Process Knowledge Extraction

Bhavya Ghai, Pranjal Sahu, Sagnik Das
Department of Computer Science, Stony Brook University, Stony Brook, NY-11794
bghai@cs.stonybrook.edu, psahu@cs.stonybrook.edu, sadas@cs.stonybrook.edu

Abstract

This project presents two novel techniques to improve existing semantic role representations to enable better understanding of the language. Firstly, We have tried to retrofit word vectors generated from LSTM model with scientific processes corpus to generate better word embeddings. Second technique uses a semi-supervised model which learns word embeddings using role as context. On testing, We found that first model outperforms existing role labeling models for scientific processes. The second model also performs well even for small annotated datasets. We have concluded by suggesting few ideas for further optimizing this model.

Problem Definition

Semantic Role Labeling (SRL) is a common NLP task that consists of detecting semantic arguments associated with a verb in a sentence and their classification into different roles (such as Agent, Patient,Instrument, etc.). In NLP literature, events are often referred to as predicates and the participants attached to the predicates as its arguments. A predicate and its arguments form a predicate-argument structure. SRL is a task that involves prediction of predicate-argument structure, i.e., both identification of arguments as well as assignment of labels according to their underlying semantic role. Given the sentence The pearls I left to my son are fake an SRL system would conclude that for the verb leave, I is the agent,pearls is the patient and son is the benefactor. Because not all roles feature in each verb the roles are commonly divided into meta-roles. Availability of lexical resources such as Propbank (Martha and Palmer,2002), which annotates text with meta-roles for each argument, has enabled significant progress in SRL systems over the last few years.

In this project, We are studying semantic role labeling problem for Scientific processes corpus. Processes are complex events with many participating entities and interrelated sub-events. Our task is to find classes of entities that are likely to fill key roles within a process namely, the undergoer, enabler, result, and action. Existing SRL systems extract semantic roles from a single sentence. In our case, we have several sentences describing a process and each of these sentences have similar entities filling similar semantic roles consistently. This allows us to design a joint inference method that can promote expectations of consistency amongst the extracted role fillers.

Motivation

Semantic Role Labeling is an important step towards understanding the meaning of a sentence. This work can be directly used for Scientific Process based Question Answering Systems. There will be a positive impact on many practical applications which could take advantage of better language understanding. These applications include Question Answering Systems, Event Summarization, textual entailment, machine translation and dialogue systems, etc. This technique can also be extended for other domains where annotated data is scarce. Semantic Representation can be considered as a higher level of abstraction of syntactic representations. If we could somehow improve existing semantic representations, then we'll be able to understand language better at a higher level. This is the sole purpose of this project.

Literature Review

Different semantic role labeling techniques have been explored by [Naik, 2016].

Contribution

In this project we have tried to incorporate two state of the art approaches to achieve better word embedding which improves the performance of the existing LSTM based process role labeller. The approaches we have used are Retrofitting [Faruqui, 2014] and Dependency-Based Word Embedding [Levy, 2014].
The main contributions of this work are as follows:

- Study the improvements over word embedding models used in process knowledge extraction.
- Applicability of the concept of Retrofitting [] in process role labelling.

- Applicability of semi-supervised word embedding approach in process role labelling.

Description

LSTM

LSTM needs vectorial representation of input in order to perform classification. We use GloVe vectors to represent each word. In cases of phrasal tokens, we multiply the individual word vectors of each word in the phrase to get the phrase vectors. These are then passed through the hidden layer of LSTM to train (and predict) role labels. And as shown in the figure, every phrasal token has one of 5 possible gold labels (None, Input, Result, Enabler, Trigger).

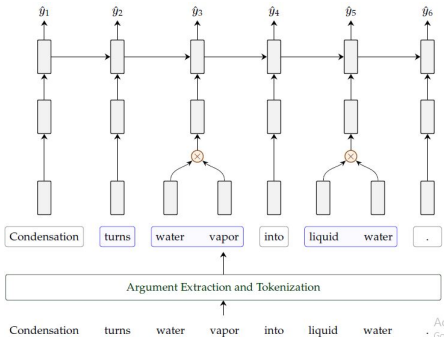

Figure 1: Role Classification using LSTM

Back-propagation is then used to learn the weights at the hidden layers. We applied rmsprop for optimization, which is more suitable for training RNNs than naive stochastic gradient descent, and less sensitive to hyper-parameters compared with momentum methods.

Retrofitting

Retrofitting is used as a post-processing step to improve vector quality and is more modular than other approaches that use semantic information while training. It can be applied to vectors obtained from any word vector training method.

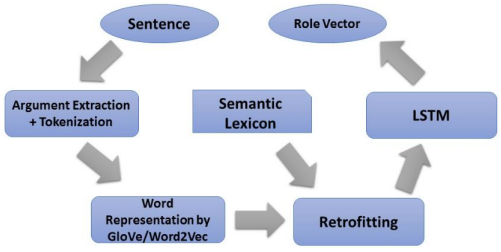

Figure 2: Retrofitting Model

Semi-Supervised Approach

The algorithm used here is a modified skip-gram model negative sampling approach which can deal with arbitrary context [Levy, 2014]. The original work is presented with dependency-based syntactic context but we use the process role annotations as the dependency context.
The main adaptation of this model over the skip-gram word2vec model is that it takes arbitrary context instead of bag-of-words linear contexts. Each word is assigned with its relevant context respective to the sentence. The approach was successful to obtain more focused embedding when syntactic context is considered.

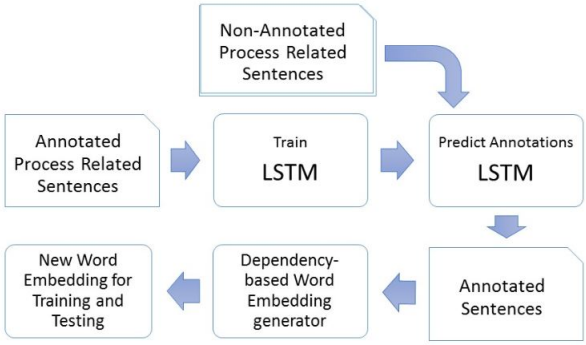

Figure 3: Semi-Supervised Model

For our work we consider the role labelling as the dependency-context. We used the existing [1] implementation to inspect the applicability of this approach in process role labelling. In this approach we perform the task in two phases- firstly we train our LSTM classifier using the annotated dataset we have. Then we feed noisy process related sentences to predict the annotations using this model. In the second phase the annotation word pairs are fed into the word embedding generator [Levy, 2014]. First and second phase steps are depicted in the first and second row of the flow diagram (Fig.).

Evaluation

Semantic Role Modeling can be considered as a multi-class classification problem. We'll be using Precision, Recall and F-Score values to evaluate our model. The goal of this work was to generate more sophisticated word embedding for process knowledge extraction from sentences. We are using a large dataset of process related sentences extracted from (i) New York Regents science exams [Clark, 2015], and (ii) helpteaching.com. The dataset consists 537 sentences and 1205 role fillers. The sentences are related to 54 unique target processes and manually annotated with five different role labelings 'Undergoer', 'Enabler',

[1] https://bitbucket.org/yoavgo/word2vecf

'Action', 'Result' and NONE. The distributions of the role annotations are given in table 1. To use this dataset with LSTM we considered only smaller argument spans and ignore the bigger spans. Thus the data we use has a reduced number of role fillers of 1021.

Table 1: Frequency of each Role

Undergoer	77
Enabler	154
Action	315
Result	194
NONE	465

We are using another large dataset to learn word embeddings from the trained classifier model. This dataset has 6905 sentences for 163 target processes. The dataset is non-annotated. We annotate the sentences using our best model and learn new word embeddings by using the annotations as the context. We use a dependency based approach over Word2Vec [Levy, 2014] to achieve this.

For retrofitting based embedding we used WordNet [Miller, 1995], FrameNet [Baker, 1998], PPDB [Ganitkevitch, 2013] and a smaller science related version WordNet as lexicons. The smaller WordNet corpus was created from a science vocabulary of myvocabulary.com. This has list of 9539 lines of related science words.

Table 2: Parameters for LSTM

Parameter	Value
Learning rate	0.001
Dropout rate	0.5
Epochs	30
Word Embedding Size	100
Depth	1

Our experiments are built upon single depth LSTM. The tuning parameters used in our experiment are listed in table 2. A five fold cross validation was performed for test role mapping to ensure that we are testing the generalization of the approach to the processes which are unseen, we generated the folds in such a way that the processes in the test fold are different from the training. We have generated word embedding by different approaches and compared all the approaches with the raw word embedding approach. In raw word embedding approach we represent each word by GloVe [Pennington, 2014] representation and we use that for training and testing the LSTM. In the retrofitting approach we retrofit the vectors with different lexicons. The semi-supervised approach uses the GloVe vectors for training the model which is used to annotate our non-annotated data. Later we use the newly annotated sentences to generate the word embeddings by a dependency-based approach [Levy, 2014].

The table 3 comapares the performance of the different methods we have used. The first method is the existing embedding approach with simple GloVe vectors. Each GloVe+¡lexicon¿ methods are different settings when we retrofitted GloVe with different lexicons. Semi-Supervised. states our dependency based word embedding approach.

Table 3: Performance Comparison

Word Embedding	Precision	Recall	F1-Score
GloVe	79.75	70.88	68.51
GloVe+FrameNet	70.51	68.17	65.57
GloVe+PPDB	71.67	76.72	69.08
GloVe+SciWordNet	76.83	77.45	72.22
GloVe+WordNet	71.89	78.48	70.17
Semi-Sv.	69.27	59.53	56.45
Semi-Sv.+SciWordNet	65.56	54.49	51.10

We obtained better results over the raw approach using three retrofitted approaches of PPDB, SciWordNet and WordNet. Where the SciWordNet performs the best, having four point F1 increase over the existing approach. The reason of this betterment is the goodness of embeddings achieved after retrofitting. When we retrofit using the SciWordNet, science related words are retrofitted thus the embeddings appear to be more meaningful in the process context. Though the semi-supervised approaches are underperforming in this scenario but we can claim that with a very large dataset it can produce a significant progress. We also claim that the semi supervised approach can be a viable option to choose in this context and can be a good approach for further investigation.

Conclusion & Future Work

In this project, we implemented two techniques for learning Word embeddings i.e. Retrofitting & Semi-supervised learning. We compared our Retrofitting model with existing GloVe based LSTM model. We found that retrofitting provided better F-Score than existing model. We also tested semi-supervised model & we feel that the results could have better if we had larger dataset. In future, Semi-supervised model can be tested with larger dataset which could justify its potential. Furthermore, the hyperparameters can also be tuned for better performance. The number of roles used to represent the sub-processes can be tweaked. We have used 5 roles to represent processes but they mayn't be sufficient for large number of processes. Different techniques such as clustering can be used to find optimal number of roles.

References

Naik, Chetan 2016. An Exploratory Study on Process Representations. Master's Thesis, Dept. of Computer Science, Stony Brook Uni., Stony Brook, NY.

Faruqui, Manaal, et al. "Retrofitting word vectors to semantic lexicons." arXiv preprint arXiv:1411.4166 (2014).

Budai, Kinga, et al. "Learning relations using semantic-based vector similarity." Intelligent Computer Communication and Processing (ICCP), 2016 IEEE 12th International Conference on. IEEE, 2016.

Levy, Omer, and Yoav Goldberg. "Neural word embedding as implicit matrix factorization." Advances in neural information processing systems. 2014.

Guo, Jiang, et al. "Revisiting Embedding Features for Simple Semi-supervised Learning." EMNLP. 2014.

Louvan, Samuel, et al. "Cross-Sentence Inference for Process Knowledge."

Christensen, Janara, Stephen Soderland, and Oren Etzioni. "Semantic role labeling for open information extraction." Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading. Association for Computational Linguistics, 2010.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global Vectors for Word Representation." EMNLP. Vol. 14. 2014.

Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. The Journal of Machine Learning Research, 9:18711874, 2008.

Paul Kingsbury Martha and Martha Palmer. 2002. From treebank to propbank. In In Proceedings of LREC-2002.

Peter Clark. Elementary school science and math tests as a driver for ai: Take the aristo challenge. to appear, 2015.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In Proceedings of NAACL.

Levy, Omer, and Yoav Goldberg. "Dependency-Based Word Embeddings." ACL (2). 2014.

Baker, Collin F., Charles J. Fillmore, and John B. Lowe. "The berkeley framenet project." Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1. Association for Computational Linguistics, 1998.

Miller, George A. "WordNet: a lexical database for English." Communications of the ACM 38.11 (1995): 39-41.