Domain-specific Hierarchical Subgraph Extraction: A Recommendation Use Case

Sarasi Lalithsena
Kno.e.sis Center
Wright State University
Dayton, OH, USA
sarasi@knoesis.org

Sujan Perera
IBM Almaden Research Center
San Jose, CA, USA
sujan.perera@ibm.com

Pavan Kapanipathi
IBM Research AI
Yorktown, NY, USA
kapanipa@us.ibm.com

Amit Sheth
Kno.e.sis Center
Wright State University
Dayton, OH, USA
amit@knoesis.org

***Abstract*—Hierarchical relationships play a key role in knowledge graphs. Particularly, large and well-known knowledge graphs such as DBpedia contain significant number of facts expressed with hierarchical relationships in comparison to the other types of relationships. These hierarchical relationships are extensively harnessed by applications such as personalization, question answering, and recommendation systems. However, the presence of large number of facts with hierarchical relationships makes the applications computationally intensive. Additionally, the applications can be domain-specific and may not require all the hierarchical facts available, but only require those that are specific to the domain. In this paper, we present an approach to extract domain-specific hierarchical subgraph from large knowledge graphs by identifying the domain-specificity of the categories in the hierarchy. Given a domain, the domain-specificity of categories are determined by combining different types of evidence using a probabilistic framework. We show the effectiveness of our approach with a recommendation use case for movie and book domains. Our evaluation demonstrates that the domain-specific hierarchical subgraphs extracted by our approach can reduce the baseline subgraph by 40% to 50% without compromising the accuracy of the recommendations. Furthermore, the presented approach outperforms the recommendation results obtained with a state-of-the-art domain-specific subgraph extraction technique which uses supervised learning.**

***Keywords*-domain-specific knowledge graph; recommendation systems; hierarchical relationships; probabilistic soft logic**

I. Introduction

Hierarchical relationships are one of the key components of knowledge graphs (KGs). They induce a structure of generalization and specialization of concepts[1] in a KG. For example, Fig. 1(b) shows a subgraph of the Wikipedia Category Hierarchy (WCH) comprising of entities and categories. In the Fig. 1(b), the entity *Franco Zeffirelli* connects to the category *Italian Film Directors* using the hierarchical relationship *broader* and the category *Italian Film Directors* connects to the category *European Film Directors* using the same relationship.

Commonly used KGs such as DBpedia, Yago, and Freebase contain a significant number of facts expressed with hierarchical relationships. Fig. 2 shows the dominance of number of facts expressed with hierarchical relationships compared to the facts expressed with other relationships within 150 most frequently used DBpedia relationships. They play a major role in intelligent applications such as personalization [2], question answering [3], and recommendation systems [4]–[7]. Particularly, in recommendation systems, the hierarchical relationships between movies such as *Stormbreaker* and *Fair game* via genre *Category: American Spy Films* have been exploited to suggest new movies [4], [5].

These applications encounter two primary challenges in consuming KGs: (1) they are computationally intensive due to the very large number of facts in the KGs [8]; and (2) while most applications that utilize KGs are domain-specific and require knowledge that is specific to the domain of interest, many of the KGs are generic and comprise of facts representing multiple domains. Hence, naive usage of such KGs may introduce noise. In order to overcome these challenges, existing applications extract relevant subgraphs by navigating a predefined number of hops (n-hops) from a set of entities that represent the domain [4], [6].

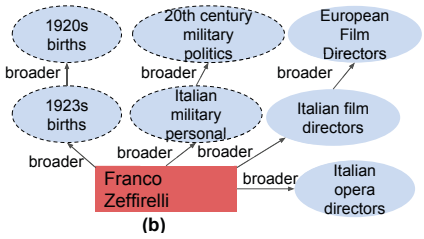
For example, Fig. 1(b) shows a subgraph extracted by traversing up to 2-hops from the entity *Franco Zeffirelli* in WCH. According to the subgraph, entity *Franco Zeffirelli* belongs to categories *Italian Film Directors*, *Italian Opera Directors*, *Italian military personal*, and *1923s births*. Out of these categories, the first two categories are relevant to the movie domain and the last two are irrelevant and might not be necessary for a movie recommendation system. Furthermore, Fig. 1(a) shows the exponential growth of the number of paths reached of a movie-specific subgraph created by navigating up to 4-hops in WCH, starting from $3,121$ movie entities in the MovieLens[2] dataset. Hence, the simple n-hop expansion technique to extract a domain-specific subgraph has not shown to be effective either in reducing size or noise from a large KG. Therefore, given a domain of interest, the primary goal of this work is to extract a domain-specific hierarchical knowledge graph that reduces the size of the KG without compromising the accuracy of

[1]Concepts represent both entities (articles) and categories on Wikipedia [1].

[2]https://grouplens.org/datasets/movielens/

Hops	Paths Reached
1	50953
2	18M
3	345M
4	72B

(a)



(b)

Figure 1: (a) Number of paths reached via seed movie enties; M - million; B - billion; (b) Hop-based path navigation. Entities and Categories are indicated by rectangles and ovals respectively. Out of domain categories for the movie domain are indicated by ovals with dotted lines.

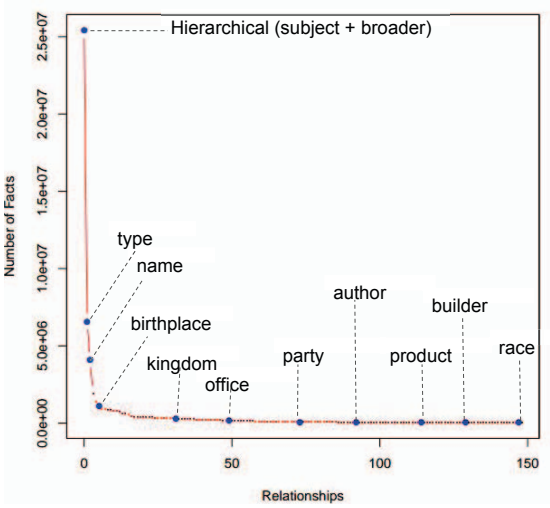


Figure 2: Number of facts for each relationship.

the applications.

We propose an evidence-based approach for extracting a domain-specific subgraph from a hierarchical KG. We selected WCH as the testbed to develop and evaluate our proposed approach due to its prominence and usage in existing domain-specific applications. WCH is the main hierarchical knowledge source for many KGs such as DBpedia and YAGO and consists of 7.5 million entities and categories connected via 25 million hierarchical relationships.[3] Our approach collects evidence for supporting or opposing a category's relevance to a given domain in the KG. These pieces of evidence are based on type, lexical, and structural semantics of the categories. To systematically combine the different sources of evidence and to manage the uncertainty associated with each of the source, we use the probabilistic soft logic (PSL) framework [9]. PSL is proven to work well in such environments and has generated state-of-the-art results [10], [11]. To demonstrate the effectiveness of the subgraphs extracted by our approach, we pick a recommendation use case which is an important application leveraging KGs. We show that it is possible to extract a domain-specific subgraph from a hierarchical knowledge graph, reducing around 40% - 50% of the paths compared to the subgraph created with a n-hop based navigation technique. This is done without compromising the accuracy of the recommendation algorithm and in most cases, domain-specific subgraphs extracted by our approach also improve the accuracy of the recommendation system. We also demonstrate that the recommendation results obtained by the subgraph created with our approach outperforms the results obtained by a subgraph created with a supervised learning technique.

The rest of the paper is organized as follows. In Section II, we describe our approach, followed by evaluation in Section III. Section IV details the related work and Section V concludes with suggestions for future work.

[3]http://wiki.dbpedia.org/downloads-2016-04

II. Approach

In order to extract a domain-specific hierarchical subgraph, our approach considers a set of domain entities as the input. These domain entities represent the domain of interest. For example, to create a movie-specific hierarchical subgraph a set of movie entities would be the input to our approach. Given the domain entities and the hierarchical graph, a domain-specific hierarchical subgraph is extracted by expanding the domain entities to only the categories that are specific to the domain. Existing approaches [4] consider connectedness to the domain entities in the hierarchical graph as the proxy to estimate the domain-specificity. For example, Fig. 3 shows a subgraph extracted by navigating 2 to 3 hops starting from five movies in WCH. While, this subgraph contains categories which describe the genre of the movie (*American LGBT related films*, *America spy films*) and the director of the movie (*Films Directed by Doug Liman*), it also contains out-of-domain categories such as *Museums in Popular Culture*, *LGBT Culture in US*, *Education*, etc. With respect to this subgraph, the domain-specific hierarchical subgraph should retain categories describing genre and director of the movies and eliminate the out-of-domain categories. In order to assess and retain the domain-specific categories, we have identified three different types of semantics of categories in the hierarchy that can be quantified and systematically leveraged, they are; 1) type semantics, 2) lexical semantics, and 3) structural semantics. These provide evidence for assessing the domain-specificity of categories. We discuss these three types of semantics in Section II-A and Section II-B describes how the evidences obtained via these types of semantics are aggregated to calculate the domain-specificity of the categories using a probabilistic framework.

A. Evidence types towards domain-specificity

A category on Wikipedia can span across multiple topics. Our intuition is that the domain-specificity of a category can be estimated based on the relevancy of its associated topics to the domain. The relevancy can be determined through the type and lexical semantics of the category label.

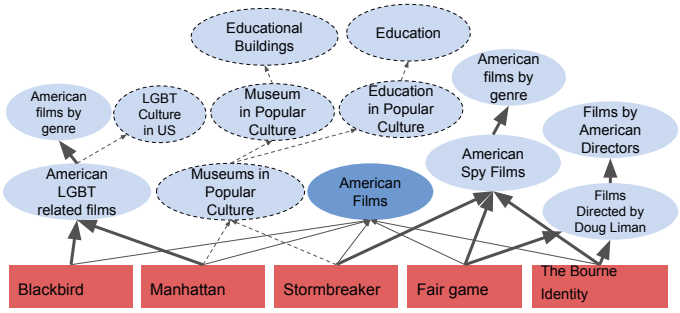


Figure 3: n-hop expansion subgraph with hierarchical relationships. The graph is created by navigating 2 to 3-hops from five movies (marked in boxes). The in-domain and out-of-doman categories are marked by ovals with a solid lines and dotted lines respectively.

1) Type semantics: The topics of a category can be made explicit by identifying the types of the entities mentioned in the category label. For example as shown in Fig. 4(a) the category 1) *Films Directed by James Cameron* has types *Film* and *Film Director*; 2) *Documentary films about horror* has types *Documentary Films* and *Horror Film*; and 3) *Kingdom and countries of Austria-Hungary* has types *Monarchy* and *Country*. These types can be utilized to show that first two categories are specific to the movie domain while third category is not specific to the movie domain.

2) Lexical semantics: The topics of a category label can also be derived using their lexical semantics. For example, categories *1997 anime* and *biographical work* have topics *animation films* and *biographical films* respectively. However, they cannot be identified via type semantics as current computational techniques fall short in identifying entities within them. The lexical semantics of these labels can be used to group such categories with more descriptive categories and collectively identify their topics. Fig. 4(b) shows such grouping where category label *1997 anime* and *biographical works* are grouped with other category labels which share a similar lexical pattern. This grouping enrich the semantics of these categories and enable the identification of their respective topics (i.e *animation films* and *biographical film*). The domain-specificity calculated for these topics are considered as reflective to the domain-specificity of the categories in the group.

Section II-B2 details the techniques used to identify type and lexical semantics and calculate domain-specificity of these topics.

3) Structural semantics: Abstract categories are generally shown to have less importance in most domain-specific applications that utilize knowledge graphs [3], [12]. For example, in a recommendation system, categories such as *American films* or *English-language films* may provide less or no impact in recommendations whereas specific categories such as *American action films* or *War epic films* are important for better recommendations. We utilize the structural semantics of hierarchies to quantify the abstractness of a category [13], [14]. Specifically, we consider the

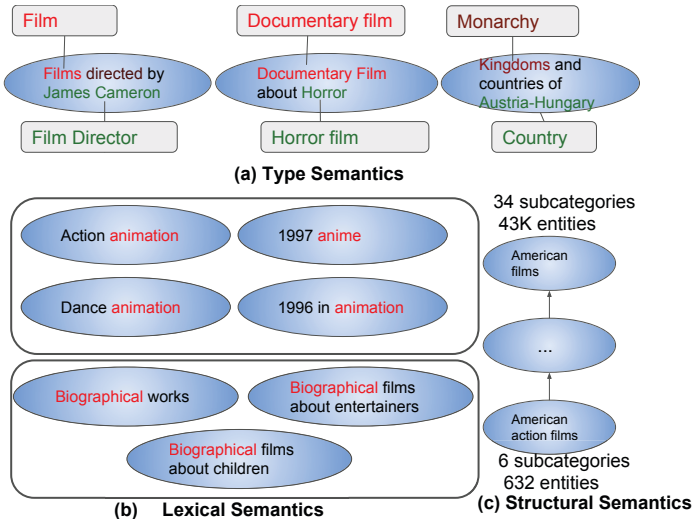


Figure 4: Evidence types for categories.

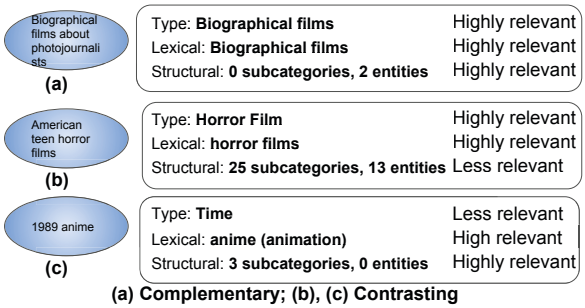


Figure 5: Complementary and contrasting evidences.

outdegree of a category to determine its abstractness. For example, Fig. 4(c) shows that the category *American films* is linked to $43k$ entities and 34 sub categories in the graph and *American action films* is only linked to 632 entities and 6 subcategories. Hence, measuring the specificity of a category using structural features provides another piece of evidence regarding the domain-specificity of a given category.

B. PSL Framework for category ranking

The three types of semantics can provide complementary and contrasting signal towards determining the domain-specificity of a category. Fig. 5 shows evidences collected for a few categories. The evidences collected for the category in (a) are complementary. The type and lexical semantic evidence collected for (b) are complimentary while structural evidence provides a contrasting signal to them. In (c) type sematic provides a contrasting signal to other two types of evidences. Our approach uses a probabilistic framework to aggregate these complementary and contrasting evidences in a principled way. Specifically, we use probabilistic soft logic (PSL), which is a statistical relational learning framework with a declarative language as the interface. Section II-B1 provides a brief overview of PSL and Section II-B2 describes the category ranking using PSL.

1) A Brief Introduction to PSL: PSL is a declarative language which uses first order logic to specify probabilistic models. PSL mainly has: (1) predicates - P, (2) atoms - P(A, B), and (3) weighted rules. A weighted rule will be of the form $w : X \rightarrow Y$, where X can be a conjunction, disjunction or an individual atom. Eq. 1 shows a simple rule where P is a predicate, A, B, and C are variables, and w is the weight.

$$w : P(A, B)\tilde{\wedge}P(B, C) \rightarrow P(A, C) \tag{1}$$

The grounding of each atom happens when the variables are instantiated with individuals. A grounded atom $P(a, b)$, where a and b are the individuals for A and B, can take a continuous value ranging from 0 to 1. The capability to deal with continuous values instead of a boolean value for atoms makes PSL more useful and practically applicable for many scenarios [10]. The value of each atom can either be observed or unknown.

PSL uses lukasiewicz t-norm [9] to provide a relaxation for the logical connectives $\wedge$ and $\vee$. If p and q are two atoms, it assigns truth values to $\wedge$ using $max(0, p + q - 1)$ and to $\vee$ using $min(1, p+q)$. Using lukasiewicz t-norm, PSL assigns a distance to satisfaction for each grounded rule. The distance to satisfaction to the grounded rule for the rule in Eq. 1 will be $max((P(a, b)\tilde{\wedge}P(b, c)) - P(a, c), 0)$.

PSL rules with grounded atoms and continuous values to represent probabilities define features for Markov networks, which are probabilistic graphical models used for collective inferencing. In particular, the PSL implementation used in this work employs Hinge-loss Markov Random Fields (HL-MRFs), which are probabilistic models over continuous random variables. HL-MRFs infer the values for the unknown variables in the model by looking at the most probable explanation (MPE). MPE tries to find an interpretation I (the most possible assignment of the soft truth values) such that it minimizes the distance to satisfaction for grounded rules R. The probability distribution over interpretation I for a set of grounded rules R can be formulated as,

$$f(I) = \frac{1}{Z} \exp[-\sum_{r \in R} w_r (d_r(I))^p] \tag{2}$$

where w_r is the weight of rule r, d_r is the rules's distance to satisfaction, p is the distance exponent, which can be 1 or 2, and Z is a normalization constant. For a detailed description of PSL, see [9].

2) PSL rules for scoring domain-specificity of categories: The proposed framework combines type, lexical, and structural semantic evidence obtained from categories using PSL to assess the domain-specificity of a category. Here, we have expressed these in the form of rules in the PSL model. The domain-specificity of categories will be the value to be inferred (unknown) and the types of evidence will be the observed (known) values. Now, we present the rules and their expansions obtained using different techniques.

PSL rules for type semantics: Type semantics harnesses the semantics of topics mentioned in the category labels. In order to extract the topics, we perform entity annotation on labels, specifically using the DBpedia Spotlight annotation tool [15] and a regular expression-based annotator to identify the time mentions. We consider the types (*rdf:type*) of the annotated entities as topics. When the type of the annotated entity is *owl:Thing*, we consider the entity itself as the topic. The domain-specificity of a topic is calculated by measuring its similarity to the domain term. The domain-specificity of a category is calculated by averaging the domain-specificity of these topics. The domain term is the *rdfs:label* of the class representing the domain in DBpedia. For example, for the movie domain, we consider the DBpedia class *dbo:Film* and use its label, which is *film*, as the domain term. The rule defined using type semantics is shown in Eq. 3,

$$\begin{aligned} semtype(Cat, TypeSet)\tilde{\wedge}semtypesim_s(TypeSet, Dom) \\ \rightarrow domainspec(Cat, Dom) \end{aligned} \tag{3}$$

where $semtype$, $semtypesim$, and $domainspec$ are predicates. Cat and Dom denote the category variable and domain variable respectively. $semtype$ predicate defines the relationship between a category and a set of topics. $semtypesim$ and $domainspec$ predicates capture the similarity of set of topics to the domain and membership value of a category to the domain respectively. s denotes the similarity measure which is used to calculate the similarity between a topic and a domain term. We use two most prominent state-of-the-art techniques as similarity measures; 1) word2vec similarity [16]: embedding-based semantic similarity measure and 2) UMBC similarity [17]: hybrid semantic similarity measure. The hybrid similarity measure uses information derived from both the knowledge base and the corpus. The rule defined in Eq. 3 will be expanded to two rules using each similarity measure. These rules state that the membership of the category to the domain is determined by its topics and similarity of those topics to the domain term.

PSL rules for lexical semantics: Lexical semantics groups the categories based on their labels and then identifies topics for each group. These topics act as a reference for measuring the domain-specificity of the members of that group. In order to perform the grouping, we use k-means clustering to cluster the category labels. The category labels are represented as vectors and these vectors are obtained by averaging the word2vec embedding vectors of the words present in the category. This simple averaging method has proven to be a strong baseline for multiple tasks [18].

The topics for each group are obtained using two methods. First, we take the most frequent entity/type obtained via DBpedia Spotlight annotations in a given cluster as its topic. Second, we extract the most frequent bigram of a given cluster as topics. Finally, we measure the semantic similarity of these two topics to the domain term using UMBC and

word2vec similarity measures. PSL rule derived from lexical semantics is shown in Eq. 4.

$$lexclutype(Cat, C) \tilde{\wedge} lexclussim_{t,s}(TC_C, Dom) \rightarrow domainspec(Cat, Dom) \quad (4)$$

where C denotes a cluster and TC_C denotes the topic for the cluster C. The predicate $lexclutype$ represents the membership of a category to the cluster C. $lexclussim$ predicate is used to represent the similarity of the topic to the domain term using a similarity measure. s denotes the similarity measure which can be either Word2vec or UMBC and t denotes the technique used to obtained the topic which is either bigram or DBpedia Spotlight annotations. The rule defined in Eq. 4 will be expanded to four rules using different combinations of the two techniques used to obtain the topics and the two similarity measures. The above rules state that the membership of the category to the domain is determined by the cluster that it belongs and similarity between the topics derived for that cluster and the domain term.

PSL rules for graph structural features: In order to measure the structural specificity of a category, we use the inverse of the out-degree of a category. Here, the out-degree refers to the sum of the number of entities assigned to the category and the number of sub-categories subsumed by the category. This is shown in the PSL rule 5,

$$graphspec(Cat) \rightarrow domainspec(Cat, Dom) \quad (5)$$

where $graphspec$ is the predicate which captures the specificity value. Hence, the structural specificity directly determines the domain-specificity of the category.

Any additional types of evidence can be easily incorporated into the PSL model. PSL rules require a weight for each of the rules and these weights can either be prespecified or can be learned. In order to learn the weights, PSL needs training data for each rule. However, it is hard to create training data in a generic way which works for any domain. So, we determine the weights of the rules via an empirical experiment, as describe in the evaluation section.

The domain-specific subgraph is created by restricting the subgraph created by navigating n-hops to the top-K domain-specific categories determined by the PSL model.

III. Evaluation

Following the related work on domain-specific subgraph extraction [8], we evaluate the effectiveness of the domain-specific hierarchical subgraphs using an existing KG-based recommendation algorithm outlined in Section III-A1. The evaluation metrics used are described in Section III-B. To demonstrate that the domain-specific subgraph extraction works for multiple domains, we performed the evaluation on both movie and book domains.

A. Evaluation Setup

1) Recommendation algorithm: We implemented the recommendation algorithm in [6] for the evaluation. This algorithm recommends items connected within 1-hop based on their similarities determined using a similarity function on the KGs. For a fair comparison, we replaced this similarity function with the measure proposed by [19] which measures the similarity of two entities x and y connected via n-hops as given below,

$$sim(x, y) = \sum_{n=1}^{N} \frac{1}{2n^2} \times \sum_{path \in Paths_n(x,y)} \sum_{e \in edges(path)} w(e)$$

where N is the maximum number of hops between two entities, $Paths_n(x, y)$ returns all the paths between two entities within n-hops, $edges(path)$ returns all the edges in the $path$, and $w(e)$ is the weight of edge e. We consider that the edges have equal weights.

2) Baseline: Recommendations with EXP-DSHG$_n$: The baseline uses above recommendation algorithm with a domain-specific subgraph created by navigating n-hops from set of given domain entities as proposed in [4], [6]. We experimented with $n = 2$ and $n = 3$ for the evaluation.

3) Our Approach: Recommendations with PSL-DSHG$_n$: For our approach, we use the n-hop domain-specific hierarchical subgraph of the WCH created using proposed method. One of the parameters in our algorithm is the set of weights for the identified rules in Section II-B2. These weights are found by executing a grid search on a randomly selected subset of evaluation dataset. To reduce the complexity of the grid search, we assume same weight values for the rules that are only differ due the similarity measure being used. Hence, the goal of this experiment was to find four weight values for; 1) the type semantic rules, 2) the lexical semantic rules derived using DBpedia annotations, 3) the lexical semantic rules derived using bigrams, and 4) the structural semantic rule. First, the grid search fixes the weights of rules 2 and 3 and vary the weights of rules 1 and 4 between 0 and 10 with step size 1. The resulting ranked list of categories are empirically evaluated by three users for their domain-specicifity. The next step vary the weight of rule 2 with the best combination of weights found in the first step. Last step repeats this for rule 3 after finding the weights for rules 1, 2, and 4. This emperical study showed that the ranked lists of categories generated with weight values *8* for rules of type 1, *6* for rules of type 4, *6* for rules of type 3, and *1* for rules of type 2 are better than the other ranked lists. Hence, we used the subgraph created with these rule weights for the experiments presented here.

4) Supervised Approach: Recommendations with SUP-DSHG$_n$: Mirylenka et. al trains a binary classifier to determine the domain relevance of Wikipedia categories [20]. Given a root category of the domain (category Film), they perform a breadth first traversal in WCH upto given max

depth and identifies relevant and irrelevant categories to the domain of interest. They showed that the model trained on a particular domain can be used to determine the domain relevancy of a another domain. Hence, we trained a model for computing domain by using provided training data and created the domain-specific subgraphs for movie and book domains. We used the recommendation algorithm outlined in Section III-A1 on the subgraph created with this supervised approach SUP-DSHG$_n$ for comaprison with our approach.

5) Datasets: We used two well-known dataset for our evaluation; 1) MovieLens dataset for the movie domain, which consists of $1,000,209$ ratings for $3,883$ movies by $6,040$ users, and 2) DBbook[4] dataset for the book domain, which has $72,372$ ratings for $8,170$ items by 6181 users. Following recent work on recommendation systems [6], we removed users who have less than 20 ratings because users who have rated few items can significantly impact the performance of recommendation systems. This resulted in reducing the MovieLens dataset to $5,886$ users with approximately $0.9M$ ratings, and the DBbook dataset to $17,802$ ratings by 812 users. For each user, we take 60% of the ratings as training data (items rated by the user) and the remaining 40% as testing data (items to be recommended).

B. Evaluation Metrics

Our primary goal is to reduce a large KG to a domain-specific subgraph while preserving the accuracy of the application that utilizes the domain-specific subgraph. Based on this, we evaluated our approach on the two aspects described below for the recommendation use case.

- Graph reduction: Graph reduction measures the reduction of *PSL-DSHG*$_n$ and *SUP-DSHG*$_n$ compared to *EXP-DSHG*$_n$ in terms of the number of categories, and the number of reachable paths within *n*-hops starting from domain entities.
- Impact on accuracy: We used two measures to calculate the accuracy of the recommendations. First, we calculate the standard measure of `precision@m` as used in [6]. `precision@m`, in comparison to other metrics such as recall, is the most suitable evaluation metric for recommender systems, particularly where the number of recommended items are pre-obtained [21]. However, it only measures the binary relevancy of the items up to the m^{th} rank, it does not capture whether recommendations obtained with *PSL-DSHG*$_n$ replaces any highly relevant items selected using *EXP-DSHG*$_n$ with relatively less relevant items. To quantify this, we developed a measure by leveraging the ratings provided by users for each item in the gold standard datasets. This measure takes the average of the ratings of a user from the gold standard dataset, calculate the deviation of the predicted rating for each relevant `top-`m item from the average and take the mean of the deviation for all relevant items. A higher positive deviation value reflects better results since it denotes that relevant items have higher ratings. This measure `ratingdev` for a given user u is formalized as,

$$ratingdev(u) = \frac{\sum_{r=1}^{R} itemrating_r - avgrating_u}{|R|}$$

where R is the number of top relevant items for user u, $itemrating_r$ is the rating given by the user u for item r, and $avgrating_u$ is the average rating for user u.

[4]http://challenges.2014.eswc-conferences.org/index.php/RecSys

	Categories	Paths
EXP-DSHG$_2$	6413	18M
PSL-DSHG$_2$*(3500)*	3844(40%)	1.62M(91%)
PSL-DSHG$_2$*(4000)*	4315(33%)	10.1M(44%)
PSL-DSHG$_2$*(4500)*	4782(25%)	10.26M(43%)

Table I: Graph reduction statistics of *PSL-DSHG*$_2$*(*K*)* in comparison to *EXP-DSHG*$_2$ in the movie domain; M denotes millions, K denotes `top-`K categories.

	Categories	Paths
EXP-DSHG$_3$	12348	320M
PSL-DSHG$_3$*(6500)*	6534(47%)	106M(67%)
PSL-DSHG$_3$*(7500)*	7508(39%)	115M(64%)
PSL-DSHG$_3$*(9000)*	9015(27%)	151M(52%)

Table II: Graph reduction statistics of *PSL-DSHG*$_3$*(*K*)* in comparison to *EXP-DSHG*$_3$ in the movie domain; M denotes millions.

C. Evaluation Results on PSL-DSHG$_n$ with EXP-DSHG$_n$

In this section, we compare the results obtained with *PSL-DSHG*$_n$ and *EXP-DSHG*$_n$ using the metrics detailed above.

1) Graph reduction: Tables I, II, III, and IV summarize the graph reduction results. These results are generated on 2 and 3 hop subgraphs for the movie and book domain. Our approach retains the `Top-`K domain-specific categories in the subgraph. If there are more than one category with the same domain-specific score as the K^{th} rank category, all those categories are included in the subgraph. Hence in Table I for *PSL-DSHG*$_2$*(*4500*)* where K is $4,500$, the movie subgraph extracted contains $4,782$ categories.

Table I shows the number of categories and paths reached in *EXP-DSHG*$_2$ and three *PSL-DSHG*$_2$s extracted by selecting different `top-`K for the movie domain. The reduction percentage from *EXP-DSHG*$_2$ to *PSL-DSHG*$_2$ is presented in parentheses beside the raw number of categories and paths. Tables II, III, and IV portray the results for movie 3-hop, book 2-hop, and book 3-hop graphs.

The reduction of number of categories by 25% has led to a reduction of the number of paths by 43% for movie 2-hop subgraphs (the last row in Table I) and number of paths reduced by 52% for category reduction of 27% for movie 3-hop subgraphs (the last row in Table II).

The book 2-hop subgraph was only able to reduce the number of paths by 18% with the reduction of categories by 27% as shown in Table III. This is comparatively lesser than the movie domain and it is due to the cardinality

	Categories	Paths
EXP-DSHG$_2$	8603	2.2M
PSL-DSHG$_2$(5000)	5155(40%)	1.4M(36%)
PSL-DSHG$_2$(5500)	5847(32%)	1.6M(27%)
PSL-DSHG$_2$(6000)	6297(27%)	1.8M(18%)

Table III: Graph reduction statistics of *PSL-DSHG$_2(K)$* in comparison to *EXP-DSHG$_2$* in the book domain; M denotes millions.

	Categories	Paths
EXP-DSHG$_3$	18680	22M
PSL-DSHG$_3$(6500)	6868(63%)	9M(73%)
PSL-DSHG$_3$(7500)	7504(60%)	12M(45%)
PSL-DSHG$_3$(8500)	8916(52%)	14M(35%)

Table IV: Graph reduction statistics of *PSL-DSHG$_3(K)$* in comparison to *EXP-DSHG$_3$* in the book domain; M denotes millions.

of links between the domain entities and categories for each domain on Wikipedia. Specifically, there are fewer links between concepts in the book domain in comparison to the movie domain. On an average, a movie entity is connected to 2 categories whereas a book entity is connected to 1.1 categories on Wikipedia. However, as we increase the number of hops for the book domain, domain-specific subgraphs were able to increase the reduction. The book 3-hop subgraph was able to reduce the number of paths by 35% by reducing the number of categories by 52% as shown in Table IV. An important aspect to note in both the domains is the significant increase in the number of reachable paths when the hop size changes from 2 to 3.

2) Accuracy:

precision@m: The graph reduction percentages reported above are only meaningful if the domain-specific subgraphs do not compromise the accuracy of the generated recommendations in comparison to the n-hop expansion subgraphs. To assess this, we calculate the precision for `top-`m for each user and then average it over all the users. Fig. 6 compares the `precision@m` for all the users from the MovieLens dataset obtained for both *EXP-DSHG$_2$* and *PSL-DSHG$_2$* for same K values used for graph reduction. Movie 2-hop subgraph extracted with `top-`4500 categories shows the best performance. It increased the precision by 1.5% for the `top-5` and `top-10` recommendations over the baseline. This indicates that merely selecting n-hops to create a subgraph can negatively impact the results. In other words, selecting the domain-specific categories and paths can lead to better performance of the overall application.

Fig. 7 shows the results for the 3-hop subgraphs. It shows the best performance for the subgraph extracted with `top-`9000 categories and it also outperforms the baseline.

The recommendation results obtained for the 2-hop subgraphs on book domain are shown in the Fig. 8. The subgraph extracted with the `top-`5500 categories shows the similar performance as the baseline subgraph. With respect to the book 3-hop subgraphs shown in Fig. 9, the subgraph extracted with `top-`7500 categories shows the best performance with an increase in precision by 3.3%

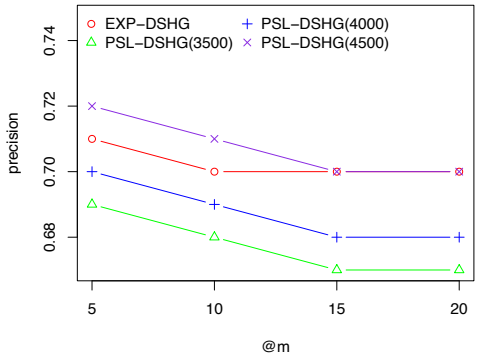


Figure 6: Precision for *PSL-DSHG$_2(K)$* and *EXP-DSHG$_2$* for movie domain; K denotes `top-`K categories.

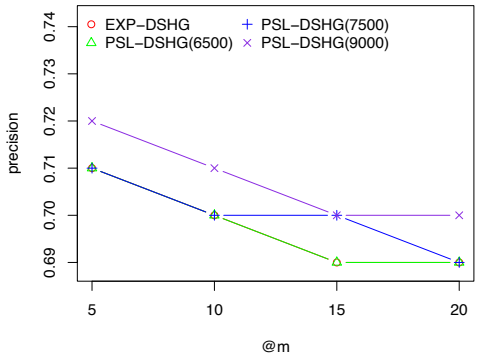


Figure 7: Precision on *PSL-DSHG$_3(K)$* and *EXP-DSHG$_3$* for movie domain.

for `top-`1 recommendations. To conclude, all the domain-specific subgraphs extracted by our approach perform better or equally well when compared to the subgraph created by expanding set of domain entities by merely navigating n-hops.

ratingdev: We calculate the $ratingdev$ for different ranks and average it over all the users. We pick the best performing *PSL-DSHG$_n$* in terms of `precision@m` to present the results in comparison to the *EXP-DSHG$_n$*. Tables V and VI show the deviation from the average ratings for the movie and book domains calculated with 2-hop and 3-hop subgraphs. For both the domains, deviation from the average rating performs equally well or outperforms the

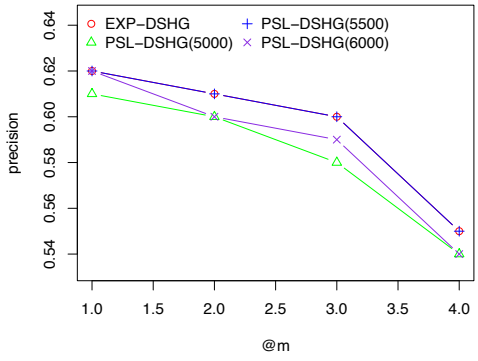


Figure 8: Precision for *PSL-DSHG$_2(K)$* and *EXP-DSHG$_2$* for book domain.

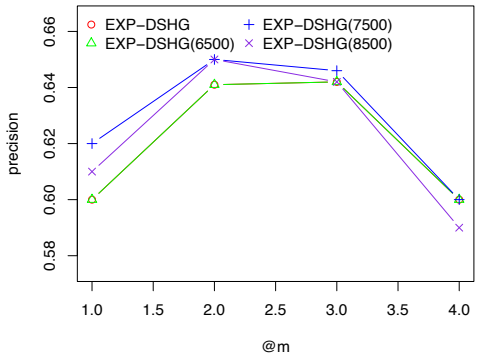


Figure 9: Precision for *PSL-DSHG*$_3(K)$ and *EXP-DSHG*$_3$ for book domain.

	2-hop		3-hop	
	EXP-DSHG$_2$	***PSL-DSHG***$_2$	***EXP-DSHG***$_3$	***PSL-DSHG***$_3$
5	0.87	0.876	0.87	0.87
10	0.852	0.857	0.851	0.852
15	0.842	0.849	0.842	0.844
20	0.837	0.842	0.836	0.84

Table V: Deviation from average rating for Movie for *EXP-DSHG* and best performing *PSL-DSHG*; *PSL-DSHG*$_2$*(*4500*)* and *PSL-DSHG*$_3$*(*9000*)*

results obtained with *EXP-DSHG*$_n$, except for top 3 results for the book 3-hop graph. Hence, we can conclude that our approach does not compromise the quality of recommendations by replacing highly rated recommendations. In most cases, our approach improves the recommendations by replacing low rated recommendations with better rated ones.

To summarize, the best performing movie-specific *PSL-DSHG*$_2$ outperforms the *EXP-DSHG*$_2$, with a 43% reduction in the number of paths, and the *PSL-DSHG*$_3$ outperforms the *EXP-DSHG*$_3$, with a 52% reduction in number of paths. The best performing book-specific *PSL-DSHG*$_2$ performs equally well with a 27% reduction in the number of paths and *PSL-DSHG*$_3$ outperforms with a 45% reduction in the number of paths with respect to the corresponding *EXP-DSHG*$_n$.

D. Evaluation Results on PSL-DSHG$_n$ *with the SUP-DSHG*$_n$

In this section, we compare the reduction and `precision@m` results for subgraphs generated by our approach and a supervised approach.

1) Graph Reduction: In our approach, *PSL-DSHG* is generated by identifying the domain-specific categories from a n-hop expansion subgraph from the domain entities. While *SUP-DSHG* follows a similar procedure, its expansion graph is generated by navigating n-hops starting from a category representing the domain (e.g., category Film) rather than the domain entities. In order to compare this approach to ours, the graph expanded has to comprise of all the domain entities for recommendation. Therefore, we set the n to 7 and 8 for movie and book domains where the expanded graph contains all the domain entities in our evaluation datasets. This resulted in $77,033$ categories for movie expansion subgraph and $45,784$ for book expansion for subgraph. The graph reduction statistics for *PSL-DSHG* and *SUP-DSHG* is presented in Table VII. For 2-hop subgraphs in both movie and book domains, we pick the best performing *PSL-DSHG* to report the results. As shown in the Table VII, movie path reductions are significantly higher in the *PSL-DSHG* in comparison to *SUP-DSHG*. But, book path reductions are higher in the *SUP-DSHG*.

	2-hop		3-hop	
	EXP-DSHG$_2$	***PSL-DSHG***$_2$	***EXP-DSHG***$_3$	***PSL-DSHG***$_3$
1	0.619	0.623	0.613	0.632
2	0.617	0.617	0.622	0.629
3	0.610	0.617	0.632	0.625
4	0.613	0.613	0.627	0.627

Table VI: Deviation from average rating for Book for *EXP-DSHG* and best performing *PSL-DSHG*; *PSL-DSHG*$_2$*(*5500*)* and *PSL-DSHG*$_3$*(*7500*)*

	PSL approach		**Supervised approach**	
	Categories	Paths	Categories	Paths
		Movie		
EXP-DSHG$_n$	6413	18M	77033	17M
**x-DSHG*	4782(25%)	10.2M(43%)	10576(86%)	16M(6%)
		Book		
EXP-DSHG$_n$	8603	2.2M	45784	2.0M
**x-DSHG*	5847(31%)	1.6M(27%)	8521(81%)	1.0M(50%)

Table VII: Graph reduction statistics for *PSL-DSHG*$_2$ and *SUP-DSHG* with expansion graphs for movie and book domain; M denotes millions; *x refers either to PSL or SUP (depends on the column title).

2) Accuracy - precision@m: We compare the `precision@m` results of *PSL-DSHG* in comparison to the *SUP-DSHG* and the expansion graph of *SUP-DSHG* created by starting at the domain category (*EXP-DSHG*). The results are portrayed in Fig. 10 where *PSL-DSHG*$_2$ performs the best with an improvement of 3% for both `top-5` and `top-15` movie recommendations in comparison to *SUP-DSHG*. The performance of the recommendation system that utilize *SUP-DSHG* not only deteriorates in comparison to *PSL-DSHG* but also in comparison to its own expansion subgraph. This is also corroborated with the results shown in Fig.11 for the book domain. *PSL-DSHG*$_2$ performs the best with an improvement of 17% and 19% for `top-3` and `top-4` in comparison to the *SUP-DSHG*. *SUP-DSHG* is a good example that emphasizes the importance of the evaluation metrics selected and the requirements stated for a good graph reduction problem in this work. To reiterate, while reducing the graph to a domain-specific graph is the primary goal of the work, it is also important not to compromise the performance of the application that utilizes the reduced graph (in our case, the recommendation system). *SUP-DSHG* performs better in reducing the graph for the book domain, however, the approach fails to provide equivalent or better recommendation performance in comparison to the baseline. On the other hand, our approach, shows significant graph reductions without compromising (and in most cases improving) the performance of the recommendation systems.

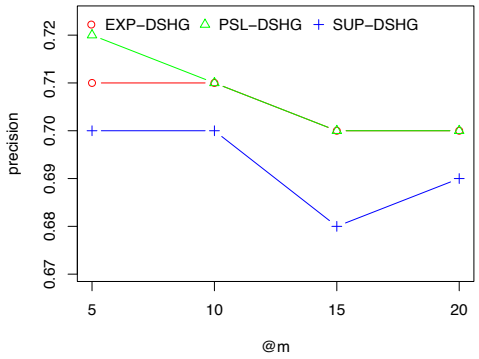


Figure 10: Precision for *PSL-DSHG*, *SUP-DSHG*, and *EXP-DSHG* for movie 2-hop.

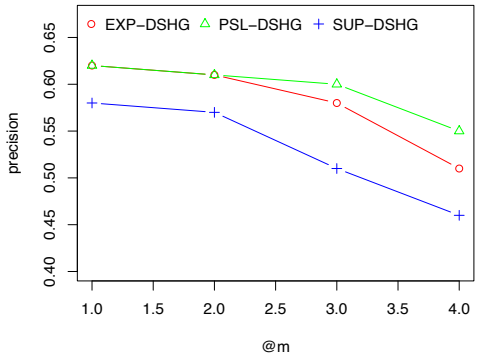


Figure 11: Precision for *PSL-DSHG*, *SUP-DSHG*, and *EXP-DSHG* for book 2-hop.

IV. Related Work

Many applications which leverage KGs typically rely on n-hop expansion to extract the subgraph. A number of recommendation algorithms such as [4]–[7] use n-hop expansion subgraphs to capture item relatedness between the items rated by the user and the items to be recommended. Named entity disambiguation [22], [23] is another type of application which leverages the subgraph to generate the context for each ambiguous entity mention. In general, entity relatedness techniques used by above applications leverage the features of the two items to be compared and the proposed approach restricts the features based on the domains so these techniques will be more meaningful within in-domain settings.

As an n-hop expansion subgraph still contains a significant portion of the KG, certain applications try to manually pick relevant relationships for their domain. Our previous work [8] exploits the semantics of the relationship to automatically rank relationships based on their domain-specificity to a given domain and then pick the subgraph with only domain-specific relationships. For instance, it would find that the relationship *actor* is more specific to the movie domain than the relationship *spouse*. Hence, the entities connected with the *spouse* relationship should be ignored when creating a movie domain-specific subgraph. This method is not applicable for hierarchical relationships as hierarchical relationships carry uniform semantics to connect both in-domain and out-of domain categories while non-hierarchical relationships carry diverse semantics.

As we have used the WCH, there are a number of studies [24], [25] dealing with extracting taxonomies from WCH. However, none of these studies have focused on addressing the domain-specific aspect of WCH. Given a set of keywords, Doozer [26] creates a domain model from the WCH. Doozer tries to capture the domain relevance by identifying semantically relevant entities (Wikipedia articles) and then traversing the WCH until finding a least common subsumer. However, in our work we have shown that even starting with only domain entities can lead to subgraphs with an irrelevant portion for the domain. In bootstraping domain ontologies, [20] have used a classifier to classify whether a given category is relevant to a given domain. We have compared our approach against [20] for identifying categories relevant to a domain and showed that our approach performs better than them in extracting domain-specific hierarchical subgraphs from Wikipedia. However, [20] also focused on extracting a richer domain model which is not the focus of our work.

V. Conclusion

We proposed an approach to extract a domain-specific subgraph from a generic hierarchical knowledge graph. We used Wikipedia category hierarchy as the test bed. Our approach uses type, lexical, and structural semantics of Wikipedia categories as evidences and aggregate them using PSL to determine the domain-specificity of a category. To demonstrate that our approach can work on multiple domains, we used datasets from two diverse domains, i.e., movie and book. We showed that our approach is able to reduce the size of the subgraph by 40% - 50% in terms of number of paths compared to the subgraph created by simple n-hop navigation-based approach. Furthermore, to show the effectiveness on applications using KGs, we evaluated the quality of the domain-specific subgraph extracted with a recommendation use case. Our evaluation showed that the recommendation results improved in majority of the scenarios which demonstrated that harnessing relevant, domain-specific information in KGs can in turn improve the performance of the applications in comparison to using the entire KGs. We also compared our approach with a state-of-the-art domain-specific subgraph extraction approach which uses a supervised learning technique and showed that our approach outperforms accuracy of recommendation results obtained via supervised technique with a significant graph reduction. We believe that this work has major impact in utilizing knowledge graphs for domain-specific applications, specially with the extensive growth in the creation of knowledge graphs. The reduction in size with no compromise in

the performance of applications will lead to fast and quick processing of KGs for corresponding applications.

In the future, we will explore how to extract the domain-specific subgraph for more fine-grained domains such as presidential campaigns.

ACKNOWLEDGMENT

This research was supported in part by NSF awards EAR-1520870 "Hazards SEES: Social and Physical Sensing Enabled Decision Support for Disaster Management and Response" and IIP-1542911 "Market Driven Innovations and Scaling up of Twitris- A System for Collective Social Intelligence". Any opinions, findings, and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer *et al.*, "Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic Web*, pp. 167–195, 2015.

[2] P. Kapanipathi, P. Jain, C. Venkataramani, and A. Sheth, "User interests identification on twitter using a hierarchical knowledge base," in *European Semantic Web Conference*. Springer, 2014, pp. 99–113.

[3] C. Welty, J. Murdock, A. Kalyanpur, and J. Fan, "A comparison of hard filters and soft evidence for answer typing in watson," *The Semantic Web–ISWC 2012*, pp. 243–256, 2012.

[4] V. C. Ostuni, T. Di Noia, E. Di Sciascio, and R. Mirizzi, "Top-n recommendations from implicit feedback leveraging linked open data," in *Proc. of the 7th ACM conference on Recommender systems*. ACM, 2013, pp. 85–92.

[5] C. Musto, P. Basile, P. Lops, M. De Gemmis, and G. Semeraro, "Linked open data-enabled strategies for top-n recommendations." in *CBRecSys@ RecSys*, 2014, pp. 49–56.

[6] T. Di Noia, R. Mirizzi, V. C. Ostuni, D. Romito, and M. Zanker, "Linked open data to support content-based recommender systems," in *In Proc, of the 8th International Conference on Semantic Systems*, 2012, pp. 1–8.

[7] G. Piao and J. G. Breslin, "Measuring semantic distance for linked open data-enabled recommender systems," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. ACM, 2016, pp. 315–320.

[8] S. Lalithsena, P. Kapanipathi, and A. Sheth, "Harnessing relationships for domain-specific subgraph extraction: A recommendation use case," in *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 2016, pp. 706–715.

[9] S. H. Bach, M. Broecheler, B. Huang, and L. Getoor, "Hinge-loss markov random fields and probabilistic soft logic," vol. arXiv:1505.04406 [cs.LG], 2015.

[10] P. Kouki, S. Fakhraei, J. Foulds, M. Eirinaki, and L. Getoor, "Hyper: A flexible and extensible probabilistic framework for hybrid recommender systems," in *Proc. of the 9th ACM Conference on Recommender Systems*, 2015.

[11] J. Pujara, H. Miao, L. Getoor, and W. Cohen, "Knowledge graph identification," in *Proceedings of the 12th International Semantic Web Conference*, 2013, pp. 542–557.

[12] A. Tonon, M. Catasta, G. Demartini, P. Cudré-Mauroux, and K. Aberer, "Trank: Ranking entity types using the web of data," in *International Semantic Web Conference*. Springer, 2013, pp. 640–656.

[13] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," *arXiv preprint*, 1995.

[14] N. Seco, T. Veale, and J. Hayes, "An intrinsic information content metric for semantic similarity in wordnet," in *Proceedings of the 16th European conference on AI*, 2004.

[15] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer, "Dbpedia spotlight: shedding light on the web of documents," in *Proceedings of the 7th international conference on semantic systems*. ACM, 2011, pp. 1–8.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[17] L. Han, A. Kashyap, T. Finin, J. Mayfield, and J. Weese, "Umbc ebiquity-core: Semantic textual similarity systems," in *Proc. of the Second Joint Conference on Lexical and Computational Semantics*, 2013, pp. 44–52.

[18] T. Kenter, A. Borisov, and M. de Rijke, "Siamese cbow: Optimizing word embeddings for sentence representations," *arXiv preprint arXiv:1606.04640*, 2016.

[19] J. P. Leal, "Using proximity to compute semantic relatedness in rdf graphs," *Computer Science and Information Systems*, vol. 10, no. 4, pp. 1727–1746, 2013.

[20] D. Mirylenka, A. Passerini, and L. Serafini, "Bootstrapping domain ontologies from wikipedia: A uniform approach." in *IJCAI*, 2015, pp. 1464–1470.

[21] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in *Recommender systems handbook*. Springer, 2011, pp. 257–297.

[22] R. Usbeck, A.-C. N. Ngomo, M. Röder, D. Gerber, S. A. Coelho, S. Auer, and A. Both, "Agdistis-graph-based disambiguation of named entities using linked data," in *International Semantic Web Conference*. Springer, 2014.

[23] I. Hulpuş, N. Prangnawarat, and C. Hayes, "Path-based semantic relatedness on linked data and its use to word and entity disambiguation," in *International Semantic Web Conference*. Springer, 2015, pp. 442–457.

[24] S. P. Ponzetto and M. Strube, "Taxonomy induction based on a collaboratively built knowledge repository," *Artificial Intelligence*, 2011.

[25] T. Flati, D. Vannella, T. Pasini, and R. Navigli, "Two is bigger (and better) than one: the wikipedia bitaxonomy project," in *ACL (1)*, 2014.

[26] C. Thomas, P. Mehra, R. Brooks, and A. Sheth, "Growing fields of interest-using an expand and reduce strategy for domain model extraction," in *Web Intelligence and Intelligent Agent Technology, 2008*, vol. 1. IEEE, 2008, pp. 496–502.