler and they are passed sequentially during the testing. BERT contains a class for token classification which is being used in this experiment. This model is formed by fine-tuning BERT model by adding a classification layer on top of the model. The classifier contains a linear layer that takes in the input of the last hidden state of the sequence. The BERT model is loaded and the set of possible labels are also provided.

The model learns faster using GPU. AdamW optimizer along with some weight decay regularization is used to the weight matrices. Fine tuning only the linear layer can also give a good performance model due to context capturing power of BERT. The learning rates are reduced linearly through the epochs. The model gave good results with a very few epochs. The model gives an accuracy score of 74.065 and an F1-score of 66.93.
Consider an input sentence from the COLX 563 dataset “Kevin Dunn has the latest”. This sentence gives an output as follows: [('Kevin', 'B-ARG0'), ('Dunn', 'I-ARG0'), ('has', 'B-V'), ('the', 'B-ARG1'), ('latest', 'I-ARG1'), ('.', 'O')]. The model outputs a list of tuples containing the word in the sentence along with the IOB tagged argument that word forms. To predict the main predicate and its corresponding arguments from a sentence, the given input is first tokenized and converted to tokenized form as performed for other sentences before training the sentences. This tokenized data is then converted to tensors. This tensor is given as input to the model to predict the predicate and its corresponding arguments along with the relationship the argument holds with the predicate.

Figure 4 shows that loss value decreases for subsequent epochs. It can be inferred that more epochs reduce the loss values. The validation accuracy was also showed a similar pattern that shows that the model is not over fitting.

**Table 1.** A table showing the accuracy and F1-score obtained using BERT-Base and BERT-Large models

| Model      | Accuracy | F1-score |
|------------|----------|----------|
| BERT-Base  | 74.06    | 66.93    |
| BERT-Large | 77.28    | 71.42    |

**5. Conclusion and future work**

Based on the study performed, it can be seen that BERT can directly be applied to the sentence to perform the entire process of semantic role labelling. This model makes use of BERT as the main component and achieves good results with just a linear layer and a small set of approximately 1600 sentences. If more sentences are provided for training, the model will give better results. Larger datasets will be providing more diverse training samples and improves performance of the model. This provides a baseline for future research.
References

[1] Peng Shi and Jimmy Lin 2019 Simple BERT models for relation extraction and semantic role labeling, CoRR, vol. abs/1904.05255.

[2] B. R. Amrutha and K. R. Bindu, "Detecting Hate Speech in Tweets Using Different Deep Neural Network Architectures," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 2019, p 923-926, doi: 10.1109/ICCS45141.2019.9065763.

[3] Daniil Larionov, Artem Shelmanov, Elena Chistova and Ivan Smirnov 2019 Semantic role labeling with pretrained language models for known and unknown predicates Proceedings of Recent Advances of Natural Language Processing p 620–630

[4] Vijayan V K, Bindu K R and Parameswaran L 2017 A comprehensive study of text classification algorithms International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017

[5] Shexia He, Zuchao Li, Hai Zhao and Hongxiao Bai 2018b Syntax for semantic role labeling, to be, or not to be In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) p 2061–2071.

[6] T. Rajasundari, Subathra P and Dr. (Col.) Kumar P N 2017 Performance Analysis of Topic Modeling Algorithms for News Articles Journal of Advanced Research in Dynamical and Control Systems (vol. 2017) p 175-183

[7] Zuchao Li, Shexia He, Jiaxun Cai, Zhuosheng Zhang, Hai Zhao, Gongshen Liu, Linlin Li, and Luo Si 2018 A unified syntax-aware framework for semantic role labeling In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing p 2401–2411.

[8] Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum 2018 Linguistically-informed self-attention for semantic role labeling In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing p 5027–5038.

[9] Zuchao Li, Shexia He, Hai Zhao, Yiqing Zhang, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou 2019 Dependency or span, end-to-end uniform semantic role labeling In Proceedings of the 33rd AAAI Conference on Artificial Intelligence

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova 2018 BERT: Pre-training of deep bidirectional transformers for language understanding arXiv:1810.04805.

[11] Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer 2017 Deep semantic role labeling: What works and what’s next In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) p 473–483

[12] Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto 2018 A span selection model for semantic role labeling In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing p 1630–1642