An Evaluation of the Impact of Training Data Locality on the Effectiveness of Knowledge Graph Embeddings Models

Marc Mellotte, Conor Hayes

Data Science Institute, National University of Ireland Galway
{first.last}@nuigalway.ie

Abstract. As the volume and variety of data that many modern organisations deal with continue to grow, graphs are becoming increasingly important and relevant as a means of organising this data. This work looks at a possible way to improve the training of some state-of-the-art machine learning models in the area of knowledge graph embeddings. Where the interest of the user is on the ability to predict the existence of a particular link type as opposed to predicting links generally, subsets or sub-graphs could possibly be used to train the model more effectively than the entire graph. We evaluate the performance of two state-of-the-art knowledge graph embedding models on the task of predicting a specific link type. The models are first trained with all of the available training data and subsequently with subsets or sub-graphs based on the locality of the link type we wish to predict. We find that there is evidence that using less training data can in some cases actually improve the performance of the model. Finally, we look at some graph features and examine if there is any correlation between these and the accuracy/performance of the machine learning models. While no strong correlation is found, the results point to further work being required to understand this phenomenon.

1 Introduction

Modern organisations deal with ever-increasing amounts of data from multiple sources and in many different formats and structures. In a widely quoted statistic, Grimes reports that 80% of the most business-relevant data comes in unstructured form, mostly text[8]. A key challenge then is the ability to organise and establish links between diverse data types. One way to achieve this is through the use of Natural Language Processing (NLP) to extract “facts“ from unstructured text and and graph technology to link these facts together into an overall knowledge base or Knowledge Graph.

Once such knowledge graphs are constructed, challenges such as how to keep them up to date and how to mine them for insights arise. The task of predicting links between nodes (people, organisations, etc) in the graph can lead to insights where the relation was not believed to exist previously. However, as opposed to generally predicting links for a graph overall, we are interested in the ability to predict specific *link types*. For example, in a knowledge graph of movies, we might be interested in predicting who directed a movie, i.e. the "director" relation between a person and a movie; instead of generally predicting links between actors, movies, places, etc. Similarly, a large commercial organisation might be interested in predicting the existence of links between customers and new product lines. A pharmaceutical research organisation may want to predict a potential link between a drug and a specific protein, which may indicate the genetic profile of the population segment that the drug might be effective for. In all of these cases, we are interested in the effectiveness of predicting *specific link types* within the overall graph.

2 Related Work

Knowledge Graphs (KGs) are knowledge bases that store factual information in the form of graph data [17]. As such, a knowledge graph is a directed, multi-relational graph. These graphs are typically represented as a set of triples in the form of (subject, predicate, object) where each triple represents a single fact and the predicate is the link or relation between the subject and object entities. Figure 1 shows a very simple sub-graph and Table 1 describes the triples. Such facts are frequently, but not necessarily, encoded using the Resource Description Framework (RDF) triples format.

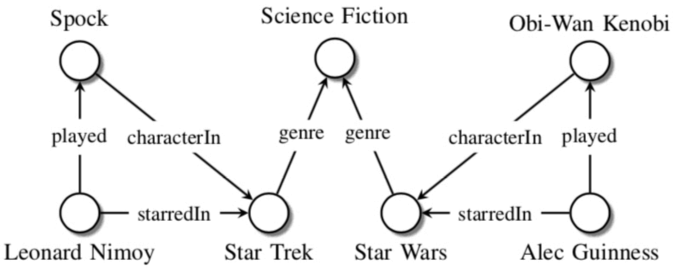

Fig. 1. Sample knowledge graph [17]

Subject (head)	**Predicate (relation)**	**Object (tail)**
Star Wars	genre	Science Fiction
Alec Guinness	starredIn	Star Wars
Obi-Wan Kenobi	characterIn	StarTrek

Table 1. Example triples

KGs continue to grow in popularity. In addition to well-know instances on the Web such as the Google knowledge graph [23], (which has incorporated an earlier KG, Freebase [3]), WordNet [14], DBPedia [2], BabelNet [16] and NELL [7], KGs are finding increasing use within within organisations such as Amazon [11]and were added to the Gartner Hype Cycle for Emerging technologies in 2018 [20].

One of the main problems with all knowledge graphs to date is that they are very far from complete. For example, Freebase has no place of birth for over 70% of the person entities it contains. Even amongst the 100k most frequent person entities on Freebase, almost a quarter (24%) do not have a profession recorded [28]. This incompleteness has lead to much research effort in the area of knowledge graph completion. Knowledge graph completion generally breaks down into the sub-tasks of entity creation or extraction and link prediction [17]. The former involves adding new entities or nodes to the graph while the latter involves adding new links between existing entities.

Statistical Relational Learning: The prediction of missing links or relations has been the focus of Statistical Relational Learning (SRL), a sub-field of machine learning that focuses on graph data. SRL covers a number of areas including collective classification, link prediction, link-based clustering, social network modelling and object identification [21]. According to [17], there are two main classes of SRL techniques. The first contains those that capture the correlation between nodes (or links) based on the *observable features of the graph.* The second captures the correlation using *latent variables.* Models based on observable graph features can be divided into two approaches: a nearest neighbour approach based on node similarity using measures such as cosine similarity, mutual information, Dice coefficient, distance-based measurements, binary classifiers and more [13], [22] or a topological approach where prediction is based on patterns extracted from local or global topology such as Common Neighbours, Jaccard Index, Adamic Adar [1], Resource Allocation Index [30], Katz Index [10], Hitting Time [6], Rooted PageRank [12] and SimRank [9]. In models based on latent variables, each node in the knowledge graph is assumed to have a latent feature vector which must be learned; the relationships between nodes are explained by these latent features [17]. As an example, Nickel et al. [17] propose that Alec Guinness received the Academy Award because he was a "good" actor. The property "good" here is a latent feature of the Alec Guinness entity because it is not directly observable in the data. A key task of all latent feature models is to learn the latent features, or "embeddings", for the nodes in the graph. The "embeddings" refers to how entities and relations in a knowledge graph are embedded into continuous vector spaces. This gives advantages such as simplifying manipulation of the graph, while at the same time preserving its inherent structure [27].

Knowledge Graph Embeddings: Wang et al [27] summarise the approaches to KG embeddings into two classes :"Semantic Matching Models" and "Translational Distance Models". Translational distance models exploit distance-based scoring functions. The Translating Embeddings or TransE approach[4] models each entity as a vector in space R^d and each relation as a vector transformation that maps the subject to the object of the relationship. The accuracy of a relation or link is measured as the distance between the entities after the transformation has been applied. TransE would allow for $AlfredHitchcock + DirectorOf \approx Psycho$ and $JamesCameron + DirectorOf \approx Avatar$. Semantic matching models differ from translational-distance models in that they use a similarity scoring approach as opposed to distance-based scoring. The likelihood of a fact or triple is calculated using the latent semantics of the entities and relations. RESCAL [19] and its derivatives/extensions such as DistMult [29], Holographic Embeddings (HolE) [18] Complex Embeddings (ComplEx) [26] are examples of this approach.

Model Evaluation: A common approach to evaluating the accuracy of embeddings models is described by Bordes et al [5]. For all training and testing triples *(s, p, o)*, the subject entity, *s*, is removed and the likelihood of all possible triples/facts *(e, p, o)* is calculated where $e \in D_e$ and D is the entire dictionary of entities in the graph. Triples other than the original correct one are referred to as the *corrupted triples.* The results are then sorted in decreasing order of their likelihood according to the model and the rank of the correct triple is recorded. The process is repeated for the object entity *o.* Evaluation is then similar to that used for question answering in information retrieval where the *Mean Rank* of the correct triple is measured.

Bordes et al [4] report the *mean* of the predicted *rank* of the subject and object entities and the *Hits@10*, i.e. the proportion of correct triples in those ranked as the top 10. One potential issue with this metric is that there may be triples amongst the corrupted triples that are correct. For example, if the original triple was the one relating to the actor Alex Guinness in Table 1, i.e. *Alec Guinness, characterIn, Star Wars*, then we would remove the subject entity (*Alec Guinness*) and predict the accuracy of all other entities in the graph (i.e. Freebase in this case) against the same predicate (relation) and object entity. However, there were more characters in the same movie so some of the corrupted triples will actually be true. In order to avoid this scenario, Bordes et al [4] removed all valid triples that appear in either the training, test or validation data sets from the corrupted triples. They referred to this as the “Filtered” setting and report metrics against both the original, “Raw”, data and the “Filtered” version. They then reported both the **Mean Rank** and **Hits@n** (e.g. Hits@1, Hits@3 & Hits@10) for both the original triples (*raw*) and those with the valid triples removed (*filtered*). Nickel et al [18] followed this evaluation approach and measured the quality of the ranking using the **Mean Reciprocal Rank** (MRR) which is less sensitive to outliers than the mean rank. Again, this is reported for both the *raw* and *filtered* settings.

3 Research Question

This work seeks to address whether more (relevant) training data is always better (i.e. produces higher accuracy in tests) when training a knowledge graph embedding model to predict a specific link/relation type. Our research question is thus - *Does locality (i.e. the local sub-graph around the relation type we are interested in) of training data have a disproportionate effect on the accuracy of the model and, if so, how much local data can effectively train the model*? Our null hypothesis is that more data will always lead to better accuracy as this is commonly the case for machine learning models. From some of the most recent work in the knowledge graph embedding space, we choose two models for this study - ComplEx and DistMult. Both models perform very well in benchmark tests in the literature [17]. We select two datasets that are representative of real-world knowledge graphs upon which to carry out the experiments - Freebase [3] & NELL[7]. We adopt the version of Freebase that has been refined for link prediction experiments by Totuanova and Chen [25]. This dataset, often referred to as FB15k-237, contains approximately 15k entities with 237 relations. Table 2 shows a summary of the two datasets used in this work.

Dataset	**Entities**	**Relations**	**Training Set Relations**	**Test Set Relations**	**Validation Set Relations**
fb15k-237	15k	237	272k	18k	20k
NELL239	48k	239	74k	3k	3k

Table 2. Dataset Statistics

Sub-Graph Selection: Figure 2 describes visually the first level of the sub-graph. At the centre of the diagram, the graph nodes shown in dark colour with the bold link between them represent an instance of the relation that we are interested in. Starting with this relation, we gather all of the relations that connect into or out of either end of this one. We refer to these as “Level 1” relations. This set consists of the relations and nodes inside the circle with the darker shading.

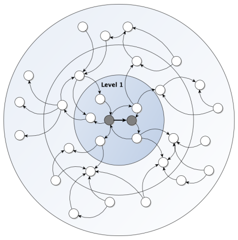

Fig. 2. Graph Locality Level 1

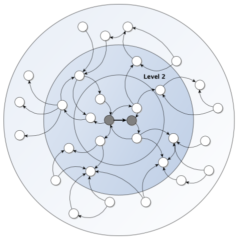

Fig. 3. Graph Locality Level 2

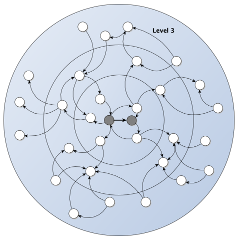

Fig. 4. Graph Locality Level 3

The next sub-graph extracted is called the "Level 2" sub-graph. The process for extracting it is similar to that described for Level 1. Again we start with the relation type of interest and select all of the adjacent relations. For Level 2, we also select all of the relations adjacent to the Level 1 relations. In other words, we select all of the relations that are 2 hops from the relation type of interest. This is shown visually by the darker shaded circles in Figure 3. Level 2 contains all of the relations in Level 1 plus those immediately adjacent to them. Lastly, the Level 3 sub-graph contains all of the relations in Level 2 as well as the set of relations immediately adjacent to these, i.e. all relations that are 3 hops from the relation type of interest. This is shown visually by the darker-shaded circles in Figure 4.

If we were interested in predicting the existence of the */film/film/genre* relation type from the Freebase dataset, we would find that there are 3756 instances of it in the training data. When the sub-graph levels are extracted we get sub-graphs of the sizes shown in Table 3. As we can see, the Level 3 sub-graph is close to the the size of the full training graph/dataset (full graph has approximately 272k relations).

Level	**# Relations**
0	272115
1	31552
2	46586
3	227982

Table 3. Sub-graph Levels Example Sizes

Normally, the evaluation of a model is against the full set of test data. In this case, we are only interested in the performance of the trained model on the specific relation we are interested in. For example, if we are interested in the relation */film/film/genre*, we filter the test dataset to **only** contain this relation. We are not interested in the model performance on the prediction of any of the other relations or the overall graph.

4 Implementation and Evaluation Details

All of the experiments in this work were carried out on a Linux server with Intel(R) Core(TM) i70.4790K CPU 4.00GHz processor, 32 GB RAM, and an nVidia Titan X GPU. The operating system of the server was Ubuntu Server 16.04.5 LTS. The code was written in Python 3.5. The KGE models were developed on top of the TensorFlow[24] (GPU) framework. The implementation of the KGE models has been provided by SK Mohamed et al. [15]. This implementation is not (yet) available as an open source release. It is planned to release the software code developed for this work as open-source once the underlying KGE libraries have also been released.

Evaluation Metrics: We evaluate our results using the *Mean Reciprocal Rank (MRR)* and the *Hits@10* on the *Filtered* dataset (*MRR_FIL*). The MRR measures the mean of the reciprocal of the rank of the correct triple as predicted by the model. Hits@10 records the number of correct triples in the top 10 predicted by the model.

5 Results

In the experiments, both models performed poorly when trained with the Level 1 sub-graphs when compared with the same model trained with the Level 0 graph. The models trained with Level 1 sub-graphs resulted in average accuracies ranging from 50% to over 80% less than the same model trained on the Level 0 graph. This contrasted with significantly better relative performance seen when the models were trained on Level 2 & 3 sub-graphs. We focus on these results here, starting with the ComplEx model (Table 4).

ComplEx Summary				
Dataset	**Filter Level**	**Difference in Training Data (vs Level 0)**	**Difference in Accuracy (vs Level 0)**	**Outperforms Level 0 for**
Freebase	2	-71%	-24%	0-30k relations
Freebase	3	-27%	-29%	20-50k relations
NELL	2	-89%	**10%**	All of Level 2
NELL	3	-67%	**55%**	All of Level 3

Table 4. Summary Results using ComplEx Model

The most immediately interesting results are where the ComplEx models trained on a sub-graph outperformed those trained on the full Level 0 graph. As Table 4 shows, the *ComplEx model trained with either the Level 2 or 3 NELL sub-graph outperformed the same model trained with the entire Level 0 NELL sub-graph.* This suggests that there are scenarios where *less training data can lead to a more accurate link prediction model* for specific links of interest.

We do not see the same overall outperforming of the Level 0 model for a full Level 2 or 3 when ComplEx was trained with Freebase data. We do notice that the drop in accuracy of Levels 2 & 3 from Level 0 is far less than the corresponding reduction in the amount of training data used. For example, the models trained with Level 2 sub-graphs experienced an average drop in accuracy of 24% versus Level 0 for a reduction in training data of 71%. This suggests that there isn't a linear relationship between the amount of training data and the model performance.

These averages for the full level also mask some additional findings within that level. For example, the Freebase Level 2 sub-graphs contained a range of relations from approximately 2k to 200k relations. When this was broken out into ranges of relation counts, we saw that ComplEx models trained on Level 2 sub-graphs with a size of up to 30k relations (approximately) outperformed the model trained on the overall Level 0 graph. Drilling into the Level 3 results in a similar way shows that the ComplEx models trained on Level 3 sub-graphs of sizes from 20-50k relations also outperformed the overall average.

Results of the experiments conducted with the DistMult model are summarised in Table 5. As was the case with the ComplEx model, we see scenarios where the model trained with a sub-graph outperforms the model trained with the full Level 0 graph. In this case, the Freebase Level 2 and NELL Level 3 average accuracy (*MRR_FIL*) outperforms the overall Level 0 average.

DistMult Summary				
Dataset	**Filter Level**	**Difference in Training Data (vs Level 0)**	**Difference in Accuracy (vs Level 0)**	**Outperforms Level 0 Level 0 for**
Freebase	2	-71.22%	**5.5%**	All Level 2
Freebase	3	-27%	-16%	0-200k relations
NELL	2	-89%	-38%	70-80k relations
NELL	3	-67%	**14%**	All of Level 3

Table 5. Summary Results using DistMult Model

For the other scenarios - DistMult trained with Freebase Level 3 or NELL Level 2 - we again see that the reduction in accuracy versus Level 0 is significantly less than the reduction in the training data used, which again supports the suggestion that there is not a linear relationship between the two here. As before, we can drill into these results to get a detailed breakdown by ranges of relations within both Level 2 & 3. For the Freebase Level 3 average, the breakdown by range of relations shows that DistMult models trained with Level 3 sub-graphs of size up to 150k relations outperform the overall Level 0 average. It is when the Level 3 sub-graph sizes go beyond 200k relations (up to a max of 300k relations) that the average accuracy drops well below the overall average. Looking at the NELL Level 2 averages breakdown, we see that DistMult models trained with 70-80k relations outperformed the overall average. However, most of the NELL Level 2 sub-graph ranges performed less well than the overall average.

5.1 Correlating Model Accuracy with Graph Features

The results discussed so far show that there are some scenarios where the models trained with a subgraph outperform those trained with the overall graph in the prediction of specific link types. Why might this be? Is there some feature of these subgraphs that distinguishes them from others? To examine this, the average clustering coefficient and the density were calculated and recorded alongside the other results. The charts in Figure 5 show the variation in both the average clustering coefficient and the density of the subgraphs against the accuracy of the trained ComplEx model (using *MRR_FIL* for accuracy).

Table 6 below summarises the observations from these charts. There is a slight trend of models showing higher accuracy being those trained with sub-graphs of a lower clustering coefficient. The density of the sub-graphs remains relatively constant regardless of the resulting accuracy of the trained models.

Model	**Dataset**	**Level**	**Clustering Coefficient**	**Density**
ComplEx	Freebase	1	Slightly down	Almost level
ComplEx	Freebase	2	Down	Almost level
ComplEx	Freebase	3	Scattered	Level
ComplEx	NELL	1	Level	Level
ComplEx	NELL	2	Down	Level
ComplEx	NELL	3	Slightly down	Level
DistMult	Freebase	1	Scattered	Level
DistMult	Freebase	2	Down	Level
DistMult	Freebase	3	Scattered	Level
DistMult	NELL	1	Level	Level
DistMult	NELL	2	Scattered	Almost level
DistMult	NELL	3	Up	Level

Table 6. Summary of Clustering Coefficient and Density variation of sub-graphs versus accuracy of the resulting trained model

It would seem that the density of the sub-graph is independent of the accuracy of the model trained on that data. The additional relations in the subgraphs of higher density do not seem to be providing any additional value in training the model. With clustering coefficient, there is a slight pattern of the subgraphs used to train models with higher accuracy having a lower average clustering coefficient. However, the pattern is not strong enough to draw confident conclusions from.

6 Conclusion

The findings from the experiments show a pattern where models trained with sub-graphs of certain sizes either outperformed the model trained with the entire graph or at least showed a decrease in accuracy that was significantly smaller than the decrease in the amount of training data used. However, there was no consistent pattern in terms of the optimal sub-graph to use to train the model. What is it then that causes some “levels” of sub-graph to perform better than others in training the model? Two specific graph properties were identified as potential candidates - average clustering coefficient and density; however our results do not point towards either property being clearly correlated with the accuracy of the resulting trained model. Further work is required to look at additional graph features/characteristics that may be correlated with model accuracy, such

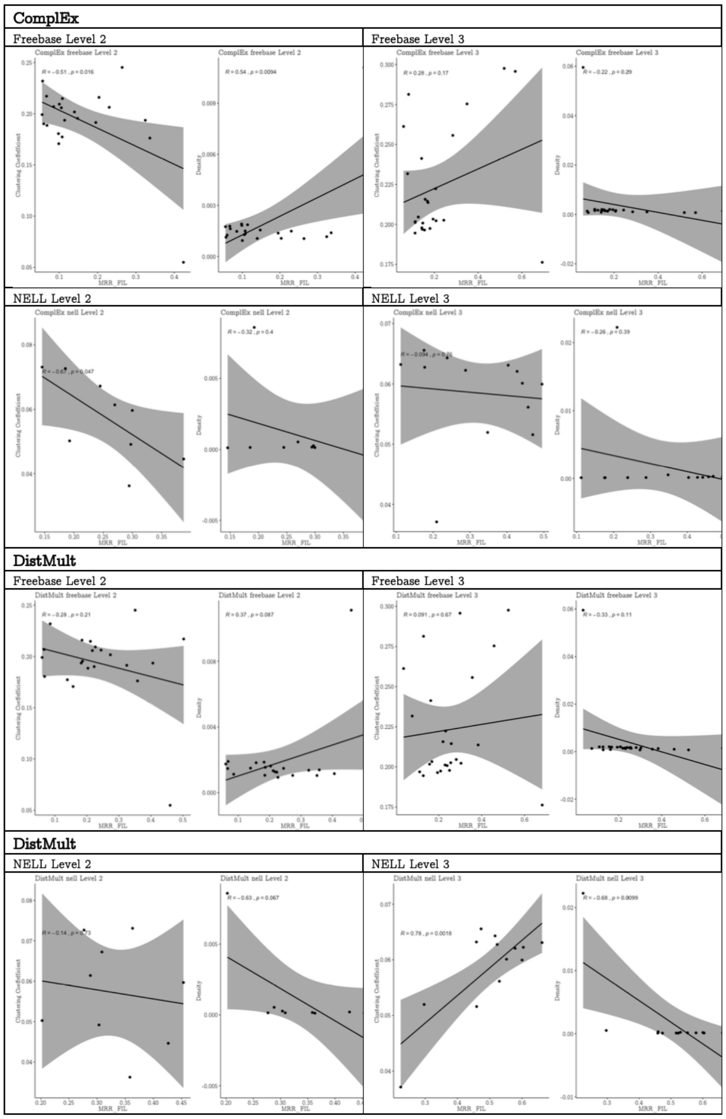

Fig. 5. Average clustering coefficient (left) and density (right) against MRR_FIL (accuracy) for ComplEx & DistMult models trained on Level 2 and 3 Freebase & NELL sub-graphs

as Average Node Degree, Edge Betweenness or Graph Connectivity. Finally, edge/link direction could be considered in the selection of the sub-graph to train the model. The experiments in this work used sub-graph levels that consisted of adjacent links/relations that were extracted independently of the direction of the relation. Future work will examine if relation directionality should be considered when constructing sub-graphs.

Acknowledgements: This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2, co-funded by the European Regional Development Fund.

References

1. Adamic, L., Adar, E.: Friends and neighbors on the web. Social Networks 25, 211–230 (07 2003)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: The Semantic Web, pp. 722–735. Springer Berlin Heidelberg, Berlin, Heidelberg (11 2007)
3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD '08. p. 1247. ACM Press (2008)
4. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. pp. 2787–2795. NIPS'13 (2013)
5. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: Twenty-Fifth AAAI Conference on Artificial Intelligence (2011)
6. Brightwell, G., Winkler, P.: Maximum hitting time for random walks on graphs. Random Structures & Algorithms 1(3), 263–276
7. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, Jr., E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence. pp. 1306–1313. AAAI'10 (2010)
8. Grimes, S.: Unstructured Data and the 80 Percent Rule (2008), `http://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/`
9. Jeh, G., Widom, J.: Simrank: A measure of structural-context similarity. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 538–543. KDD '02 (2002)
10. Katz, L.: A new status index derived from sociometric analysis. Psychometrika 18(1), 39–43 (Mar 1953)
11. Krishnan, A.: Making search on Amazon easier (8 2018), `https://blog.aboutamazon.com/innovation/making-search-easier`
12. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In: Proceedings of the Twelfth International Conference on Information and Knowledge Management. pp. 556–559. CIKM '03 (2003)

13. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the Fifteenth International Conference on Machine Learning. pp. 296–304. ICML '98 (1998)
14. Miller, G.A., A., G.: WordNet: a lexical database for English. Communications of the ACM 38(11), 39–41 (11 1995)
15. Mohamed, S.K., Nováček, V.: Link Prediction Using Multi Part Embeddings. In: The Semantic Web - 16th International Conference, {ESWC} 2019, Portorož, Slovenia, June 2-6, 2019, Proceedings. pp. 240–254. NUI Galway (2019)
16. Navigli, R., Ponzetto, S.P.: BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence 193, 217–250 (12 2012)
17. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. Proceedings of the IEEE 104(1), 11–33 (2016)
18. Nickel, M., Rosasco, L., Poggio, T.: Holographic Embeddings of Knowledge Graphs. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. pp. 1955–1961. AAAI'16, AAAI Press (2016)
19. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on International Conference on Machine Learning. pp. 809–816. ICML'11 (2011)
20. Panetta, K., Gartner: 5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies, 2018 - Smarter With Gartner (2018), https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/
21. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning 62(1-2 SPEC. ISS.), 107–136 (2006)
22. Rossi, R.A., Mcdowell, L.K., Aha, D.W., Neville, J.: Transforming Graph Data for Statistical Relational Learning. Journal of Artificial Intelligence Research 45, 363–441 (2012)
23. Singhal, A.: Introducing the knowledge graph: things, not strings. Official google blog (2012)
24. Tensor Flow Team, G.R.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems (2015), www.tensorflow.org.
25. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: 3rd Workshop on Continuous Vector Space Models and Their Compositionality. ACL - Association for Computational Linguistics (July 2015)
26. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. pp. 2071–2080. ICML'16 (2016)
27. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering 29(12), 2724–2743 (Dec 2017)
28. West, R., Gabrilovich, E., Murphy, K., Sun, S., Gupta, R., Lin, D.: Knowledge base completion via search-based question answering. In: Proceedings of the 23rd International Conference on World Wide Web. pp. 515–526. WWW '14 (2014)
29. Yang, B., Yih, S.W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the International Conference on Learning Representations (ICLR) 2015 (May 2015)
30. Zhou, T., Lü, L., Zhang, Y.C.: Predicting missing links via local information. The European Physical Journal B 71(4), 623–630 (Oct 2009)