International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 9 ǁ September 2022 ǁ PP. 376-383

A novel time-based seed generation algorithm by means of a knowledge graph

VSK Sindhura[1],YJahnavi[2*]
[1] *Associate consultant oracle health insurance (OHI), Telangana,India*
[2] *Department of Computer Science, Dr V S Krishna Govt Degree and PG College (Autonomous), Visakhapatnam, Andhra Pradesh,India*

Abstract:
Entity Set Expansion (ESE) is extensivelyexploited in numerousimplementations such as query design, dictionary construction, and new tag identification. ESE is a challenge that elongates a minor set of seed entities into anadditionalwhole set whose entities have basic qualities. A number of methods have been projected for ESE, and these methods can be abridged into three categories, text corpus-based, web-based, and others. The first two categories mainly use distribution information or context seed pattern to develop entities. Others influence the knowledge graph, which is an important tool for storing and retrieving graph-structured data such as Wikipedia and Geonames as an aid for enhancing ESE performance in the text or on the web. In the proposed model, we not only divide the entity into seed pairs, but also provide accuracy, ranking, and time delay.

Keywords*: Knowledge graph, Entity set expansion,Seed generation algorithm,Meta path*

Date of Submission: 03-09-2022 Date of acceptance: 17-09-2022

I. INTRODUCTION:

Big data is a field that accords with ways of methodicallyanalyzing, extracting information from data sets that are too huge or multifaceted for traditional data processing application software to manage. Data with numerousinstancesput forwardgreater statistical power, while data with greater complexity can manage to a greater false discovery rate. Big data issues comprise data collection, data storage, data analysis, retrieval, transmission, visualization, querying, updating, information privacy, and data source. Big data was initiallyconnected with three key perceptions: volume, variety and velocity. Other perceptionsfutureaccredited to big data are truthfulness and value.Data sets are acceleratingrapidlydue to the factthat they are progressivelyaccumulated by inexpensive and abundant information appliances.In accordance with an IDC review forecast, the global volume of data will raisemore and more rapidly from 4.4 zetta bytes to 44 zetta bytes between 2013 and 2020. By 2025, IDC anticipates that there will be 163 zetta bytes of data. One of the questions for large enterprises thatdecidethe owner of the big data initiatives stirring the whole organization[1-3].

Presently the term big data pertain to the function of behavior analytics, predictive analytics, or particularsophisticated data analytics algorithms that pullout value from data. The most important feature of this innovative systemis the huge amounts of available data. Analyzing data sets can determinefurtherassociations to identifyenterprisetendencies, block disease, fight crime, and so on. Business managers, Doctors, Scientists etc.,consistently struggle with big data sets in fields such as Internet search, fin tech, business and urban informatics[4].

Big data warehousesareinhabited in numerousmodels. Various companies built these big data warehouses with a particularrequirement. Parallel big data management systems are also existingsince the 1990s. Winter Corp has announced the enormous database proceedings for many years. Teradata association released the DBC 1012 parallel processing system in 1984. Teradata systems were the primaryforaccumulation and examine 1 terabyte of data in 1992. In 2017, numeroushundred petabyte-class Teradata relational databases are mounted, the gigantic of which surpasses 50 PB. There are various categories of data such as structured, semi-structured and unstructured data. The large quantities of these data are available in various forms for processing. For storing various amounts of data and handling these different types of stored big data, LexisNexis Group introduced a C++based distributed file sharing structure. Unstructured data types such as JSON, Avro and XML need to be stored and has to be processed on multiple servers[5].

Relational database management systems and software packages used to envisage data often struggle to handle big data. The function may necessitateenormously parallel software consecutivelyworking on large

number of servers [6].The MapReduce approachdelivers a parallel processing model, and a relatedoperation has been released for dispensationof large quantities of data [7].The approach was very efficacious, so others requiredto reproduce the algorithm. Consequently, the implementation of the MapReduce framework was carried out by an open-source Apache project called Hadoop. This type of architectureattempts to produce computing performance limpid to the end user using a front-end application server [8]. There exist numerous other algorithms for processing diversevarieties of data [9-15].A data lake permits a body to moveemphasis from centralized management to a shared model to retort to the altering dynamics of information management. This permitsrapiddiscrimination of data into the data lake, dropping overhead. Comprehensive examinations on real datasets prove the usefulness and efficacy of MP ESE. MP ESE can accomplish very good performance in general extension tasks. The proposed method is also appropriate from the point of assessment of the steadiness between effectiveness and efficiency, and also attains anadditionalsuitable weight for the Meta path [24].

II. LITERATURE SURVEY:

In our daily life, when data is growing as fast as possible, this can lead to various kinds of problems. Today, big data will play a vital role in most applications such as data storage, data analytics, search engines, etc. In this work, we will expand the initial entities from a large entity set to improve the efficiency and accuracy of search data. Thus, we have included in the literature review work that shows different approaches to entity expansion.

There are anextensive variety of descriptions and definitions about the knowledge graph that is used to optimize search results by Google. A knowledge graph is a knowledge base system that is formed from semi-structured knowledge with semantic properties or taken from the web by a combination of statistical and linguistic methods. Various refinement methods have been proposed in order to deduce and enhance missing knowledge to the knowledge graph as well as to enhance its expediency, numerous existing refinement methods have been proposed [17][18].

In recent years, there have also been extensive researches on knowledge graph utilization tasks, particularly in the natural language question and answer system [19-21] that can be represented in different ways. For example, Zou et al. proposed a systematic framework from a graph-based perspective for answering natural language questions over an RDF repository. A heterogeneous information network (HIN), comprising of various types of objects and relationships that has important properties such as rich semantics and complex structure. A statistical model can be created for the knowledge graph data and can be represented using a tensor. Nickel et al., revieweda relational machine learning methods for the knowledge graph [22][23].

A Meta path is anarrangement of relationship types linking object types,for the reason thatdiverse types of objects and links coexist in HIN and have diverse semantic meanings; some data mining tasks based on dissimilarmetapaths may get various results. Sun et al propose the concept of Meta path and examine top-k similarity search based on Meta path in HIN.

Then, Shi et al., propose a general relevance measurement framework based on the Meta path in HIN, which can evaluate the association of objects with the identical or diverse types in an unified framework. In addition to the underlying resemblance measure problem, Sun et al., use a meta-path to manage clustering with different semantics and amalgamate meta-path selection to cluster objects in HIN with user-driven clustering. Yu et al. [25] initiate latent functions based on metapaths that signify the association between users and items along various types of paths, and then determine recommendation models at personalized level and also at global level.

Recently, Shi et al. [26] recommendtraining the significance of HIN from a probabilistic viewpoint and proposing a reproductive model to originate a new path-based significance measure called PReP. Although the Meta path has demonstrated its strong capabilities in numerous mining functions, it has not yet been approved to put right the ESE problem. In this work, we first use the Meta path to apprehend the rich semantic meaning among entities to solve the ESE problem in the knowledge graph.

Shi et al.,recommend a probabilistic Co-Bootstrapping technique to resolve the extensionlimits problem and the semantic drift problem. Krishnan et al. [27] practice semantic information from Wikipedia and recommend a technique called Select-Link-Rank to produceexpanded query extensions. In order to exploit the intrinsicassociationamong entities and attributes, Zhang et al. [28] recommend a combined model for entity set extension and attribute extraction.

Bing et al. develop a novel framework to attain the objective of Wikipedia entity extension and attribute extraction. Recently, several researchers are starting to use extraneous semantic information to expand the functioning of entity set proliferation for a text or web data source. For example, Qi et al.,present semantic knowledge using Wikipedia and diminish seed uncertainty. Sadamitsu et al.,practice topic information to mitigate semantic drift. Jindal et al., present some negative examples for extension class constraints.

Yu et al. [29] suggest a set of evaluation models based on meta-paths to characterize semantic connotation for entity query. QBEES is intended for entity resemblance search founded on entity characteristics. Fetahu et al. recommend a dual approach to entity retrieval as well as offline preprocessing and an optimized retrieval model. Kahng et al. suggest a probabilistic entity retrieval model for RDF graphs by means of paths in the graph.

Chen et al. [30] recommend a system for entity examination and debugging. They do not use the knowledge graph and the HIN method. They consider that an entity can be labelled by entities, types, or relationships that are immediately associated. However, they overlook those entities or relationships with which they are indirectly associated, which may also define the entity to some stretch, even comprisesignificantknowledge.

Although Yu et al. [29] acquire the Meta path, but the Meta paths in their approaches have to be furnishing by domain expert users, which is perceptibly time-consuming and expensive, specifically for large datasets.

In this paper, we not only extend entities and establish relationships between them, but also provide significant Meta paths and time delay, accuracy and Meta path efficiency evaluation.

III. METHODOLOGY:

Implementation is the phase when the conceptual design is thrown into a functioningprocedure. It can therefore be examinedas the essential phase to achieve anefficaciousnovelstructure and to give the user confidence that the novel system will operate and be efficacious. The implementation phase includes meticulousdesigning, examination of the existing system and its confines of the implementation,outline of methods to accomplishmodifications, and assessment of the approaches.Apache Tomcat is a Servlet container used in the official reference implementation for Java Servlet and Java Server Pages technologies. Web servers like Apache Tomcat support only web components, while application server supports both web components and business components. SQL Server is used to communicate between client and server systemthat utilizes transact-SQL. SQL Server resourcefullyassignsexisting resources such as disk I/O, network bandwidth, and memory among numerousmanipulators. Other functions like TIMESTAMP and TIMESTAMP with TIMEZONE for saving TIME are also verified. Code setting flexibility intervals of twelve rules such as integrity constraint independence, guaranteed access rule, data representation, data description, correct handling of null values, security, logical independence, physical independence, display update, embedding, complex data sublanguage and rule updates are also need to be checked whether they are meeting the basic requirements. Real-time information is available in the resource management module. All packages are scheduled through this module. The list of documents, images, user numbers can be displayed at any time. The search history can also be usedalong with the time delay on the desktop. Accuracy, order, and corresponding time-related status can also be checked with the resource planning module. The algorithm has been represented as follows:

Algorithm: Time based SMPG
Input: *Text Dataset TD, Keyword k*
Output: *Seed pairs SP,SPC,DT*

1: ***create*** a root node for tree *TR*;
2: *ls*<= *set of links*;
3: ***while****TR* can be expanded ***do***
4: *MX*<= *nodes* with maximum no. of tuples;
5: ***foreach*** tuple *tpl* € *MX****do***
6: get pair *tpl*.(*s*,*d*);
7: ***for each*** adjacent *a* of *tpl.d* on *TD****do***
8: *p*<= *path* from *tpl.d* to *a*
9: ***if****a* not being visited &*p* € *ls****then***
10: ***if****p* not in *MX.child*;
11: create a new child node *nc* with key *MX.key+p*;
12: ***end if***
13: ***if*** (*s*,*a*) € *seedpairs****then***
14: ⌐*(s,a|CP)*<= 1;
15: ***add****(s,a)* to child *nc*;
16: ***else***
17: ⌐(*s*,*a*|*CP)*<= 0;
18: ***end if***
19: add *a* tovisitedset*(s ..... d)*;
20: insert *tuple<(s,a),⌐(s,a|CP),(s,d)>* into *treenode*;
21: add *tpl.s*to source set of *nc*;

22: update the *SPC* of node *nc*;
23: ***end if***
24: ***endfor***
25: ***end for***
26: ***for each*** node *aMX* in *TR**do***
27: *DT*<= *delay time*;
28: *SPC*<= *accuracy*;
29: *SIT*<= *find*& store the session created time for *session id*;
30: *SET*<= *find*& store the session last accessed time for *session id*;
31: *DT = SET-SIT*;
32: *i*<= document in which *k* occured;
33: ∑*i*<= total no.of documents;
34: *SPC = i/*∑*i*;
35: ***if****aMX.SPC*>threshold *v**then***
36: add seed pairs that *CP* connects in *SP*;
37: ***else***
38: set *SPC*<= 0 & add the *seed* pairs;
39: ***end if***
40: ***end for***
41: ***end while***
42: return ***SP,SPC,DT***;

Here, we proposed a novel algorithm called Time based seed meta path generation. In order to automatically discover the meta paths between seeds, we design SMPG algorithm. In our work, we proposed time based SMPG that not only discover the meta paths but also shows the time delay. Here our basic idea is that TSMPG searches the keyword *k* from all the seeds and find the significant paths that connect with the seed pairs and shows the intrinsic relations among the seeds.

The process of meta path generation is to traverse among the text dataset *TD* and thus a tree like structure is introduced called *TR*. It stores a list of entities with similar values and set of visited entities. Here the format of the list is as follows *<(s, d), ⌐(s, d| CP),(s.....d)>* where *(s,d)* denotes the source node and destination node of the current path *CP*. The score or sparsity score *SPC* is used to measure the accuracy of the path. The edge between the entities is known as link. The root node of the tree *TR* contains all entity pairs with seeds. Now the expansion starts in a step-by-step manner to discover important meta paths. At each step we check the status of *SPC* is greater than the threshold value *v*. If the constraint is fulfilled, we take the corresponding meta path or we will move forward for the termination. While proceeding further, we have to select the node with maximum number of tuples to get more accurate entities. Here, we are not going for minimum, because of thereason, the less the number of tuples the discriminability will be larger.

Specifically, in TSMPG we take the source set to record all source entities. The predefined threshold value v represents the smallest value of accuracy of the meta paths. The more the fraction of precision the more momentous the meta path. This discloses the implied semantic connotation of seed entities.

In addition to avoid repeated traversing we record the nodes that visited along the path *CP* in *(s.....d)* of the tuple *<(s, d), ⌐(s, d| CP),(s.....d)>*. Here *⌐(s, d| CP)* is similarity that characterizes*d* is in destination node set of source *s*, it is 1 if so and 0 otherwise. Here *⌐(s, d| CP)*designates whether the meta path attaches the seed pairs. The score *SPC* of tree node is that the document in which the *k* appears to the sum of all documents present.

Let us understand the algorithm with simple example where the set of seeds is {*HCL, country info, elections*} for simplicity. All these seed entities are the documents that are present. And they are marked as {*a, b, c*}. Now take a key word to search say it as *k*. Then the seed combinations include {[*k(a,b)],[k(b,c)],[k(a,c)],[k(a,b,c)*]}. The root node of tree contains all entity pairs and has initial *SPC = 0*. The expansion passes through the link: belongs to and gets the new child nodes. For each new node the *TSMPG* records *SP* and *SPC*. So, we choose the maximum number of tuples to have less discriminability. This process continuous until the given *k* is not in set of seed pairs.

We put forward the comprehensive steps of *TSMPG* Algorithm. Initially, the root node of the tree has been created in *Step 1* and roughly predefined constants are given in *Step 2*. Then the tree has to be grown and the significantmeta paths are found in *Steps 3-41*. At each expansion, the tree node with the maximum number of tuples is chosen in *Step 4*. *Step 10*estimates whether the neighbor node isn't visited earlier, whether the link

is in the set of the specified link type. If so, an expansion has to be made and the entity pair in seed combination pairs has to be inspected in *Step 15*. The connected entity pair has to be recorded. The new tuple has to be inserted into the corresponding tree node in *Step 20*. The source node of the entity pair to the source set of the tree node has to be added in *Step 21* and the updating of *SPC*has been done in *Step 22*. Some constants for variables such as accuracy, time delayis defined in *Steps27-30*. Calculation of the accuracy and delay time are performed in step *31-34* and seed pairs that current meta path connects are added from *Steps 35-42*and at last we are going to get the seed pairs *SP*, accuracy *SPC*, delay time *DT* of the given keyword *k*.

The detailed information about *MP ESE*has been introduced, which includes the following two steps. Firstly, an algorithm called Time based Seed Meta Path Generation (*TSMPG*) has been formedto systematicallydeterminesignificantmeta paths amongst seeds and provide the delay time and accuracy. Finally, heuristic learning method and PU learning method are implementedto mergeexcavatedmeta paths for the further *ESE*.

The time complexity of *MP ESE* algorithm includes two steps. The first oneinevitably finds themeta paths, i.e., the time complexity of determining significant meta paths among seeds is associated to *m* and *D*, where *m* indicates the number of seeds and *D* indicates the average degree of the entities in *TD*. The ranking entities complexity forestimating the likeness for all the entities and the seeds set, where *m*indicates the number of seeds, and to determinesignificantmeta paths.

The proposed algorithm can be used in Dictionary Construction, Word Sense Disambiguation, Query Refinement, Query Suggestion, Search Engine.It is used to establish the relationship among the keyword & documents.We can also find the likelihood of the user based upon the searching of a document represented in terms of the Accuracy, Time Delay and Ranking.

IV. EXPERIMENTAL ANALYSIS AND RESULTS:

The system is developed for checking the working nature using the seed entities, time delay and accuracy. If everything works properly then the separation of entities and calculation of Accuracy, Time Delay is displayed.

This paperindicates the expansion of the entities into seed entities and provide the meta path i.e., the Accuracy. The other part of the developed work displays the time.Once the user given the keyword that shows the time delay, accuracy and seed pairs and corresponding document that it contains. All the major functionalities are provided on the admin side, where we can check the number of users, graphical representations of accuracy, time delay and ranking.

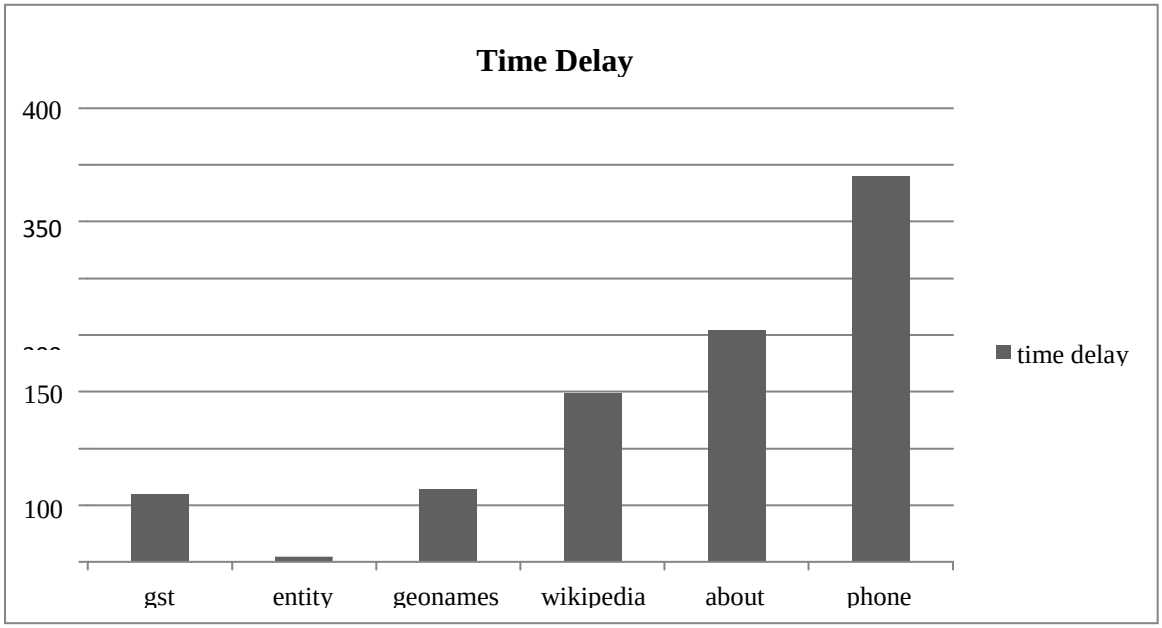

Fig.1: Graphical representation of time delay

Fig.1 shows the search history of all usersand the time delay for the given document vs keyword in microseconds(ms) graphically. Here the time delay is calculated based on the session initiated and the session last accessed time. In the above graph, the time delay for the word entity is less when compared to the other.

Fig. 2: Ranking results

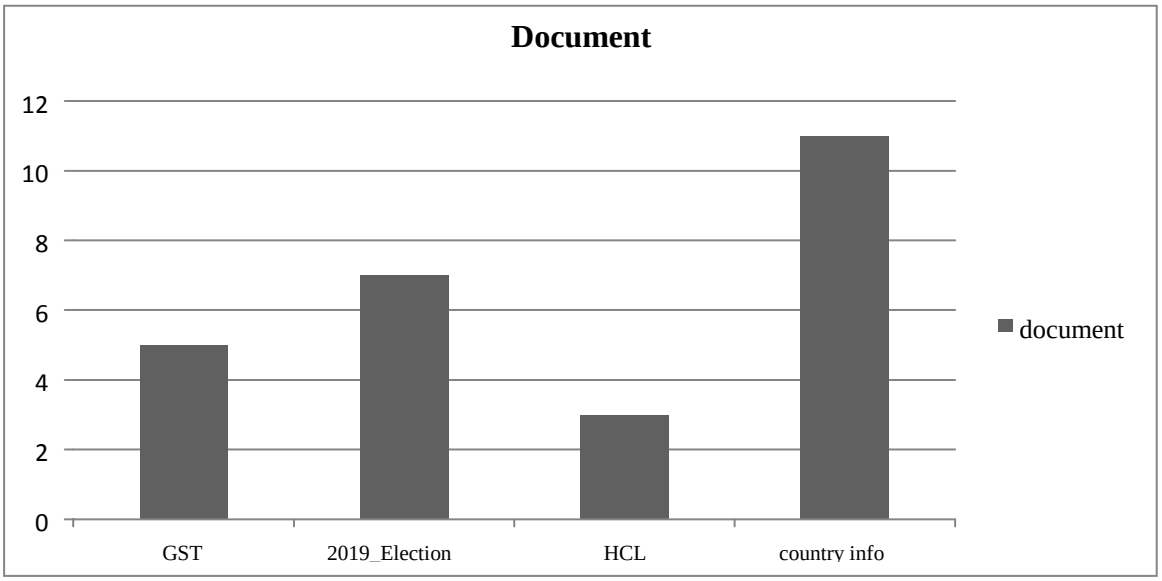

Fig. 2 shows the ranking for the given keyword or document graphically. The ranking is allocated in such a way based on the searching of the given document or keyword. Here in our experiment the country info takes the highest rank and the HCL takes the lowest rank.

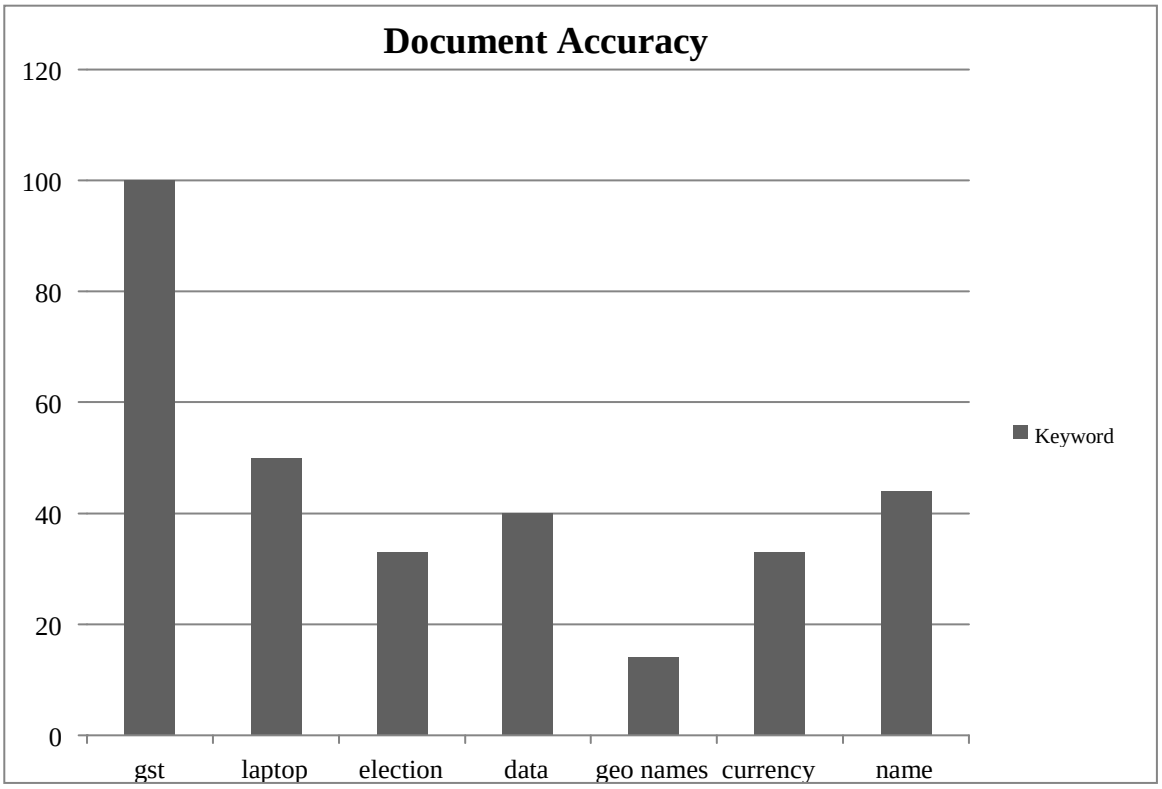

Fig.3: Graphical representation of accuracy

Fig.3shows the accuracy of the given document or keyword graphically. The problem of entity set expansion in a knowledge graph was studied. A knowledge graph as a heterogeneous information network and a Meta Path

Expansion Entity Set Expansion approach called MP ESE were proposed, which uses the meta path to exercise the implicit common properties of seeds.

V. Conclusion:

In order to inevitablydeterminesignificantmeta paths among seeds, MP ESE proposes a new algorithm called TSMPG, and then we propose a heuristic strategy and a PU learning method to allocate the prominence of meta paths. MP ESE uses weighted meta paths to improve entities. Experiments demonstrate the effectiveness and efficiency of MP ESE. In addition, we estimate the effects of seed size, delay time, accuracy, and order. Overwideexperimentations, we found that the proposed time-based TSMPG can determine seed pairs, meta paths between these pairs, and accuracy. We also found that the demonstrated MP ESE can attainexcellent performance in over-all extension tasks. Even on these overlapping class extension tasks, the proposed method is also appropriate from the point of view of the equalize between efficiency and effectiveness.

Through extensive experiments, the proposed TSMPG can discover seed pairs, the meta paths between these pairs, and the accuracy has been found. The presented MPESE can achieve good performance in general extension tasks. Even on these overlapping class extension tasks, the proposed method is also adequate from the point of view of the balance between efficiency and effectiveness. The optimal choice of seed combination and the correct determination of seed size can be studied in the future. We will also explore another weight learning method to get a more appropriate weight for the meta path.

References:

[1]. Wang, Junliang, et al. "Big data analytics for intelligent manufacturing systems: A review." Journal of Manufacturing Systems (2021).
[2]. Jahnavi, Y. et. al., A Novel Processing of Scalable Web Log Data using Map Reduce Framework, Springer Conference (2022).
[3]. Gao, Peng, Jingyi Li, and Shuai Liu. "An introduction to key technology in artificial intelligence and big data driven e-learning and e-education." Mobile Networks and Applications 26.5 (2021): 2123-2126.
[4]. Liang, Chen, et al. "Emergence and evolution of big data science in HIV research: Bibliometric analysis of federally sponsored studies 2000–2019." International Journal of Medical Informatics 154 (2021): 104558.
[5]. Costa, RogérioLuís de C., et al. "A survey on data-driven performance tuning for big data analytics platforms." Big Data Research 25 (2021): 100206.
[6]. Zou, You, et al. "Parallel computing for genome sequence processing." Briefings in Bioinformatics 22.5 (2021): bbab070.
[7]. Lin, Jerry Chun-Wei, et al. "Scalable mining of high-utility sequential patterns with three-tier MapReduce model." ACM Transactions on Knowledge Discovery from Data (TKDD) 16.3 (2021): 1-26.
[8]. Lin, Jerry Chun-Wei, et al. "Scalable mining of high-utility sequential patterns with three-tier MapReduce model." ACM Transactions on Knowledge Discovery from Data (TKDD) 16.3 (2021): 1-26.
[9]. JahnaviYeturu, "FPST: a new term weighting algorithm for long running and short-lived events", Int. J. Data Analysis Techniques and Strategies (Inderscience Publishers), Vol. 7, No. 4, 2015.
[10]. JahnaviYeturu, "A Cogitate Study on Text Mining", International Journal of Engineer-ing and Advanced Technology, Vol. 1, No. 6, pp. 189-196, 2012.
[11]. JahnaviYeturu, Analysis of weather data using various regression algorithms, Int. J. Data Science (Inderscience Publishers), Vol. 4, No. 2, 2019.
[12]. Jahnavi, Y., Elango, P., Raja, S.P. et al. A new algorithm for time series prediction using machine learning models.Evol.Intel.(2022). https://doi.org/10.1007/s12065-022-00710-5).
[13]. JahnaviYeturu, Statistical data mining technique for salient feature extraction, Int. J. Intelligent Systems Technologies and Applications (Inderscience Publishers), Vol. 18, No. 4, 2019.
[14]. Jahnavi, Y. and Radhika, Y. Hot topic extraction based on frequency, position, scatter-ing and topical weight for time sliced news documents, 15th International Conference on Advanced Computing Technologies, ICACT 2013.
[15]. Yeturu, Jahnavi, et al. "A Novel Ensemble Stacking Classification of Genetic Variations Using Machine Learning Algorithms." International Journal of Image and Graphics (2021): 2350015.
[16]. Hogan, Aidan, et al. "Knowledge graphs." Synthesis Lectures on Data, Semantics, and Knowledge 12.2 (2021): 1-257.
[17]. Kejriwal, Mayank. "Knowledge Graphs."Applied Data Science in Tourism.Springer, Cham, 2022.423-449.
[18]. Paulheim, Heiko. "Knowledge graph refinement: A survey of approaches and evaluation methods." Semantic web 8.3 (2017): 489-508.
[19]. Lu, Jinting, et al. "BEAT: Considering question types for bug question answering via templates." Knowledge-Based Systems 225 (2021): 107098.
[20]. Hu, Xin, JiangliDuan, and Depeng Dang. "Natural language question answering over knowledge graph: the marriage of SPARQL query and keyword search." Knowledge and Information Systems 63.4 (2021): 819-844.
[21]. Zhang, Jing, et al. "Neural, symbolic and neural-symbolic reasoning on knowledge graphs." AI Open 2 (2021): 14-35.
[22]. Xiao, Wenyi, et al. "Neural PathSim for Inductive Similarity Search in Heterogeneous Information Networks." Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021.
[23]. Yang, Carl, et al. "Heterogeneous network representation learning: A unified framework with survey and benchmark." IEEE Transactions on Knowledge and Data Engineering (2020).
[24]. C. Shi, Y. Li, J. Zhang, Y. Sun and P. S. Yu, "A survey of heterogeneous information network analysis," in IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 1, pp. 17-37, 1 Jan. 2017, doi: 10.1109/TKDE.2016.2598561.
[25]. X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: A heterogeneous information network approach," in WSDM. ACM, 2014, pp. 283–292.
[26]. Y. Shi, P.-W. Chan, H. Zhuang, H. Gui, and J. Han, "Prep: Path-based relevance from a probabilistic perspective in heterogeneous information networks," arXiv, 2017.
[27]. A. Krishnan, D. Padmanabhan, S. Ranu, and S. Mehta, "Select, link and rank: Diversified query expansion and entity ranking using

Wikipedia," in Springer, 2016, pp. 157– 173.
[28]. Z. Zhang, L. Sun, and X. Han, "A joint model for entity set expansion and attribute extraction from web search queries," in AAAI, 2016, pp. 3101–3107.
[29]. X. Yu, Y. Sun, B. Norick, T. Mao, and J. Han, "User-guided entity similarity search using meta-path selection in heterogeneous information networks," in CIKM. ACM, 2012, pp. 2025–2029.
[30]. J. Chen, Y. Chen, X. Du, X. Zhang, and X. Zhou, "Seed: A system for entity exploration and debugging in large-scale knowledge graphs," in ICDM. IEEE, 2016, pp. 1350–1353.