Improving Semantic Relation Prediction using Global Graph Properties

Eric Liang
esliang@andrew.cmu.edu

Yun Cheng
yuncheng@andrew.cmu.edu

Yuxin Pei
yuxinp@andrew.cmu.edu

1 Introduction/Motivation

In the past several years natural language processing (NLP) has been a rapidly developing field with various different applications. One of the crucial tasks in NLP, that we will also focus on, is trying to predict relationships between different words (semantic relation prediction) in the English language. If semantic relation prediction is able to be improved, this could help improve many different applications of NLP, such as sentiment analysis. One way semantic relation prediction can be done is if we imagine representing the relationship between words as a directed graphical model and utilize global graph properties. We therefore are trying to solve the task of improving semantic relation prediction using different global graph properties. Specifically we hope to try to improve semantic relation prediction by not only utilizing different global graph features, but also by taking advantage of the graphical model structure and use different sampling methods and proposal distributions to improve parameter estimation for this prediction task. For this problem, the data we will be using comes from the WN18RR dataset [3]. This dataset is an improved version of the WN18 dataset [1] that is a popular semantic relation prediction based on WordNet 3.0 (a semantic graph database) [4]. The dataset consists of 11 different semantic relation types and over 40,000 entities(a word or group of words that refer to the same object). Thus from the dataset we can define tuples that consist of a semantic relationship (edges in our graph), and two entities (nodes) that are involved with this semantic relationship. Thus our prediction task will take as input the relationship and only one of the two entities from the tuple. Our problem's output will be the other entity that is not given to us as input. In order to analyze our results we plan on using several different metrics that are commonly used in relation prediction tasks. These metrics are also used in our baseline which is based off of previous works (which we will discuss further in our background), so we keep the same metrics for easy comparison with our results. The metrics we plan on using to analyze our results are the Mean Reciprocal Rank and the proportion of true entities found in the top k of the ranking lists, where $k \in \{1, 10\}$ (we refer to this metric as H@k).

2 Background/Literature review

There have been papers in the past that have looked into the task of predicting semantic relations. One recent work from [1], looks into converting word embeddings in different phrases in order to model and predict semantic relations between words. While their problem is similar to ours, they state that perhaps there are other ways to structure semantic relations such that the prediction of semantic relations can be improved. We noted this lack of structuring semantic relations in their work and saw in our research that in other NLP areas, for example predicting text similarity, it was possible to have improved metrics when structuring texts as graphical models [7]. Although their problem is different, it still uses similar concepts as our problem to build the graphical model. Thus this motivates our problem where, we hope to improve on semantic relation prediction by modeling semantic relations as graphical models, such that by using global graph properties, we can improve on previous results. However one aspect of our problem that we have not addressed is how we represent semantic relations as a graphical model (a directed model). One paper we researched represents semantic relations as a graphical model and solved the problem of automatically detecting relations from text that has

Preprint. Under review.

been syntactically parsed [6]. This motivates how we define our graphical model, specifically in our case the words in the semantic relationships would be the nodes, and the edges would be different semantic relations. For example one semantic relation is a hypernym ("is-a") relationship between the words "purple" and "color". Since purple is a color, there would be an edge going from the node purple to the node color which represents the hypernym relation. [5] combines the idea of graphical models and applies it for semantic relation prediction, similar to our problem. Their approach uses an exponential random graph model (ERGM) to do parameter estimation and prediction for the semantic relation graphical model. Although they were able to achieve better results compared to other methods that did not use graphical models, we note that there are still improvements that could be made to their algorithm. For their ERGM, they utilize a proposal distribution that is based off of only three semantic relation specific association operators. Having a limit on these operators causes for their proposal distribution to possibly not be the most suitable proposal distribution. In addition when normalizing in the ERGM, they use Monte Carlo MLE (MCMLE) and mention sampling negative samples, without trying any other types of sampling, so this could possibly also be optimized. Thus we plan on extending their algorithm by improving the ERGM method's proposal distribution and trying different sampling methods (discussed in next steps sections).

3 Methods/Model

We followed [5] and implemented M3GM as our baseline. M3GM is a framework that combines global and local properties of semantic graphs to constrain local predictions to structurally sound ones. It is built on the ERGM to estimate weights on local and global graph features and follows the Monte Carlo Maximum Likelihood Estimation (MCMLE) approach for parameter estimation. Since nodes generally have little intrinsic distinction in classical ERGM application areas like social network while synsets in semantic graphs contain useful information for predicting graph structures, M3GM introduces relation-specific association operators into the scoring function: $\psi_{\text{M3GM}}(G) = \exp\left(\boldsymbol{w}^T\mathbf{m}(G) + \sum_{r\in\mathcal{R}}\sum_{s,t\in E(r)}\mathcal{A}^{(r)}(s,t)\right)$ where $\mathbf{m}(G)$ is the global graph motifs (total edge counts, number of cycles of length in $\{2,3\}$, number of nodes of degree in $\{1,2,3\}$, number of nodes of degree at least $\{1,2,3\}$, number of paths of length 2, and transitivity $u\to v\to w$ with $u\to w$), $\boldsymbol{w}$ is the parameter to estimate, and $\mathcal{A}^{(r)}$ is one of three association operators: (1) TransE which embeds r into a vector representing the difference between the source and target and computes the association score under a translational objective $\mathcal{A}^{(r)}_{\text{TRANSE}}(s,t) = -\|\mathbf{e}_s + \mathbf{e}_r - \mathbf{e}_t\|$; (2) BiLin which embeds r into a full-rank matrix and computes the association score by a bilinear multiplication $\mathcal{A}^{(r)}_{\text{BILIN}}(s,t) = \mathbf{e}_s\mathbf{W}^T\mathbf{e}_t$; (3) DistMult which embeds r into a diagonal full-rank matrix and reduces the computation to a dot product $\mathcal{A}^{(r)}_{\text{DISTMULT}}(s,t) = \langle\mathbf{e}_s,\mathbf{e}_r,\mathbf{e}_t\rangle$. M3GM uses negative sampling to approximate the normalization term of ERGM by obtaining negative samples via $\tilde{G} = G\setminus\{s\xrightarrow{r}t\}\cup\{s\xrightarrow{r}\tilde{t}\}$ i.e. replacing an edge $s\xrightarrow{r}t$ with T edges with $\tilde{t}$ sampled from the proposal distribution Q proportional to the local association score of edges not present in G: $Q(\tilde{t}\mid s,r,G)\propto\mathcal{A}^{(r)}(s,\tilde{t})$ where $s\xrightarrow{r}\tilde{t}\notin G$. Furthermore, M3GM replaces the maximum likelihood objective with a margin-based objective: $\ell(\boldsymbol{\Theta},\tilde{G};G) = (1-\max(\psi_{\text{M3GM}}(G),0)+\max(\psi_{\text{M3GM}}(\tilde{G}),0))$ so that the log score of negative sample $\tilde{G}$ should be below the log score of G by a margin of at least 1. The overall loss is the sum over $N = |E|\times T$ negative samples plus an L_2 regularization term: $\ell(\boldsymbol{\Theta};G) = \lambda\|\boldsymbol{\Theta}\|_2^2 + \sum_{i=1}^{N}\ell(\boldsymbol{\Theta},\tilde{G}^{(i)};G)$

4 Preliminary Results and Evaluation of Preliminary Results

For baseline model, we trained the association model **TransE** and finetuned with M3GM. The results are included in the Table 1. Note that a plot was not used, because with our three different metrics for our problem and baseline model, it is not suitable to use a plot to display the information. As expected, TransE, which takes into account of node-specific information by introducing relation-specific association operators, improves upon the rule-based predictor (higher values for all three metrics). M3GM, which reranks the top-k candidates with the global graph score, is successful in that it further improves upon the association model, with even higher scores for all three metrics. In the next section we discuss how we hope to improve on this work and also what was unsuccessful/the flaws of the current algorithm.

	MRR (↑)	h@10(↑)	h@1(↑)
Rule	35.26	35.27	35.23
TransE	43.05	50.48	39.55
M3GM	46.86	55.64	42.15

Table 1: Preliminary Results

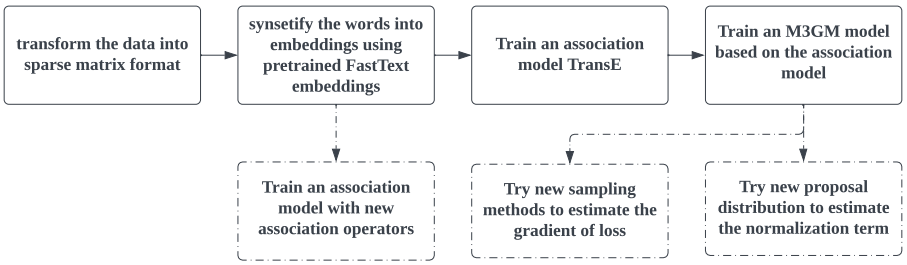

Figure 1: Baseline model (solid) and next steps (dashed).

5 Next Steps

For next steps, we plan on improving the ERGM framework. One way is by improving the proposal distribution. As mentioned in Section 3, the proposal distribution is defined based on three semantic relation specific functions called association operators. Our motivation behind improving this, is we believe that using only these three functions limits certain aspects of semantic relations that the algorithm is accounting for. Many different association operators exist in semantic relations, such as the Structured Embeddings (SE) [2], so we believe that having the limit of the three operations causes for the proposal distribution to not be optimal for the problem. Thus to improve this, we plan on extending and modifying the association operators so that the proposal distribution used is better for our problem. Another improvement we plan on making is the sampling method used in the ERGM framework to estimate the normalization term. The motivation behind our improvement is that in lecture we saw that there were other types of sampling such as Gibbs, Rejection, or Importance sampling that could result in better results or could be computationally easier for our task. Thus we plan on attempting to improve the sampling portion of the ERGM framework by trying different sampling methods. More details of our next steps can be seen in Figure 1. To evaluate our final method, we plan on using the same metrics as previously mentioned which are mean reciprocal rank and H@k. For the timeline regarding our next steps, we note that we have about six and a half weeks before the final project deadlines. Thus we plan on spending about 4-5 weeks working on improving the proposal distribution and sampling (coding and debugging), which would leave about 1-2 weeks for writing our results and finalizing submission details. We also plan to meet together twice a week so that we are able to adjust our plans accordingly if any problems arise. For the job being done by each team member, we plan on dividing it such that we are all involved with coding and writing our results, more details for this can be found in the next section.

6 Teammates and Work Division

According to our rough timeline above, we plan on having Eric work on improving the sampling methods, while Catherine (Yun) and Abbey (Yuxin) will work on improving the proposal distribution. We thus will spend about 2-3 weeks working on completing our respective tasks (with meetings that occur twice a week as previously mentioned). During this time, we may move our responsibilities depending on the workload of unseen debugging challenges. From there we will spend the next 1-2 weeks combining our code and debugging our two portions. During this time we will all be responsible for connecting our portions of the code, so that at the end of this time period we have our finished improvements. During the final week before the deadline, we also will all be responsible for writing our results for the final submission.

References

[1] BORDES, A., USUNIER, N., GARCIA-DURAN, A., WESTON, J., AND YAKHNENKO, O. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems* (2013), C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., vol. 26, Curran Associates, Inc.

[2] BORDES, A., WESTON, J., COLLOBERT, R., AND BENGIO, Y. Learning structured embeddings of knowledge bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence* (2011), AAAI'11, AAAI Press, p. 301–306.

[3] DETTMERS, T., MINERVINI, P., STENETORP, P., AND RIEDEL, S. Convolutional 2d knowledge graph embeddings. In *AAAI* (2018).

[4] FELLBAUM, C. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[5] PINTER, Y., AND EISENSTEIN, J. Predicting semantic relations using global graph properties. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels, Belgium, Oct.-Nov. 2018), Association for Computational Linguistics, pp. 1741–1751.

[6] RIEDEL, S., YAO, L., MCCALLUM, A., AND MARLIN, B. M. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Atlanta, Georgia, June 2013), Association for Computational Linguistics, pp. 74–84.

[7] YIH, W.-T., TOUTANOVA, K., PLATT, J. C., AND MEEK, C. Learning discriminative projections for text similarity measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning* (Portland, Oregon, USA, June 2011), Association for Computational Linguistics, pp. 247–256.