Entity linking for structured data using knowledge graph embeddings and energy functions for semantic matching

Advait Savant
Prof. Zeidenberg

I. Introduction

We have a set of relational databases, with a set of themes defining the scope of the data content. For each relational database as a dataset, we have of attributes consisting of a consistent theme. Every tuple in the dataset represents information about an entity/object with every attribute depicting a certain relation/property of the concerned entity. Across the set of tuples in the dataset, we assume that the tuples are related to each other based on the semantics of some underlying data generating process.

For every attribute column in the dataset, we have entity mentions in that column. We would assume that these entity mentions would be sharing a common pattern, likely governed by the head of that column, which enables them to be put in the same column. For example, a dataset can consist of names of cars, or names of governors of a state among it's attributes. Here, the implicit semantic relation between the entities in the column, is that each entity is a type of car or each entity is a governor of the state. Our task is, given an attribute and instances of the attribute consisting of entity mentions across tuples, for a given entity mention, we would want to correctly associate it with the corresponding real world entity.

As a toy example, we have a test column consisting of the following names; Newton, Einstein, Bose, Lagrange, Nakamura, Boltzmann, Hamilton etc. We would want to associate the entity mention Hamilton here with the physicist William Rowan Hamilton rather than other people with the same name such as Alexander Hamilton, a founding father of the United States, or other entities named Hamilton, like the city Hamilton in New Zealand. We can facilitate such an association by linking the entity mention with the corresponding entity id in the wikidata entry.

We will construct effective preliminary models for dealing the same. We will also deal with row based disambiguation and the limitations of notation in wikidata. In practise we work with themes associated with the hoovers database. We intend to develop a system to expand a given relational database into a smarter database by connecting it's entity mentions with a larger relevant knowledge base, allowing the user access a wider space of information. We look into considerations in state space search, machine learning based approaches for the same.

II. Challenges and considerations

Consider the problem with standard supervised learning approaches here. For entity mention in a column, e.g. 'Hamiltion', there would be one corresponding embedding in standard text based word embeddings like Bert based embeddings or word2vec,Glove. For effective disambiguation, we would need a different embedding for each entity corresponding to an entity mention. Also, importantly, since each entity is uniquely mapped to a wikidata QID, in what is an annotation process not necessarily based on factors conducive to pattern recognition, the underlying function to be learned might not be subjected to interpolation and extrapolation by learning representations based on some training data, depending on some inductive bias of a learning algorithm like a decision tree, a deep neural net or an SVM.

Consider a mapping to learn from set A to set B. Elements of set A can be seen as a part of a continuous embedding space. Elements in set B are discrete. Elements in set A are uniquely assigned to elements in set B in an arbitrary order, pattern recognition for extrapolation/interpolation here would not be very useful.

While the assignment of QIDs follows a systematic, sequential pattern, (namely, Q1 represents the concept of ”universe.” Q2 is assigned to ”Earth.” Q3, Q4, Q5, and so on are assigned to other entities, progressively based on the order of their creation), the process does not offer meaningful information for the task of column entity disambiguation.

The QID itself does not encapsulate any information about the characteristics or attributes of the entity it represents, thus learning a model based on QID patterns would not provide a way to disambiguate between entities in a meaningful manner.

However, the structured data and relationships that are associated with each QID in Wikidata are useful for entity disambiguation tasks and can be leveraged in a supervised learning context.

We also need permutation invariance of the input in our model representation, since the disambiguation of an entity mention is not correlated with the order of other entity mentions presented in it's context.

III. Solution approach 1: Entity linking for relational data by learning representations through heterogeneous knowledge graph reasoning

Consider an attribute A with entities $m_1, m_2, .., m_n$. For every entity, we generate a list of candidate entities based on fuzzy string matching techniques. [1][2][3].

m_1 will be associated with $\phi(m_1)$ consisting of $e^1_1, e^1_2, e^1_3, ..., e^1_{k_1}$

m_2 will be associated with $\phi(m_2)$ consisting of $e^2_1, e^2_2, e^2_3, ..., e^2_{k_2}$

We here find it useful to define a knowledge graph. A knowledge graph G(V,E) will consist of nodes V and set of edges E wherein each edge e_i from E will denote a connection between two vertices v^i_r and v^i_l from V. In a heterogeneous knowledge graph, we will have edges which have types and denote relations between the node entity objects. Let the type of the relation be from the set R consisting of $r_1, ...r_m$. For every edge, the left node entity, the relation type and the right node entity constitute an instantiation of a relation denoted by (e^l_i, r_i, e^r_i) The relations can be represented through properties in data through the induced knowledge graph. Consider our previous example on column A consisting of Hamilton, Newton and related physicists/mathematicians.

In wikidata, Entity William Rowan Hamilton has the code (Q11887). Entity Physicist has the code (Q169470). Property occupation has the code (P106).

The corresponding relation is (William Rowan Hamiltion, occupation, Physicist). Now, based on our considerations of the semantic structure of the entities in the column, we expect the following relation to hold in all likelihood for some entity e_i in our column, (e_i,occupation, Physicist).

We intend to set up a model to learn to use this information in order to perform the disambiguation across various candidate entity sets.

For entities and relation types, we use vector embeddings obtained via the traversal of a large induced knowledge graph.[4] [5]

We have an energy function $f(e^r_i, r_i, e^l_i)$ in order to represent relations [6] [7]. The energy functions are designed to effectively represent the semantics of a relation tuple (e^r_i, r_i, e^l_i). A low value of which can indicate a high likelihood of the tuple being a valid relation in the knowledge graph. We will use these trained embeddings and parameterized functions as initial representations for our task.

We set up the graph neural network model as follows.

We explain how information flows across the candidate entities with a concrete example. We will consider a window of mentions of size q, Let us consider set $\phi(m_1)$ of size 3, $\phi(m_2)$ of size 4, $\phi(m_3)$ of size 2. As for $\phi(m_1)$, it has three candidate entities For the entity e^1_1, e^1_2, e^1_3. Let us say that e^1_2, e^2_3 and e^3_1 are the appropriate entities.

We have a model graph M_G associated with our model which builds on the top of the graph W_G which follows from wikidata.

In M_G, each entity in $\phi(m_1)$ will be connected to every entity in $\phi(m_2)$ and $\phi(m_3)$. We follow the message passing and aggregation format across layers in a graph neural network for reference. The modeling choice is in line with the structure of the data. In order to make predictions on a candidate entity, we find the underlying relation governing the column. In order to find the underlying relation governing the column, we can aggregate information about the candidate entities of the neighbor mentions and make predictions over the relations associated with an entity. We can then aggregate the predictions in the relation space.

By relation space, in terms of embeddings, we mean the following. If we are to find a similarity score based on an entity embedding as it is and a relation type embedding by itself, it will not be semantically meaningful. We can find similarities between entities and between relation types. We can associate entities and their transformations based on relation types to map to other entities.

e^2_1 will aggregate and transform messages from the embeddings of its M_G neighbors and learn a transformation given its own embedding. Every layer is parameterized by a linear transformation followed by an activation for learning non linear mappings. Consider e^2_1. For every entity in $\phi(m_1)$, it will be connected to each entities in $\phi(m_2)$ and $\phi(m_3)$.

The forward pass for e^2_1 will work as follows. We have layer 1 and layer 2. For layer 2, e^2_1 will have some input embedding $h_1[e^2_1]$. e^2_1 will receive messages from $\phi(m_2)$ and $\phi(m_3)$. These messages will be obtained via local attention driven aggregation of trainable parameterized transformations and a subsequent sum over the transformed embeddings from each of the entities of $\phi(m_2)$ and $\phi(m_3)$. It will then take its own embedding and the messages received and map the same to embedding $h_3[e^2_1]$. This will be used to compute a measure of similarity with the relation type embeddings of each of the relations which e^2_1 is a part of.

Our approach helps us to mitigate computational issues based on an expensive softmax while continuing to enable us to consider the full range of relations.

For the case of William Rowan Hamilton, h_3 should give a high similarity with the embedding for occupation. The similarity score can be a parameterized bilinear form [8][9] or a simple dot product.

In layer 1, each of the entities in $\phi(m_2)$ will receive messages from each entity in $\phi(m_1)$ and $\phi(m_3)$ based on the transformations of the initial embeddings. Incorporation of *noise filtering* techniques and an *attention mechanism* for graphs should help in learning the message passing scheme for desired signal entities and learn transformations to predict over the relation space.

The original entities Hamilton, Newton and Lagrange, their corresponding embeddings will be two hops away from each other with physicist being a central node and occupation being the relation type. In layer one, each of these entities will receive information from the other physicist entities and other candidate entities which are not signals. As each entity, signal and non-signal(appropriate and not relevant) receives information from other entites, a signal entity can learn a message to propagate to other signal entities in layer 2 picking up on the commonality of the physicist occupation relation with the various signal entities which it is getting information from in layer 1. The signal entities will be trained to do the same based on structured parameterizations. We will predict a maximum similarity on a dummy relation for non-signal entities.

Once we have predicted a candidate relation for each candidate entity, we will use a dynamic programming algorithm to find the maximal appropriate relation. We will then use the energy function, which we can chose to fine tune, in order to make inferences of the appropriate right entity in the relation tuple. We will chose the maximal appropriate right entity again based on dynamic programming. Having done this, we can infer in each candidate set, in order to find the most appropriate left entity and complete our disambiguation task.

A. *Specificity and global coherence constraints*

In order to prevent the selection of trivial properties such as human, we do the following.

We have the predicted relation $r*$ and the predicted right entity $e*$. For $\phi(e_1)$, if we select e_1^2, we have $(e_1^2, r*, e*)$ as a valid relation in M_W. We would not want relations $(e_1^3, r*, e*)$, $(e_1^2, r*, e*)$ to be present in M_W since that defeats the purpose of disambiguation. This can be the case if $r*$ is a trivial property.

We would want there to be $(e_l, r*, e*)$ for some e_l in each ϕ. Across the sets ϕ, we would want to have coherence held by the energy function values in view of shared $r*, e*$. Within the sets ϕ, we do not want coherence associated with $r*, e*$ based energy function values for left entities in the same set. We introduce penalty terms for the same.

IV. Solution Approach 2: Entity linking for relational data by learning representations through heterogeneous knowledge graph reasoning

Key idea: Instead of predicting on relations in M_W of the candidate entites in M_G, create an abtract node for each mention, make a sequential prediction mapping from the abtract node over to candidate entities. At every stage, the set of selected entities should be having an induced path which is maximally interconnected and with a minimal distance constraint

We can conceptualize an approach to the problem entirely in the entity space reasoning that the entity embeddings developed are themselves contigent on the relation types which the entities partake in. As a caveat, we can include the an aggregation of the relation types for each entity in our workflow, we can touch upon the same later.

For every mention m_i, we have an abstract node m_i. m_i will to connected to the candidate entities which it is a part of $\phi(m_i)$. In a window of mentions of an arbitrary size q, m_i will be also be connected to abstract nodes $m_{j_{j \neq i}}$ for j $\epsilon 1, .., q$.

We have W_{em} the denoting the parametrization for entity to mention connections. We have W_{mm} denoting the parametrization for mention to mention connections and W_{ee} denoting the parametrization for entity to entity node connections.

Consider a concrete example. Let us consider again the set $\phi(m_1)$ of size 3, $\phi(m_2)$ of size 4, $\phi(m_3)$ of size 2. As for $\phi(m_1)$, it has three candidate entities For the entity e_1^1, e_2^1, e_3^1. Let us say that e_2^1, e_3^2 and e_1^3 are the appropriate entities. m_1,m_2 and m_3 will be abstract nodes connected to each other.

The computational graph rooted at m_1 which will govern its forward pass will be as follows. In layer 0, each entity will receive messages from other entities based on W_{ee}. Now, in layer 1, each of the mentions will receive messages from their respective candidate entities via W_{em}. layer 2, each m_1 will receive a message from its own candidate entities via W_{em} and the other mentions via W_{mm}. This will enable us to generate an embedding for m_1 based on parametrized transformations.

We will then learn to compute a similarity score between the embedding for m_1 and the embeddings for its candidate entities. We can represent the paramtrization as a bilinear form enabling us to model our problem as an edge classification problem in graph neural networks. Based on the mention - entity similarity, we can make an selection over the candidate entities for m_1. We denote this as $\hat{e}_1$. Let the true entity be y_1.

We can repeat the procedure for m_2 and m_3.

We will have a one-to- loss wrt $\hat{e}_1$ and y_1.

We also have a loss which would constrain $\hat{e}_1$, $\hat{e}_2$ and $\hat{e}_3$ to be close to each other in the knowledge graph M_W. This is akin to the neighborhood condition [10] and the functionally analogous to the minimal path condition [11].

This should help us in our training since we know that Hamiltion and Newton are two stops from each other by virtue of the edges (William Rowan Hamiltion, occupation, Physi- cist) and (Newton, occupation, Physi- cist). We would expect the city of Hamilton in New Zealand to have relatively lesser similarity with the Newton embedding. We develop a weighted multi objective optimization criterion based on the same considerations and experiment with a contrastive setting for the second loss.

Forward pass:
Layer 0: Entity to Entity Message Passing

$$h_{e_j}^{(1)} = \sigma \left(W_{ee}^{(1)} \sum_{k \in N(e_j)} \alpha_{jk}^{(1)} h_{e_k}^{(0)} \right), \tag{1}$$

Layer 1: Entity to Mention Message Passing

$$h_{m_i}^{(1)} = \sigma \left(W_{em}^{(1)} \sum_{j \in \phi(m_i)} \alpha_{ij}^{(1)} h_{e_j}^{(1)} \right), \tag{2}$$

Layer 2: Mention to Mention Message Passing

$$h_{m_i}^{(2)} = \sigma \left(W_{mm}^{(1)} \sum_{l \in N(m_i)} \alpha_{il}^{(1)} h_{m_l}^{(1)} + W_{em}^{(2)} \sum_{j \in \phi(m_i)} \alpha_{ij}^{(2)} h_{e_j}^{(1)} \right), \tag{3}$$

where the attention weights can be computed using

$$\alpha_{jk}^{(t)} = \text{softmax} \left(\text{Attention\_Parameters}_{jk}^{(t)} \cdot [h_{e_j}^{(t-1)}; h_{e_k}^{(t-1)}] \right), \tag{4}$$

Here, σ denotes a non-linear activation function, and t is used to denote the layer number. The different attention parameters allow the model to differentiate the importance of messages coming from different entity sets, potentially applying different normalization strategies across these sets.For attention parameters across set A and set B, it is not feasible to learn attention parameters for each combination AB. Instead, we will learn parameters for candidate entity set A as a function of it's entities, similarly for candidate entity set B, then meaningfully combine them to obtain the parameters for AB, we try addition, multiplication, parameterized combination. We can also try a more generic attention mechanism for graph neural nets.

Loss and optimization:
1. Entity-Mention Alignment Loss (L_{ema}):

$$L_{ema}(m_i, e_j) = \sum_{i=1}^{N} D(\phi(m_i), y_i)$$

2. Graph Structure Loss (L_{gs}):

$$L_{gs}(e_i, e_j) = \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} F(d(e_i, e_j))$$

The multi-objective optimization utilizes these loss functions with a weighted approach:

$$L_{total} = \alpha L_{ema} + \beta L_{gs}$$

Side note: There will certain candidate entity sets which will be un ambiguous, of size 1. For those mentions, entity disamguation is trivial. How can we effectively use the label information of such mentions in our training procedure? This is analogous to the problem of label propagation.

Side note: we are in the process of evaluating our GNN based approaches, as we test and tune them for scalability and generalization.

Side note : Subsequently, we are also investigating a transformer based approach. Removing positional encodings for permutation invariance. Compositional attention [12] could be useful for our scheme since it could enable entities to search for other entities in the context of a relation. Based on certain considerations, large language models can be seen to function as knowledge bases, encoding certain relation triplet information [13]. Meta learning based on task driven prompts for entity mention categorization(in terms of the relation) and subsequent disambiguation using energy based inference could be potentially tried [14].

V. State space search approach baseline

$$m_1 - > e_1^1 : r_1, ..r_k, .., e_d^1 : r_1, ..., r_d$$

$$m_2 - > e_2^2 : r_{2,1,1}, ..r_{2,1,k}, .., e_d : r_1, ..., r_d$$

To find: maximal common relations
Mapping: $(M * E * K * E') - > (M - > E)$

$$T_{naive} = O(D^N K^2)$$

$$T_{DP} = O(D^2 N K^2)$$

References

[1] Yixin Cao et al. “Neural collective entity linking”. In: *arXiv preprint arXiv:1811.08603* (2018).

[2] Octavian-Eugen Ganea and Thomas Hofmann. “Deep joint entity disambiguation with local neural attention”. In: *arXiv preprint arXiv:1704.04920* (2017).

[3] Zhaochen Guo and Denilson Barbosa. “Robust named entity disambiguation with random walks”. In: *Semantic Web* 9.4 (2018), pp. 459–479.

[4] Aditya Grover and Jure Leskovec. “node2vec: Scalable feature learning for networks”. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016, pp. 855–864.

[5] Zhen Wang et al. “Knowledge graph embedding by translating on hyperplanes”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 28. 1. 2014.

[6] Antoine Bordes et al. “A semantic matching energy function for learning with multi-relational data: Application to word-sense disambiguation”. In: *Machine Learning* 94 (2014), pp. 233–259.

[7] Antoine Bordes et al. “Learning structured embeddings of knowledge bases”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 25. 1. 2011, pp. 301–306.

[8] Richard Socher et al. “Reasoning with neural tensor networks for knowledge base completion”. In: *Advances in neural information processing systems* 26 (2013).

[9] Bishan Yang et al. “Embedding entities and relations for learning and inference in knowledge bases”. In: *arXiv preprint arXiv:1412.6575* (2014).

[10] Yixin Cao et al. “Bridge Text and Knowledge by Learning Multi-Prototype Entity Mention Embedding”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1623–1633. DOI: 10.18653/v1/P17-1149. URL: https://aclanthology.org/P17-1149.

[11] Wenhan Xiong, Thien Hoang, and William Yang Wang. *DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning*. 2018. arXiv: 1707.06690 [cs.CL].

[12] Sarthak Mittal et al. *Compositional Attention: Disentangling Search and Retrieval*. 2022. arXiv: 2110.09419 [cs.LG].

[13] Fabio Petroni et al. *Language Models as Knowledge Bases?* 2019. arXiv: 1909.01066 [cs.CL].

[14] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].