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 this project focuses on is to predict relationships between different words (semantic relation prediction) in the English language. The progress in semantic relation prediction is expected to help improve many different applications in natural language processing, such as sentiment analysis and question and answer generation. One direction of research in semantic relation prediction is to represent the relationship between words (entities) as a directed graphical model and utilize local and global graph properties. In this project, we are interested in improving semantic relation prediction using global graph properties. Specifically, we proposed improving semantic relation prediction by utilizing different global graph features and taking advantage of the graphical model structure and using different sampling methods and proposal distributions to improve parameter estimation for this prediction task.
For this project, we will be using the WN18RR dataset [3]. This dataset is an improved version of the WN18 dataset [1], which is a popular semantic relation prediction dataset 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). Therefore, from the dataset we can define tuples that consist of a semantic relationship (edges in our graphical model), and two entities (nodes in our graphical model) that are involved with this semantic relationship. For example, one tuple that could exist is *(hypernymy, bird, pigeon)*, which is interpreted as bird is a hypernym of pigeon (note that a hypernym semantic relation is equivalent to a "is-a" relationship so a pigeon "is-a" bird). In our prediction task, the model will take the relationship and only one of the two entities from the tuple as input. Then the model will predict the other entity as output. In order to analyze our results, we plan on using several different metrics that are commonly used in relation prediction tasks. These metrics are used in our baseline which is based off of previous works (which we will discuss further in Section 2). We also keep the same metrics for easy comparison with our results throughout the report. The metrics we plan on using to analyze our results are the Mean Reciprocal Rank (which we will refer to as MRR) and the proportion of true entities found in the top k of the ranking lists, where we define the parameter k to be 1 or 10 (we will refer to these two metrics as h@1 and h@10 respectively).

2 Background

Briefly summarizing our midway report, we trained the association model **TransE** [1] and finetuned with M3GM. We followed [8] and implemented M3GM as our baseline. M3GM is discussed in detail further in Section 4. To give a brief overview of the architecture, M3GM is a framework that utilizes both local and global properties of semantic graphs to constrain local predictions to structurally sound ones. The results for our baseline from our midway report 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 in [1], we observed that the use of TransE, which takes into account of node-specific information by introducing relation-specific association operators, has made some improvement upon the rule-based predictor with higher performance in all

Preprint. Under review.

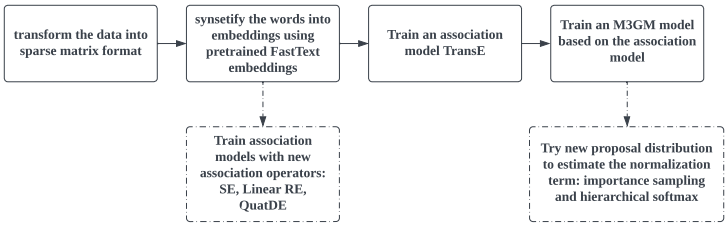

Figure 1: Baseline model (solid) and our proposed methods (dashed).

metrics. M3GM, which re-ranks the top-k candidates with the global graph score, further improves upon the TransE association model with even higher scores for all three metrics, resulting in the most superior performance compared to TransE and our rule-based predictor.

	MRR (↑)	h@10(↑)	h@1(↑)
Rule	35.26	35.27	35.23
TransE	42.43	49.19	39.17
M3GM (TransE)	42.82	49.14	39.81

Table 1: Preliminary Results on Baseline Models

We also stated in our midway report that we plan on improving the ERGM framework from three aspects, as illustrated in Figure 1. One way is by improving the proposal distribution. As we will discuss further in detail in Section 4, the proposal distribution is defined based on three semantic relation-specific functions called association operators. Our motivation behind improving this is to experiment with more association models (for example, the Structured Embeddings (SE) [2]) inspired by the literature on semantic relation prediction and compare the empirical results with the original M3GM architecture. 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 the lecture, we saw other types of sampling such as Gibbs, Rejection, and Importance sampling that could result in better results or could be computationally more efficient for our task. Therefore, we try different sampling techniques to improve the ERGM framework. For easier visualization of where our improvements happen in the workflow of the M3GM model and to elaborate on the structure of our improvements, we have detailed this in Figure 1. Finally, to evaluate our final method, we use the same metrics as previously mentioned: mean reciprocal rank and h@1 and h@10.

3 Related Work

There have been multiple lines of research that have looked into the task of predicting semantic relations. One recent work by Bordes et al. looks into converting word embeddings in different phrases in order to model and predict semantic relations between words [1]. While their problem is similar to ours and are able to predict multi-word relationships, as stated in their work, using their approach is computationally expensive. Therefore, there are potentially other ways to structure semantic relations such that the prediction of semantic relations can not only improve, but also be computationally easier. In addition to this gap in the computational difficulty of their task, we also noted a lack of structuring semantic relations in their work. Specifically, in other natural language processing areas, when looking at text similarity, it was possible to have improved metrics when structuring texts as graphical models to predict their similarity [10]. Despite of the difference in the specific tasks and objectives, we are inspired to set up the semantic relation prediction problem by modeling semantic relations as graphical models such that by using global graph properties, we can improve on previous results.

One of the key aspects of our problem is how we plan on representing or structuring semantic relations as a directed graphical model for the semantic relation prediction task so that we can utilize various local and global graph properties as previously stated. Some previous work have converted semantic

relations into graphical models in order to solve some natural language task utilizing properties of the graphical model. One such paper is [9], where they represent semantic relations as a graphical model and solved the problem of being able to automatically detect relations from text that has been syntactically parsed. Although the task is slightly different from ours, we can take inspiration from their work and define our graphical model similar to how they have done so. Specifically, in our case, the words in the semantic relationships are the nodes of the graph and the edges represents different semantic relations. For example, to represent the triple *(hypernym, purple, color)*, i.e. *purple is a color*, in our graphical model, *purple* and *color* will be the source and target nodes and there will be an edge going from purple to color which represents the *hypernym* relation. In this way, we are able to understand how to build a foundation of converting semantic relations into a graphical model.
Another work worth noting by Yuval Pinter et al. applies the idea of graphical models in the task of semantic relation prediction, which has provided great inspiration for our project [8]. Their approach uses an exponential random graph model (ERGM) for predicting the semantic relation in the graphical model. To do this, they followed the Monte Carlo Maximum Likelihood Estimation (MCMLE) approach to estimate the normalization term of the ERGM. Although they were able to achieve better results compared to other methods that did not use graphical models, we note that there are various aspects of their algorithm and architecture that could be potentially improved. First, they utilized a proposal distribution that is based off of the semantic relation specific functions called association operators. They only experimented with three such operators in literature, which motivates us to improve the proposal distribution by investigating and experimenting with more association models inspired from the existing literature. In addition to improving the proposal distribution, when performing parameter estimation for the normalization term in the ERGM, they use Monte Carlo Maximum Likelihood Estimation (MCMLE) formulation with negative sampling. We are interested in trying other types of sampling methods and comparing the effects on the overall model performance for potential improvement. In conclusion, we proposed to improve their algorithm by searching for better proposal distribution and trying different sampling methods to improve the parameter estimation in the ERGM. We are going to discuss our proposed approach specifically in Section 4.1 and Section 4.3.

4 Methods

For baselines, we followed [8] and implemented the rule-based model, TransE, and M3GM. 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, as previously mentioned. 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 as seen here: $\psi_{\text{M3GM}}(G) = \exp\left(\boldsymbol{w}^T \mathbf{m}(G) + \sum_{r\in\mathcal{R}} \sum_{s,t\in E(r)} \mathcal{A}^{(r)}(s,t)\right)$ 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:

- **TransE** embeds r into a vector representing the difference between the source and target and computes the association score under a translational objective $\mathcal{A}^{(r)}_{\text{TransE}}(s,t) = -\|\mathbf{e}_s + \mathbf{e}_r - \mathbf{e}_t\|$;
- **BiLin** embeds r into a full-rank matrix and computes the association score by a bilinear multiplication $\mathcal{A}^{(r)}_{\text{BiLin}}(s,t) = \mathbf{e}_s \mathbf{W}^T \mathbf{e}_t$;
- **DistMult** 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 (in other words we define it as such): $Q(\tilde{t} \mid s, r, G) \propto \mathcal{A}^{(r)}(s,\tilde{t})$ where

$s \xrightarrow{r} \tilde{t} \notin G$. Furthermore, M3GM replaces the maximum likelihood objective with a margin-based objective: $\ell(\mathbf{\Theta}, \tilde{G}; G) = (1 - \max(\psi_{\text{M3GM}}(G), 0) + \max(\psi_{\text{M3GM}}(\tilde{G}), 0))$ so that the log score of negative sample $\tilde{G}$ should be below the log score of G by a margin of at least one. The overall loss is the sum over $N = |E| \times T$ negative samples plus an L_2 regularization term:$\ell(\mathbf{\Theta}; G) = \lambda\|\mathbf{\Theta}\|_2^2 + \sum_{i=1}^{N} \ell(\mathbf{\Theta}, \tilde{G}^{(i)}; G)$. Now that we have summarized the model that we implemented, we will dive further into the improvements that we tried to make to this model.

4.1 Other Association Models: SE, LineaRE, and QuatDE

As in Figure 1, our first attempt to improve the original M3GM method is to try other association models to obtain better proposal distribution for parameter estimation. We refer to the Structured Embeddings (SE) [2], LineaRE [7], and QuatDE [5] as three superior knowledge bases embeddings for relation prediction. Reusing the notations above, we define the following three new association operators corresponding to the scoring function of the three embeddings:

- **SE** transforms the entity embedding vectors by the corresponding left and right hand relation matrices $\mathbf{W}_{\text{left}}^{(r)}, \mathbf{W}_{\text{right}}^{(r)}$ for the relation r and outputs the association score as the 1-norm distance in the transformed embedding space, i.e.

$$\mathcal{A}_{\text{SE}}^{(r)}(s,t) = \|\mathbf{W}_{\text{left}}^{(r)}\mathbf{e}_s - \mathbf{W}_{\text{right}}^{(r)}\mathbf{e}_t\|_1$$

- **LineaRE** is similar to SE with an extra bias vector $\mathbf{b}^{(r)}$ for relation r and expects that each dimension of the relation can be represented as a straight line in a rectangular coordinate system, i.e. $\mathbf{W}_{\text{left}}^{(r)}(i) \cdot \mathbf{e}_s(i) + \mathbf{b}^{(r)}(i) = \mathbf{W}_{\text{right}}^{(r)}(i) \cdot \mathbf{e}_t(i)$ for each dimension i. Thus, the association score is naturally defined as

$$\mathcal{A}_{\text{LINEARE}}^{(r)}(s,t) = \|\mathbf{W}_{\text{left}}^{(r)} \circ \mathbf{e}_s + \mathbf{b}^{(r)} - \mathbf{W}_{\text{right}}^{(r)} \circ \mathbf{e}_t\|_1$$

  where $\circ$ denotes the Hadamard (element-wise) product.

- **QuatDE** improves upon QuatE [11] which uses operation on quaternion space to capture the interaction between entitiy pair with a dynamic mapping strategy to enhance the character interaction with Hamiltonian product between the triple explicitly. Specifically, with representation as quaternion ordered pairs, i.e. $\mathbf{e}_s = a_s + b_s\mathbf{i} + c_s\mathbf{j} + d_s\mathbf{k}$ and similarly for $\mathbf{e}_t, \mathbf{e}_r$, we denote $S^{(r)}(\mathbf{e})$ as a dynamic mapping function driven by entity e and relation r and define

$$S^{(r)}(\mathbf{e}_s) = \mathbf{e}_s \otimes \tilde{\mathbf{e}}_s \otimes \tilde{r}, \quad S^{(r)}(\mathbf{e}_t) = \mathbf{e}_t \otimes \tilde{\mathbf{e}}_t \otimes \tilde{r}$$
$$U^{(r)} = \frac{a_r + b_r\mathbf{i} + c_r\mathbf{j} + d_r\mathbf{k}}{\sqrt{a_r^2 + b_r^2 + c_r^2 + d_r^2}}$$

  where $\tilde{\mathbf{e}}_s, \tilde{\mathbf{e}}_t$ are normalized entity transfer vectors, $\tilde{r}$ is normalized relation transfer vectors, and $\otimes$ denotes the Hamilton Product (Quaternion Multiplication) $q_1 \otimes q_2 = (a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2) + (a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2)\mathbf{i} + (a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2)\mathbf{j} + (a_1d_2 + b_1c_2 - c_1b_2 + d_1a_2)\mathbf{k}$. Finally, the association score is computed as

$$\mathcal{A}_{\text{QUATDE}}^{(r)}(s,t) = S^{(r)}(\mathbf{e}_s) \otimes U^{(r)} \cdot S^{(r)}(\mathbf{e}_t)$$

4.2 Extending Graph Motifs

A key observation made in the original paper is that by incorporating recurrent significant subgraph or patterns as features in ERGMs, referred as motifs, the extracted features help constrain local predictions to structurally sound ones. For example. the edge count includes knowledge of proximity between vertices. Since cycle is an invalid structure in the graph, treating cycles as features can also help inhibit its occurrence in prediction. To incorporate the relation nature of the task, we extended the motifs to further include combinatorial features including transitivity, reciprocity, and three-cycles.

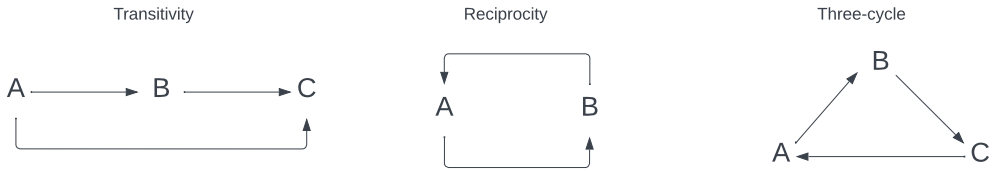

Figure 2: Three types of combinatorial graph motifs

4.3 Building upon Negative Sampling

One of the improvements we try to make that was mentioned previously was trying to improve the sampling method that was used in the algorithm. As previously mentioned, the M3GM framework utilizes negative sampling to approximate the normalization term of ERGM. Note that negative sampling is similar to stochastic gradient descent where during the training process we do not update all the weights after each iteration. Instead as seen from [8], negative sampling is a variation/based off of noise contrastive estimation, thus that is why we have positive and negative samples in this technique. Negative sampling, as previously mentioned, when sampling from the training data also samples points from the data that are deemed negative training examples. In other words when sampling from the training data, negative sampling is looking for training tuples or semantic relationships that are less common. This motivated us to try different techniques that are similar to see if they would improve the algorithm. We noted that the original framework does negative sampling without experimenting or proving that it is better to differentiate between positive/negative samples and draw negative samples instead of using classical sampling approaches that do not account for negative samples. Therefore this motivated our first idea/technique, which is that we wanted to try to use a common classical approach, such as the Monte Carlo technique, importance sampling, to approximate the normalization term. Thus we can use this experiment to test if it was necessary to do negative sampling and if drawing negative samples improved the prediction. As seen in the below sections, we display the results of using importance sampling and saw that this approach did not work as well as the negative sampling. Also in the below section of discussion and analysis, we will further discuss why we believe this method was not successful and did not perform as well as negative sampling.
After attempting this technique, another idea we had in order to improve the sampling, was to build upon the idea of negative samples in negative sampling. Specifically we were motivated by the fact that using a softmax function as the last activation, the computational cost of negative sampling would be linear in terms of the number of negative samples at each iteration. Therefore to improve the sampling, using this motivation, we came up with the idea to replace this with hierarchical softmax. One of the reasons why we wanted to use hierarchical softmax was to potentially reduce the computational cost. Hierarchical softmax utilizes a binary tree so that the computational cost is logarithimic in terms of the size of the vocabulary for a given semantic relationship. In our case the binary tree for a given semantic relationship would have the words as the leaves of the binary tree with the root word being the given word from the semantic relationship tuple as previously described. From previous literature and research for example in [6] we also saw that because negative sampling is based off of noise contrastive estimation with positive and negative samples, it usually works better in situations with training data that has frequent words and lower dimensional training data, while hierarchical softmax usually performs better with training data that has infrequent words. Therefore since our training dataset tuples does have many infrequent words, we expect the results of using hierarchical softmax to be comparable or better. Thus below in the sections of results/discussion and analysis we see if this technique is a possible improvement and further discuss the limitations of this technique compared to the original negative sampling.

5 Results

After standardizing the experiment setups, we chose to keep the same metrics as in our baseline report for easy comparison with our results throughout our final report. The metrics we used to analyze the

results are the Mean Reciprocal Rank (MRR) and the proportion of true entities found in the top k of the ranking lists, where we define the parameter k to be 1 or 10 (h@1 and h@10).
First, we present a summary of experiment results of trying SE, LineaRE, and QuatDE as new association models in Table 2. We observed that despite of subtle improvements from the rule-based baseline, none of SE, LineaRE, or QuatDE has outperformed TransE. Among the three association models we tried, LineaRE achieves the best performance in all metrics while we note that TransE is a special case of LineaRE by setting the two transformation matrices as identical, i.e. TransE defines a relation as straight lines with a constant slope of 1. By comparing the performance of the three association models with the performance of TransE, we found that the extent of improvement by the reranking of M3GM depends on the performance of the association models. For the original M3GM model, adding extra motifs also further improves the performance.

	MRR (↑)	h@10 (↑)	h@1(↑)
SE	35.70	36.37	35.23
LineaRE	37.09	39.35	35.83
QuatDE	35.24	35.23	35.23
M3GM (SE)	35.82	36.50	35.53
M3GM (LineaRE)	36.87	38.94	35.89
M3GM (QuatDE)	35.25	35.23	35.23
M3GM (TransE) + Extra Motifs	42.97	49.23	39.94

Table 2: Results for new association models and extra motifs

In addition as shown in the last row of Table 2, excluding more combinatorial graph motifs as extracted features also helps in achieving a higher performance. This can be seen because the metric values for all three metrics are higher than previous results. This matches our assumption/expectation that local combinatorial features helped the models to make better prediction. We will also discuss further next steps to possibly improve this in the next section.
For the sampling experiments as described in our methods section, we have included a plot/table of our results, as seen in Table 3, for the two new sampling techniques we modified to the algorithm. Note that both models utilized the pre-trained TransE association operator model, so that our results can be compared to prior works below. As seen in the table, across all three metrics the Hierarchical Softmax technique performs better (we can see that it achieves higher metric values) than importance sampling. Thus from our experiment it seems that the hierarchical softmax technique we described in the methods section is a better method than exchanging negative sampling for importance sampling.
For convenience we have also included a plot (Figure 3) of the numbers so that easy comparison

	MRR (↑)	h@10 (↑)	h@1(↑)
Importance Sampling	35.65	37.08	35.84
Hierarchical Softmax	46.97	55.03	41.45

Table 3: Results for new sampling techniques

(to see the difference or improvements in the different metrics) can be done between the two new sampling modifications with the original negative sampling technique that we explored in our baseline. As seen using Figure 3 we can compare our results to prior work, specifically we compare it to the previous work of the M3GM model that was trained with TransE and negative sampling. We can see that the prior work with negative sampling performs better than importance sampling, because the metric value scores are all higher. We note that this does align with our previous expectations, because as previously mentioned this provides experimental proof that using a noise contrastive approach such as negative sampling is better than importance sampling which does not account for positive or negative samples. On the other hand we can see that using hierarchical softmax seems to result in comparable results to the prior work, because the metric values are all slightly higher than the values from the original negative sampling technique. Specifically the h@10 score increases from 51.02 to 55.03, the h@1 scores increase from 39.68 to 41.45, and the MRR score increases from 43.29 to 46.86 when comparing to the prior work using negative sampling (these numbers can also be verified using the values from Table 1 and 3). This again does align with our expectations, because as stated previously in the methods using hierarchical softmax, we expected our results to be comparable

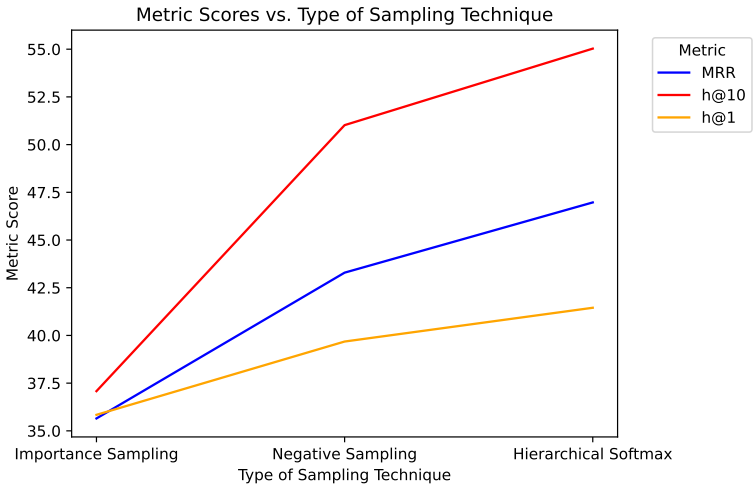

Figure 3: Comparison of each sampling technique experiment

or better, because our hierarchical softmax techinique was expected to perform better with infrequent words. We also noted that in terms of runtime, both the original negative sampling and hierarchical softmax took about eight hours. In the next section we will further analyze these results, and give more details about possible reasons for the metric value results of these experiments. Note that in addition to the quantitative results, we also took a look at the qualitative results and analyzed some patterns we noticed in the words of the predicted semantic relations. We noticed that for the original negative sampling technique, the algorithm had a pattern of ranking higher/being more accurate with semantic relations that include common words. For example some semantic relations it was able to predict correctly are hypernym relationship between travel and cruise (where the word travel appeared frequently in the data) and a hypernym relationship between position and presidency (where the word position appeared frequently in the data). On the other hand it would rank lower or predict poorly on words that appeared less frequently, for example the hypernym relationship between bird family and rheidae. Meanwhile looking at some of the same hypernym relationships, we notice that there is a general pattern where hierarchical softmax is able to rank these less frequent hypernym relatinoships higher. This observed performance does align with our expectations, because as stated in previous sections, we expected the original negative sampling to be able to perform better with relations that have words that appear frequently in the data, while on the other hand hierarchical softmax attempts to predict better on relations with infrequent words.

6 Discussion and Analysis

Now we will analyze our model and results in terms of the sampling experiments. As previously described in the results section we saw that importance sampling did not perform as well as the prior result, however one limitation or constraint to our approach is that there could have been a better proposal distribution. As mentioned in the methods section, we utilize the same proposal distribution as using in the original M3GM framework, however this assumption is a limitation because there could be a better proposal distribution based off of domain knowledge regarding semantic relationships. Therefore a possible further way to improve our method to eliminate this limitation is if future research is done on better proposal distributions that can be used for our case of semantic relation prediction. In addition in our results section we observed that the results from hierarchical softmax and the original negative sampling were comparable (specifically the original negative sampling metrics were slightly lower) and also the had similar runtimes. We believe that this also provides insight into the environment/dataset we are dealing with. As stated in our methods section, the hierarchical softmax usually performs better in cases where words are infrequent. However with the WN18RR dataset, when training we also did notice that the dataset had a couple repeat words that many different semantic relationships with other words. Therefore if these repeat words in the tuple are given to the model when training, then this could be the reason why the the hierarhical softmax

technique scores were slightly higher than the original negative sampling technique (or in other words why hierarchical softmax did not have a larger improvement in the metrics). Regarding the runtime, even though hierarchical softmax utilizes the binary tree, we also have a large vocabulary in our semantic relationship dataset, so this could be slowing down the training, which is why we see similar runtimes. In the future, it would be interesting to further verify our method by using different datasets of not only varying vocabulary size (to test the runtime difference between the two techniques), but also datasets with varying frequency of words in the relation tuples, so that perhaps we can see a larger difference in the result metrics between the two techniques.
Unfortunately, the experiments with new association operators did not bring further improvement to the M3GM framework but we can see the dependence of the improvement by the M3GM reranking on the inherent performance of the underlying association model, as the association scores are heavily used in coming up the proposal distribution for parameter estimation.
Also as mentioned in the results section, we saw that in Table 2, excluding more combinatorial graph motifs as extracted features also helps in achieving a higher performance. This matches our assumption/expectation that local combinatorial features helped the models to make better prediction. A possible future direction is to further explore the benefit of adding motifs potentially on other association models, to see if the combination could result in better metric values/ further improve our method.
In addition our work looked at different ways to improve certain aspects of the original algorithm. Through our experiments, we also found some limitations in our methods. The feature (graph motifs) updating operation is hard-coded and our codes are not optimized for GPU. As a result, even with negative sampling, our experiments take a very long time to train (more than 8 hours on average). We also expect the transition to GPU computing resources allows better convergence of more complicated association models to match the superior performance as reported in the original paper to further improve the results. Finally, the dataset and our experiments are currently solely English. We believe that it is definitely feasible to extend the model into multilingual setting. Since the extracted features are not language-dependent, we believe exploiting combinatory motifs will still be beneficial.
An interesting future direction is seeing how the different combinations of our techniques can be combined and how the combinations of techniques affect the overall model or how the interaction of the techniques affect/compare to each other's original results that we reported in the above sections.

7 Teammates and Work Division

As stated in our midway report, we were able to adhere to our plan for our timeline and work division so that all three members did an equal amount of work when it came to both programming and writing the report. According to our timeline, we sectioned off the work where Eric primarily worked on improving the sampling methods, while Catherine (Yun) and Abbey (Yuxin) primarily worked on improving the proposal distribution, specifically with Catherine working on the association operators and Abbey working on the graph motifs, as explained in previous sections. We all equally spent about four to five weeks working on our respective tasks (with meetings that occurred twice a week as previously mentioned in our midway report). As previously planned during this time, our responsibilities did shift slightly, for example we helped each other debug and fix unseen challenges when programming. The three of us also equally divided the work when typing up the final report, so that we all put in about the same amount of time. In addition we met frequently throughout the week during the final week before the final report deadline to check-in and make sure we were all on the same page when writing the report. We also made sure that all team members read through and were satisfied with the final report. One thing to note is that one deviation we had from our original plan was the video report, which as stated on piazza we no longer needed to spend time on the video since the video was no longer required for our project.

8 Access to Our Code

For our project, all of our code can be found in the following below github repository: `https://github.com/kapikantzari/10708-project`. Using the code in the link below, readers should be able to also follow along with our work and reproduce similar experiments. We have also included a README with details on how to run our code along with the experiments.

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] GAO, H., YANG, K., YANG, Y., ZAKARI, R. Y., OWUSU, J. W., AND QIN, K. Quatde: Dynamic quaternion embedding for knowledge graph completion. *ArXiv abs/2105.09002* (2021).

[6] MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G., AND DEAN, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Infomration Processing Systems* (2013).

[7] PENG, Y., AND ZHANG, J. Lineare: Simple but powerful knowledge graph embedding for link prediction. In *IEEE International Conference on Data Mining, ICDM* (2020), C. Plant, H. Wang, A. Cuzzocrea, C. Zaniolo, and X. Wu, Eds., pp. 422–431.

[8] 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.

[9] 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.

[10] 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.

[11] ZHANG, S., TAY, Y., YAO, L., AND LIU, Q. Quaternion knowledge graph embeddings. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2019), Curran Associates Inc., p. 11.