

Graph-Based Semi-Supervised Learning for Natural Language Understanding

Anonymous EMNLP-IJCNLP submission

Abstract

Semi-supervised learning is an efficient method to augment training data automatically from unlabeled data. Development of many natural language understanding (NLU) applications has a challenge where unlabeled data is relatively abundant while labeled data is rather limited.

In this work, we propose transductive graph-based semi-supervised learning models as well as their inductive variants for NLU. We evaluate the approach’s applicability using publicly available NLU data and models. In order to find similar utterances and construct a graph, we use a paraphrase detection model. Results show that applying the inductive graph-based semi-supervised learning can improve the error rate of the NLU model by 5%.

1 Introduction

Natural language understanding (NLU) technology is an important component for a dialog system and is commonly used in voice assistants (e.g., Amazon Alexa, Google Home, Siri). An NLU system takes recognized speech input and produces intent, domain, and slots for the utterance to support the user request (Tur and De Mori, 2011). For example, for a user request “turn off the lights in living room,” the NLU system might generate domain Device, intent Light-Control, and slot values of “off” for OffTrigger and “living room” for Location.

It is crucial for an NLU system to be able to add further support and improve performance in an incremental manner. An efficient method for this is semi-supervised learning (SSL), especially when only small amount of labeled data is available. In contrast with supervised learning algorithms, SSL algorithms can improve their performance by leveraging information in unlabeled data. Some recent results (Laine and Aila, 2017; Miyato et al.,

2019; Tarvainen and Valpola, 2017) have shown that semi-supervised learning could reach performance of purely supervised learning in certain scenarios.

Currently, most NLU models rely on the utterance text and its annotation to learn domain, intent, and slots of the utterance. However, this does not scale to unlabeled data. In this work, we aim to find and represent the relationship between labeled and unlabeled data in a non-Euclidean space, a graph, for SSL. We show that graph-based SSL is a high-performant method which improves an NLU model by leveraging unlabeled data.

In order to represent the labeled and unlabeled data in a graph, we used a paraphrase detection model. Nodes and edges in the graph represent utterances and paraphrase relations respectively. Given the constructed graph, a transductive graph model was applied for node classification, which in our case is intent classification (IC) for each utterance. We used an NLU Slot Gated Model (SGM) (Goo et al., 2018) to obtain slot labels. Experiments on the SNIPS data set show that we can achieve 5% error reduction on the slot error rate.

The rest of the paper is structured as follows: Section 2 reviews work related to our approach. Section 3 describes the graph-based SSL methods we propose in this paper followed by a description of the paraphrase measures used to construct a graph. Section 5 describes the experimental setup. We share results and analysis in Section 6. Section 7 shows conclusions.

2 Related Work

Over the past few years, many deep learning approaches have been extended to NLU tasks—e.g., intent classification and slot filling (Liu and Lane, 2016). Manual annotation is costly. Thus, recent work has turned to SSL in order to achieve similar

performance with much less manually annotated data compared to purely supervised learning.

Aliannejadi et al. (2017) applies graph-based supervised learning of Conditional Random Fields (CRF) for Spoken Language Understanding (SLU) on unaligned data.

Lan et al. (2018) proposes an adversarial multi-task learning method by merging a bidirectional language model (BLM) and a slot tagging model (STM). As a secondary objective, the BLM is used to learn generalized and unsupervised knowledge with abundant unlabeled data and improve the performance of STM on unseen data samples.

Cho et al. (2019) generates paraphrases and uses them to enhance the training set in a semi-supervised learning setting for NLU. The augmented data is used jointly for domain classification, intent classification and slot filling.

The recent rise of neural networks has brought significant advances in a large number of machine learning tasks. While deep learning techniques have achieved huge success, their performance on non-Euclidean data is not as good as on Euclidean data. The complexity of graph structures is a significant challenge to most of existing deep learning algorithms and this complexity has drawn the attention of community to extend deep learning algorithms to graph data which in turn inspired various methods for Graph Neural Networks (GNN) (Kipf and Welling, 2017; Velickovic et al., 2018; Yang et al., 2018; Zhang et al., 2018a; Tran, 2018; Xinyi and Chen, 2019).

GNNs can be applied in a supervised, semi-supervised, or purely unsupervised manner for different tasks. For instance, graph convolutional networks (GCN) (Kipf and Welling, 2017) could be used in a semi-supervised way for node-level classification (Kipf and Welling, 2017), in a supervised way for graph-level classification (Zhang et al., 2018b; Ying et al., 2018; Pan et al., 2016, 2017), and in an unsupervised way for graph embedding (Hamilton et al., 2017; Kipf and Welling, 2016; Pan et al., 2018; Yang et al., 2018).

To the best of our knowledge, our work is the first approach to apply a text-based graph structure for SSL for NLU. We evaluate our method on a publicly available data set in order to show its applicability.

3 Graph Methods

We propose two transductive graph models for semi-supervised learning NLU tasks, Text Graph Convolutional Network (TGCN) and Text Graph Beam Search (TeGrabS), as well as their inductive versions, Pseudo labeling with TGCN (PL-TGCN) and Pseudo labeling with TeGrabS (PL-TeGrabS).

3.1 Transductive Models

In a semi-supervised learning setting, we have the following data sets, $\mathcal{D} = \{X_{train}, X_{unlabeled}, X_{test}\}$. In inductive scenarios, labels of X_{test} and $X_{unlabeled}$ are unknown to model, and the model sees $X_{unlabeled}$ during training but X_{test} is unseen. In transductive scenarios, the model sees X_{test} and $X_{unlabeled}$ at training time.

In our task, transductive models learn paraphrase patterns among utterances from a given graph, then they are applied as auxiliary models in the NLU model pseudo-labeling pipeline. By doing this, we make the determination of input utterances labels become a parametric function of the features and thus obtain inductive variants of transductive models.

3.1.1 TeGrabS

Similar to beam search, TeGrabS is a heuristic method. For a starting node n , the algorithm keeps track of k separate transitions; for each transition at each time step, random sample a node from the current node’s neighbors as the next node. The sampling process can be regarded as a Markov chain: the transition is represented by hop from one node to another with the weight of the edge between two nodes is the transition probabilities.

$$p(n'|n) = W_{n,n'} \quad (1)$$

where n' is a candidate for next node, $W_{n,n'}$ is weight of edge between n and n' . Probability of a whole transition is modelled as equation below:

$$p(n_0, n_1, \dots, n_m) = \prod_{i=0}^{m-1} W_{i,i+1} \quad (2)$$

If a transition does not have available next node candidates (i.e., the current node does not have any neighbors), this transition will be stopped and the beam width is reduced by 1. The beam search will be terminated when all transitions either meet

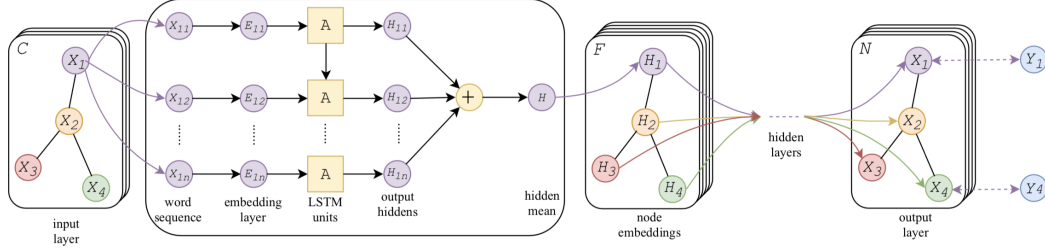


Figure 1: TGCN model architecture.

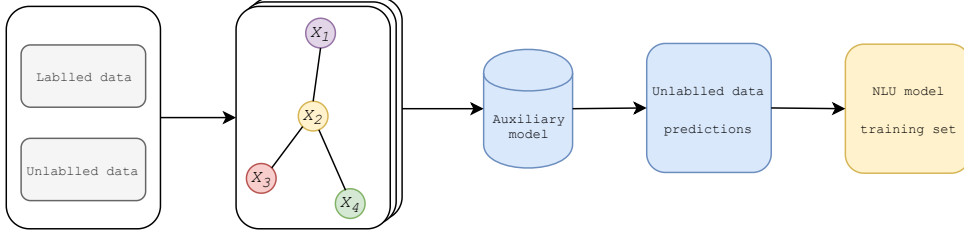


Figure 2: Inductive semi-supervised learning pipeline.

the maximum number of hops limit or are stopped due to the current node not having any neighbors. Pseudocode of TeGrabS is shown as Algorithm 1 in the appendix.

3.1.2 TGCN

GCN has been commonly applied on graph data in recent years (Kipf and Welling, 2017; Hamilton et al., 2017; Zhang et al., 2018b; Ying et al., 2018; Pan et al., 2016, 2017; Kipf and Welling, 2016; Pan et al., 2018; Yang et al., 2018). However, to the best of our knowledge, few researchers have attempted to use it for text graphs.

Here we propose our GCN-based transductive text-graph semi-supervised learning model, TGCN. The model architecture is shown in Figure 1. The input is a text graph including nodes $\{X_1, X_2, \dots, X_i\}$ and edges where each node represents a unique utterance. We use an embedding layer followed by LSTM cells as a feature extractor; X_{ij} , E_{ij} and H_{ij} are token, word embedding, and LSTM hidden state for j -th word in i -th utterance, respectively.

$$E_i = \frac{1}{n} \sum_{w \in X_i} E_w \quad (3)$$

where n is the length of utterance X_i . We compute the average sum of each token's hidden state in utterance as the node feature.

$$H_i = \frac{1}{n} \sum_{j=0}^n H_{ij} \quad (4)$$

Inspired by the original GCN architecture design (Kipf and Welling, 2017), the features are fed into a two-layer graph convolution network. The first graph convolution layer is followed by ReLU units,

$$F^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} F^{(l)} W^{(l)}) \quad (5)$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of the undirected graph with self-connections added, I_N is the identity matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and $W^{(l)}$ is a layer-specific trainable weight matrix. $\sigma(\cdot)$ is an activation function, which is ReLU in our case. $F^{(l)}$ is the matrix of activations in l -th layer, $F^{(0)} = H$.

And the output of second graph convolution layer is passed through a softmax layer to get distribution over all classes per node.

$$Y = \text{Softmax}(F^{(2)}) \quad (6)$$

3.2 Inductive Models

Usage of transductive models is limited to certain test cases that have been seen by model during training. For an NLU system to support user queries, it is crucial to be able to generalize to unseen data. Thus, we use our proposed transductive models as an auxiliary model in an inductive semi-supervised learning pipeline, which is shown as Figure 2.

The input of pipeline is the combination of a few labeled utterances X_{train} and a large amount

of unlabeled data $X_{unlabeled}$ as mentioned in previous section. We apply paraphrase detection model on both X_{train} and $X_{unlabeled}$ to find pairwise paraphrase relations between utterances. Details of the paraphrase detection model is given in Section 4.

For each utterance, we first find all its paraphrases as adjacency lists. We then build graph based on the adjacency lists. Transductive models (TeGrabS, TGCN, etc.) are applied as auxiliary model as shown in the graph, to predict labels for unlabeled data $X_{unlabeled}$ from both labeled data and graph structure. Finally, predictions of $X_{unlabeled}$ are fed into NLU model training set to re-train the NLU model and test on the unseen test set.

Based on the transductive TeGrabS and TGCN, here we propose their inductive variants, named Pseudo-Labeling with Text-Graph Beam Search (**PL-TeGrabS**) and Pseudo-Labeling with TextGCN (**PL-TGCN**) where TeGrabS and TGCN are used as auxiliary models in the aforementioned inductive semi-supervised learning pipeline, respectively.

4 Paraphrase Detection for Graph Construction

In this work, we leveraged paraphrase learning to find potential paraphrases in the data set and construct a graph. In real-world applications, this could be obtained from analyzing usage pattern, such as repetition or rephrase of user requests. In this work, we apply the paraphrase classification model on the NLU utterances to retrieve the paraphrase pairs within the data. We then construct the graph where paraphrases are connected.

In this section, we explain how the paraphrase model is trained as well as the construction of the graph.

4.1 Paraphrase Embedding Learning

In order to obtain embedding for paraphrases, we used a word averaging model. In this approach, once a word embedding matrix is learned, we average them over a sequence:

$$g(x) = \frac{1}{n} \sum_i^n W_w^{x_i} \quad (7)$$

where W_w is a word embedding matrix. Parameters are learned by minimizing an objective func-

tion with a margin, as described in [Wieting et al. \(2016a\)](#).

For embedding learning, we used the PPDB-S data set ([Pavlick et al., 2015](#)), which comprises 1.5 million paraphrase pairs.

4.2 Paraphrase Classification

Using the embeddings, we trained a model that outputs a score as an indicative for the pair to be paraphrases of each other. In the model, we used the embedding approach described in Section 4.1 and obtain an embedding e for each utterance. For a pair of utterances u_1 and u_2 , we combine their embeddings in the following way:

$$h = [e_{u_1}, e_{u_2}, |e_{u_1} - e_{u_2}|, e_{u_1} \times e_{u_2}] \quad (8)$$

where we concatenate each utterance’s embedding, element-wise difference and product between the two.

We then used a fully-connected network to output the probability for two utterances being paraphrases. We used two 100-dimension hidden layers with ReLU activation ([Nair and Hinton, 2010](#)) for the task. Further details of the embedding learning and classification model can be found in *anonymous*.

To train the paraphrase classification model, we used a back-translated paraphrase corpus ([Wieting and Gimpel, 2017](#)). For positive examples, we randomly selected 1.4M paraphrase pairs from the corpus. For negative examples, we randomly pair up utterances within the corpus so that the utterances in the pair are not paraphrases of each other. In the end, we obtained 2.8M pairs of data, with balanced positive and negative labels.

Using the method above leads to an F-score of 98.39 on a test set with balanced 20K pairs. Note that the performance of classification model is expected to regress when applied on the target task data, due to the domain mismatch between the sizable, publicly available paraphrase corpus ([Wieting and Gimpel, 2017](#)) and the NLU task data ([Coucke et al., 2018](#)).

In this work, we consider paraphrase pairs whose score returned by the model is higher than a threshold $\theta = 0.99$. Detailed statistics on the constructed graph can be found in Section 6.

5 Experimental Setup

In this section, we discuss the experimental setup used in this work. First, we describe the data sets

Intent	Utterances with slot labels
searchFlight	find me a flight from [origin](Paris) to [destination](New York)
searchFlight	I need a flight leaving [date](this weekend) to [destination](Berlin)
searchFlight	show me flights to go to [destination](new york) leaving [date](this evening)

Table 1: Examples of the SNIPS Dataset.

	No. Utterances
Train	1,310
SSL candidate (unlablled)	11,774
Dev	700
Test	700

Table 2: SNIPS data statistics.

used for training and evaluating the suggested SSL approach. We also describe the NLU model used in this work, followed by description on the comparative systems.

5.1 Data

To evaluate the proposed model, experiments were performed on SNIPS dataset (Coucke et al., 2018), which is collected from the SNIPS personal voice assistant. This data comes with a pre-cut train, dev, and test sets, which contain 13,084, 700 and 700 utterances respectively. There are 72 slot labels and 7 intent types for the training set. Example utterances from the SNIPS data is shown in Table 1.

Designing a real-world application often faces with a challenge where there is an abundant amount of unlabeled data, but only a limited amount of labeled data. In order to simulate this scenario, we split the training data portion further, so that only 10% of the labeled training data is used for model training. The rest 90% of the labeled training data would be considered as candidates for SSL. Thus, we did not rely on the annotated labels in the SSL portion of the training data, but consider this as an unlabeled data and try to learn them from SSL process. Overall data statistics is given in Table 2.

5.2 NLU System Description

The NLU model we used is a Slot-Gated attention-based bidirectional long short-term memory Model (SGM) (Goo et al., 2018). In our setting, the full-attention setup was used, which achieves the best performance in the paper. We used the default hyper-parameters from the code base.

5.3 Comparative systems

In order to explore the effectiveness of the SSL approaches we discuss in this work, we rely on two comparative systems. First system (“Baseline”) is trained only on training data in Table 2, without applying SSL. In the second system Pseudo Labelling Baseline (“PL-Baseline”), we applied SSL, more specifically, pseudo labelling, but without leveraging the graph structure. For this system, we first trained the Baseline then infer labels for unlabeled data $X_{unlabeled}$ with Baseline. After giving $X_{unlabeled}$ pseudo labels, we put it into NLU model training set, along with X_{train} , to re-train NLU model.

5.4 Evaluation

We report IC accuracy, F1 score, slot F1 and Slot Error Rate (SER) as metrics to measure the performance of the models.

SER is a metric used to combine intent classification accuracy and the slot classification accuracy in a single score. It is defined as:

$$SER = \frac{S + I + D}{S + D + C}$$

where S is number of substitution errors for intents or slots, I is the number of insertion errors for intents or slots, D is the number of deletion errors for intents or slots, and C is the number of correct slots and intents.

6 Results

In this section, we first discuss the constructed graph using paraphrase measures. We then report the inductive SSL performance with graph methods as auxiliary models.

6.1 Constructed Graph

The paraphrase graph is built based on the train and SSL candidate set as we discussed in previous sections. Figure 3 is a part of the constructed graph, from which we can observe paraphrase patterns among connected components. We have also confirmed that neighbors in this excerpt share the

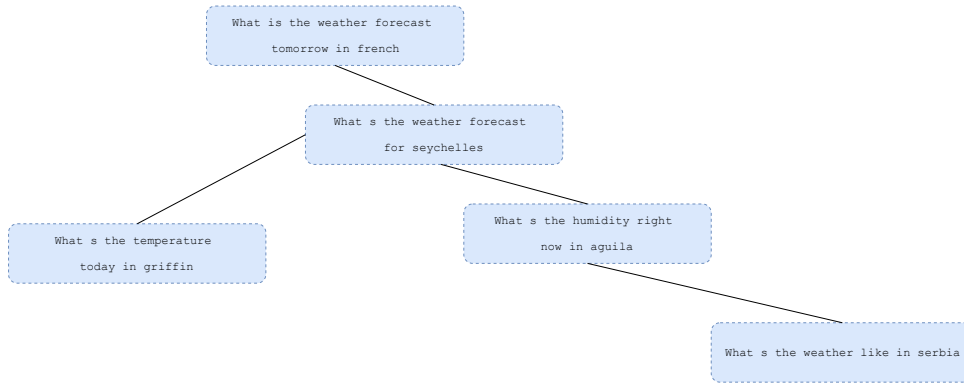


Figure 3: Excerpt of constructed graph using SNIPS data. Graph is constructed to represent utterance similarity using a paraphrase measure.

Model	IC Acc.	IC F1	Slot F1	SER
Baseline*	92.57	92.52	59.30	67.43
PL-Baseline	92.86	92.03	59.61	68.71
PL-TGCN	93.14	92.48	63.95	63.86
PL-TeGrabS	73.43	72.95	58.02	73.71

Table 3: Transductive results on SNIPS data. *Baseline for Snips dataset is Slot Gated Modeling (Goo et al., 2018).

same intent (`GetWeather`) as well as similar slots.

The whole graph contains 12,895 nodes (which indicates that there are duplicates in SNIPS dataset), 52,876 edges when we set the paraphrase threshold $\theta = 0.99$.

6.2 Inductive Results

We evaluated baselines and our proposed models on the SNIPS dataset. Intent classification and slot filling experiment results are shown in Table 3.

We can observe that PL-TGCN outperformed other models on intent classification accuracy. However, this model is slightly defeated by baseline on intent classification F1-score. Our analysis revealed that PL-TGCN tends to predict more utterances into `AddToPlaylist` instead of `PlayMusic`, compared to baseline. Since `AddToPlaylist` is the biggest intent class in test set (124/700), more predictions in this class will certainly raise accuracy, but will do little harm to F1-score, given that we are reporting F1-score averaged from all classes. However, though Baseline did good job in not assigning more false positives to `AddToPlaylist`, it is more likely to assign utterances in `AddToPlaylist` to other classes, which is actually not good. Therefore, we can conclude that PL-TGCN achieved best per-

formance on intent classification in general. It boosted the performance of slot filling through slot gate in SGM, leading to a great reduction on SER.

7 Conclusion

In this work, we proposed transductive graph-based semi-supervised learning models as well as their inductive variants for NLU. In order to find similar utterances and construct a graph, we use a paraphrase detection model. To the best of our knowledge, our work is the first approach to apply text based graph structure for an SSL of NLU. We evaluate our method’s applicability on publicly available data and model. Results show that applying the inductive graph-based semi-supervised learning can reduce the error rate of the NLU model by 5%.

References

- Mohammad Aliannejadi, Masoud Kiaeeha, Shahram Khadivi, and Saeed Shiry Ghidary. 2017. [Graph-based semi-supervised conditional random fields for spoken language understanding using unaligned data](#). *CoRR*, abs/1701.08533.
- Eunah Cho, He Xie, and William M Campbell. 2019. [Paraphrase generation for semi-supervised learning in nlu](#). In *Proceedings of the Workshop on Meth-*

- 600 *ods for Optimizing and Evaluating Neural Language*
601 *Generation*, pages 45–54. 650
- 602 Alice Coucke, Alaa Saade, Adrien Ball, Théodore 651
603 Bluche, Alexandre Caulier, David Leroy, Clément 652
604 Doumouro, Thibault Gisselbrecht, Francesco Calt- 653
605 agirone, Thibaut Lavril, Maël Primet, and Joseph 654
606 Dureau. 2018. [Snips voice platform: an embedded](#) 655
607 [spoken language understanding system for private-](#) 656
608 [by-design voice interfaces](#). *CoRR*, abs/1805.10190. 657
- 609 Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li 658
610 Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun- 659
611 Nung Chen. 2018. [Slot-gated modeling for joint slot](#) 660
612 [filling and intent prediction](#). In *Proceedings of the* 661
613 *2018 Conference of the North American Chapter of* 662
614 *the Association for Computational Linguistics: Hu-* 663
615 *man Language Technologies, NAACL-HLT, New Or-* 664
616 *leans, Louisiana, USA, June 1-6, 2018, Volume 2* 665
617 *(Short Papers)*, pages 753–757. 666
- 618 William L. Hamilton, Zitao Ying, and Jure Leskovec. 667
619 2017. [Inductive representation learning on large](#) 668
620 [graphs](#). In *Advances in Neural Information Process-* 669
621 *ing Systems 30: Annual Conference on Neural In-* 670
622 *formation Processing Systems 2017, 4-9 December* 671
623 *2017, Long Beach, CA, USA*, pages 1024–1034. 672
- 624 Thomas N. Kipf and Max Welling. 2016. [Variational](#) 673
625 [graph auto-encoders](#). *CoRR*, abs/1611.07308. 674
- 626 Thomas N. Kipf and Max Welling. 2017. [Semi-](#) 675
627 [supervised classification with graph convolutional](#) 676
628 [networks](#). In *5th International Conference on* 677
629 *Learning Representations, ICLR 2017, Toulon,* 678
630 *France, April 24-26, 2017, Conference Track Pro-* 679
631 *ceedings*. 680
- 632 Samuli Laine and Timo Aila. 2017. [Temporal ensem-](#) 681
633 [bling for semi-supervised learning](#). In *5th Inter-* 682
634 *national Conference on Learning Representations,* 683
635 *ICLR 2017, Toulon, France, April 24-26, 2017, Con-* 684
636 *ference Track Proceedings*. 685
- 637 Ouyu Lan, Su Zhu, and Kai Yu. 2018. [Semi-supervised](#) 686
638 [training using adversarial multi-task learning for](#) 687
639 [spoken language understanding](#). In *2018 IEEE In-* 688
640 *ternational Conference on Acoustics, Speech and* 689
641 *Signal Processing, ICASSP 2018, Calgary, AB,* 690
642 *Canada, April 15-20, 2018*, pages 6049–6053. 691
- 643 Bing Liu and Ian Lane. 2016. [Attention-based recur-](#) 692
644 [rent neural network models for joint intent detection](#) 693
645 [and slot filling](#). In *Interspeech 2016, 17th Annual* 694
646 *Conference of the International Speech Communica-* 695
647 *tion Association, San Francisco, CA, USA, Septem-* 696
648 *ber 8-12, 2016*, pages 685–689. 697
- 649 Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, 698
650 and Shin Ishii. 2019. [Virtual adversarial training:](#) 699
651 [A regularization method for supervised and semi-](#)
652 [supervised learning](#). *IEEE Trans. Pattern Anal.*
653 *Mach. Intell.*, 41(8):1979–1993.
- Vinod Nair and Geoffrey E Hinton. 2010. [Rectified](#)
[linear units improve restricted boltzmann machines](#).
In *Proceedings of the 27th international conference*
on machine learning (ICML-10), pages 807–814.
- Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina
Yao, and Chengqi Zhang. 2018. [Adversarially reg-](#)
[ularized graph autoencoder for graph embedding](#).
In *Proceedings of the Twenty-Seventh International*
Joint Conference on Artificial Intelligence, IJCAI
2018, July 13-19, 2018, Stockholm, Sweden., pages
2609–2615.
- Shirui Pan, Jia Wu, Xingquan Zhu, Guodong Long,
and Chengqi Zhang. 2017. [Task sensitive feature](#)
[exploration and learning for multitask graph classi-](#)
[fication](#). *IEEE Trans. Cybernetics*, 47(3):744–758.
- Shirui Pan, Jia Wu, Xingquan Zhu, Chengqi Zhang,
and Philip S. Yu. 2016. [Joint structure feature explo-](#)
[ration and regularization for multi-task graph classi-](#)
[fication](#). *IEEE Trans. Knowl. Data Eng.*, 28(3):715–
728.
- Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch,
Benjamin Van Durme, and Chris Callison-Burch.
2015. [Ppdb 2.0: Better paraphrase ranking, fine-](#)
[grained entailment relations, word embeddings, and](#)
[style classification](#). In *Proceedings of the 53rd An-*
ual Meeting of the Association for Computational
Linguistics and the 7th International Joint Confer-
ence on Natural Language Processing (Volume 2:
Short Papers), volume 2, pages 425–430.
- Antti Tarvainen and Harri Valpola. 2017. [Mean teach-](#)
[ers are better role models: Weight-averaged consis-](#)
[tency targets improve semi-supervised deep learning](#)
[results](#). In *Advances in Neural Information Process-*
ing Systems 30: Annual Conference on Neural In-
formation Processing Systems 2017, 4-9 December
2017, Long Beach, CA, USA, pages 1195–1204.
- Phi Vu Tran. 2018. [Learning to make predictions on](#)
[graphs with autoencoders](#). In *5th IEEE Interna-*
tional Conference on Data Science and Advanced
Analytics, DSAA 2018, Turin, Italy, October 1-3,
2018, pages 237–245.
- Gokhan Tur and Renato De Mori. 2011. *Spoken lan-*
guage understanding: Systems for extracting seman-
tic information from speech. John Wiley & Sons.
- Petar Velickovic, Guillem Cucurull, Arantxa Casanova,
Adriana Romero, Pietro Liò, and Yoshua Bengio.
2018. [Graph attention networks](#). In *6th Inter-*
national Conference on Learning Representations,
ICLR 2018, Vancouver, BC, Canada, April 30 - May
3, 2018, Conference Track Proceedings.
- John Wieting, Mohit Bansal, Kevin Gimpel, and Karen
Livescu. 2016a. [Towards universal paraphrastic sen-](#)
[tence embeddings](#). *International Conference on*
Learning Representations (ICLR).

700	John Wieting and Kevin Gimpel. 2017. Parant-50m:	750
701	Pushing the limits of paraphrastic sentence embed-	751
702	dings with millions of machine translations. <i>arXiv</i>	752
703	<i>preprint arXiv:1711.05732</i> .	753
704	Zhang Xinyi and Lihui Chen. 2019. Capsule graph	754
705	neural network . In <i>7th International Conference</i>	755
706	<i>on Learning Representations, ICLR 2019, New Or-</i>	756
707	<i>leans, LA, USA, May 6-9, 2019</i> .	757
708	Zhilin Yang, Junbo Jake Zhao, Bhuwan Dhingra,	758
709	Kaiming He, William W. Cohen, Ruslan Salakhut-	759
710	dinov, and Yann LeCun. 2018. Glomo: Unsuper-	760
711	vised learning of transferable relational graphs . In	761
712	<i>Advances in Neural Information Processing Systems</i>	762
713	<i>31: Annual Conference on Neural Information Pro-</i>	763
714	<i>cessing Systems 2018, NeurIPS 2018, 3-8 December</i>	764
715	<i>2018, Montréal, Canada.</i> , pages 8964–8975.	765
716	Zhitao Ying, Jiaxuan You, Christopher Morris, Xi-	766
717	ang Ren, William L. Hamilton, and Jure Leskovec.	767
718	2018. Hierarchical graph representation learning	768
719	with differentiable pooling . In <i>Advances in Neural</i>	769
720	<i>Information Processing Systems 31: Annual Con-</i>	770
721	<i>ference on Neural Information Processing Systems</i>	771
722	<i>2018, NeurIPS 2018, 3-8 December 2018, Montréal,</i>	772
723	<i>Canada.</i> , pages 4805–4815.	773
724	Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Ir-	774
725	win King, and Dit-Yan Yeung. 2018a. Gaan: Gated	775
726	attention networks for learning on large and spa-	776
727	tiotemporal graphs . In <i>Proceedings of the Thirty-</i>	777
728	<i>Fourth Conference on Uncertainty in Artificial Intel-</i>	778
729	<i>ligence, UAI 2018, Monterey, California, USA, Au-</i>	779
730	<i>gust 6-10, 2018</i> , pages 339–349.	780
731	Muhan Zhang, Zhicheng Cui, Marion Neumann, and	781
732	Yixin Chen. 2018b. An end-to-end deep learning	782
733	architecture for graph classification . In <i>Proceed-</i>	783
734	<i>ings of the Thirty-Second AAAI Conference on Ar-</i>	784
735	<i>tificial Intelligence, (AAAI-18), the 30th innovative</i>	785
736	<i>Applications of Artificial Intelligence (IAAI-18), and</i>	786
737	<i>the 8th AAAI Symposium on Educational Advances</i>	787
738	<i>in Artificial Intelligence (EAAI-18), New Orleans,</i>	788
739	<i>Louisiana, USA, February 2-7, 2018</i> , pages 4438–	789
740	4445.	790
741		791
742		792
743		793
744		794
745		795
746		796
747		797
748		798
749		799

8 Appendix

800	850
801	851
802	852
803	853
804	854
805	855
806	856
807	857
808	858
809	859
810	860
811	861
812	862
813	863
814	864
815	865
816	866
817	867
818	868
819	869
820	870
821	871
822	872
823	873
824	874
825	875
826	876
827	877
828	878
829	879
830	880
831	881
832	882
833	883
834	884
835	885
836	886
837	887
838	888
839	889
840	890
841	891
842	892
843	893
844	894
845	895
846	896
847	897
848	898
849	899

Algorithm 1 Graph Beam Search

Require: Relation graph G **Require:** Labelled nodes $N_{labelled}$ **Require:** Unlabelled nodes $N_{unlabelled}$ **Require:** Beam width k

```

1: for node  $n$  in  $N_{unlabelled}$  do
2:    $p_t(n_i, n_j) \leftarrow Weight_{Edge_{i,j}}$ 
3:    $beams \leftarrow \emptyset$ 
4:    $bestBeams \leftarrow \emptyset$ 
5:    $p_b(\emptyset, 0) \leftarrow 1$ 
6:   for  $t = 1 \dots T$  do
7:     if  $beams \neq \emptyset$  then
8:        $bestBeams \leftarrow \arg \max_k (beams, \frac{\log(p_b(beam, t-1))}{length_{beam}})$ 
9:        $beams \leftarrow \emptyset$ 
10:    for  $beam$  in  $bestBeams$  do
11:       $successors \leftarrow$  labelled neighbors of  $currentNode$ 
12:      if  $successors = \emptyset$  then
13:        continue
14:      for  $suc$  in  $\arg \max_k (successors, p_t(currentNode, suc))$  do
15:         $beam \leftarrow beam \cup successor$ 
16:         $p_b(beam, t) \leftarrow p_b(beam, t-1) + p_t(currentNode, suc)$ 
17:        if  $outDegree(suc) = 0$  then
18:           $k \leftarrow k - 1$ 
19:        if  $all(length_{beam} = 0)$  or  $k = 0$  then
20:          break
21:     $bestBeam \leftarrow \arg \max (bestBeams, \frac{\log(p_b(beam, T))}{length_{beam}})$ 
22:     $domain \leftarrow \max\{\dots\}\{n.domain \text{ for } n \text{ in } bestBeam\}$ 
23:     $intents \leftarrow \{n.intent \text{ for } n \text{ in } bestBeam\}$ 
24:    for each  $intent$  in  $intents$  do
25:      if  $intent$  does not belong to  $domain$  then  $intents \setminus intent$ 
26:     $intent \leftarrow \max\{\dots\}intents$ 

```
