Argument Prediction using Bi-LSTM and GCN on NomBank

Srushti Pawar
New York University
sxp8182@nyu.edu

Simran Makariye
New York University
simranmakariye@nyu.edu

Aryan Shetty
New York University
as17388@nyu.edu

Abstract

Our project tackles Semantic Role Labeling and thereby aims to derive context from sentences by identifying subsequent objects and subjects (ARG0, ARG1 etc.) given a predicate. The given predicates would be labeled with the appropriate sense (drawing from a defined inventory) and then assigned a semantic role. Previous approaches on the Nombank dataset mostly classify arguments using Max Entropy (Jiang and Ng, 2006) and Adaboost among other classifiers. There are also approaches which incorporate Auxiliary Structure Optimization(Liu and Ng, 2007).We treat this as a classification problem and explore the possibility of adapting features previously shown useful in PropBank-based SRL systems. Various NomBank-specific features are explored using their word embeddings, lemma embeddings, and POS embeddings. We aim to achieve SRL by utilizing the similarity between semantic and syntactic representations by exploiting syntactic information of a given sentence. Our approach involves implementing a deep learning approach that utilizes graph convolutional networks (GCNs)(Marcheggiani and Titov, 2017) since GCNs are better suited to model dependency graphs than traditional models. GCNs are neural networks that utilize the properties of their neighbors to induce features of the nodes of a graph. GCNs do not perform well when the candidate arguments are far apart. Hence, we can address this by pairing GCNs with LSTMs. In (Marcheggiani and Titov, 2017) it was also shown that stacking GCNs on top of LSTMs has shown significant improvement on conventional SRL models. Our project aims to generate similar improvement on the Nombank dataset using the above stated approach

1 Introduction

Semantic role labeling (SRL) is an integral NLP task. It is a complex problem that has generated several approaches to its countless subproblems. Semantic role labeling (SRL) can be informally described as the task of discovering who did what to whom in the context of nouns. In the context of linguistics, an argument is a noun phrase that is part of the predicate of a clause and contributes to the meaning of the clause. Our project aims to tackle identifying the predicate-argument structure of a sentence. This would enable us to garner important information and subsequently context across a variety of classes. Since syntactic representations and semantic ones are closely connected we deploy syntactic information in our approach. The argument ARG0 could be generalized to the detection of the sentence's agent/doer and ARG1 would be the 'affected entity'. Our project focuses especially on the classification of ARG1s from the Nombank dataset. We treat this as a classification problem and explore the possibility of adapting features previously shown useful in PropBank-based SRL systems. Various NomBank-specific features are explored using their word embeddings, lemma embeddings, and POS embeddings. We aim to achieve SRL by utilizing the similarity between semantic and syntactic representations by exploiting the syntactic information of a given sentence. This could be achieved by GCN-Graph Convolutional Networks. Our approach involves implementing a deep learning approach that utilizes Bi-LSTMs to achieve said goals.

A BiLSTM (short for bidirectional LSTM) is a sequence processing model that entails taking input in and scanning information in the forward direction as well as in the backward direction. It utilizes two LSTMs to tackle either direction. An LSTM in turn is a special type of Recurrent Neural Network (RNN) that is utilized for learning long-term dependencies. In our approach, we implement BiLSTMs to enhance argument tagging on the NOMBank dataset using specific features. We propose a version of graph convolutional networks (GCNs), a recent class of neural networks operating on graphs, suited to model syntactic dependency graphs. GCNs over syntactic dependency trees are used as sentence encoders, producing latent feature representations of words in a sentence. We observe that GCN layers are complementary to LSTM ones: when we stack both GCN and LSTM layers.

The BiLSTMS would provide additional information that we would use for our main goal - to identify contexts and classify arguments (ARG1). This paper entails that experience.

2 Related Work

The task of Semantic Role Labeling (SRL) is to identify predicate-argument relationships in natural language texts in a domain-independent fashion. In recent years, the availability of large human-labeled corpora such as PropBank (Palmer et al., 2005) and FrameNet (Baker et al., 1998) has made possible a statistical approach of identifying and classifying the arguments of verbs in natural language texts.

NomBank-based automatic Semantic Role Labeling (SRL) is described in (Jiang and Ng, 2006). Max-entropy classification is one of the first models tried on nombank based SRL. This uses max-entropy to predict arguments. This system is often used as a baseline for nombank based tasks to compare improvements and results.

In the paper,(Liu and Ng, 2007) a novel application of Alternating Structure Optimization (ASO) to the task of Semantic Role Labeling (SRL) of noun predicates in NomBank. ASO is a proposed linear multi-task learning algorithm, which extracts the common structures of multiple tasks to improve accuracy, via the use of auxiliary problems. This approach produced better results compared to the systems at that time.

In the paper,(Zhao et al., 2009), an integrated dependency based semantic role labeling system is proposed for English from both NomBank and PropBank. By introducing assistant argument labels and considering many more feature templates, two optimal feature template sets are obtained through an effective feature selection procedure and help construct a high performance single SRL system. From the evaluations on the dataset of CoNLL-2008 shared task, the performance of this system is quite close to the state of the art.

Various SRL systems have been developed on Propbank to predict arguments. We study how techniques used in building the PropBank SRL system can be transferred to developing the NomBank SRL system. In the paper,(Marcheggiani and Titov, 2017), using verb argument dataset, they have stated that GCN layers are complementary to LSTM ones: when stacked both GCN and LSTM layers, a substantial improvement is obtained over an already state-of-the-art LSTM SRL model, resulting in the best reported scores on the standard benchmark (CoNLL-2009) both for Chinese and English. Considering this, this would work well on noun argument dataset such as nombank as well. Hence, our proposed system is derived from the idea of stacking BiLSTM and GCN together to predict arguments.

3 Overview of NomBank Dataset

The NomBank annotation project (Meyers et al., 2004) originated from the NOMLEX nominalization lexicon developed under the New York University Proteus Project. NOMLEX lists 1,000 nominalizations and the correspondences between their arguments and the arguments of their verb counterparts. NomBank frames combine various lexical resources, including an extended NOMLEX and PropBank frames, and form the basis for annotating the argument structures of common nouns. Similar to PropBank, NomBank annotation is made on the Penn TreeBank II (PTB II) corpus. For each common noun in PTB II that takes arguments, its core arguments are labeled with ARG0, ARG1, etc, and modifying arguments are labeled with ARGM-LOC to denote location, ARGMMNR to denote manner, etc.

The Nombank dataset includes different classes of nouns, such as percentage, partitive, ability, relational, and attribute nouns. The dataset consists of features such as token names, POS tags, BIO tags, sentence number, token number, and semantic roles like predicate and support labels. The focus of the current project is on the percentage class of nouns.

4 Pre-processing

As mentioned in the earlier section, the motive was to predict the arguments given the predicate labels on a Nombank dataset. The preprocessing phase started with data exploration.The train dataset comprised of 2174 sentences while the test dataset had 150 sentences. Since deep neural networks accept input in the numeric space, label encoding was applied on the token, POS, and BIO features in order to assign a numeric value to every unique string encountered. Our problem statement relies on the implementation of Bi-LSTMs and Graph Convolutional Networks (GCNs). To implement Bi-LSTMs, we require the data to be in the form of sequences. Since the data given are already tokenized, we merged the tokens based on the sentence ID feature such that each row in our input represents the sentence information altogether. Similar pre-processing was done on the POS and BIO-Tags as well. For the implementation of GCNs, we have employed dependency head features, this feature was extracted with the aid of a publicly available Spacy Model. The spacy model generated the dependency heads for every token in the sentence. The dependency head feature was used predominantly in GCN model execution. Apart from token, POS, and BIO dependency relations, several other features were extracted to run the state-of-the-art SRL models such as Adaboost, Random forest, and Logistic Classifier. These additional features were the distance from the predicate for each token in a sentence, previous two tokens, previous two POS tags, previous two BIO tags, following two tokens, following two POS tags, and following two BIO tags. These models served as the baseline for comparison for our proposed model.

5 Model Architecture

The following section describes the implementation details of our proposed approach.

5.1 Bi-directional long-short term memory (Bi- LSTMs)

Bi-LSTMs have proven to work very effectively in modeling long sequences and thus employing them seems to be a good model to explore for an SRL task. An LSTM takes in an input sequence and returns a hidden representation of the input sequence capturing its neighboring left context information. Since we might be required to induce information from both, the left and the right context, we prefer Bi-LSTMs over LSTMs. Thus a Bi-LSTM network is a concatenation of two LSTMS, one forward pass LSTM to capture left contextual information and one backward pass LSTM to capture the right context. The rest of the paragraph describes how Bi-LSTM was used in generating an encoded representation of our features vectors

In this component, we are primarily working with three layers (Embedding, Bi-LSTM, and LSTM) and the fourth layer, which is the TimeDistributed Dense layer, to output the result. During the pre-processing stage, we transformed the discrete features such as tokens, POS tags, and BIO tags into numeric values. We use these numeric representations of features and extract respective embeddings such as token embeddings, POS embeddings, and BIO embeddings using the Keras framework's Embedding layer. The Embedding layer generates embedding vectors of 64 dimensions. We are using POS and BIO embedding in our model because some of these POS and BIO tags are highly correlated with the argument tokens and thus would help better in the argument prediction task. Thus, the three embedding layers were used for each input vector i.e token, POS, and BIO tags. Once these embeddings are generated for all three input vectors, we concatenate these three embeddings into a single vector and then feed it into our next layer. We have concatenated the three embedding vectors into one vector and have fed them into a Bi-LSTM layer to capture the context in both directions. The output of this layer is again fed to the LSTM layer. In the model architecture, the time distributed dense layers are used to apply a dense operation with a Softmax activation function to each output at every time step. This enables the model to make an argument label prediction for every token in the input data. The architecture is illustrated in Figure 1 for reference.

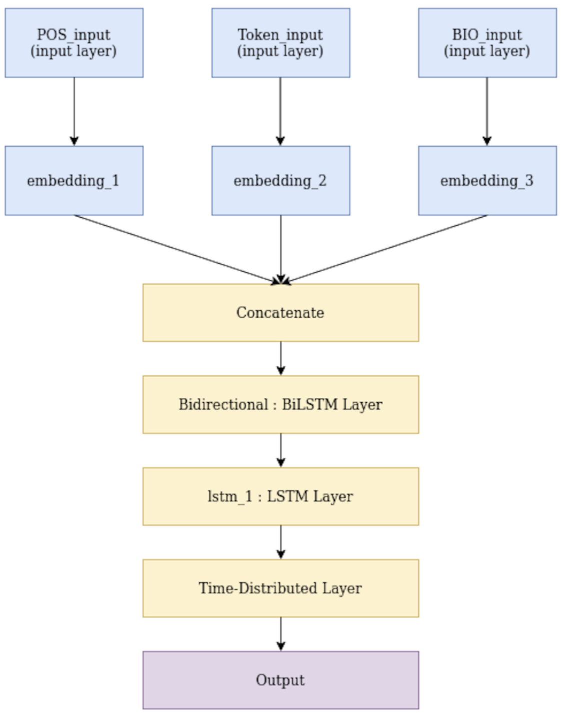

Figure 1: Model Architecture

5.2 Graph Convolutional Network (GCNs)

GCNs are neural networks operating on graphs and induce features of nodes based on the properties of their neighborhoods. Depending on how many layers of convolution are used, GCNs can capture information only about immediate neighbors (with one layer of convolution) or any nodes at most K hops away (if K layers are stacked on top of each other). It is known that syntactic information is highly correlated with semantic one. In our project, we try to explore this idea i.e. try to use the syntactic structure of a sentence to predict the argument of a sentence given a predicate.

In this project, we are leveraging the ability of graph convolutional networks (GCN) to model dependency relations in order to improve the performance of our argument prediction model for semantic role labeling (SRL). While the bi-directional long short-term memory (Bi-LSTM) model has achieved good results, we hope to incorporate syntactic information by running the GCN model in parallel and using a classifier layer to decide which output is more accurate and should be used for the final prediction. The modified architecture is illustrated in the diagram for further understanding.

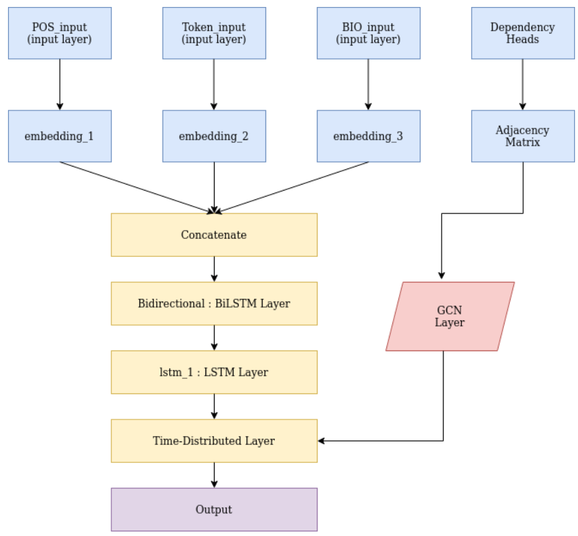

Figure 2: Modified Model Architecture

6 Results

We used the percentage class from the Nombank dataset for training and evaluating our model. Our model was trained on 2174 sentences of varying lengths and numbers of ARG1s, and tested on 150 sentences from Nombank's percentage test dataset. We employed standard splits for the training and validation sets and trained the model for 120 epochs with the goal of accurately predicting ARG1s.

Initially, we stacked an LSTM layer on top of a BiLSTM layer and used a sigmoid activation function, which resulted in a F1-score of 0.80, better than the baseline. To increase recall, which was 0.75, we removed the stacked LSTM layer and achieved a F1-score of 0.81, but precision decreased by 3%.

The modified GCN + Bi-LSTM architecture did not significantly improve the F1 scores compared to the Bi-LSTM model, but it performed better than the initial model that used an LSTM layer after the Bi-LSTM layer. Adding more GCN layers may help to improve the scores, as the GCN can capture dependency information from nodes that are up to K hops away from the predicate node. There is still potential for improvement in the GCN implementation.

We then tested our system using a softmax classifier, which significantly outperformed the previous system. The softmax with LSTM resulted in a precision of 0.88 and a F1-score of 0.82, while the softmax without LSTM achieved the best F1-score so far at 0.84.

Upon analyzing the results, we found that the softmax optimizer significantly improved our scores compared to sigmoid. We also observed that using an LSTM layer improved precision, while removing it increased recall and F1-score. The optimal balance between using an LSTM layer and not using one will depend on the specific dataset and requirements.

Currently, prioritizing F1-score, we conclude that the combination of BiLSTM without an LSTM layer and softmax activation gives the best results. Further improvement could potentially be achieved by increasing the number of training epochs and samples.

RESULTS AND ANALYSIS			
SYSTEM	PRECISION	RECALL	F1-SCORE
BiLSTM+LSTM (***sigmoid*** *activation*)	0.87	0.75	0.80
BiLSTM (*without LSTM layer*) (***sigmoid*** *activation*)	0.84	0.79	0.81
BiLSTM + GCN +LSTM (***softmax*** activation)	0.87	0.76	0.811
BiLSTM+LSTM (***softmax*** *activation*)	0.88	0.77	0.82
BiLSTM (*without LSTM layer*) (***softmax*** *activation*)	0.87	0.81	0.84

Figure 3: Performance Analysis

7 Conclusion

The experimental results indicate that the combination of a bi-directional long short-term memory (Bi-LSTM) model with a softmax classifier achieved the highest performance on the percentage class of the Nombank dataset, with an F1 score of 0.84. Further hyperparameter tuning may improve the results even more.

8 Future Works

In future work, we aim to improve the performance of the modified architecture, which combines a bi-directional long short-term memory (Bi-LSTM) model with a graph convolutional network (GCN). One potential approach is to add more GCN layers, as this can incorporate information from nodes that are up to K hops away from the predicate node. Additionally, we plan to extend our implementation to all classes of nouns in the Nombank dataset and evaluate the performance on these classes.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Zheng Ping Jiang and Hwee Tou Ng. 2006. Semantic role labeling of NomBank: A maximum entropy approach. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 138–145, Sydney, Australia. Association for Computational Linguistics.

Chang Liu and Hwee Tou Ng. 2007. Learning predictive structures for semantic role labeling of NomBank. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 208–215, Prague, Czech Republic. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank project: An interim report. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, pages 24–31, Boston, Massachusetts, USA. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Hai Zhao, Wenliang Chen, and Chunyu Kit. 2009. Semantic dependency parsing of NomBank and PropBank: An efficient integrated approach via a large-scale feature selection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 30–39, Singapore. Association for Computational Linguistics.