UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft
Faculty for Computer Science, Electrical Engineering and Mathematics
Department of Computer Science
Research Group DICE

Seminar Report

Submitted to the DICE Research Group
in Partial Fullfilment of the Completion of

Seminar on

A Hybrid Graph Model for Distant Supervision Relation Extraction : A Report

by
Varun Maitreya Eranki

Thesis Supervisor:
Diego Moussalem

Paderborn, January 27, 2020

Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen worden ist. Alle Ausführungen, die wörtlich oder sinngemäß übernommen worden sind, sind als solche gekennzeichnet.

Ort, Datum	Unterschrift

Abstract.
Write at the last

Contents

Chapter 1

Introduction

From plain text, the task of extracting semantic relations between two or more entities, is known as Relation Extraction (RE). These relations can exist in different types. For example, *"Germany is in Europe"* states a *"is in"* relationship between *"Germany"* and *"Europe"*. With this information, a triple can be formed, <*Germany, is in, Europe*>. Efficient RE is useful for applications like Knowledge Graph (KG) completion and question answering, which are in turn responsible for dependent applications.

Traditionally, supervised RE techniques produce elevated performance for RE[**?**]. They solely rely on labeled data that is manually annotated. Manual annotation is time consuming and need an army of annotators. It is generally annotated into entities and relationships(entities: *"Germany"*, *"Europe"* relationship: *"is in"*). This limitations strongly suggest need for semi-supervised or unsupervised RE techniques that are reliable enough and can mimic manual annotation.

Distant Supervision (DS) aims at solving this limitation by automatic production of labeled data by aligning KGs and plain text. In this process, it makes an assumption that, if there exists a relationship between two entities (*e1, e2*) in Knowledge Base (KB), then all the sentences that consist *e1, e2* express that relationship in some way[ZLCZ15].

This introduces noise problem. It can be countered using Deep Neural Network (DNN) models, which try to provide a significant improvement, but fail in making predictions due to lack of sufficient background information associated with entities and relationships. For example, DNNs create bias at each stage and especially with long-tail relations, background information tend to be unusable for making predictions. They are mostly constructed for customized models to join knowledge that is limited to incorporate heterogeneous background information in parallel. Some of the methods did not handle the side effect caused due to introduced noise.

Chapter 2

Concepts

Assertions: There are two probable types of Assertions that can be. Assertion could be denoted as ⟨ a,P,b ⟩. Typing Assertion means the subject of an RDF triple could be of the type class 'a' and the object could be of class 'b' type or of the datatype 'b' if it is a literal. Relation assertion states that the property of a certain triple would follow the type P(a, b) with both subject and object being of the type 'a' and 'b' respectively.

Pattern Occurrence: The data instances in a schema is said to follow a pattern having the form (C, P, D) if the subject of that instance is of type C, Property of type P and object of the type D. A pattern could be of type for example *{dbo:Film, dbo:directedBy, dbo:Director}* which means the data instances describing a film directedby a person could all fit to the pattern described.These relations could be expressed hierarchically in a Type Graph that would explain the instances at the top as most general type and instances relatively at the lower level as most specific types.

Minimal Type Patterns: A pattern ⟨ C, P, D ⟩ is a minimal type pattern for a relational assertion ⟨ a, P, b ⟩ according to a type graph G iff C and D are the types of a and b respectively and are minimal in G. For example, if you consider a pattern *{dbo:Film, dbo:directedBy, dbo:Director}* then a triple say, {Shutter Island, directedBy, Martin Scorsese} is said to be the minimal type pattern.

Local Cardinality Descriptors: According to Tomasso et. al., Local Cardinality Descriptors give the Cardinalities of RDF properties that are specific to the pattern in question. To define these descriptors individually, a concept of *Restricted property extensions* must be introduced. Tomasso et. al., also define the extension of a property P restricted to a pattern (C, P, D) as the set of pairs {a,b} such that the relational assertion {a,P,b} is part of the dataset and (C, P, D) is a minimal type pattern for {a,P,b}. When referring to these extensions, it is as well known, assumed that in RDF triples, subject and object refer respectively to the first and second element of each pair in the extension.Given a pattern Π with a property P, we can define Local Cardinality Descriptor functions as:

- **minS(π), maxS(π), avgS(π):** Respectively the minimum, maximum and average number of distinct subjects associated with unique objects in the extension of P restricted to π.

- **minO(π), maxO(π), avgO(π):** Respectively the minimum, maximum and average number of distinct objects associated with unique subjects in the extension of P restricted to π.

All these functions including the avgS(π) and avgO(π) return integer values that approximates to the real average values. To get the Global Cardinality Descriptors, the same is followed without the property extensions. The local cardinality descriptors also are used for selecting features that are used to compute similarity for resources. For example say, to compute similarity for the movies, we could automatically omit those properties that occur in pattern π with dbo:Film as sourcetype and avgS(π) = 1 as we know that avgS(π) being equal to 1 means that the average number of subjects linked with unique objects are just one. Also, the values of Local cardinality descriptors differ from the values of the Global cardinality descriptors counterpart. For the property *dbo:cinematography*, the Global cardinality descriptors minS = 1, maxS = 249, avgS = 5, minO = 1, maxO = 13, avgO = 1 while for the pattern dbo:Film,dbo:cinematography,dbo:Person,the Local cardinality descriptors minS = 1, maxS = 249, avgS = 14, minO = 1, maxO = 7, avgO = 1.

Chapter 3

Feature Selection: State-of-the-art vs Ontology Based

Feature selection is the vital part of LD based Recommender system. They aid in improving the predictive performance, to provide faster and more cost-effective predictors and also give a better understanding of the process that generates the data [**?**]. The Measures on which evaluation of Recommender systems are made such as Accuracy, Diversity etc., are entirely dependent on the kind of features that are selected and the ones that are left out. Features are again, the properties in the RDF trifecta. It is only right that those features that hardly occur or contribute very less, if at all, to the recommendation decision are irrelevant and are to be best removed in-order to benefit from overfitting the decision making. In this section, a comparison of the way features are selected in techniques that are state-of-the-art is made against a completely new Ontology based methods that the authors *Tommaso* et.al., propose. The first section describes the mentioned state-of-the-art techniques and the second section gives the new approach.

3.1 Feature selection by State-of-the-art techniques

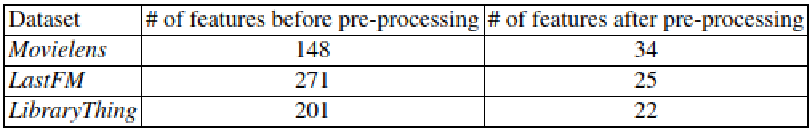
As RDF properties are considered as the features, the idea is to select those properties that are relevant. There are number of techniques that could be used for this purpose. Among them *Tommaso* et.al., considered techniques such as *Information Gain, Information Gain Ratio, Chi-Squared Test and Principal Component Analysis* for the feature selection. The outcome from these features being again fed to a Recommendation Algorithm that calculates the similarity of any two entities that have the features selected here, as their properties. Out of all the techniques tested against the datasets, they found out that the Information gain (IG) works best. IG can be given as [**?**]:

$$IG(f_i) = E(I) - \sum_{v \in dom(f_i)} \frac{|I_v|}{|I|} * E(I_v)$$

Where *E(I)* is the entropy of the data, I_v is the number of items in which the feature f_i (which could be director for movies)has a value equal to v (could be Martin Scorsese in the movie domain), and $E(I_v)$ is the entropy computed on data where the feature f_i assumes value v. The *IG* of a feature f_i is higher as the lower is the value of the entropy $E(I_v)$. Then features are ranked according to their *IG* value and the top-k ones are taken into consideration.

It may happen that in most of the cases, we may require to *Preprocess* and remove some of the features that may lack the knowledge domain and might not be relevant or missing values and hence act as anything but noise in the dataset. These features would only add uncertainty into the model and might act negatively in deciding the features. So, the authors prescribe a preprocessing strategy that would better suit here. For instance, in the Movie domain, properties such as *dbp:artDirection* or *dbp:precededBy* are very specific and have a lot of missing values. Also, properties such as *dbo:wikiPageExternalLink* and *owl:sameAs* always have unique and distinct values, so they're seldom not very informative.

This may also pose *scalability* issues that is better to avoid at any costs if there is a chance to do so. The preprocessing, based on the idea by *Heiko* and *Johannes* [**?**] is done by them as follows: A threshold tm = td = 97% both for missing and distinct values are set. The features having values missing more than tm and distinct more than td are left out and discarded. The following table gives the results of the preprocessing step as to how many number of features were significantly of little use and hence would only act negatively and hence ultimately removed. The remaining left out number of features could be used for the calculation of IG discussed above and ultimately the top-k ones are considered.

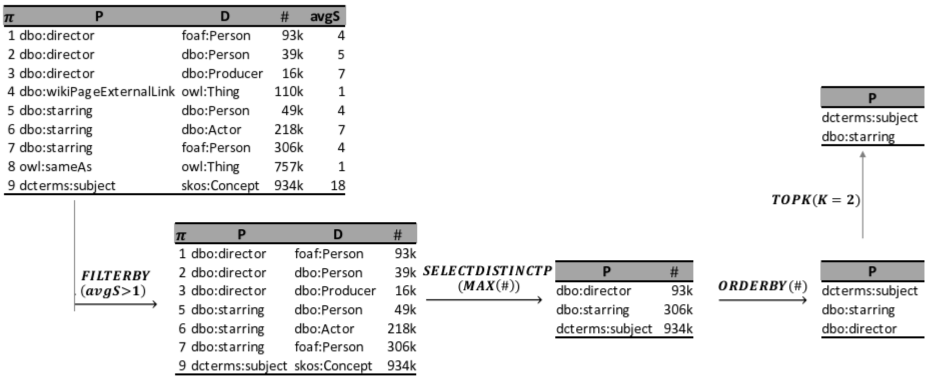


Dataset	# of features before pre-processing	# of features after pre-processing
Movielens	148	34
LastFM	271	25
LibraryThing	201	22

Table 1: Reduction on the number of features after the pre-processing step.

3.2 Feature selection - Ontology based technique

The Ontology based features uses the pattern based summaries extracted from ABSTAT [**?**] framework. ABSTAT, in general takes a linked dataset and – if specified – one or more ontologies are input and returns a summary that consists of: a type graph, a set of patterns, their frequency, local and global cardinality descriptors. The summaries returned by these approaches are complete, as in they give statistics about every element used in the vocabulary/Ontology in the dataset. Among these statistics, the pattern frequency and the cardinality descriptors are of importance and are used in the feature selection process specified by *Tommaso* et.al. The process starts by considering all patterns $\Pi = \{\pi 1, \pi 2, \pi 3, \ldots\ldots, \pi n\}$ of a given class C occurring as a source type.



π	P	D	#	avgS
1	dbo:director	foaf:Person	93k	4
2	dbo:director	dbo:Person	39k	5
3	dbo:director	dbo:Producer	16k	7
4	dbo:wikiPageExternalLink	owl:Thing	110k	1
5	dbo:starring	dbo:Person	49k	4
6	dbo:starring	dbo:Actor	218k	7
7	dbo:starring	foaf:Person	306k	4
8	owl:sameAs	owl:Thing	757k	1
9	dcterms:subject	skos:Concept	934k	18

π	P	D	#
1	dbo:director	foaf:Person	93k
2	dbo:director	dbo:Person	39k
3	dbo:director	dbo:Producer	16k
5	dbo:starring	dbo:Person	49k
6	dbo:starring	dbo:Actor	218k
7	dbo:starring	foaf:Person	306k
9	dcterms:subject	skos:Concept	934k

P	#
dbo:director	93k
dbo:starring	306k
dcterms:subject	934k

P
dcterms:subject
dbo:starring
dbo:director

P
dcterms:subject
dbo:starring

Fig. 1: Feature selection with ABSTAT
with source type `dbo:Film`

The example given below shows a subset of Π with *dbo:Film* as source type. The first step filters out (FILTERBY) properties based on the local cardinality descriptors. In particular, it filters out the properties for which the average number of distinct subjects associated with unique objects is more than $(avgS\rangle 1)$. The idea here being to take into consideration, only those properties connecting one target type with many source types. In the example, patterns $\pi 4$ and $\pi 8$ with *dbo:wikiPageExternalLink* and *owl:sameAs* property respectively are removed because there exists on average only one subject of type dbo:Film associated with a distinct object. The second step is to select all (SELECTDISTINCTP) properties of the patterns in Π by applying the maximum of the pattern frequency (#). Then, the properties are ranked (ORDERBY) in a descending order on pattern frequency and then k (TOP-K) properties are selected (k = 2).

The earlier proposed feature preprocessing is also done here to remove irrelevant properties with namespaces similar to those of relevant ones. For example,dbp:starring is similar to dbo:starring and while the latter is important feature, the former is not really of much use. So, filtering such properties is of suggestion.

Chapter 4

Recommendation Method

To Recommend an item or entity, it is essential to know if it is somewhat similar to another entity that is in past, liked or selected. That is to say if there is a reference of some entities that a user holds a liking or preference to, then a system to recommend items has a baseline as to the kind of items that the same user could probably like to have in the future. Keeping this in mind, the authors use an item-based nearest neighbor algorithm as in [**?**],where similarity measures to calculate the similarity between an item that the user prefers or already liked to the other items in the catalogue are made use of. The *Jaccard* similarity is one such effective measure used in calculating the semantic recommendation for categorical features. Tomasso et. al., give the following formulation of Jaccard Similarity between two resources i and j in a LD dataset:

$$Jaccard(i,j) = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|}$$

.

Where *N(i)* and *N(j)* are the neighbors of i and j respectively in the RDF graph. The formulation in this scenario means that the neighborhood of i (respectively j) includes all the nodes in the graph reachable starting from the resource i (respectively j) following the properties selected in the feature selection phase. Hence, neighbors are one hop features. It is also noteworthy to say that the frequent properties like *dbo:wikiPageExternalLink* or *owl:sameAs*, which have always different and unique values, would only bring noise when computing Jaccard similarity. Also, computing Jaccard using the very large number of features that can be found in linked datasets would not be efficient at runtime, considering the recommendations need to be computed almost at interactive time. These Observations provide additional evidence for the need of a feature selection method before computing recommendations.

So, given an item j and a user u, the following formulation is used to predict the ratings of items i which is unknown to the user:

$$r^*(u,i) = \frac{\sum_{j \in N(i) \cap r(u)} jaccard(i,j) \cdot r(u,j)}{\sum_{j \in N(i) \cap r(u)} jaccard(i,j)}$$

Where *r(u)* represents the items rated by the user u and *r(u,j)* the rating value given by the user u with respect to the item *i*. Therefore, the above equation considers the neighbors of i belonging to the user profile and computes an average of the user ratings to such neighbors weighted by the similarity values.

Chapter 5

Experimental Evaluation

A walkthrough of the setup, involving the datasets used and to the kind of metrics, the results from the recommendation algorithm were evaluated against and some outline of the implementation will be made through initially. Finally, the results and the significance of the values within those results are made known.

5.1 Setup, Measures and Implementation

Datasets: Three different datasets of different domains i.e., *MovieLens* [of Movie domain], *Last.fm* corpus [of Music domain], *LibraryThing* [of books domain] are made use of, to evaluate. MovieLens contains 1,000,209 ratings (1 to 5 stars) given by 6,040 users to 3,883 movies. LibraryThing contains 7,112 users,37,231 books and 626,000 ratings ranging from 1 to 10. The Last.fm corpus has data containing of 1,892 users, 17,632 artists and 92,834 relations between a user and a listened artist together with their corresponding listening counts.

Measures: Measures aside from accuracy are to be also taken into account for evaluation. Accuracy though could be the main concern and bottom-line of some of the recommender systems, there are other measures such as diversity and novelty that could also become important. Taking this into consideration, after settling out on a feature selection technique, the recommender is evaluated against the standard metrics defined to evaluate measures such as diversity and accuracy. To evaluate Accuracy, metrics *Precision (Precision@N)* and *Mean Recprocal Rank (MRR)* are made use of. Precision@N is a metric denoting the fraction of relevant items in the Top-N recommendations. MRR computes the average reciprocal rank of the first relevant recommendation item, and hence results that are particularly meaningful when users are provided with few but valuable recommendations. Diversity is another important measure that should also be heavily considered [**?**]. An accurate recommender that doesn't diversify the entities while recommending is said to have low degree of Personalization [**?**]. To evaluate aggregate diversity, metrics *Catalog coverage* (The percentage of items in the catalog recommended at least once) and *Aggregate entropy* [**?**] are made use of. While the catalog coverage is used to assess the ability of a system to cover the item catalog, namely to recommend as many items as possible, the aggregate entropy measures the distribution of the recommendations across all the items, showing whether the recommendations are concentrated on a few items or are they distributed better.

Implementation: Different ranking and filtering functions by combining local cardinality descriptors and pattern frequency to study their effect on feature selection are made. Some of them being:

AbsMaxS: Instead of frequency of patterns that are considered for selection (SELECTDISTINCTP) as stated in Section 3.2, the maxS is considered for the selection.

AbsOccAvgS: Here, the maximum of the pattern frequency is up for the selection (SELECTDISTINCTP) and filtering (FILTERBY) is done by avgS.

AbsOcc*MaxS: Here, the product of maxS and pattern frequency is considered for selection (SELECTDISTINCTP).

All these have the prefix "Abs" since the idea proposed here i.e., Ontology based feature selection is an extension to the ABSTAT system on which some of the same authors had previously worked on. In a way, the naming convention can be said to have split into two parts with the prefix "Abs" denoting the ABSTAT system and the remaining part which would denote the selection and ranking done with the concept of Local cardinality descriptors which is the extension work done by the authors here. Both for Ontology based method and the state-of-the-art IG method, different configurations in the experimental settings are considered and as follows:

noRep: Here, the first N features selected are considered and if the results contain both dbo: and dbp: feature (say, for example as dbo:starring and dbp:starring),the features that appear later in the ranking or comparatively ranked below are ousted.

withRep: Both dbo:starring and dbp:starring are taken into consideration and in the order that they appear in the ranking.

Onlydbp: If the resulset contained both dbo:starring and dbp:starring, only the dbp: one is taken into consideration.

Onlydbo: Same as above, only that dbo: ones are considered. However, in the experiments of MovieLens and Lastfm, the dbo: features are not present. In such cases, the results become exactly the same as the noRep configuration which is reflected in the table shown below.

Baseline: The *Term frequency-inverse document frequency*, abbreviated as *TfIdf* is a common technique to identify most relevant terms (properties) for a document (class). In this context, TfIdf with respect to the document is the set of patterns here having the same subject-type and the term is the property. Formally, TfIdf can be said as the number of properties occurring in a document that corresponds to Tf and the logarithm of the ratio between the total number of documents and the number of documents containing the property that corresponds to Idf. While Tf is proportional to the number of properties occurring in a document, Idf tries to penalize those properties that occur very frequently and that those rarely occur.

5.2 Results and Discussion

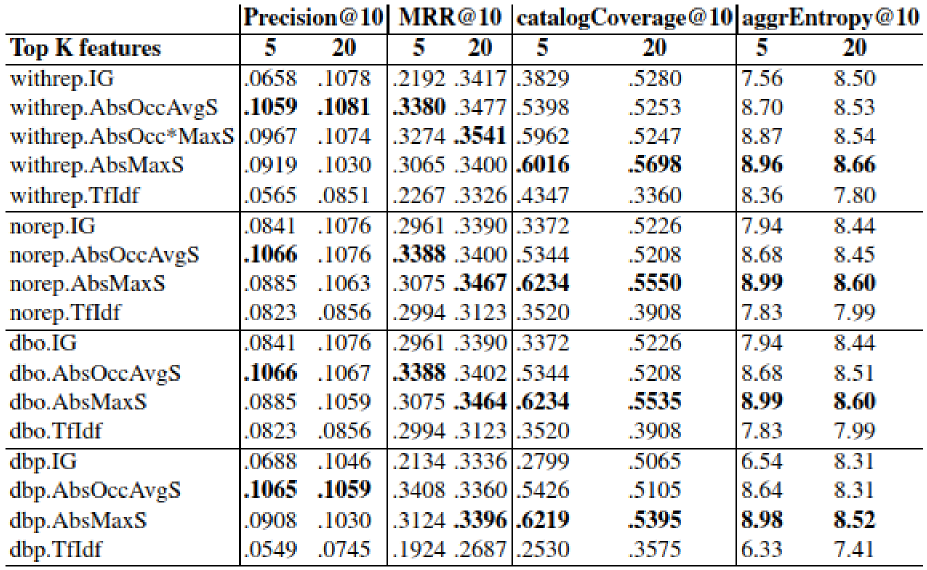
Tables 2,3 and 4 show the experimental results obtained on respectively, MovieLens, Last.fm and LibraryThing datasets in terms of metrics defined for the measures Accuracy and diversity i.e., Precision, MRR, catalogCoverage and aggrEntropy. Results are computed over lists of top-10 items recommended by configurations but results for k = 5,20 are shown here. The values are highlighted in bold for those that have statistically significant difference. For Last.fm dataset, the differences are not statistically significant, so the two methods are equivalent in selecting features.

Discussion:

Some of the observations made from the evaluation results can be summarised as follows:

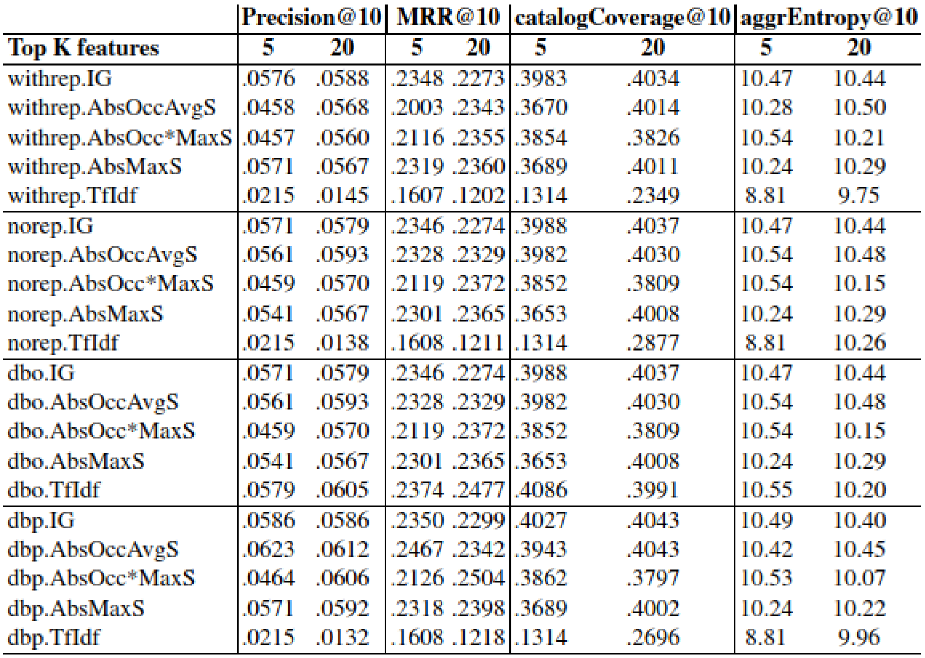
In Overall, Ontology based feature selection leads to better results when measured against accuracy and diversity for both MovieLens and LibraryThing. IG based selection leads to a marginally better result for Last.fm.

With respect to MovieLens, the results of the Ontology based technique is better with all the



	Precision@10		MRR@10		catalogCoverage@10		aggrEntropy@10	
Top K features	**5**	**20**	**5**	**20**	**5**	**20**	**5**	**20**
withrep.IG	.0658	.1078	.2192	.3417	.3829	.5280	7.56	8.50
withrep.AbsOccAvgS	**.1059**	**.1081**	**.3380**	.3477	.5398	.5253	8.70	8.53
withrep.AbsOcc*MaxS	.0967	.1074	.3274	**.3541**	.5962	.5247	8.87	8.54
withrep.AbsMaxS	.0919	.1030	.3065	.3400	**.6016**	**.5698**	**8.96**	**8.66**
withrep.TfIdf	.0565	.0851	.2267	.3326	.4347	.3360	8.36	7.80
norep.IG	.0841	.1076	.2961	.3390	.3372	.5226	7.94	8.44
norep.AbsOccAvgS	**.1066**	.1076	**.3388**	.3400	.5344	.5208	8.68	8.45
norep.AbsMaxS	.0885	.1063	.3075	**.3467**	**.6234**	**.5550**	**8.99**	**8.60**
norep.TfIdf	.0823	.0856	.2994	.3123	.3520	.3908	7.83	7.99
dbo.IG	.0841	.1076	.2961	.3390	.3372	.5226	7.94	8.44
dbo.AbsOccAvgS	**.1066**	.1067	**.3388**	.3402	.5344	.5208	8.68	8.51
dbo.AbsMaxS	.0885	.1059	.3075	**.3464**	**.6234**	**.5535**	**8.99**	**8.60**
dbo.TfIdf	.0823	.0856	.2994	.3123	.3520	.3908	7.83	7.99
dbp.IG	.0688	.1046	.2134	.3336	.2799	.5065	6.54	8.31
dbp.AbsOccAvgS	**.1065**	**.1059**	.3408	.3360	.5426	.5105	8.64	8.31
dbp.AbsMaxS	.0908	.1030	.3124	**.3396**	**.6219**	**.5395**	**8.98**	**8.52**
dbp.TfIdf	.0549	.0745	.1924	.2687	.2530	.3575	6.33	7.41

Table 2: Experimental results on the Movielens dataset. Bold values indicates that the difference with the other methods are statistical significant (T-test with p-value < 0.0001).



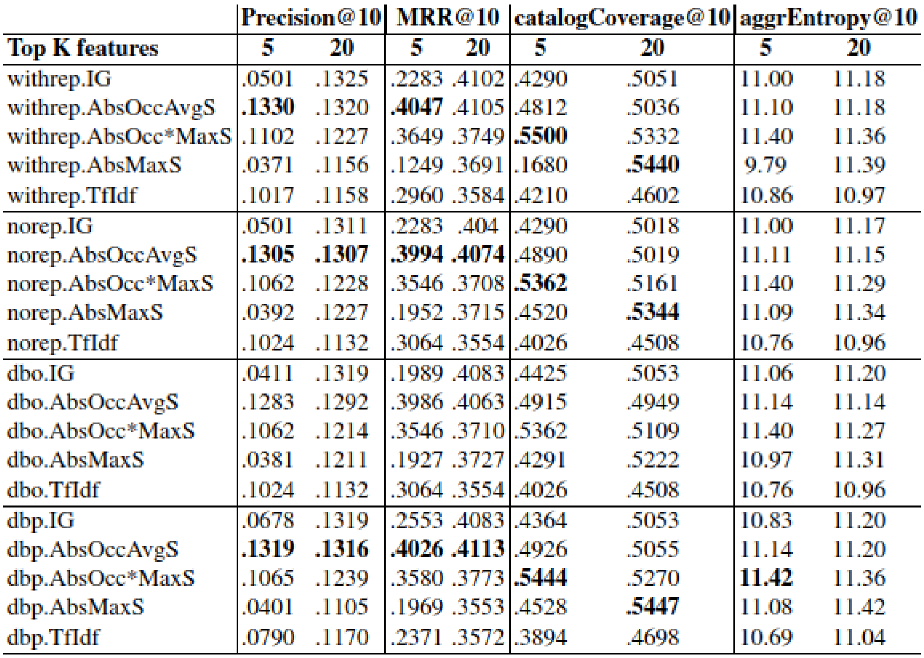
	Precision@10		MRR@10		catalogCoverage@10		aggrEntropy@10	
Top K features	**5**	**20**	**5**	**20**	**5**	**20**	**5**	**20**
withrep.IG	.0576	.0588	.2348	.2273	.3983	.4034	10.47	10.44
withrep.AbsOccAvgS	.0458	.0568	.2003	.2343	.3670	.4014	10.28	10.50
withrep.AbsOcc*MaxS	.0457	.0560	.2116	.2355	.3854	.3826	10.54	10.21
withrep.AbsMaxS	.0571	.0567	.2319	.2360	.3689	.4011	10.24	10.29
withrep.TfIdf	.0215	.0145	.1607	.1202	.1314	.2349	8.81	9.75
norep.IG	.0571	.0579	.2346	.2274	.3988	.4037	10.47	10.44
norep.AbsOccAvgS	.0561	.0593	.2328	.2329	.3982	.4030	10.54	10.48
norep.AbsOcc*MaxS	.0459	.0570	.2119	.2372	.3852	.3809	10.54	10.15
norep.AbsMaxS	.0541	.0567	.2301	.2365	.3653	.4008	10.24	10.29
norep.TfIdf	.0215	.0138	.1608	.1211	.1314	.2877	8.81	10.26
dbo.IG	.0571	.0579	.2346	.2274	.3988	.4037	10.47	10.44
dbo.AbsOccAvgS	.0561	.0593	.2328	.2329	.3982	.4030	10.54	10.48
dbo.AbsOcc*MaxS	.0459	.0570	.2119	.2372	.3852	.3809	10.54	10.15
dbo.AbsMaxS	.0541	.0567	.2301	.2365	.3653	.4008	10.24	10.29
dbo.TfIdf	.0579	.0605	.2374	.2477	.4086	.3991	10.55	10.20
dbp.IG	.0586	.0586	.2350	.2299	.4027	.4043	10.49	10.40
dbp.AbsOccAvgS	.0623	.0612	.2467	.2342	.3943	.4043	10.42	10.45
dbp.AbsOcc*MaxS	.0464	.0606	.2126	.2504	.3862	.3797	10.53	10.07
dbp.AbsMaxS	.0571	.0592	.2318	.2398	.3689	.4002	10.24	10.22
dbp.TfIdf	.0215	.0132	.1608	.1218	.1314	.2696	8.81	9.96

Table 3: Experimental results on the LastFM dataset.

configurations (noRep, withRep,Onlydbp and Onlydbo) with Top-5 and Top-20 features when measured against accuracy. When measured against diversity, the technique still holds its own in almost all the cases and works best with the AbsMaxS configuration.

With respect to LibraryThing, the results of the Ontology based technique is still better with all the configurations when measured for accuracy. Specifically, the results show doubled accuracy when compared to the IG based method for the Top-5 feature selections. When measured for diversity, Ontology based technique still offers better results than the IG counterpart.

Only when coming to the Last.fm corpus, the results show an inherent similarity or changes very little between the techniques.

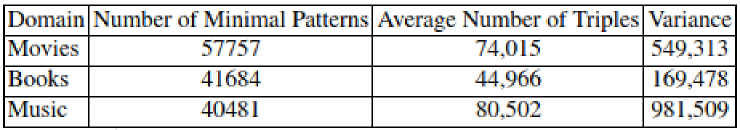


	Precision@10		MRR@10		catalogCoverage@10		aggrEntropy@10	
Top K features	**5**	**20**	**5**	**20**	**5**	**20**	**5**	**20**
withrep.IG	.0501	.1325	.2283	.4102	.4290	.5051	11.00	11.18
withrep.AbsOccAvgS	**.1330**	.1320	**.4047**	.4105	.4812	.5036	11.10	11.18
withrep.AbsOcc*MaxS	.1102	.1227	.3649	.3749	**.5500**	.5332	11.40	11.36
withrep.AbsMaxS	.0371	.1156	.1249	.3691	.1680	**.5440**	9.79	11.39
withrep.TfIdf	.1017	.1158	.2960	.3584	.4210	.4602	10.86	10.97
norep.IG	.0501	.1311	.2283	.404	.4290	.5018	11.00	11.17
norep.AbsOccAvgS	**.1305**	**.1307**	**.3994**	**.4074**	.4890	.5019	11.11	11.15
norep.AbsOcc*MaxS	.1062	.1228	.3546	.3708	**.5362**	.5161	11.40	11.29
norep.AbsMaxS	.0392	.1227	.1952	.3715	.4520	**.5344**	11.09	11.34
norep.TfIdf	.1024	.1132	.3064	.3554	.4026	.4508	10.76	10.96
dbo.IG	.0411	.1319	.1989	.4083	.4425	.5053	11.06	11.20
dbo.AbsOccAvgS	.1283	.1292	.3986	.4063	.4915	.4949	11.14	11.14
dbo.AbsOcc*MaxS	.1062	.1214	.3546	.3710	.5362	.5109	11.40	11.27
dbo.AbsMaxS	.0381	.1211	.1927	.3727	.4291	.5222	10.97	11.31
dbo.TfIdf	.1024	.1132	.3064	.3554	.4026	.4508	10.76	10.96
dbp.IG	.0678	.1319	.2553	.4083	.4364	.5053	10.83	11.20
dbp.AbsOccAvgS	**.1319**	**.1316**	**.4026**	**.4113**	.4926	.5055	11.14	11.20
dbp.AbsOcc*MaxS	.1065	.1239	.3580	.3773	**.5444**	.5270	**11.42**	11.36
dbp.AbsMaxS	.0401	.1105	.1969	.3553	.4528	**.5447**	11.08	11.42
dbp.TfIdf	.0790	.1170	.2371	.3572	.3894	.4698	10.69	11.04

Table 4: Experimental results on the LibraryThing dataset. Bold values indicates that the difference with the other methods are statistical significant (T-test with p-value < 0.0001).

When investigating for the reason for different behaviors based on selected domain, authors then evaluated two orthogonal dimensions related to their corresponding sub-graphs. The two dimensions reflect the two different aspects related to the Feature selection technique. While, the Ontology based technique is heavily rooted with the underlying schema, IG and other techniques (chi-square and Gain ratio) mainly consider the data and the triples that represent them.

Hence, for the three domains, authors measured: (i) Number of minimal patterns and, (ii) The average number of triples per resource and the corresponding variance. Former signifies as to when there is a higher number of minimal patterns present, it only means a richer and more diverse ontological representation of the domain. Latter provides info as to when variance in the number of triples associated to resources is at high, it only means that there could be an unbalanced representation of items to recommend. Hence, items with a higher number of triples associated result "more popular" in the knowledge graph compared to those with only a few. This may reflect in the rising of a stronger content popularity bias while computing the recommendation results. From the table5, it could be said that the music domain is the one with lowest number of minimal patterns and the highest variance, book domain have the lowest values in terms of variance and Movie domain show intermediate values in terms of variance and has highest number of minimal patterns.



Domain	Number of Minimal Patterns	Average Number of Triples	Variance
Movies	57757	74,015	549,313
Books	41684	44,966	169,478
Music	40481	80,502	981,509

Table 5: Ontological and data dimensions of the three datasets

Hence, the authors conclude that the values assert that a higher sparsity in the knowledge graph data may give chance to statistical methods to beat ontological ones. In other words, it seems that the higher the sparsity of the knowledge graph at the data level, the lower the influence of the ontological schema in the selection of the most informative features to build a pure content-based recommendation system.

Chapter 6

Conclusion and Future work

To Summarize succinctly, authors *Tommaso* et.al., in the work *"Using Ontology-based Data Summarization to Develop Semantics-aware Recommender Systems"* propose a different approach for data summarization for feature selection that could be used for recommender systems. Unlike some of the previous works in the area, the proposed method here is firmly rooted in the Ontology of the data rather than the data itself which based on the experiments carried over the data in three domains mainly Movie, Books and Music provide better results in terms of Accuracy and Diversity of the recommended items than the existing state-of-the-art techniques such as Information gain in those domains. Apart from providing better results, the proposed technique also boasts advantage in terms of those scenarios where the entire data may not be available to make use of.

These findings calls for a firm case for expanding and exploring the proposed technique further. The technique must be evaluated over the other application domains apart from the mentioned ones to test the versatility of the approach. Also, the datasets that are not particularly rich in Ontology are to be tested with, to see how it particularly fares against that of Information Gain. Further work also calls out for evaluating against other measures such as serendipity and unexpectedness etc., The authors intend to take the next step for the proposed method by considering similarities with multi-hop between entities. These would result in a more complex subgraphs that are estimated to be relevant for LD-based RSs.

Bibliography

[ZLCZ15] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1753–1762, 2015.