Characterizing Concepts in Taxonomy
For Entity Recommendations

A thesis submitted in partial fulfilment
of the requirements for the degree of
Master of Science

By

SIVA KUMAR CHEEKULA
B.Tech., Jawaharlal Nehru Technological University, India, 2009

2017
Wright State University
I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION
BY Siva Kumar Cheekula ENTITLED Characterizing Concepts in Taxonomy for Entity Re-
commendations BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF Master of Science.

Amit P. Sheth, Ph.D.
The thesis director

Mateen Rizki, Ph.D.
Chair, Department of Computer Science and Engineering

Committee on
Final Examination

Amit P. Sheth, Ph.D.

Krishnaprasad Thirunarayan, Ph.D.

Derek Doran, Ph.D.

Robert E. W. Fyffe, Ph.D.
Vice President of Research and
Dean of the Graduate School
ABSTRACT


Entity recommendation systems are enormously popular on the Web. These systems harness manually crafted taxonomies for improving recommendations. For example, Yahoo created the Open Directory Project for search and recommendation, and Amazon utilizes its own product taxonomy. While these taxonomies are of high quality, it is a labor and time-intensive process to manually create and keep them up to date. Instead, in this era of Web 2.0 where users collaboratively create large amounts of information on the Web, it is possible to utilize user-generated content to automatically generate good quality taxonomies. However, harnessing such taxonomies for entity recommendations has not been well explored. We exploit the Wikipedia category structure as a taxonomy and explore three prominent characteristics of concepts in the taxonomies for entity recommendations. The three characteristics we explore are: 1) Specificity, 2) Priority, and 3) Domain Relatedness of concepts in the taxonomy. We demonstrate the utility of specificity and priority of concepts in the taxonomies in achieving high quality recommendations by evaluating our recommender system on two diverse datasets.
Contents

1 Introduction 1

2 Related Work 8
  2.1 Linked Open Data in Recommendations 8
  2.2 Hierarchical Knowledge in Recommendations 10
    2.2.1 Manually Created Taxonomies 10
    2.2.2 Automatically Created Taxonomies 11
  2.3 Role of Taxonomies in other Use Cases 12
    2.3.1 Semantic Similarity 12
    2.3.2 Information Retrieval 13

3 Approach 14
  3.1 Taxonomy Creation from Wikipedia 14
  3.2 Entity Recommendation Algorithm 16
    3.2.1 Spreading Activation on Taxonomy 17
    3.2.2 Recommending Entities 19
  3.3 Characteristics of Concepts in Taxonomy 20
    3.3.1 Specificity 20
3.3.2 Domain Relatedness ........................................ 23
3.3.3 Priority ...................................................... 25

4 Evaluation ......................................................... 29
  4.1 Experimental Setup ......................................... 29
    4.1.1 Datasets ................................................. 29
    4.1.2 Evaluation Metrics ...................................... 30
  4.2 Discussion .................................................. 33
    4.2.1 Impact of Specificity on Recommendations .......... 34
    4.2.2 Impact of Priority on Recommendations .............. 35
    4.2.3 Impact of Domain Relatedness on Recommendations . 36
    4.2.4 Non-accuracy metrics .................................. 38
    4.2.5 Precision-Recall ....................................... 39

5 Conclusion and Future Work .................................. 43

Bibliography ......................................................... 45
List of Figures

1.1 Hierarchical relationships in taxonomies ............................................. 2
1.2 Approach for recommending entities from Wikipedia Category Graph ........ 5

3.1 Snippet of infobox properties from the Wikipedia article ‘Gravity (film)’. 15
3.2 Snippet of categories from the Wikipedia article ‘Gravity (film)’. ............... 15
3.3 Example of taxonomy structure. ......................................................... 17
3.4 Example of spreading activation on taxonomy. ....................................... 19
3.5 Specificity as a function of hyponyms in taxonomy ............................... 21
3.6 Inter-domain categories of film Chappie (2015 Film) ............................. 23
3.7 Categories for the film The Matrix. ..................................................... 25
3.8 Similarity of entity and category profiles. ............................................ 27

4.1 Specificity - Recall@K ................................................................. 40
4.2 Specificity - Precision vs Recall ....................................................... 41
4.3 Priority - Recall@K ................................................................. 41
4.4 Priority - Precision vs Recall ....................................................... 42
List of Tables

4.1 Data-set details after applying preprocessing steps. .......................... 30
4.2 Results of baseline spreading activation function ............................. 33
4.3 Recommendations performance on specificity measures ..................... 35
4.4 Rank of the film *The Matrix’s* categories as per the two priority measures. 37
4.5 Results for the domain relatedness characteristic. .......................... 37
4.6 Performance of recommendations for various measures of edge priority. 38
Acknowledgments

I would like to take this opportunity to thank everyone who has helped me throughout this wonderful graduate journey. First and foremost, I would like to thank my advisor Dr. Amit P. Sheth for the opportunity to work with him and his guidance. Working with him was a great experience and I am thankful to his efforts in imparting research and professional skills in his students.

I would like to thank my committee members Dr. Krishnaprasad Thirunarayan and Dr. Derek Doran for their time and feedback. I would specifically like to thank my mentor Dr. Pavan Kapanipathi for his continuous guidance, and insightful discussions; which helped in shaping this work into what it is today. A good friend and an influential person who has always motivated me to do the best of my ability. Thank you again for your kind support and guidance. I would also like to thank all my friends at Kno.e.sis, without whom this journey would not have been easier. Specifically, I would like to thank Sujan, Sarasi, Revathy, Vaikunth and Adarsh for the insightful discussions and the support.

This acknowledgement would be incomplete without thanking my family and friends who has always believed in me. Thank you my dear sister Sindhu, and brother-in-law Narsimha Reddy for the encouragement and support. To my friends and the extended family Kavya, and Surya, thanks for being there all the time to support me. Last but not the least, thanks to my parents Venkata Reddy, and Padmavathi who has always trusted and motivated me. Their immense support in difficult times has had greater role in achieving my goals. Thank you!

This thesis is based upon work supported by the National Science Foundation under Grant No. EAR 1520870 titled Hazards SEES: Social and Physical Sensing Enabled Decision Support for Disaster Management and Response. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.
Introduction

The World Wide Web as we know it today has evolved from a publish-only platform to one that enables users to interact with the content. There has been an exponential growth in the content on the Web along with advances in technology\(^1\). Entities such as countries, people, organizations, films, and books underlie this enormous Web of Data (WoD) and also are integral to the daily activities of users. 50% of search queries received by search engines contain references to entities [1]. Navigating users through numerous entities on the Web has been a growing challenge, and applications that personalize entity recommendations have seen growing popularity among users and application developers. In a recent development, search engines such as Google, and Bing have begun recommending entities relevant to the ones found in user search queries [2].

While search engines featuring recommended entities is a recent development, the practice of recommending entities such as books, and films has been in place since the early days of the Web and are commonly referred to as recommender systems [3, 4, 5]. Recommendation systems are becoming increasingly popular since their role in e-commerce applications has expanded to applications that host films, books, music, news articles, etc. Users on social media platforms such as Facebook, Linkedin, and Twitter get recommendations for people they may be interested in adding to their network. Entity recommendation has become such a prominent feature that, for

\(^1\)45.5 billion documents on Web of Data as of January 06 2017 (source: http://www.worldwidewebsize.com).
example, 75% of Netflix’s views comes from its recommendation engine\(^2\).

Taxonomies as background knowledge have played a significant role in recommendation systems [6, 7, 8, 9]. Yahoo created Open Directory Project [2] for search and recommendation, Pandora harnesses its own music genre taxonomy\(^3\), and Amazon utilizes a product taxonomy [6]. Hierarchical relationships in taxonomies, which describe \(is-a\) relationships between an entity and a class, between a class and super-class are utilized as background knowledge in entity recommendations. For example, as shown in Fig. 1.1, if a user has given ratings of 4 and 5 (on 1-5 scale) to the films \textit{Crank} and \textit{Chappie}, we may recommend the user the films from \textit{American Action Thriller Films} category or its parent category in the taxonomy i.e. \textit{American Action Films} which abstractly represent user interests. The user may find entities \textit{Blade} and \textit{Against The Dark} interesting as they belong to categories in the taxonomy she previously has shown interest in. Prior work utilizing taxonomies as background knowledge has shown that taxonomies can address


\(^3\)https://www.pandora.com/about/mgp
Despite the benefits of taxonomies, one is often faced with two major challenges when utilizing taxonomies for developing entity recommendation algorithms. The challenges are as follows:

1. The taxonomies that were explored have either been created manually or automatically from entity descriptions. Creating taxonomies manually is laborious, time consuming, and is an expensive effort. On the other hand, taxonomies created automatically from entity descriptions lack semantics and coverage as the descriptions provide only limited information about an entity.

2. Taxonomies are underexploited in recommender algorithms, i.e., hierarchical relationships from taxonomies have been only one piece among knowledge sources utilized in algorithms developed so far [12, 13], thus limiting the insights into the structural properties of taxonomies and their role in recommendation systems.

We believe that the potential of a taxonomic structure for entity recommendation needs to be further explored. Specifically, harnessing openly available taxonomies can alleviate the challenges in creating and updating taxonomies for recommendations. Therefore, in this work, we investigate three primary characteristics of concepts in an openly available taxonomy for entity recommendations. Three characteristics of taxonomies that we explore in this work are:

- **Specificity**: Specificity measures the notion of abstractness of a concept in the taxonomy. For example, from the taxonomy we can infer that the concept “Science Fiction Adventure Films” is more specific than “English-language Films”. While recommending movies that are instances of “Science Fiction Adventure Films”, “English-language Films” instances may add more noise as they may be too generic. Therefore, we hypothesize that the specificity
of concepts in the taxonomy can have an impact on entity recommendations. However, in the Wikipedia taxonomy the concepts *English-language Films* and *Science Fiction Films* are at the same hierarchical level, which is not intuitively satisfactory. Since we understand that a concept’s hierarchical level may not truly reflect their abstractness in crowd-sourced taxonomies, we investigate measures to gauge abstractness of concepts in crowd-sourced taxonomies.

- **Priority:** Collaborative taxonomies are mostly directed acyclic graphs (DAGs) rather than a tree. In such cases, each concept may have multiple concepts as its super-class in the taxonomy. For example, the movie *Titanic* is linked to a super-class (categories) such as *English-language films*, *1990’s romantic drama films*, and *Films about women*. Intuitively, each of these concepts has varied significance as a super-class to the movie *Titanic*. Therefore, we experiment with different measures to prioritize the super-class subclass relationships for recommendations.

- **Domain Relatedness:** In a taxonomy, concepts can be exclusive to one domain (in-domain) or can span multiple domains (inter-domain). For example, considering the Wikipedia category system, along with in-domain categories like *Action Films*, and *Thriller Films*, there exists inter-domain categories like *Fictional versions of real people* and *Brain-computer interaction in fiction* which could be used across domains such as Films, Television series or Books. Since an entity may have both types of concepts as parents in a taxonomy, we explore the contribution of the domain relatedness of a concept for recommending entities of the same domain.

Our investigation focuses on finding the appropriate measures to compute these three characteristics by harnessing the taxonomy structure and then compares their impact on recommending entities. Contribution of our work are as follows:
1. We explore a taxonomy derived from crowd-sourced knowledge base for entity recommendations.

2. Define three characteristics of concepts in taxonomy that can impact entity recommendations.

3. Experiment seven measures representing the three characteristics using an adaptation of spreading activation algorithm and discuss the impact of the characteristics on entity recommendation system.

We explore the role of the three characteristics in entity recommendations by utilizing them with a taxonomy-based entity recommendation algorithm, which we developed in our prior work [14] using a taxonomy derived from Wikipedia Category Graph. Our recommendation algorithm, as illustrated in Fig. 1.2, utilizes DBpedia\(^4\) category graph with an adaptation of spreading activation theory to recommend entities. The steps involved in the recommendation algorithm are briefly explained below.

**Figure 1.2: Approach for recommending entities from Wikipedia Category Graph**

- **Preprocessing DBpedia Category Graph:** In a one-time preprocessing step, the DBpedia category graph is transformed into a taxonomic structure with multiple inheritance.

\(^4\)DBpedia is an extract of semi-structured/structured data from Wikipedia
• **Determining Interest Categories:** Explicit ratings of entities, provided by a user as input to the algorithm, are spread to the corresponding categories of the DBpedia taxonomy by adapting a spreading activation algorithm. The categories in the taxonomy are scored reflecting the degree of a user’s interest.

• **Recommending Entities:** The score of the categories from the previous step is again spread to the entities not rated by the user. The highest scored entities are recommended to the user.

In order to explore the three characteristics’ impact on recommendation, we include each of their formulations in the spreading activation phase, a key step of our recommendation algorithm. By doing so, we expect the concepts in the taxonomy to be prioritized with respect to their characteristics and improve the quality of the recommended entities. As to the measures of the three characteristics in the taxonomy, we devise new measures as well as adopt existing measures of characteristics from the literature. We gauge the impact of each measure on recommendations by comparing recommendation results against a baseline recommendation algorithm which only includes the natural parameters of a spreading activation function.

We evaluate the performance of recommendation using the two standard datasets - MovieLens\(^5\) (movies) and LibraryThing\(^6\) (books) - well known in recommendation system evaluations [12, 15]. As our recommendation algorithm aims at determining the most relevant entities of interest to a user, we measure the quality of recommended entities as the capability of listing the most interesting entities in the top ranks (top K) of the predicted entities. To do so, we implement the evaluation approach introduced by Cremonesi et al. [15], which evaluates top K recommendation systems in terms of accuracy metrics. Additionally, we include recommendation system specific evaluation metrics such as *Diversity* and *Coverage*, which focus on non-accuracy objectives of recommendations.

---

\(^5\)http://grouplens.org/datasets/movielens/
\(^6\)https://www.librarything.com
Our evaluation on the two datasets shows that among the three characteristics the specificity and priority of the concepts in the taxonomy significantly impact the taxonomy-based entity recommendations while domain relatedness has little to no impact. We notice that the in-domain categories are automatically prioritized over inter-domain categories in our baseline recommendation algorithm, and thus adding the domain relatedness measure in the recommendation algorithm did not improve the quality of recommendations. On the other hand, a significant improvement in the diversity and coverage of recommended entities is observed when the specificity or priority of concepts is used with a taxonomy-based recommendation algorithm.

The rest of the thesis is organized as follows. In Chapter 2, we discuss the broader research work that utilized knowledge bases for recommendations and applications where taxonomies have been employed. In Chapter 3, we explain the approach taken to extract a taxonomy from the Wikipedia Category Graph and provide details of seven different measures as part of the investigation of characteristics. Chapter 4 presents the experimental setup and results of our experiments on the two datasets. Finally, Chapter 5 concludes our work and presents future experiments.
Related Work

Recommendation systems span diverse domains ranging from music [16, 17], books [18], and movies [19, 12] to items in e-commerce shops [6]. The range of techniques used in recommendation systems include (1) content-based, where prior ratings of a user and her demographic information are considered [19, 20], (2) collaborative, where the popularity and ratings of entities from similar or affiliated other users are considered [21], and (3) hybrid, which mix both content-based and collaborative ideas [22]. Infusion of knowledge about entities into these recommendation approaches has been gaining attention [12, 2, 6, 23]. In our work, we have focused on exploring hierarchical knowledge in recommendation systems. In this chapter, we present a survey of knowledge-based recommendation systems, and the role of hierarchical relationships in entity recommendations as well as other use cases.

2.1 Linked Open Data in Recommendations

Broadly, our work harnesses Linked Open Data for entity recommendations. This space has been well explored and techniques developed have utilized all types of relationships for recommendations [17, 19, 12, 24]. Passant [17] developed a graph-based measure called "linked data semantic distance (LDSD)" to recommend entities. This measure considers DBpedia as an undirected graph and does not differentiate any properties. Di Noia et al. [19] have also utilized
DBpedia in order to recommend movies based on the content of the user interested entities. Similar to Passant’s work, Di Noia et al. also utilizes all the properties of DBpedia and introduce a measure to determine the similarity between the entity to be recommended and the entities rated by users. Ostuni et al. [12] present a hybrid recommendation system named Sprank that utilizes both neighborhood users’ ratings and a user’s explicit ratings for recommendations. They create a bipartite graph between users and entities (rated and unrated) of length 4 from DBpedia. From the analysis of the paths, path-based features are extracted without regard to the type of the relationships in the path. Then they apply a learning to rank algorithm to recommend the most relevant items to the user. Piao et al. [25], like Passant, developed four additional semantic distance-based measures on knowledge graphs for recommendation algorithms. Piao’s measures of semantic distance between two entities considers the number of common resources connected by a property to them in knowledge bases. In their evaluation, three out of the four new variants of linked data semantic distance measures performed well compared to Passant’s LDSD measure. Sarasi et al. [24] have extracted a domain-specific sub-graph (DSG) from DBpedia using statistical and semantic-based metrics, to be used in recommendation algorithm. The recommendations are based on domain-specific relationships in the DSG. However, from these approaches, it is hard to analyze the value created by each property on Linked Open Data for recommendations. We, on the other hand, have tried to explore different features of hierarchical relationships on DBpedia, and define three characteristics which were previously not explored for recommendations.

Linked Open Data (LOD), as a knowledge-base, is not only used for entity recommendations, but also for tasks like location prediction of twitter users [26], implicit entity linking in tweets [27] and to create faceted entity summary [28]. On the other hand, LOD is also used to create a semantic infrastructure for real-time processing of microblog posts [29, 30]. The translation of unstructured text as Linked Open Data enabled querying and flexible analysis of microblog posts.


2.2 Hierarchical Knowledge in Recommendations

The integration of hierarchical knowledge in recommendation is becoming an emerging area of interest. They have also shown significant improvement in the performance of recommendation systems, specifically while handling the cold-start problem [31]. The knowledge bases considered are either manually created [6] or are automatically derived from the content descriptions of the items [31].

2.2.1 Manually Created Taxonomies

Kanagal et al. [22] have developed a taxonomy-aware latent factor model (TF) that combines taxonomies and latent factors using additive models. Along with estimating latent factors for users and entities, latent factors are also estimated for each internal node in a manually created taxonomy (the Yahoo! product taxonomy). Hierarchical levels in the taxonomy are leveraged to estimate user-entity preferences among nodes at the each level and to efficiently rank entities (among millions of them) by selecting top-k nodes at each level of the taxonomy beginning at the highest one. Menon et al. [32] present an algorithm that mitigates the data sparsity challenge by incorporating information from hierarchical structures in a matrix-factorization model, a well-known approach in collaborative-filtering recommendation systems. Predicting the probability of a user clicking an online advertisement (click-through rate) by posing web-pages and ads as users-item pairs, Menon’s algorithm borrows the information of sibling advertisements where historical data of a pair (web-page, ad) is sparse. Sibling advertisements or web-pages were obtained through shared parent categories (such as advertiser, campaign) predefined in a manually created hierarchy of web-pages and ad campaigns. By borrowing information from sibling nodes in the hierarchy, the highly prevalent data sparsity challenge arising from billions of web-pages and ad campaign pairs could be addressed. In another work, Ziegler et al. [6] proposed a taxonomy-driven product recommendation in which users’ interest (represented by numeric ratings) in entities is spread
across their hierarchical category path beginning at the leaf level. By doing so, Ziegler obtains a user profile consisting of categories of entities and a score representing users’ degree of interest. Similarly, a profile is created for each entity by spreading a constant value throughout the hierarchy. While user profiles are used to find similar users, entity profiles are used to calculate similar entities; both of which are combined while recommending entities. Unlike Ziegler’s approach which uses a domain-specific tree structured taxonomy, we use a multi-domain taxonomy which allows multiple inheritance. Hence, the characteristics that we investigate, such as domain relatedness and priority, are not relevant to concepts in Ziegler’s taxonomy-drive approach.

2.2.2 Automatically Created Taxonomies

Automatically creating taxonomies involves exploring the description of entities, content associated with entities from user-generated content such as product reviews, tweets, and blogs. For example, the tweet "Saving Private Ryan is the best World War II film" enables categorizing "Saving Private Ryan" under the category "World War II films". In this direction, Zhang et al. [31] have proposed a recommendation algorithm that automatically learns a taxonomy from textual descriptions of entities and their purchase data. A nonparametric generative model, which the authors refer to as hierarchical latent factor model (HF model), is implemented to create a hierarchical structure of categories and to tag entities to appropriate leaf level category. Experimental results show that the HF model achieves a better AUC score compared to state-of-the-art latent factor models. In addition, their evaluation against a human-induced taxonomy suggests that automatically created taxonomy achieves better AUC score. In a similar recommendation algorithm presented by Zhu et al. [10], user-generated content associated with entities has been utilized to automatically create topic hierarchies. An extended latent factor model that includes topic hierarchy information, called THRec, has been developed to generate entity recommendations. Two additional versions, THRec-tr and THRec-tp, are also proposed which exploit features such as topic popularity and topic relatedness derived from the topic hierarchy and user-generated content.
The majority of these algorithms report significant contribution of hierarchical knowledge in addressing the data sparsity and cold-start challenges. However, creating these hierarchical knowledge bases manually is tedious and time consuming, whereas automatic creation is completely data driven (based on item descriptions) and hence might lack coverage, i.e., the number of categories that could be assigned to an entity completely depends on the coverage of information present in the entity description. In our work, we instead use a hierarchical structure derived from Wikipedia. Wikipedia has domain experts as volunteers contributing and making sure the quality of the content is good. The sheer number of volunteers (24,457,390)\(^1\) makes it less tedious for creating the knowledge base and keeping it up to date.

### 2.3 Role of Taxonomies in other Use Cases

Prior to gaining popularity in recommendation systems, hierarchical knowledge-based solutions were widely explored in use cases such as measuring the semantic similarity of word pairs, and documents retrieval in Web search. In this section, we briefly discuss use cases that explored hierarchical knowledge since we adapt some of them in investigating the characteristics.

#### 2.3.1 Semantic Similarity

Hierarchical knowledge bases such as WordNet have played a significant role in measuring the similarity of word pairs as relationships between the corresponding categories obtained from their hierarchical relationships. For example, measuring similarity of pair *cars* and *bikes* may benefit by incorporating knowledge that they both are means of commuting. Resnik [33] has utilized the hierarchical grouping of words in WordNet taxonomy to measure the similarity of word pairs. The similarity of a word pair is quantified as the information content of the categories that subsume both the words in WordNet taxonomy. Resnik measures the information content of each category in the taxonomy by considering the frequency of words that belong to the category and its subcategories in

\(^1\)http://en.wikipedia.org/wiki/Wikipedia:Wikipedians
a text corpus. Unlike Resnik, Seco et al. [34] rely only on information in the hierarchical structure to measure the information content of categories and present an approach to measure the semantic similarity of word pairs. The proportion of nodes subsumed by a category to the total number of nodes in the hierarchy is considered to measure the information content of a category, called intrinsic information content (IIC). Experimental evaluation shows that the intrinsic information content measure, which relies only on the hierarchy to measure the similarity of word pairs, performs well compared to Resnik’s information content measure which relies on an external text corpus. In our work, as we investigate approaches to measure specificity of concepts in the crowd-sourced taxonomy, we explore utility of Resnik and Seco’s information content measures and present evaluation on recommendations performance.

2.3.2 Information Retrieval

Information retrieval systems for Web search are among the other major applications that harness hierarchical knowledge bases. Varelas et al. [35] utilizes hierarchical knowledge to augment words in search queries and documents with similar words to generate extended query and document vectors. This allows document matching algorithms go beyond syntactic-based matching of documents to search queries. On the other hand, hierarchical category structures were explored in order to narrow the wide range of document categories a search query could yield. Klein et al. [36] have developed a category ranking algorithm to narrow categories and display results from fewer non-redundant matching categories. Features such as the similarity of terms in the category and search query, related categories from the category hierarchy and documents associated with them are part of category ranking algorithm. By ranking and selecting few categories to retrieve documents from, Klein’s algorithm provides a cohesive and organized results for the user to traverse.
3

Approach

In this chapter, we first discuss our approach to transforming the Wikipedia Category Graph to a taxonomy, and the recommendation algorithm that harnesses it. Next, we continue the discussion of the three characteristics of concepts in taxonomies and techniques we have explored to measure them.

3.1 Taxonomy Creation from Wikipedia

Wikipedia, a collaborative encyclopedia, is a source of human readable and semi-structured knowledge. Curated by the users, it is a well-known openly available knowledge source leveraged by a variety of applications. A document or article on Wikipedia is dedicated to discuss a subject (e.g. Barack Obama). Besides an unstructured description of a subject, a Wikipedia article also has structured information such as an infobox and article categories. The infobox of a Wikipedia article contains attribute-value pairs that are specific to that subject (Fig. 3.1). On the other hand, categories on Wikipedia (Fig. 3.2) cluster relevant articles together. The massive Wikipedia article collection is organized hierarchically using the category graph which has hierarchical relationships between categories. The structured infobox, and category structure are transformed and made available as

---

1 https://en.wikipedia.org/wiki/Barack_Obama  
part of DBpedia [37] dataset on Linked Open Data [38]. This dataset has been utilized as background knowledge in tasks such as semantic similarity [39], and recommendations [16, 12, 17].

The disadvantage of Wikipedia (and hence DBpedia) is that the category structure is not organized as a formal ontology. This is because users are allowed to link any Wikipedia category to an entity (Wikipedia article) without regard to whether the categorical relationship is reasonable or not. By addressing this inconsistency, if a hierarchy could be extracted from DBpedia, it could be invaluable for entity recommendations. For example, if we can determine that a user is interested in
category:films by james cameron⁴ and category:horror films⁵, it would be reasonable to recommend entities that are children of both categories (namely, horror movies by James Cameron to the user).

We followed our previous approach [40] to create the taxonomy from Wikipedia Category Graph. This process can be summarized as follows:

1. All Wikipedia administrative categories⁶ are removed based on a keyword set similar to [41].

2. The category “Main Topic Classification” is selected as the root of the taxonomy as it subsumes 98% of the categories.

3. A hierarchy level is assigned to every category reflecting its shortest path (distance) to the root node. Our intuition is that, the closer the category is to the root category, the more abstract it is.

4. All the back edges that do not comply with the taxonomy structure (i.e., edges pointing to an abstract level node from a lower/more specific level) are removed.

The derived taxonomy has 0.8M vertices, 1.2M edges, and 16 hierarchical levels. The cardinality of relationships in the taxonomy is many-to-many and hence our taxonomy is a Directed Acyclic Graph (DAG).

### 3.2 Entity Recommendation Algorithm

Our algorithm for recommending entities involves two steps, which are discussed in detail in Sections 3.2.1 and 3.2.2. First, while making recommendations, our algorithm assigns weights to categories in the taxonomy by utilizing the explicit feedback such as ratings, and likes given to entities by the user. In doing so, the mechanism adopts spreading activation theory, which is
widely popular for its applicability in semantic networks. We implement the spreading activation functions to propagate numeric ratings of entities throughout the taxonomy and assign weights to categories. In the next step, we derive a ranked list of entities to be recommended to a user utilizing the weights assigned to the respective categories of each entity. To facilitate this process, all the entities (Wikipedia articles) have been added to the taxonomy at the leaf level. As shown in Fig. 3.3, each entity at the leaf level may have edges with one or more categories in the taxonomy.

3.2.1 Spreading Activation on Taxonomy

Built on the premise that human memory is represented as a semantic network of concepts [42], spreading activation theory is implemented to perform computations on associative and semantic networks. For example, in a network formed by scientific articles with citations as edges, the spreading activation algorithm was applied to retrieve relevant articles given a set of seed articles as input [43, 44]. Typically, in a spreading activation algorithm a computation method, known as activation function, is applied to each node of the network. Parameters of the activation function could be values on the incoming and/or the outgoing edges of a node along with other
application-specific variables. Controlled by the activation function, the seed values such as ratings
given by users to entities are spread across the network to identify nodes of interest. The activation
function usually includes a decay factor that decays initial activation value as it spreads further
from the seed nodes. The output of the activation function, known as activation value, is used as a
reference to retrieve/rank nodes in the network for application specific use. We adopt this approach
of spreading activation on the taxonomy extracted from Wikipedia Category Graph to identify
categories of users’ interest. Formally, a simple activation function is as follows:

\[ A_i = \sum_{j \in C(i)} A_j \times W_{ij} \times D \]  

(3.1)

where \(i\) is the node to be activated, \(A_i\) is the activation value of node \(i\), \(C(i)\) is the set of child
nodes of \(i\), \(W_{ij}\) is the weight of the edge connecting \(i\) and \(j\), and \(D\) is a decay factor.

In our algorithm, we apply the activation function to each category of the taxonomy in a
bottom-up fashion. As shown in Fig. 3.4, ratings given to entities by the user are spread to each
category of the taxonomy using an activation function similar to Eq. 3.1. This activation process
begins at the bottom level of the taxonomy and moves up the hierarchy level by level. At the end of
spreading activation process, the activation values of categories reflect the degree of users’ interest
in them.

As categories of an entity could either be generic (e.g. film) or specific (e.g science-fiction film),
entities at the leaf level of the taxonomy may have edges with categories at different hierarchical
levels. Due to this, it is possible for entities to activate categories redundantly through direct and
hierarchical relationships. For example, entity \textit{American Ninja} is associated with abstract category
\textit{American Films} and specific category \textit{American Action Films} which are hierarchically connected.

In the spreading activation phase, abstract category \textit{American Films} receives activation directly
from entity *American Ninja* as well as from sub-category *American Action Films* through their hierarchical relationship. We term these multiple activations from the same entity as redundant activations. To avoid this redundancy, we ignore input from entities which have already activated one of the category’s hyponyms while applying activation function on a category.

### 3.2.2 Recommending Entities

In the previous phase, user ratings for entities that live at the leaf level of the taxonomy were spread across the taxonomy to activate categories. In order to recommend entities to the user, we sum the activation values of all categories of an entity and assign the resulting score to it. For example, in Fig. 3.4, score of entity *Against The Dark*, which is unrated by user, will be 7.32. Ranked based on their score, such previously unrated entities form the list of entities to be recommended to the user. Formally, the entity scoring method is given as follows.
\[ A_i = \sum_{j \in N(i)} A_j \]  
(3.2)

where \( N(i) \) is the set of categories associated with entity \( i \) in the taxonomy.

### 3.3 Characteristics of Concepts in Taxonomy

The Wikipedia category structure has two types of edges: category-category and category-article. In this section, we present various approaches that leverage these two edge types to measure the Specificity, Priority, and Domain Relatedness of categories in the taxonomy.

#### 3.3.1 Specificity

Specificity measures the granularity of the information conveyed by a category. This characteristic of a category can be pivotal to narrow the search of the entities to be recommended. For example, recommending entities tagged with generic categories like *Film* yields many results, while specific categories like *Science Fiction Adventure Films* yield fewer results. Hence, we hypothesize that incorporating specificity of categories as a parameter in the spreading activation function may help improve the recommendation task by penalizing the activation value with respect to the abstractness of a category. The formulation of the new spreading activation function that considers specificity of a category is as follows:

\[ A_i = S(i) \times \sum_{j \in C(i)} A_j \times W_{ij} \times D \]  
(3.3)

where \( S(i) \) is the function that defines the specificity of category \( i \) in the taxonomy. \( S(i) \) is replaced by one of the following three measures of specificity.
Intrinsic Information Content (IIC): Seco et al. [34] have presented a specificity measure for nodes in the taxonomy based on the number of nodes subsumed by a node. Referred to as Intrinsic Information Content (IIC), specificity is measured as a ratio of the number of nodes subsumed by a category to the total number of nodes in the taxonomy. The intuition being that the higher the ratio of subsumed nodes to the total number of nodes, the more generic a node is. For example, in Fig. 3.5, node A subsumes all the nodes and is the most generic node in the taxonomy, compared to node D which subsumes only two nodes. The computed specificity of nodes A and D according to Seco’s IIC measure is 0 and 0.47 respectively. Seco’s formulation of the specificity measure of a node in the taxonomy is given as follows:

$$S(i) = 1 - \frac{\log(|M_i| + 1)}{\log N} \quad (3.4)$$

where $M_i$ is the set of nodes subsumed by category $i$ in the taxonomy, and $N$ is the number of categories in the taxonomy.

Information Content (IC): In an information theoretic formulation, Resnik [33] has developed a specificity measure based on the information content of a node in the taxonomy. According to
the study, the specificity of a node increases as its information content increases. The proposed measure of information content uses an external corpus of entities of categories in the taxonomy. The information content of a node is defined as the probability of encountering entities of the node given a set of entities. For example, the information content of the category "American action films" would be the probability of finding an entity of that category among all the films. Here, entities of a node include entities directly associated with it as well as entities associated with the nodes subsumed by it. The formalization of the probability based information content is as follows:

\[ S(i) = -\log p(i) \]  

(3.5)

where \( p(i) = \frac{Freq_i}{N} \) and \( Freq_i \) is the number of unique entities of category \( i \) and of its hyponyms (e.g., in Fig. 3.3, the frequency of the node American Action Films is 4); \( N \) is the total number of entities in the corpus, which is the set of entities of target domain such as films.

**Out Degree:** Out degree of a node in the taxonomy, i.e. the number of outgoing edges of a node, may denote the specificity of it. For example, a generic category like English Language Films has 30 sub-categories and 51K entities on Wikipedia, whereas a more specific category like American Science Fiction Action Films has only 9 sub categories and 338 entities. Hence, based on the intuition that specificity increases as out degree decreases, we implement a measure of specificity using the out degree of the nodes in our crowd-sourced taxonomy. Since categories in our taxonomy have two edge types, i.e., category-subcategory and category-entity, they have two types of out degrees. Hence, we implement a variant of the spreading activation function that measures specificity based on the out degree from the two edge types. The formulation of the new spreading activation function is as follows:
Figure 3.6: Inter-domain categories of film Chappie (2015 Film)

\[ A_i = \left( \frac{1}{|C(i)|} \sum_{j \in C(i)} A_j \times W_{ij} \times D + \frac{1}{|E(i)|} \sum_{j \in E(i)} A_j \times W_{ij} \times D \right) \quad (3.6) \]

where \( C(i) \) and \( E(i) \) are, respectively, a set of subcategories and entities of node \( i \). Note that this activation function is new and is not a replacement for specificity parameter \( S(i) \) in activation function in Eq. 3.3.

### 3.3.2 Domain Relatedness

In a category structure like Wikipedia, which hosts entities from multiple domains, there exist categories that span multiple domains. For example, as shown in Fig. 3.6, the film Chappie has categories Fictional Versions of Real People and Brain-computer interfacing in fiction, which are also used with entities of other domains such as Books and Television series. We hypothesize that incorporating a category’s relevance to the target domain of recommendation may have an impact on the quality of the recommended entities. For example, we may achieve better quality in the entities recommended in the film domain by ranking strict film domain categories higher than categories that span multiple domains because the film domain categories may have higher influence on the recommended entities. We have devised two measures in order to quantify a category’s relevance to the target domain of interest. The spreading activation function that includes domain relatedness as
A parameter is as follows:

\[ A_i = D(i) \times \sum_{j \in C(i)} A_j \times W_{ij} \times D \]  

(3.7)

where \( D(i) \) refers to the domain relatedness measure of category \( i \). \( D(i) \) is replaced by one of the two measures of domain relatedness described below.

**Entity Fraction:** This measure of domain relatedness considers the type of entities (\textit{rdf:type} property in DBpedia) subsumed by a category in the taxonomy. We quantify domain relatedness as the fraction of target domain entities among all the entities subsumed by a category in the taxonomy. For example, when recommending films, the category \textit{Brain-computer interfacing in fiction} has 8 entities of type \textit{Film} among 68 total entities subsumed by it (39 directly associated and 29 through hyponyms). The smaller fraction of Film domain entities may denote that category \textit{Brain-computer interfacing in fiction} may not only be related to the Film domain. Based on this intuition, fraction of entities subsumed by a category denote the domain relatedness of categories in the taxonomy. Formally, entity-based domain relatedness measure of category \( i \) is given as follows.

Let \( E \) denote a set of entities that belong to the target domain of recommendation.

\[ D(i) = \frac{|L_i^e|}{|L_i|} \]  

(3.8)

where \( e \in E, L_i \) and \( L_i^e \) denote the list of all entities, and target domain entities subsumed by category \( i \), respectively.

**Shortest Path:** In the Wikipedia Category Graph, it is possible that categories of different domains
are connected at further distances. The distance between categories of the same domain may be shorter compared to the distance between categories from two different domains. Based on this notion, we use the length of shortest path between a domain’s root category (e.g. film) and a category to quantify the domain relatedness of the category. The shorter the distance, the more relevant a category is to the domain. Hence, we calculate the inverse of the shortest path’s length as the domain relatedness measure, which is formulated below:

\[ D(i) = \frac{1}{L_{ri}} \]  

(3.9)

where \( L_{ri} \) is the length of shortest path between category \( i \) and root category \( r \). The root category \( r \) could be a category like film or books in the category graph that subsumes most of the respective domain’s categories.

### 3.3.3 Priority

Our transformed Wikipedia category taxonomy is a Directed Acyclic Graph (DAG). Thereby, in the taxonomy, categories and entities have multiple super categories. For example, as shown in Fig. 3.7, the film The Matrix has English Language Films, Science Fiction Action Films and Films
about Telepresence as its categories. Intuitively, these categories have varied levels of importance to the film *The Matrix*. Therefore, we hypothesize that incorporating the variation in importance of categories may contribute to searching categories of users’ interest. In this direction, we explore two measures that prioritize super categories of an entity in the taxonomy. The spreading activation function that incorporates the priority of an edge is as follows:

\[
A_i = \sum_{j \in C(i)} A_j \times D \times P_{ij}
\]  

(3.10)

where \(P_{ij}\) is the priority of the edge between category \(i\) and entity or sub-category \(j\).

The two measures we explore for priority are *Preferential Path Constraint*, and *Similarity* which are discussed below.

**Preferential Path Constraint (PPC):** Wikipedia categorization guidelines\(^7\) suggest that an article’s most significant categories be listed first in its categories section. This provides us with an order that reflects category importance to an entity or a category. Referred to as *Preferential Path Constraint*, this crowd-sourced ordering of significant categories has been previously exploited by Kapanipathi et al. [40] for user interest modeling. We explore the same approach to find its reliability for entity recommendations. The formulation of this priority measure is below.

\[
P_{ij} = \frac{1}{P_{ji}}
\]  

(3.11)

where \(P_{ji}\) is the priority of category \(i\) to subcategory or entity \(j\).

\(^7\)https://en.wikipedia.org/wiki/Wikipedia:Categorization
Similarity: Categories associated with entities can be used to measure the importance of entity-category and category-category relationships. We build category-based profiles for entities and categories, then measure the similarity of the profiles. The intuition being that a category may be important to an entity $e$ if the category subsumes entities similar to $e$. Similarly, a category that subsumes entities not similar to the entity $e$ may not be of much importance to $e$. For example, as shown in Fig. 3.7, the category *Science fiction action films* may be more important to the entity *The Matrix* than the category *English language films* as the former subsumes entities similar to *The Matrix* while the latter subsumes a more diverse set of entities.

When measuring the similarity of entity-category profiles, the entity profile consists of the categories associated with it, while the category profile consists the categories of entities associated with it, excluding the entity in question. For example, as shown in Fig. 3.7, the profile of entity *The Matrix* consists of all its categories, and the profile of *English language films* consists of categories associated with its entities, except *The Matrix*. We use the jaccard similarity index to measure the similarity of entity and category profiles. The formulation of the same is given below:

$$P_{ij} = \frac{|i_c \cap j_c|}{|i_c \cup j_c|}$$ (3.12)
where \( i \) is set of categories associated with entities of category \( i \); \( j \) is set of entity \( j \) categories.
4

Evaluation

This chapter describes the evaluation of the impact of the three characteristics, namely Specificity, Priority and Domain Relatedness, on entity recommendations. As our spreading activation based recommendation algorithm aims at determining the most relevant entities of interest to a user, we measure the quality of recommended entities as the capability of listing the most interesting entities in the top ranks of the predicted entities. In the following sections, we present the details of the datasets, evaluation approach, the metrics used in the evaluation, and finally discuss the impact of the characteristics on recommendation results.

4.1 Experimental Setup

4.1.1 Datasets

To evaluate the impact of the three characteristics on entity recommendations, we used the datasets Movielens \(^1\) and LibraryThing \(^2\), which are widely used in recommendation system evaluation. As our algorithm is based on DBpedia \([19, 12, 15]\), it is necessary that entities in the datasets are mapped to the appropriate entities in DBpedia. We utilized the mappings open sourced by Di Noia et al. \([19]\), and removed entities which have no mapping. Additionally, it is important to note that

---

\(^1\)https://movielens.org/

\(^2\)https://www.librarything.com/
sparse data (due to users who have rated very few entities) significantly impacts the recommendation performance in content-based approaches [45]. As ours is one such algorithm, similar to [19, 12], we eliminated users who have ratings for fewer than 20 entities. Table 4.1 shows the details of the datasets after applying the above preprocessing steps and eliminating 9% of users from Movielens, and 43% of users from LibraryThing datasets.

4.1.2 Evaluation Metrics

**Precision-Recall:** Cremonesi et al. [15] argued that the top K recommender systems which focus on listing relevant entities in the top ranks should be evaluated by means of accuracy metrics, not by the popular error metrics (such as RMSE) as they only act as a proxy in the evaluation of top K recommendation systems. Acknowledging the same, we also adapt the top K recommendation system evaluation.

After applying the preprocessing steps discussed in Section 4.1.1 on the original datasets, we randomly selected 1.4% of the entire dataset and used it as a probe set $T$. The remaining 98.6% of items are used as training set $M$. For each of the test item $i$ with 5-stars in probe set $T$, we performed the following.

- We randomly select 1,000 items unrated by the user with an assumption that those were uninteresting to the user.
- A ranked list of 1,001 items (i.e. 1,000 unrated plus the test item $i$) is formed using our
recommendation algorithm.

- Then, we evaluate by the selection of the top $K$ items. If the test item $i$ is in the top $K$ list, we have a hit, otherwise, a miss.

The probability of a hit increases as $K$ increases. We consider recall as the ratio of total hits and the size of the 5-star test items from probe set $T$. The set of 5-star test items in probe set $T$ are denoted by $N$. Formally, precision and recall are as follows:

$$\text{recall} @ K = \frac{\#\text{Hits}}{|N|} \quad \text{precision} @ K = \frac{\text{recall} @ K}{K} \quad (4.1)$$

where $\#\text{Hits}$ is the sum of hits within top $K$ items.

**Non-accuracy metrics:** Recommendation systems evaluation goes beyond the accuracy of recommended items; they are also evaluated on other important objectives of recommendations in the application. For example, while recommending entities, one might choose to recommend entities that are less known to their users, or choose to recommend a list of diverse entities in order to keep the user engaged with several types of content. Referred to as non-accuracy metrics, recommendation systems are evaluated on objectives like the **Diversity**, **Coverage**, and **Novelty** of the recommended entities. While optimizing our algorithm for such objectives is not in the scope of this work, we intend to understand the impact of the proposed characteristics on the **Diversity** and **Coverage** of recommended entities. Since **Novelty** is not a suitable metric for content-based recommendation approaches, we do not consider the metric further in our evaluation. Let $U$ and $E$ represent the set of users and entities, respectively, the definitions and formulation of the non-accuracy metrics are as follows.

- **Diversity:** Recommending a set of interesting yet similar entities to users may limit their ability to explore other type of entities. In order to understand the diversity of
recommendations, recommendation systems often employ a suitable diversity metric. By doing so, one might tune the diversity of recommendations with a possible trade-off in accuracy of recommended entities. To understand the impact of the three characteristics on diversity, we consider the categories of entities and calculate the dissimilarity of entities in the top K list of recommended entities. The formulation of the diversity metric is as follows.

\[ Div_{@k} = \frac{1}{|U|} Div_{u@k} \]  \hspace{1cm} (4.2)

\[ Div_{u@k} = \frac{1}{2} \sum_{i \in L_u^k} \sum_{j \in L_u^k} 1 - sim(i, j) \]  \hspace{1cm} (4.3)

where \( Div@k \) is the diversity of top K entities across all users, and \( L_u^k \) is the list of top K entities recommended to user \( u \).

- **Coverage:** Also known as catalog coverage, this metric is used to understand the percentage of entities covered at top K positions through recommendations across all users.

\[ Cov_{@k} = \frac{|L_u^k|}{|E|} \]  \hspace{1cm} (4.4)

where \( L_u^k \) is set of unique entities recommended to all users until rank \( k \).

- **Inter-list Dissimilarity (ILD):** Unlike diversity metric, which measures dissimilarity within a user’s recommendations, inter-list dissimilarity computes the dissimilarity of top K recommendations between users. We include this metric in our evaluation to gauge if our recommendation approach biases recommended entities towards a certain type of content. We expect the dissimilarity of recommended entities to be considerably higher when the recommendation algorithm (including the three characteristics) does not bias the recommendations towards certain categories in the taxonomy. The formulation of the metric
Table 4.2: Results of baseline spreading activation function

<table>
<thead>
<tr>
<th>Domain</th>
<th>Recall@k</th>
<th>ILD@k</th>
<th>Div@k</th>
<th>Cov@k</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>@10</td>
<td>@20</td>
<td>@10</td>
<td>@20</td>
</tr>
<tr>
<td>Movies</td>
<td>0.1150</td>
<td>0.1733</td>
<td>0.7892</td>
<td>0.7867</td>
</tr>
<tr>
<td>Books</td>
<td>0.1342</td>
<td>0.1753</td>
<td>0.8920</td>
<td>0.8956</td>
</tr>
</tbody>
</table>

is as follows.

\[
ILD@k = \frac{1}{|U|} \sum_{u \in U} ILD_u@k
\]  

(4.5)

\[
ILD_u@k = \frac{1}{|V|} \sum_{v \in L^k_u} 1 - sim(i^k_v, j^k_u)
\]  

(4.6)

where \( V \subset U \), and is a set of randomly selected 100 users; \( sim(i^k_v, j^k_u) \) is the jaccard similarity of top K recommended entities of users \( v \) and \( u \).

4.2 Discussion

In order to understand the impact of the three characteristics on recommendations, we first compute performance of a baseline spreading activation function (Eq. 3.1) which does not consider any of the characteristics of concepts in the taxonomy. We present the results in Table 4.2. As expected, the baseline recall and dissimilarity of recommended entities is considerably low at top 10 and 20. This is because, as observed, the baseline spreading activation function ranks generic categories higher than categories which reflect users’ specific interests. For example, English-language films and American science fiction novels are two of the highly ranked categories found in user profiles after the spreading activation phase. As generic categories are ranked high, entities at top 10 and 20...

33
lists are found to belong them; hence, the entities recommended are similar across many users.

4.2.1 Impact of Specificity on Recommendations

We investigated three approaches measuring the specificity of categories in the taxonomies. Among the three, Resnik’s information content (IC) based specificity measure uses an external corpus of entities, and Seco’s intrinsic information content (IIC) measure considers the hyponyms of a category in measuring its specificity. The out degree based measure, the third one which we introduced, uses the out degree of a category in the taxonomy. In Table 4.3, we present the entity recommendation results of the three specificity measures as evaluated on various metrics of top K recommendations.

Results demonstrate that considering the specificity of categories in the taxonomy improves recommendation performance significantly. Among the three measures of specificity, the category’s out degree based specificity measure displays the highest performance improvement over the baseline. While Seco’s specificity measure shows minor improvement in recall, Resnik’s specificity measure negatively impacted the baseline performance. These two measures did not have much impact on abstract categories, primarily because of their log-based nature, and hence the magnitude of the penalty applied by them on abstract categories is very low. As a result, though the activation values of abstract categories are penalized, abstract categories activation value has remained high compared to specific categories and influenced recommendations. On the other hand, penalty applied by out degree measure had high impact on the abstract categories since the penalty is inversely proportional to the out degree of categories. For example, category English-Language Films has 51k entities as it’s children in the taxonomy, and specificity of it as calculated by out degree measure (Eq. 3.6) is very low; subsequently it decreases the activation value of the category English-Language Films significantly. Therefore, out degree based specificity measure is found effective in identifying specific categories of interest to users’ and in recommending appropriate entities.
<table>
<thead>
<tr>
<th>Domain</th>
<th>Specificity</th>
<th>Recall@k</th>
<th>ILD@k</th>
<th>Div@k</th>
<th>Cov@k</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>@10</td>
<td>@20</td>
<td>@10</td>
<td>@20</td>
</tr>
<tr>
<td>Movies</td>
<td>Resnik et al</td>
<td>0.0951</td>
<td>0.1553</td>
<td>0.7811</td>
<td>0.7490</td>
</tr>
<tr>
<td></td>
<td>Seco et al</td>
<td>0.1306</td>
<td>0.1947</td>
<td>0.8027</td>
<td>0.8037</td>
</tr>
<tr>
<td></td>
<td>Out Degree</td>
<td>0.1995</td>
<td>0.2937</td>
<td>0.8205</td>
<td>0.8285</td>
</tr>
<tr>
<td>Books</td>
<td>Resnik et al</td>
<td>0.0411</td>
<td>0.0932</td>
<td>0.8627</td>
<td>0.8463</td>
</tr>
<tr>
<td></td>
<td>Seco et al</td>
<td>0.1370</td>
<td>0.1863</td>
<td>0.8882</td>
<td>0.8916</td>
</tr>
<tr>
<td></td>
<td>Out Degree</td>
<td>0.3616</td>
<td>0.4493</td>
<td>0.9725</td>
<td>0.9726</td>
</tr>
</tbody>
</table>

Table 4.3: Recommendations performance on specificity measures

Note that Resnik’s specificity measure at top 10 and 20 is less than that of the baseline recommendation algorithm. According to Resnik’s measure, specificity is based on a category’s information content, which only considers the entities subsumed by it. As a result, the top ranked (according to the activation value) categories after the spreading activation phase are specific only with respect to the number of entities they subsume, but our analysis found them to be abstract categories which have fewer entities as children. Therefore, the top recommendations were influenced by abstract categories (which have very few entities) that do not reflect user interests. Because of this, the recommended entities were found not relevant to user interests. Also, Resnik’s specificity measure has the highest diversity in recommended entities since the highly activated abstract categories which subsume only a few entities influence recommendations.

4.2.2 Impact of Priority on Recommendations

We hypothesize that prioritizing the categories of entities in taxonomies may prove beneficial to entity recommendations. We have experimented with two measures of priority: 1) preferential...
path constraint, which is based on the Wikipedia convention of listing categories according to their significance to an entity and 2) similarity, a measure we introduced that calculates the similarity of an entity and category profiles to gauge the significance of a category to an entity.

Reported in Table 4.6, the similarity measure achieves significantly better recall than the Wikipedia convention preferential path constraint while making recommendations. Since the similarity measure considers the profiles created from categories of entities and their super-categories, it is observed that the significance order derived by it is more semantically relevant than that is assigned by volunteers of Wikipedia. For example, in Table 4.4 we compare a few top categories derived using both the priority measures for the film *The Matrix*. It shows that among the top categories listed by both priority measures, the similarity-based prioritization of categories appears more relevant to recommendations than the list of categories prioritized by the preferential path constraint. Therefore, the similarity measure is able to improve the quality of personalized entity recommendations.

4.2.3 Impact of Domain Relatedness on Recommendations

The domain relatedness characteristic quantifies the degree of a category’s relevance to the target domain of recommendation. We hypothesize that, among all categories of an entity, placing emphasis on categories that are most relevant to the domain may be valuable to recommendations. For this purpose, we investigated two measures, namely Entity Fraction and Shortest Path. We applied these two measures on the baseline spreading activation function (Eq. 3.7).

Presented in Table 4.5, the two domain relatedness measures had no significant impact on the recommended entities, and thus recall at top 10 and 20 is approximately similar to the baseline performance. This is because the baseline spreading activation function automatically prioritizes in-domain categories over inter-domain categories. Hence, both the measures we have developed do not have impact on the baseline approach.
<table>
<thead>
<tr>
<th>Rank</th>
<th>Category - Similarity</th>
<th>Category - PPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gun Fu Films</td>
<td>1999 Films</td>
</tr>
<tr>
<td>2</td>
<td>Films about telepresence</td>
<td>English Language Films</td>
</tr>
<tr>
<td>3</td>
<td>Martial arts science fiction films</td>
<td>1990s science fiction films</td>
</tr>
<tr>
<td>4</td>
<td>Films shot in Sydney</td>
<td>American action thriller films</td>
</tr>
<tr>
<td>5</td>
<td>Cyberpunk films</td>
<td>American science fiction action films</td>
</tr>
<tr>
<td>6</td>
<td>Films directed by the Wachowskis</td>
<td>Cyberpunk films</td>
</tr>
<tr>
<td>7</td>
<td>Best film empire award winners</td>
<td>Dystopian films</td>
</tr>
<tr>
<td>8</td>
<td>Silver pictures films</td>
<td>Films about telepresence</td>
</tr>
<tr>
<td>9</td>
<td>American science fiction action films</td>
<td>Films directed by the Wachowskis</td>
</tr>
</tbody>
</table>

Table 4.4: Rank of the film *The Matrix*’s categories as per the two priority measures.

<table>
<thead>
<tr>
<th>Domain</th>
<th>DR</th>
<th>Recall@k @10</th>
<th>Recall@k @20</th>
<th>ILD@k @10</th>
<th>ILD@k @20</th>
<th>Div@k @10</th>
<th>Div@k @20</th>
<th>Cov@k @10</th>
<th>Cov@k @20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movies</td>
<td>Entity Fraction</td>
<td>0.122</td>
<td>0.183</td>
<td>0.793</td>
<td>0.795</td>
<td>0.854</td>
<td>0.880</td>
<td>0.079</td>
<td>0.126</td>
</tr>
<tr>
<td></td>
<td>Shortest Path</td>
<td>0.135</td>
<td>0.196</td>
<td>0.779</td>
<td>0.785</td>
<td>0.856</td>
<td>0.882</td>
<td>0.080</td>
<td>0.123</td>
</tr>
<tr>
<td>Books</td>
<td>Entity Fraction</td>
<td>0.137</td>
<td>0.197</td>
<td>0.888</td>
<td>0.892</td>
<td>0.838</td>
<td>0.867</td>
<td>0.061</td>
<td>0.093</td>
</tr>
<tr>
<td></td>
<td>Shortest Path</td>
<td>0.134</td>
<td>0.181</td>
<td>0.860</td>
<td>0.856</td>
<td>0.834</td>
<td>0.863</td>
<td>0.054</td>
<td>0.085</td>
</tr>
</tbody>
</table>

Table 4.5: Results for the domain relatedness characteristic.
<table>
<thead>
<tr>
<th>Domain</th>
<th>Priority</th>
<th>Recall@k</th>
<th>ILD@k</th>
<th>Div@k</th>
<th>Cov@k</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>@10</td>
<td>@20</td>
<td>@10</td>
<td>@20</td>
</tr>
<tr>
<td>Movies</td>
<td>PPC</td>
<td>0.0233</td>
<td>0.0447</td>
<td>0.7621</td>
<td>0.7433</td>
</tr>
<tr>
<td></td>
<td>Similarity</td>
<td>0.1782</td>
<td>0.2709</td>
<td>0.8411</td>
<td>0.8366</td>
</tr>
<tr>
<td>Books</td>
<td>PPC</td>
<td>0.0795</td>
<td>0.1096</td>
<td>0.8922</td>
<td>0.8840</td>
</tr>
<tr>
<td></td>
<td>Similarity</td>
<td>0.3178</td>
<td>0.4356</td>
<td>0.9686</td>
<td>0.9693</td>
</tr>
</tbody>
</table>

Table 4.6: Performance of recommendations for various measures of edge priority.

### 4.2.4 Non-accuracy metrics

In addition to the recommendation system’s performance evaluation metrics, we intend to understand the role of the three characteristics on two other important objectives Diversity and Coverage. While using the diversity measure we gauge the diversity of entities recommended to a user, and using the coverage metric we determine the overall coverage of entity catalog (e.g. all films on Netflix) through recommendations made to all users.

We present the evaluation results of the diversity and coverage at top 10 and 20 positions in Table 4.3 - 4.6. Results demonstrate the consistent improvement in diversity and coverage of the recommendations along with the improvement in recall at top K positions. Among the different measures explored for Specificity and Priority, out degree and similarity measures show significant improvement over the baseline (Table 4.2) diversity and coverage of the recommendations. An exception is that, in case of diversity, Resnik’s specificity measure has the highest diversity of recommended entities, though it’s recall is lower than the baseline. This is because the recommendations from Resnik’s measure of specificity are influenced by categories which are abstract and subsume only a few entities. Therefore, the top K recommendations show higher
diversity. The consistent influence of these abstract categories on recommendations is also seen in the diminished $ILD@k$ of Resnik’s specificity measure over the baseline. Since the recommended entities are always influenced by abstract categories, entities were quite similar across users’ recommendations. While improvement in diversity from Resnik’s specificity measure is not considerable, specificity based on the out degree of categories could be a valid measure as it improves baseline diversity significantly.

Prioritizing categories in the taxonomy for recommendations also helps in improving the non-accuracy metrics of the recommendation system. Similarity, the best performing priority measure, has shown significant improvement in the diversity, coverage, and dissimilarity of recommended entities. On the other hand, PPC measure had quite a negative impact on the non-accuracy metrics, identical to its impact on the recall of the baseline.

From the results of non-accuracy objectives evaluation, we understand that by exploring the characteristics of concepts in taxonomies we can significantly improve on both the performance and non-accuracy objectives of the recommendation system.

### 4.2.5 Precision-Recall

Results discussed in the prior sections demonstrate the improvement in recall when the characteristics of concepts in the taxonomy are considered for recommendations. Similarly, precision of the recommendations also had significant improvement over the baseline. Since precision is only a function of recall (Eq. 4.1), we observe consistent improvement in precision as recall of the recommendations improve. According to Eq. 4.1, precision of the out degree based specificity measure is 0.03 and 0.02 at top 10 and 20 positions, respectively, in the books domain. While this may be considered low, it would be valuable to note an assumption of the evaluation approach that 1,000 randomly selected entities are uninteresting to the user. This assumption of the evaluation approach, as noted in [15], tends to underestimate the accuracy of our recommendations as user may
be interested in the other entities in the top K recommendations besides the test entity. In Fig. 4.1 through 4.4, we present the comparison of recall up to top 20 for the three characteristics, as well as decline in precision as recall increases. While specificity and priority characteristics improve the performance significantly over the baseline, the respective best performing measures, i.e., \textit{out degree} and \textit{similarity}, have significant difference compared to the other measures of the characteristics.

Figure 4.1: Specificity - Recall@K
Figure 4.2: Specificity - Precision vs Recall

Figure 4.3: Priority - Recall@K
Figure 4.4: Priority - Precision vs Recall
Conclusion and Future Work

Entity recommendation systems as well as recommendation algorithms that use hierarchical relationships from taxonomies as background knowledge bases are becoming increasingly popular. Utilizing taxonomies as background knowledge bases addresses the cold-start and data sparsity challenges predominant in recommendation systems. However, developing a taxonomy-based recommendation system could become challenging as creating and updating a taxonomy manually is a time-intensive and laborious task. Second, insights into what characteristics of taxonomies to consider in recommendation algorithms are not available since taxonomies have only been one piece among the several types of knowledge bases in recommendation systems and their potential as the only knowledge base has not been notably explored.

In this thesis, we primarily focused on automatically generated taxonomies from a crowd-sourced encyclopedia. Utilizing automatically generated taxonomies alleviates the manual effort in creating taxonomies, and creating it from a crowd-sourced encyclopedia helps in keeping it up to date with minimal effort. We have explored three characteristics of concepts in a taxonomy that can impact recommendation systems - (1) specificity, which measures the abstractness of a category, (2) domain relatedness, which measures the relatedness of the category to the domain of interest (e.g. movies, music), and (3) priority, which measures the suitability of a category to an entity or to a sub-category in the taxonomy. We adapted spreading activation theory to develop a
taxonomy-based entity recommendation system and investigated each of the characteristics of the
taxonomy. In our evaluation on the datasets of two domains, i.e. movies and books, we have shown
that the characteristics of concepts in the taxonomy significantly impact entity recommendation
performance.

In the future, we will work on implementing a hybrid recommendation system that utilizes the
explored taxonomic characteristics to compete with the state-of-the-art recommendation systems.
Additionally, applying a combination of the characteristics on state-of-the-art recommendation
systems to augment their performance will also be a likely use of our work. An approach that
combines the specificity and priority of categories is a precursor to the advancement in the
aforementioned future tasks.

Another interesting line of research will be in devising personalized scores of these
characteristics to improve recommendations. For example, an expert in “cars” could be
recommended entities based on more specific (in detail) categories of cars. These categories
generally would have higher specificity scores. On the other hand, a non expert who is interested in
cars can be recommended cars which are generic and popular.

The domain relatedness of categories is found to have much less impact on recommendation
performance as the baseline algorithm itself has prioritized in-domain categories over inter-domain
categories. However, we find that the two domain relatedness measures, per our analysis, distinguish
between in-domain and inter-domain categories. As inter-domain recommendations are on the rise,
i.e. applications recommend entities from one domain based on the users’ interests in another domain
such as recommending movies based on books, it would be interesting to devise a recommendation
algorithm that benefit from our domain relatedness measures in inter-domain recommendation
algorithms.
Bibliography


