Knowledge graph curation from text via ontologies

Vaibhava lakshmi R
Department of Computer Technology, Madras Institute of Technology
Chennai, India
vaibhavi18092002@gmail.com

Gerard Deepak
Manipal Institute of Technology, Bengaluru, Manipal Academy of Higher Education
Manipal, India
gerard.deepak.christuni@gmail.com

Santhanavijayan A
Department of Computer Science and Engineering, National Institute of Technology
Tiruchirapalli, India
vijayana@nitt.edu

Radha S
School of Advanced Sciences, Vellore Institute of Technology
Chennai, India
radha.s@vit.ac.in

***Abstract*—Huge-scale knowledge graphs have become widespread on the web and are useful for a plethora of tasks. Knowledge graphs which are powered by machine learning use Natural Language Processing to build an extensive view of nodes, edges and labels by semantic enrichment in order to automate tasks like data structuring, text analysis and merging of data models. In this paper, we put forth a semantically-driven and knowledge-centric strategy which integrates machine learning and semantic inference for curating knowledge graphs from text using ontologies. Our framework amalgamates knowledge from the sources: Wikidata, DBpedia and Mediawiki to instill heterogeneity. The proffered framework transcends the baseline models by yielding an accuracy value of 98.22Cumulative Gain (NDCG) value of 0.97.**

***Index Terms*—Knowledge graphs, Semantic inference, Ontologies, Natural Language Processing**

I. Introduction

Despite the availability of a huge amounts of services and tools, for inspecting scholarly data, there is very exiguous support in the backdrop of tasks which involve sense making for exhaustive and precise depiction of the objects within a domain and their semantic connections. Among the extant techniques, Knowledge graphs are a great means to represent data in an organized fashion. Research in Artificial Intelligence significantly involves incorporation of structured human knowledge, which can be best represented by using knowledge graphs. A knowledge graph, otherwise called the "semantic network" is a directed and labeled graph which delineates a network of interconnected bodies – i.e., situations, events, concepts or objects [1]. It is a knowledge base that uses topology to consolidate data and converts data into machineunderstandable Knowledge. Some extant knowledge graphs are DBpedia, NELL, Wikidata, YAGO, Microsoft Satori, Amazon Product Graph, Facebook Entity Graph and Freebase. Knowledge graphs are built on purpose for frequently changing nature of knowledge. They provide a pliable basis for digital operations by adopting new data, requirements and definitions. Hence, enterprises are endeavoring to build and maintain knowledge graphs to bolster sundry downstream applications. They entrench cognizance into the business domain and can help disclose unrevealed conceptual links between heterogeneous items [2]. In the domain of Explainable Artificial Intelligence, knowledge graphs provide new prospects to overcome the issue of explainability for AI [3, 6]. Knowledge graphs play a focal role in the biomedical field in overcoming issues like scrutinizing treatment options for sundry diseases and recognizing relation between biomolecules and diseases [4]. With the advent of the Web over the 2000s, and the emergence of e-commerce and social media, knowledge graphs have come up as chief models for storing and querying disparate pieces of data that have real-world semantics [5]. Lately, knowledge graphs have been widely used in the case investigation field, where the particular downstream job is to give efficacious support to an investigator or analyst working to assess evidence and manage line of queries in the examination process [7]. Hence, the semantic networks have far-flung applications in a multitude of domains. Although, the practical applications of the semantic networks in certain industries are recent, it can be observed that knowledge graph technology has become highly beneficial to the development of a plenitude of industries and also helped to get rid of sundry technical issues which couldn't be solved before [8].

A. Motivation

There is a need for knowledge graphs in order to ensure that the Web 3.0 is properly modeled. The Web 3.0 is augmenting and organizing knowledge from sundry contributors. A community of knowledge has to be formulated and regulated. But knowledge is sparse in the Web 3.0. There is no human cognition. Hence, a knowledge-centric and semantically-driven framework for knowledge graph generation from text is required. The baseline models are not compliant with the cohesive structure of the Web 3.0 where the information density is extensively high. As a result, the proposed semantically-

Identify applicable funding agency here. If none, delete this.

driven and knowledgecentric model works based on Machine Intelligence which is amalgamation of Machine Learning and Semantic Inference and helps overcome the challenges faced using the baseline models.

B. Contribution

In this paper, we propose the knowledge-centric and semantically-driven paradigm for building knowledge graphs from text using ontologies. The framework predominantly involves amalgamation of knowledge from DBpedia and Wikidata to enrich the knowledge tree generated by MetaTag Harvester. Metatags are created which are formalized into a knowledge tree to enhance the auxiliary knowledge of the framework and hence, lessen the cognitive gap betwixt the knowledge going into the localized model and the knowledge in the World Wide Web. Finally, the generated knowledge graph is made more dense by TF-IDF vectorization.

C. Organization

This paper has the following sections: Section II is on related works and the extant methodologies. In section III, we have elucidated the architecture of the proffered framework. In section IV, the results for the proposed framework are discussed and compared with the other baseline models. Finally, section V contains the conclusion and future works to be done.

II. Related works

Shaoxiong Ji et. al. [1] have proffered a complete categorization on curation of knowledge graphs. Knowledge graph embedding and knowledge acquisition elaborately discussed. Also, sundry transpiring topics like meta relational learning and temporal knowledge graphs are discussed.To aid ensuing research on Knowledge graphs, a tailored combination of open source libraries and datasets on disparate tasks. Ali Hur et. al. [2] have reviewed about sundry extant automated methodologies for creating and maintaining Knowledge graphs. Also, a multitude of research problems pertaining to curation of knowledge graphs are discussed. Manas Gaur et. al.[3] have explained about the explainability and interpretability via usage of knowledge graphs in education and healthcare domain. Sundry learning algorithms that instill Knowledge and their applications for the aforementioned two domains are discussed. Taejin Kim et. al.[4] have proffered an Open Information Extraction without a priorly built dataset based on unsupervised learning. The proposed technique gets knowledge from a large bundle of text documents about COVID-19 rather than a general knowledge base and incorporates it to the current knowledge graph. Rik Koncel-Kedziorski et. al. [5] have put forth a novel graph transforming encoder that can leverage the associative composition of such knowledge graphs without thrusting linearization or hierarchical constraints. They have provided an endlong trainable system for graph-to-text convertion that can be applied to the domain of scientific text. Mikuláš Zelinka et. al. [6] have proffered a new Sequence-to-Sequence framework for yielding basic operations of KG. Furthermore, they have introduced a new dataset for extraction of KG created upon game transitions based on text. Phuc Do et. al. [7] have used a cross-lingual shift technique to create a Vietnamese knowledge graph.The proffered framework outperformed the experimentation done using MinIE algorithm on spark cluster. Aman Mehta et. al. [8] have proffered an endlong KG generation framework, that identifies and furnishes entities and relationships from text and delineates them to the homogeneous DBpedia namespace. For mapping of predicates, a Deep learning framework is put forth to model semantic similarity.

III. Implementation

A. Dataset

TABLE I: Details of datasets of different sub-domains

Sub-domain	Concepts	Sub-concepts	Individuals
Social media policy	1142	2546	54654
State policy innovation	2586	3345	9134
State policy diffusion	1414	2489	3586
National defense policy	5512	8932	11186
National security policy	5514	7893	12186
Environmental policy	7412	11812	17886
Sustainability policy	7123	12863	21486
Child development policy	7123	11384	70896
GSS Geography policy	30816	5812	8893
Economic and cooperative development policy	2386	6369	7142

The experimentation was conducted on a single and large integrated dataset, which is a combination of 12 individual datasets [9-20] related to generic domain of public policy. Public policy is a specific yet broad spectrum domain and hence includes sundry independent domains such as social media policy, state policy, nation defense security policy, planning policy, child care development fund policy, geographical policy, environmental and sustainability policy. The social media policy dataset provides straight link to the social media policy documents of higher education institutions. The state policy innovation and diffusion database contains details on the year of adoption in all the 50 American states for about 700 policies. The OECD National Defence and Security Policy documents database contains details on the official national security and defence policy documents published by the OECD's members. The planning policy documents dataset (2016), published by the Barrow Borough Council has information on progress in furnishing the Local plan and other planning documents. The Child Care and Development Funds policy database provides details on the states' and territories' child care subsidy programs. The GSS Geography Policy dataset has documents that provide information on using the geographic reference data so that the official statistics could be compared geographically. In [16], a unique dataset having all written communications published by German Bundestag betwixt 1949 and 2017 is published. Although these datasets were collected during different years, they were all strategically integrated based on commonly recurring categories by

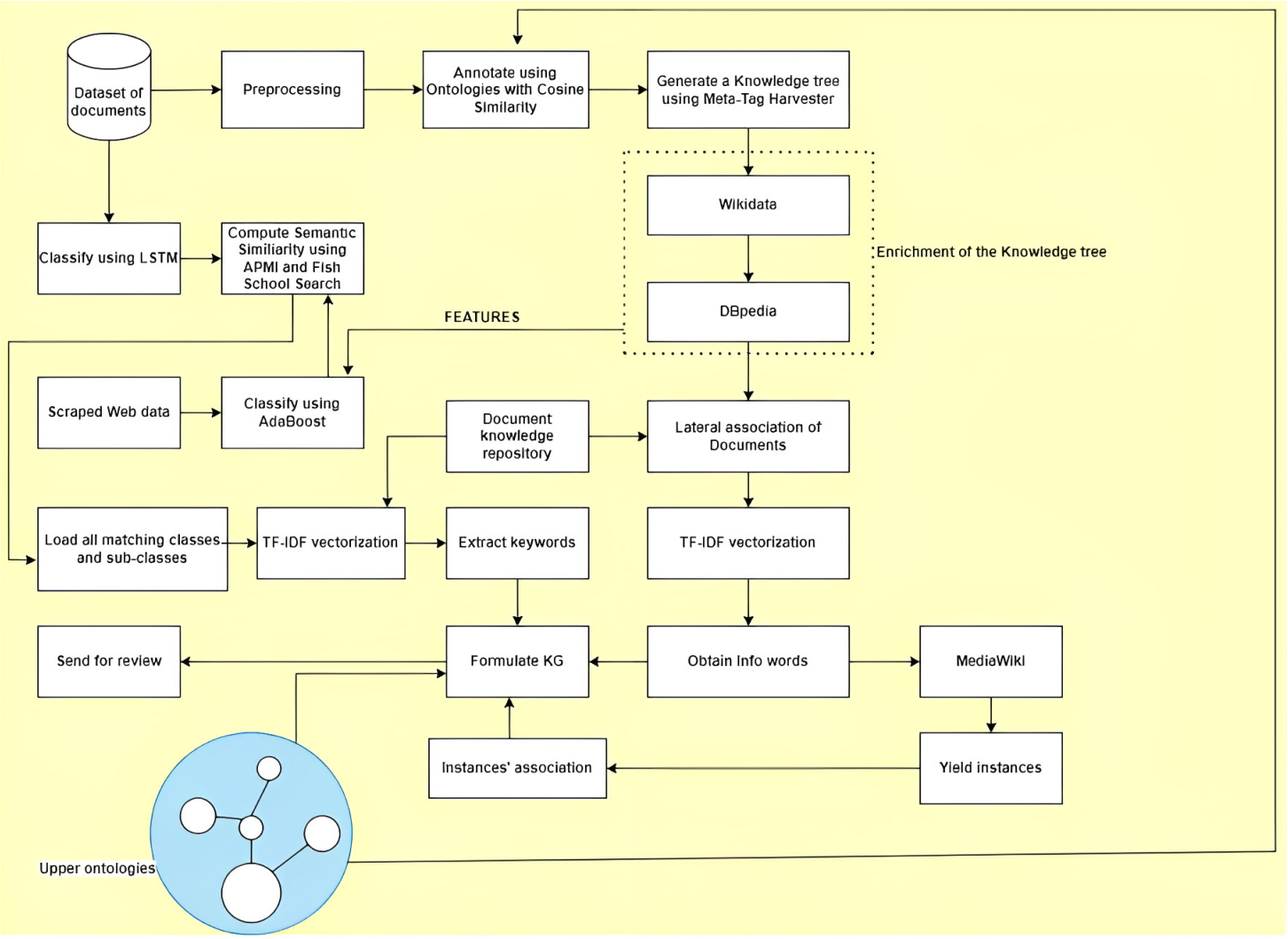

Fig. 1: Architecture of the proposed KGCTO framework

crawling Linked Open Data from the Wikidata and Mediawiki. These datasets were categorized and annotated using a customized web annotater. The initial domain ontology consists of the concepts, sub-concepts and individuals of a sub-domain as mentioned in Table.1.

B. Proposed architecture

Fig.1 illustrates the architecture of the proffered knowledge-centric, ontologydriven and conglomerated framework for knowledge graph generation. The dataset which consists of documents is subjected to pre-processing. Preprocessing encompasses Tokenization, Lemmatization, Stop-word removal, Named Entity Recognition and Word Sense Disambiguation. The dataset is subjected to pre-processing so as to eliminate inconsistencies in the form of punctuations, stop words and typos. For pre-processing, Python's NLTK (Natural Language Tool Kit) was used and for lemmatization, the WordNet 3.0 Lemmatizer was used. A white space special character and period punctuated tokenizer was used. For stop word removal, RegEx based stop word matching algorithm was incorporated. Thesaurus based NER (Named Entity Recognition) was achieved. The dataset is subjected to pre-processing so as to eliminate inconsistencies in the form of punctuations, stop words and typos. For pre-processing, Python's NLTK (Natural Language Tool Kit) was used and for lemmatization, the WordNet 3.0 lemmatizer was used. A white space special character and period punctuated tokenizer was used. For stop word removal, RegEx based stop word matching algorithm was incorporated. Thesaurus based NER (Named Entity Recognition) was achieved. At the end of pre-processing phase, the yielded content is subjected to annotations. The dataset is annotated by including the upper ontologies. These Upper ontologies are furnished from the categories of the dataset of documents. These upper ontologies formalized are used for annotating the pre-processed categories of the dataset by using the traditional Cosine similarity measure with a threshold of 0.60. The Cosine similarity measure is used to determine the similarity between

two entities. It can mathematically expressed as:

$$\cos(\mathbf{t}, \mathbf{e}) = \frac{\mathbf{te}}{\|\mathbf{t}\|\|\mathbf{e}\|} = \frac{\sum_{i=1}^{n} \mathbf{t}_i \mathbf{e}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{t}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{e}_i)^2}} \quad (1)$$

where 't' and 'e' are two entities. The cosine similarity determines the similarity between the instances and the ontologies and the instances which are relevant to the entities in the dataset. The reason for choosing a threshold of 0.60 is to have neither a large nor too small value, so as to achieve large number of highly relevant associations. Once the pre-processed dataset is annotated, a knowledge tree is generated using the MetaTag Harvester. The MetaTag Harvester generates tags in the form of metadata. The tags generated are linked based on the similarity values between them. Hence, the tags form the nodes of the tree. Once the knowledge tree is created, it is subjected to enrichment using two distinct knowledge sources using Wikidata and DBpedia. Wikidata is a lexico-syntactical and hierarchical constitution of Open Linked Data which is community contributed and community verified. DBpedia is a domainspecific aggregation of large knowledge subjected to several domains. Hence, the aforementioned two strong knowledge sources are used for enrichment of the knowledge tree. The enriched knowledge tree is further used for lateral association of documents by inclusion of the documents from a document knowledge repository. The documents in the document knowledge repository are several web scraped text documents relevant to the set of domains and sub-domains which are taken from the upper ontologies as well as the dataset comprising of the original documents. This knowledge repository comprises of 6,84,122 documents which are relevant to the original dataset and these documents are associated with that of the entities in the DBpedia by looking up the documents through keyword-based mapping using an agent which runs using the Pearson correlation coefficient and Cosine similarity measures. An automatic agent is modeled and the knowledge repository is linked. All the documents comprising the entities in the enriched knowledge tree are looked up using this agent and the informative terms from the lateral association of documents are obtained using the TF-IDF (Term Frequency - Inverse Document Frequency) vectorization and agent-based mapping technique. The TF-IDF is used for identifying informative terms in the dataset of documents. The TF-IDF weightage to the terms based on their frequency of appearance in a document. The higher the TF-IDF score, the more pertinent the term is. As the pertinence of a term decreases, the TF-IDF value approaches 0. It is mathematically expressed as:

$$TF - IDF = TF(t, d) \times IDF(t) \quad (2)$$

where

$$TF(t, d) = log_2(1 + freq(t, d)) \quad (3)$$

and

$$IDF(t) = log_2(\frac{N}{count(d \in D : t \in d)}) \quad (4)$$

where 't' is a term in the document 'd'. 'D' is the set of all documents. This is further subjected to formulation of a knowledge graph. Due to the large size of the dataset of documents, they can't be handled as it is. Hence, they are classified using the LSTM (Long Short Term Memory) classifier, which is an auto-handcrafted Deep learning classifier for classification of the dataset. The LSTM uses gating procedure that influences memorization. It consists of 3 gates: input, forget and output gates. The memory is stored by these gates in analog format. The forget gate determines what information to be kept and which one to be ignored. The input gate regulates the amount of information to be written onto the internal cell state. The output gate decides the value of the succeeding hidden state. This state contains details on antecedent inputs. We built the LSTM model with one hidden layer having 18 nodes. A dropout value of 0.25, decay rate of 0.97 and a momentum value of 0.7 was used. A learning rate of 0.75 was used. Weights were initialized between 0 and 0.25. Sigmoid activation function was made use of. Batch size was set to 32 for a total of 50 epochs.

The classified instances are yielded. The number of classes are not explicitly visible in the pipeline due to the use of a Deep learning classifier which makes sure that classes are implicitly taken care of. Furthermore, scraped web data having extortionate amount of information obtained from the web in an automated fashion, are classified using the AdaBoost classifier, by assimilating Wikidata and DBpedia enriched knowledge tree as features. The feature selection is done using the Pearson correlation coefficient with a step deviation of 0.45. The Pearson correlation coefficient is used to calculate linear correlation betwixt two sets of data. It can be mathematically expressed as:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 (y_i - \overline{y})^2}} \quad (5)$$

where
x_i = values of the x-variable in a sample
$\bar{x}$ = mean of the values of the x-variable
y_i = values of the y-variable in a sample
$\bar{y}$ = mean of the values of the y-variable

The step deviation of 0.45 is a lagged value when compared to the stringent step deviation value of 0.25. It helps to associate more number of features to classify the scraped web data using the AdaBoost. The reason for choosing AdaBoost is that it is a feature-controlled classifier. The outcomes of both the AdaBoost and the LSTM classifier are subjected to Semantic Similarity computation using the APMI (Adaptive Pointwise Mutual Information) measure. The APMI measure gives the relevance between two entities 'x' and 'y' as given in the equation below:

$$APMI(x; y) = \log \frac{p(x, y)}{p(x)p(y)} \quad (6)$$

where p(x) and p(y) denote probability of occurrence of enti-

ties ‘x’ and ‘y’ respectively and p(x,y) denotes the probability of occurrence of ‘x’ and ‘y’ together. The APMI measure is initially set to a value of 0.60. Further, the APMI computation between the classes identified from AdaBoost and LSTM classifier is optimized using the Fish School Search optimization algorithm which is a swarm-based optimization technique. FSS considers APMI with 0.60 threshold as the objective function and then computes much finer and optimal results. The APMI is used because it is a multi-staged, pointwise mutual information driven strategy which is in turn based on the NPMI which is a staged pointwise mutual information driven model with a threshold of 0.65. The initial solution set is transfigured into feasible solution sets so as to yield high quality matching classes and sub-classes. All the matching classes and sub-classes are loaded. The classes are linked to the matching sub-classes using a hash table data structure. They are used with that of the TF-IDF discovered entities from the document knowledge repository. Then the Semantic similarity is calculated using APMI and in this case, a value of 0.5. It is stringent because the relevance is exponentially increasing. Then the keywords or the informative terms which are extracted from the TF-IDF vectorization are directly submitted for formulation of the knowledge graph. Along with the extracted keywords, the upper ontologies are also integrated and the informative terms obtained in the previous step are associated. The informative terms obtained from the TF-IDF are further passed to Mediawiki API to yield large number of sub-classes and instances which are again associated with the knowledge graph directly by again computing the APMI measure with a 0.60 threshold. The finally formulated knowledge graph is sent for review by domain experts.

IV. Performance Evaluation and Results

TABLE II: COMPARISON OF PERFORMANCE OF THE PROPOSED KGCTO FRAMEWORK WITH OTHER APPROACHES

Search technique	Precision	Recall	Accuracy	F-measure	FDR
DLKGG	90.22	92.12	91.17	91.16	0.10
KGGT	89.44	92.09	90.76	90.74	0.11
KGFTG	91.15	92.62	91.88	91.87	0.09
CTMDM	92.49	93.09	92.79	92.78	0.08
DLPM	93.45	94.18	93.81	93.81	0.07
Proposed KGCTO	97.36	99.09	98.22	98.21	0.03

The performance of the proffered KGCTO framework is evaluated using Precision, Recall, Accuracy and F-measure for quantifying the relevance of the results, and the FDR (False Discovery Rate) for determining the false positives count which are furnished by the system.

$$Accuracy = \frac{TP + FP}{TP + FP + TN + FN} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

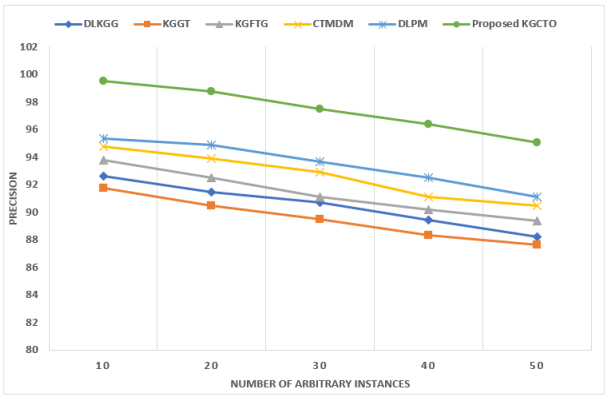

Fig. 2: Precision versus number of arbitrary instances curve

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

where
TP = number of positive classes correctly predicted
FP = number of positive classes incorrectly predicted
TN = number of negative classes correctly predicted
FN = number of negative classes incorrectly predicted

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{10}$$

$$FDR = 1 - Precision \tag{11}$$

Also, the NDCG (Normalized Discounted Cumulative Gain) loss metric is used for measuring diversity in the results. It is mathematically given as:

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{12}$$

where

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{13}$$

and

$$IDCG_p = \sum_{i=1}^{rel_i} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{14}$$

In equations (8) and (9), reli denotes the relevance score of an entity ’i’. In order to rate the performance of the proffered model, it is baselined with the DLKGG [4], KGGT [5], KGFTG [6], CTMDM[7] and DLPM [8] models respectively.

It is indicated from Table.2 that the proffered KGCTO framework furnishes highest precision, recall, accuracy and F-measure percentages of 97.36, 99.09, 98.22 and 98.21 percentages respectively and the lowest FDR of 0.03. The reason why the KGCTO framework outperforms the baseline models is due to the fact that it incorporates ontologies for curating the knowledge graph from text. Ontologies are cognizable entities conceived semi-automatically based on human reasoning and verification. Hence, there is a high degree of cognizance

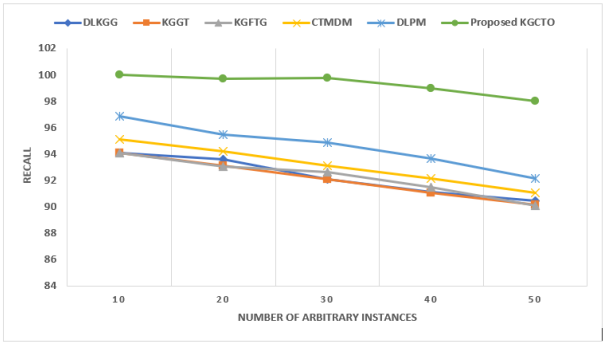

Fig. 3: Recall versus number of arbitrary instances curve

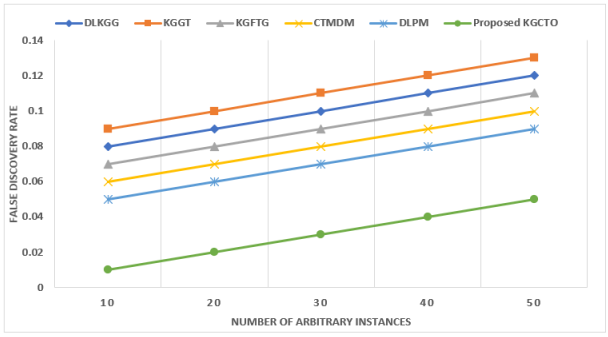

Fig. 4: FDR versus number of arbitrary instances curve

instilled in the ontologies and they provide perfect description of knowledge. The ontologies are graph-like structures that are light-weighted compared to knowledge graphs. Therefore, ontologies serve as perfect building blocks to derive the knowledge graphs. Apart from the incorporation of ontologies, the text dataset is annotated using the ontologies using Cosine similarity measure. Hence, knowledge inclusion takes place selectively and Cosine similarity being a perfect semantic similarity measure helps in achieving proper anchorage of the right ontological terms with that of the annotated dataset. Hence, proper alignment of the ontological nodes is regulated. Apart from this, metatags are generated which is formalized into a knowledge tree to enhance the auxiliary knowledge of the framework and hence lessen the cognitive gap betwixt

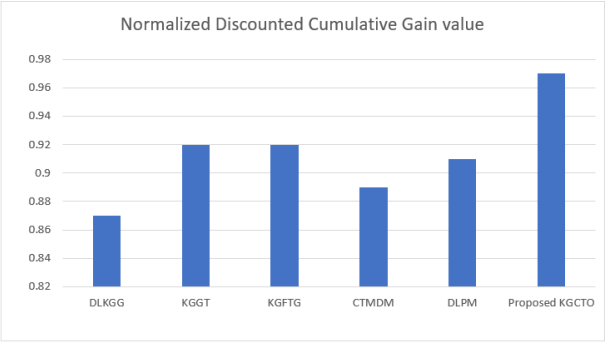

Fig. 5: Normalized Discounted Cumulative Gain curve

the knowledge going into the localized framework and the knowledge in the World Wide Web. Furthermore, the enrichment of the knowledge tree based on two different heterogeneous knowledge sources: Wikidata and DBpedia ensures large scale anchorage of entities which are crowd-sourced, communitycontributed and community-verified and accepted. Two disparate knowledge sources incorporate high degree of heterogenic variations. Owing to this, there is large scale knowledge instilled in the localized framework. Apart from this, the use of LSTM, which is a deep learning classifier with the feature of autohandcrafted selection classifies the instances of the dataset documents into independent classes which are further used to model the knowledge graph. The framework also uses the scraped web data which is classified using the AdaBoost classifier, which is a feature-controlled machine learning classifier. There is a strong need for using machine learning classifier here, so as to get rid of deviants from the features. Therefore, the incorporation of two distinct learning-based classifiers improves the predictive capability of the framework. Also, the TF-IDF based discovery of informative terms and instantiation using Mediawiki which is a knowledge source which is community-contributed and community-verified along with lateral association of matching classes and subclasses and encompassment of upper ontologies makes the knowledge graph much denser. Apart from this, the semantic similarity is calculated using the APMI measure using Fish School algorithm that makes sure that an optimal solution set is yielded. The precision versus number of arbitrary instances curve and recall versus number of arbitrary instances curve are depicted in Fig.2 and Fig.3 respectively. Fig.4 represents False Discovery Rate versus number of arbitrary instances curve. The proffered KGCTO framework occupies the highest position in Fig.2 and Fig.3 and the lowest position in fig.4 irrespective of the number of arbitrary instances, indicating that its performance is finer in comparison to the baseline models. Although the DLKGG framework is deep learning-based, it is highly dependent on a single domain. It uses BERT, which incorporates a strong learning environment. But the knowledge incorporated in the framework is very exiguous. Learning takes place solely from the large dataset of documents itself. Hence, learning process will be prone to outliers that will reduce the Precision, Recall, Accuracy and F-measure percentages and hence increase the FDR value. Due to the dearth of knowledge reasoning and knowledge regulation mechanisms, the performance of the DLKGG model is not at par with the KGCTO model. In the KGGT framework, large-scale text alone is incorporated and the learning ecosystem is very strong. However, no competitive techniques are incorporated to regulate the knowledge. Knowledge regulatory factors and schemes to select preferential knowledge is the actual need. This framework doesn't incorporate auxiliary knowledge sources. There are too many deviants in the results owing to the presence of outliers. Hence, this framework also underperforms in comparison to the proffered KGCTO framework. The KGFTG model involves dynamic knowledge graph generation based on text-based games and sequence

to sequence architecture using already extracted knowledge as data points. Only ready-made capsulated knowledge is incorporated in this framework. There are no knowledge understanding and regulating mechanisms used in this model. Hence, this framework yields lesser precision, accuracy, F-measure and Recall percentages and higher FDR value in contrast to the proposed framework. The CTMDM framework uses the cross-lingual transfer method along with the distributed MinIE algorithm on Apache Spark. Although, YAGO, DBLP, and DBpedia knowledge sources are used, Apache Spark being a big data ecosystem for processing data, doesn't have the cognizable capability of deriving knowledge from the data. No knowledge reduction or reasoning methods are used. Hence, the KGCTO framework transcends this model also. The DLPM model which uses a deep learning-based predicate mapping strategy is certainly better than the aforementioned baseline models, but underperforms when compared to the KGCTO framework. The predicates used in this framework are not just text-based, but can be URLs and links too. Hence, they can instill noise, and hence, affect the overall performance. Also, DBpedia is the only knowledge source used, hence there isn't any heterogeneity in the knowledge incorporated. Knowledge structuring strategy is included, but knowledge regulating technique is not used. Hence, DLPM framework underperforms owing to all these aforementioned factors. The comparison of the Normalized Discounted Cumulative Gain (NDCG) values is depicted in Fig.5. for the DLKGG, KGGT, KGFTG, CTMDM, DLPM and the proposed KGCTO frameworks respectively. From Fig.5 it can be inferred that the proposed KGCTO framework has the highest NDCG value of 0.97 in contrast to the baseline models owing to the inclusion of high number of auxiliary knowledge sources such as Mediawiki for instantiation, DBpedia and Wikidata for enrichment of the generated knowledge tree. Also, efficient regulation mechanisms incorporating APMI and cosine similarity measures improve optimization. The DLPM framework has a lesser NDCG value of 0.91 owing to the reason that it has only a single knowledge source incorporated and the predicates increase noise and outliers. The CTMDM framework has a lesser NDCG of 0.89 due to the paucity of sufficient knowledge sources. The KGGT and KGFTG frameworks are completely dataset-dependent and incorporate very exiguous knowledge, hence yielded lesser NDCG values of 0.92 each. The DLKGG framework has the least NDCG value of 0.87 due to complete dearth of knowledge incorporation.

V. CONCLUSION

Knowledge graphs can have widespread uses in governance of data, detection of fraudulent acts, management of knowledge, search, chatbot, recommendation, as well as intelligent systems across different organizational units. Hence, an efficient way for generating and curating knowledge graphs is necessary. In this paper, we have proffered the KGCTO framework for generating knowledge graphs from text by incorporation of knowledge and semantic intelligence. Sundry auxiliary knowledge sources like DBpedia, Wikidata and Mediawiki are used. Apart from this, the semantic similarity measure computed using the APMI measure using Fish school search algorithm makes sure that an optimal solution set is yielded. Hence, the proffered model yields an accuracy of 98.22outperforming all the baseline frameworks. As a part of future work, we intend to incorporate more auxiliary knowledge sources and use a more efficient scheme for aggregation of ontologies for experimentation.

REFERENCES

[1] Ji, S., Pan, S., Cambria, E., Marttinen, P., Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems, 33(2), 494-514.

[2] Hur, A., Janjua, N., Ahmed, M. (2021, December). A Survey on State-of-the-art Techniques for Knowledge Graphs Construction and Challenges ahead. In 2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE) (pp. 99-103) IEEE.

[3] Gaur, M., Desai, A., Faldu, K., Sheth, A. (2020). Explainable AI Using Knowledge Graphs. In ACM CoDS-COMAD Conference.

[4] Kim, T., Yun, Y., Kim, N. (2021). Deep learning-based knowledge graph generation for COVID-19. Sustainability, 13(4), 2276.

[5] Koncel-Kedziorski, R., Bekal, D., Luan, Y., Lapata, M., Hajishirzi, H. (2019). Text generation from knowledge graphs with graph transformers. arXiv preprint arXiv:1904.02342.

[6] Zelinka, M., Yuan, X., C˙ot´e, M. A., Laroche, R., Trischler, A. (2019). Building dynamic knowledge graphs from text-based games. arXiv preprint arXiv:1910.09532.

[7] Do, P., Phan, T., Le, H., Gupta, B. B. (2020). Building a knowledge graph by using cross-lingual transfer method and distributed MinIE algorithm on apache spark. Neural Computing and Applications, 1-17.

[8] Mehta, A., Singhal, A., Karlapalem, K. (2019, May). Scalable knowledge graph construction over text using deep learning based predicate mapping. In Companion Proceedings of The 2019 World Wide Web Conference (pp. 705-713).

[9] Laura A. Pasquini (2017). Social media policy document database [Dataset]. http://doi.org/10.6084/m9.figshare.4003401.v1

[10] Harvard Dataverse (2021). State Policy Innovation and Diffusion (SPID) Database v1.2 [Dataset]. http://doi.org/10.7910/DVN/CVYSR7

[11] Harvard Dataverse (2021). OECD National Defence and Security Policy Documents [Dataset]. http://doi.org/10.7910/DVN/5LURUV

[12] Barrow Borough Council (2016). Planning Policy Documents Updated 2016 [Dataset]. https://data.gov.uk/dataset/4c05eb0e-28ca-4ba3-9185- 0f51e6dbac22/planning-policy-documents-updated-2016

[13] Minton, Sarah; Giannarelli, Linda; Dwyer, Kelly; Tran, Victoria; Kwon, Danielle (2020). Child Care and Development Fund (CCDF) Policies Database, United States, 2009-2018 [Dataset].

[14] Minton, Sarah; Giannarelli, Linda (2018). Child Care and Development Fund (CCDF) Policies Database, United States, 2009-2016 [Dataset]. http://doi.org/10.3886/ICPSR36866.v3

[15] Office for National Statistics (2017). GSS Geography Policy [Dataset]. https://data.gov.uk/dataset/84604cb7-1d85-41d9-8646-500af1de89e0/gssgeography-policy

[16] Harvard Dataverse (2021). Every single word - A new dataset including all parliamentary materials published in Germany [Dataset]. http://doi.org/10.7910/DVN/7EJ1KI

[17] The Charity Commission (2013). Environmental and sustainability policy [Dataset]. https://www.gov.uk/government/publications/environmentaland-sustainability-policy

[18] World Bank (2018). Environmental policy [Dataset]. https://govdata360.worldbank.org/indicators/h84e40ea2

[19] Organisation for Economic Co-operation and Development (2020). Environmental Policy Stringency Index [Dataset]. https://knoema.com/EPS/environmental-policy-stringency-index

[20] World Bank (2018). Policy and institutions for environmental sustainability [Dataset]. https://govdata360.worldbank.org/indicators/hb3bc2bb